



UWEM 1.0
[Unified Web Evaluation Methodology]
version 1.0

[traduction en français]

Association BrailleNet
INSERM – UPMC B23
9 quai Saint Bernard
75252 Paris Cedex 05
www.brailenet.org
www.accessiweb.org
Tel : 01 44 27 26 25
info@accessiweb.org

Suivi de Version

Auteur	Version	Notes	Date
Association BrailleNet	1.0	Traduction en français de UWEM 1.0 (5 juillet 2006, version G)	Juillet 2006

Avertissement: Cette traduction en français de UWEM 1.0 a été réalisée par l'association BrailleNet en juillet 2006. L'association BrailleNet ne peut pas être tenue responsable des erreurs ou des imprécisions de traduction ainsi que leurs conséquences éventuelles. La [version officielle de UWEM 1.0 \(anglais\)](http://www.wabcluster.org/uwem1/) [http://www.wabcluster.org/uwem1/] est consultable en ligne sur le site du WAB Cluster.

Ce document (traduction en français de UWEM 1.0) est aussi consultable en ligne en format html à : <http://www.accessiweb.org/fr/uwem/index.html>



<http://www.braillet.net>



<http://www.accessiweb.org>

Table des matières

1. Résumé	5
2. Introduction	7
2.1 Définition de la méthodologie	8
2.2 Public concerné par ce document.....	8
2.3 Technologies concernées par ce document.....	9
2.4 Expertise nécessaire pour l'évaluation de l'accessibilité.....	9
2.5 Remerciements.....	10
2.6 Plus d'informations sur le WAB Cluster	10
3. Procédure d'évaluation et conformité.....	11
3.1 Types d'évaluation.....	11
3.2 Déclarations de conformité.....	12
3.3 Conformité des outils	13
4. Définition et échantillonnage d'un site Web.....	14
4.1 Définitions	14
4.2 Procédure pour définir un site Web.....	14
4.3 Procédure pour générer des échantillons pour l'évaluation	15
5. Liste des tests de conformité WCAG 1.0.....	18
Tests UWEM 1.0 pour la recommandation 1 des WCAG 1.0	20
Point de contrôle 1.1	20
Point de contrôle 1.2	25
Point de contrôle 1.3	25
Point de contrôle 1.4	26
Tests UWEM 1.0 pour la recommandation 2 des WCAG 1.0	28
Point de contrôle 2.1	28
Point de contrôle 2.2	30
Tests UWEM 1.0 pour la recommandation 3 des WCAG 1.0	32
Point de contrôle 3.1	32
Point de contrôle 3.2	33
Point de contrôle 3.3	34
Point de contrôle 3.4	36
Point de contrôle 3.5	39
Point de contrôle 3.6	41
Point de contrôle 3.7	45
Tests UWEM 1.0 pour la recommandation 4 des WCAG 1.0	47
Point de contrôle 4.1	47
Tests UWEM 1.0 pour la recommandation 5 des WCAG 1.0	51
Point de contrôle 5.1	51
Point de contrôle 5.2	52
Point de contrôle 5.3	53
Point de contrôle 5.4	54
Tests UWEM 1.0 pour la recommandation 6 des WCAG 1.0	57
Point de contrôle 6.1	57
Point de contrôle 6.2	59
Point de contrôle 6.3	61
Point de contrôle 6.4	63
Point de contrôle 6.5	64
Tests UWEM 1.0 pour la recommandation 7 des WCAG 1.0	66

Point de contrôle 7.1	66
Point de contrôle 7.2	69
Point de contrôle 7.3	72
Point de contrôle 7.4	75
Point de contrôle 7.5	77
Tests UWEM 1.0 pour la recommandation 8 des WCAG 1.0	79
Point de contrôle 8.1	79
Tests UWEM 1.0 pour la recommandation 9 des WCAG 1.0	81
Point de contrôle 9.1	81
Point de contrôle 9.2	81
Point de contrôle 9.3	82
Tests UWEM 1.0 pour la recommandation 10 des WCAG 1.0	84
Point de contrôle 10.1	84
Point de contrôle 10.2	85
Tests UWEM 1.0 pour la recommandation 11 des WCAG 1.0	87
Point de contrôle 11.1	87
Point de contrôle 11.2	87
Point de contrôle 11.4	91
Tests UWEM 1.0 pour la recommandation 12 des WCAG 1.0	93
Point de contrôle 12.1	93
Point de contrôle 12.2	94
Point de contrôle 12.3	94
Point de contrôle 12.4	99
Tests UWEM 1.0 pour la recommandation 13 des WCAG 1.0	101
Point de contrôle 13.1	101
Point de contrôle 13.2	102
Point de contrôle 13.3	102
Point de contrôle 13.4	103
Tests UWEM 1.0 pour la recommandation 14 des WCAG 1.0	104
Point de contrôle 14.1	104
6. Agrégation des résultats des tests	105
7. Rapport des résultats des tests	106
8. Rapport d'analyse évolutive	107
9. Glossaire	108
10. Références	109
11. Annexe A : licence du document	111
12. Annexe B : modèle de rapport (expert)	113
12.1 Introduction	113
12.2 Résumé	113
12.3 Contexte de l'évaluation	113
12.4 Liste des ressources	114
12.5 Résultats des vérifications	114
12.6 Références	115
12.7 Annexes	115
13. Annexe C: Schémas RDF	116
14. Annexe D: Contributeurs	117
15. Annexe E : Licence des documents du W3C	118
16. Notes	120
17. Téléchargement	123
18. Remerciements	124

1. Résumé

Ce document est le résultat d'un travail mené conjointement par 3 projets européens au sein d'un groupe appelé [WAB Cluster](#). 23 organisations participent à ces 3 projets européens dont les sigles sont [EIAO](#), [Support EAM](#) et [BenToWeb](#).

L'objectif de ce groupe est de développer une méthodologie unifiée pour l'évaluation du Web (Unified Web Evaluation Methodology (UWEM)). Cette version 1 de UWEM présente une méthodologie pour l'évaluation de la conformité des sites Web aux [WCAG 1.0](#) du [W3C](#). La prochaine version de UWEM tiendra compte de la transition des WCAG 1.0 vers les WCAG 2.0. A ce sujet, nous espérons que cette version de UWEM élaborée au sein du WAB Cluster pourra servir au travail sur les WCAG 2.0 fait par le W3C dans le cadre de son initiative [WAI](#).

Les organisations des 3 projets engagés dans l'élaboration de UWEM ont pour objectif que les outils d'évaluation et les méthodes (développés soit pour évaluer de manière globale le Web, soit pour évaluer séparément chaque site Web) soient compatibles et cohérents entre eux et avec WAI. Leur but est d'augmenter la valeur des évaluations en les basant sur une interprétation commune des WCAG 1.0. C'est pourquoi UWEM propose une interprétation harmonisée des WCAG 1.0.

UWEM fournit une procédure d'évaluation consistant en un système de principes et de pratiques pour l'évaluation de l'accessibilité du Web soit par un expert humain, soit de manière automatique par des interfaces de machines.

Le but de la version 1.0 de cette méthodologie est d'être complètement conforme aux recommandations WCAG 1.0. Actuellement, UWEM se limite aux recommandations de priorité 1 et 2 et présente une méthode unique à la fois pour l'évaluation par un expert et par des outils.

La méthodologie est utilisable pour les évaluations détaillées d'une page Web, d'un site complet (quelle que soit sa taille) ou de plusieurs sites. Elle couvre les problématiques de sélection d'un échantillon Web représentatif, de clarification des points de contrôle des WCAG 1.0, de l'interprétation et de l'intégration/agrégation des résultats, des informations nécessaires pour le rapport d'évaluation.

UWEM sera utilisée par les projets appartenant au WAB Cluster pour la création d'un observatoire du Web (projet EIAO), pour le développement de modules d'évaluation (projet BenToWeb) et pour la mise en place d'un schéma européen de certification (projet Support EAM). Plus d'informations sur le WAB cluster, UWEM et les projets impliqués sont disponibles sur le site <http://www.wabcluster.org/>.

Ce document (UWEM version 1.0) est organisé de la façon suivante :

- La [section 2 \(Introduction\)](#) indique les pré-requis, le public visé et les techniques pour le document et sert à décrire les propriétés élémentaires de UWEM.
- La [section 3 \(Procédures d'évaluation et conformité\)](#) décrit les procédures d'évaluation de UWEM et de conformité.
- La [section 4 \(Définition et échantillonnage d'un site Web\)](#) explique les méthodes de délimitation et de sélection permettant de déterminer les listes de ressources pour la conformité d'un site Web. Les procédures sont développées afin que les échantillons identifiés soient représentatifs du niveau d'accessibilité du site Web.

- La [section 5 \(Liste des tests de conformité WCAG 1.0\)](#) sur les tests de conformité décrit comment effectuer des tests pour une évaluation de la conformité aux points de contrôle de priorité 1 et 2 des WCAG 1.0.
- La [section 6 \(Agrégation des résultats des tests\)](#) donne un modèle pour agréger les résultats des tests à partir des procédures décrites dans la section 5. Le modèle contient également l'estimation des barrières d'accessibilité dans un site Web ainsi que dans des groupes de sites Web.
- La [section 7 \(Rapport des résultats des tests\)](#) décrit le rapport des résultats des tests à l'aide de EARL et propose également un modèle de rapport pour les résultats d'une évaluation expert.
- La [section 8 \(Rapport d'analyse évolutive\)](#) donne une méthode de rapport spécialement conçue pour les décideurs politiques à l'aide d'une méthode basée sur des scorecard.
- La [section 9 \(Glossaire\)](#) et la [section 10 \(Références\)](#) fournissent un glossaire des termes et une liste des références utilisées dans ce document.
- L'[annexe A](#) est la licence du document pour UWEM.
- L'[annexe B](#) est un modèle pour le rapport des résultats de l'évaluation adapté pour l'évaluation expert à l'aide de UWEM.
- L'[annexe C](#) donne des schémas RDF.
- L'[annexe D](#) contient la liste des personnes ayant contribué à la création de UWEM et à la réalisation de ce document.
- L'[annexe E](#) fournit la licence des documents W3C pour les travaux du W3C référencés dans ce document UWEM.

2. Introduction

Sommaire:

- [2.1 Définition de la méthodologie](#)
- [2.2 Public concerné par ce document](#)
- [2.3 Technologies concernées par ce document](#)
- [2.4 Expertise nécessaire pour l'évaluation de l'accessibilité](#)
- [2.5 Remerciements](#)
- [2.6 Plus d'informations sur le WAB Cluster](#)

La méthodologie Unified Web Evaluation Methodology (UWEM) devrait faire en sorte que les outils d'évaluation et les méthodes (développés soit pour évaluer de manière globale le Web, soit pour évaluer séparément chaque site Web) soient compatibles et cohérents entre eux et avec WAI. Ce document est le résultat d'un travail mené conjointement par 3 projets européens au sein d'un groupe appelé [WAB Cluster](#). 23 organisations participent à ces 3 projets européens dont les sigles sont [EIAO](#), [Support EAM](#) et [BenToWeb](#).

UWEM décrit une méthodologie pour l'évaluation de la conformité aux recommandations WCAG 1.0 du W3C. Des raisons pratiques ont conditionné ce choix. En effet, les recommandations WCAG 1.0 sont largement acceptées et forment un facteur stable d'accessibilité depuis mai 1999. En 2002 déjà, l'Union Européenne recommandait qu'elles soient adoptées par le secteur public des états membres. Dans plusieurs pays, elles sont intégrées comme référence à la législation nationale rendant obligatoire l'accessibilité des sites Web du secteur public. La prochaine version de UWEM tiendra compte de la transition des WCAG 1.0 vers les WCAG 2.0.

L'objectif de UWEM est de fournir une base unique permettant à la fois l'évaluation de l'accessibilité du Web par un expert humain et de manière automatique par des interfaces de machines [\[1\]](#). L'évaluation même de UWEM est aussi prévue pour contribuer au travail de W3C/WAI sur l'évolution de leurs recommandations. L'équipe de W3C/WAI a évalué et fourni une contribution à des ébauches antérieures de ce document afin de minimiser la fragmentation éventuelle du contenu technique. Cela n'implique en aucun cas l'approbation par le W3C ou WAI de tout ou partie de ce document.

Une partie des informations présentées dans ce document à la [section 5 \(Liste des tests de conformité WCAG 1.0\)](#) sont des annotations des documents du W3C. Nous avons utilisé en particulier les deux documents suivants :

- [WCAG 1.0](#),
- [Techniques for Web Content Accessibility Guidelines 1.0 \(WCAG1.0-TECHS\)](#), et d'autres documents de techniques liés à celui-ci.

Selon la Intellectual Rights FAQ (FAQ du W3C sur les droits intellectuels) [\[2\]](#), la section 5 de UWEM relève d'une règle qui permet la copie sans modification du document W3C [\[3\]](#) ayant été utilisé. C'est pourquoi toutes les références aux recommandations et aux points de contrôle du document WCAG 1.0 sont dûment citées et l'URL vers le document original est incluse. Le W3C n'est pas responsable de tout contenu ne se trouvant pas à l'url originale et les annotations de ce document ne sont pas normatives.

2.1 Définition de la méthodologie

La méthodologie Unified Web Evaluation Methodology fournit une procédure d'évaluation consistant en un système de principes et de pratiques pour l'évaluation de l'accessibilité du Web soit par un expert humain, soit de manière automatique par des interfaces de machines. Le but de la version 1.0 de cette méthodologie est d'être complètement conforme aux recommandations priorité 1 et 2 des WCAG 1.0.

UWEM a pour but d'augmenter la valeur des évaluations en les basant sur une interprétation commune des WCAG 1.0 et sur une série de tests suffisamment robustes pour donner aux parties intéressées confiance dans les résultats. Les personnes produisant du contenu Web peuvent également souhaiter évaluer leur propre contenu et UWEM vise aussi à être appropriée pour ces utilisateurs.

La méthodologie est conçue pour répondre aux exigences suivantes :

- **Conformité à WAI:** conformité aux documents de recommandations et de techniques existants de WAI.
- **Indépendance:** les tests sont rédigés de manière à être aussi indépendants des outils et des navigateurs que possible.
- **Interprétation unique:** les tests doivent pouvoir être interprétés d'une seule façon.
- **Répliquabilité:** différents évaluateurs de l'accessibilité du Web effectuant les mêmes tests sur le même site doivent pouvoir obtenir les mêmes résultats dans une tolérance donnée.
- **Traduction:** la méthodologie doit pouvoir être traduite dans les différentes langues.
- **Conformité avec le règlement européen:** conformité avec le règlement (CE) 808/2004 du Parlement Européen et du Conseil du 21 avril 2004 concernant les statistiques communautaires sur la société de l'information.

Nous avons inclus dans la méthodologie des informations à propos de :

- La définition et l'échantillonnage d'un site Web.
- Le rapport, l'interprétation et l'intégration/agrégation des résultats des tests.

2.2 Public concerné par ce document

Le publics concerné par ce document inclue mais n'est pas limité à :

- Des projets d'évaluation globale de l'accessibilité du Web (Commission Européenne, gouvernements nationaux, groupes de personnes handicapées, EDEAN, et institutions de recherche.
- D'éventuelles autorités de certification.
- Des personnes produisant du contenu Web et qui souhaitent évaluer ce contenu.
- Des développeurs d'outils d'évaluation et de réparation.
- Des décideurs politiques et des propriétaires de sites Web.
- D'autres organisations qui font des évaluations de sites Web.

La Commission Européenne, les gouvernements nationaux et les autres organismes souhaitant évaluer de manière globale l'accessibilité du Web pourront utiliser UWEM afin d'effectuer des évaluations et comparer leurs résultats de façon significative.

UWEM est une méthodologie d'évaluation et n'a pas pour objectif de fournir des conseils aux personnes produisant du contenu Web et qui souhaitent rendre ce contenu conforme aux WCAG 1.0. Ces informations sont données dans les documents de techniques des WCAG 1.0 (WCAG1.0-TECHS) disponibles sur le site de W3C/WAI.

2.3 Technologies concernées par ce document

UWEM traite principalement des méthodes pour évaluer les documents basés sur les technologies suivantes :

- HTML 4.01,
- XHTML 1.0 et 1.1,
- CSS 2.x,
- Autres objets intégrés dans les ressources HTML et XHTML.

2.4 Expertise nécessaire pour l'évaluation de l'accessibilité

La suite de documents sur l'évaluation du W3C (W3C Evaluation suite) [\[4\]](#) décrit en détail l'expertise nécessaire pour l'évaluation de l'accessibilité des contenus Web pour les personnes handicapées.

Les activités d'évaluation nécessitent diverses expertises et perspectives. Les individus évaluant l'accessibilité des contenus Web ont besoin d'une formation et d'une expérience dans une grande gamme de disciplines. Une méthode collaborative peut apporter de meilleurs résultats aux individus qui évaluent des contenus Web.

La suite sur l'évaluation de W3C/WAI dit [\[5\]](#): une évaluation efficace de l'accessibilité du Web exige plus que de seulement lancer un outil d'évaluation sur un site Web. Des évaluations complètes et efficaces nécessitent des évaluateurs connaissant les technologies du Web, les outils d'évaluation, les barrières rencontrées par les personnes handicapées, les aides techniques et les méthodes utilisées par les personnes handicapées, ainsi que les recommandations et techniques sur l'accessibilité.

Ce document décrit la méthodologie d'évaluation expert et automatique. L'évaluation automatique peut réduire considérablement le temps et les efforts nécessaires pour une évaluation mais il est très important de savoir que de nombreuses vérifications de l'accessibilité nécessitent l'avis d'un expert humain et doivent être effectuées manuellement. Ceci est expliqué de façon plus détaillée dans les tests de la section 5.

Des informations complémentaires sur l'utilisation d'outils sont disponibles dans la section de la suite sur les évaluations du W3C/WAI sur la sélection des outils "selecting tools" [\[6\]](#).

2.5 Remerciements

Les organisations suivantes ont travaillé à l'élaboration de ce document UWEM (version 1.0):

Accessibility Foundation (Pays-Bas, co-ordinateur du [WAB Cluster](#)); Agder University College (Norvège, co-ordinateur de [EIAO](#)); Association BrailleNet (France, co-ordinateur [Support EAM](#)); Fraunhofer Institute for Applied Information Technology FIT (Allemagne, co-ordinateur [BenToWeb](#)); Vista Utredning AS (Norvège); Forschungsinstitut Technologie-Behindertenhilfe der Evangelischen Stiftung Volmarstein (Allemagne); The Manchester Metropolitan University (Royaume-Uni); Nettkroken as (Norvège); FBL s.r.l. (Italie); Université des technologies de Varsovie; Faculté d'ingénierie de production, Institut de l'organisation des systèmes de production (Pologne); Université d'Aalborg (Danemark); Intermedium as (Norvège); Fundosa Technosite (Espagne); Université de la ville de Dublin (Irlande); Universität Linz, Institut integriert studieren (Autriche); Katholieke Universiteit Leuven, Recherche et Développement (Belgique); Accessinmind Limited (Royaume-Uni); Multimedia Campus Kiel (Allemagne); Department of Product and Systems Design, University of the Aegean (Grèce); Université de York (Royaume-Uni); ISdAC International Association (Belgique); FernUniversität in Hagen (Allemagne).

Nous remercions l'équipe de la Web Accessibility Initiative du World Wide Web Consortium pour tous les commentaires utiles qu'ils ont faits sur les versions d'ébauches de ce document.

2.6 Plus d'informations sur le WAB Cluster

Les projets participant au WAB Cluster sont financés par l'Union Européenne dans le second appel IST FP6 (2003) de l'objectif stratégique de l'intégration numérique. Le [site Web du WAB Cluster](#) est consultable en ligne. Plus d'informations sur les projets sont disponibles sur les sites Web des projets :

- [EIAO](#).
- [Support EAM](#).
- [BenToWeb](#).

3. Procédure d'évaluation et conformité

Sommaire:

- [3.1 Types d'évaluation](#)
- [3.2 Déclarations de conformité](#)
- [3.3 Conformité des outils](#)

Cette section décrit les procédures et leurs objectifs pour l'évaluation expert et automatique et fournit des informations sur les déclarations de conformité basées sur UWEM 1.0. Les procédures d'évaluation de UWEM 1.0 ont les objectifs suivants :

- Améliorer la répliquabilité des résultats de l'évaluation à l'aide de la définition d'un schéma d'échantillonnage reproductible (voir la [section 4](#)).
- Clarifier l'interprétation, faciliter l'évaluation automatique et améliorer la répliquabilité des résultats de ses tests (voir la [section 5](#)).
- Permettre l'agrégation des résultats de l'évaluation de pages Web, de sites Web et même d'une série de sites Web (voir la [section 6](#)). UWEM permettra également les agrégations selon des zones géographiques ainsi que des secteurs économiques (voir la [section 8](#)).
- Permettre le rapport des résultats des tests (voir la [section 7](#) et la [section 8](#)).
- Permettre l'évaluation de sites Web à grande échelle pour identifier les zones de problèmes.
- Evaluer la conformité aux vérifications de la section 5. Cela inclut les procédures d'évaluation allant jusqu'à des tests très spécifiques pour l'évaluation expert et automatique.

3.1 Types d'évaluation

Les tests d'accessibilité peuvent être effectués par des experts, par des tests automatiques ou par des utilisateurs. Les différentes méthodes d'évaluation ont chacune des forces et des faiblesses.

La figure 1 décrit trois différentes méthodes d'évaluation parmi lesquelles deux (l'évaluation automatique et l'évaluation expert) sont traitées dans UWEM. La figure montre, par exemple, que l'évaluation automatique (par l'outil 1 ou l'outil2) peut seulement vérifier la conformité à une sous-liste de points de contrôle des WCAG 1.0 (tels que la série de tests indiqués comme "complètement automatisable" dans la section 5), ce qui signifie en outre que seulement une sous liste de toutes les possibles barrières d'accessibilité peuvent être identifiées de façon fiable à l'aide de tests automatiques. C'est pourquoi le traitement de l'évaluation automatique comme indicateur global de l'accessibilité est bas, même si cela permet d'identifier de nombreuses barrières de façon fiable. Il peut être aussi appliqué de façon efficace pour tester un nombre important de ressources Web dans un site Web ainsi que dans des sites multiples. Les outils 1 et 2 sont ici deux outils d'évaluation complètement automatiques dont le but est de vérifier les questions d'accessibilité avec des fonctionnalités qui peuvent être identiques ou similaires. Certains outils peuvent également être une aide dans un processus d'évaluation expert. Les outils donnent des résultats fiables pour une série de tests et peuvent non seulement accélérer le processus en effectuant certaines tâches de façon automatique, mais aussi en indiquant où se trouvent certaines barrières et indiquer les endroits où les évaluateurs experts doivent diriger leur évaluation.

Les tests utilisateurs permettent d'identifier les barrières qui ne peuvent être repérées par d'autres méthodes de tests et permettent également d'estimer l'accessibilité des scénarios testés. Cependant, les tests utilisateurs sont assez spécialisés. C'est pourquoi les tests utilisateurs ne conviennent généralement pas à des tests de conformité puisqu'ils ne permettent pas de tester tous les aspects des tests de la section 5. La meilleure méthode pour assurer la conformité à l'accessibilité du Web ainsi qu'à UWEM est d'utiliser une méthode mixte incluant toutes les méthodes d'évaluation : automatique, expert et tests utilisateurs.

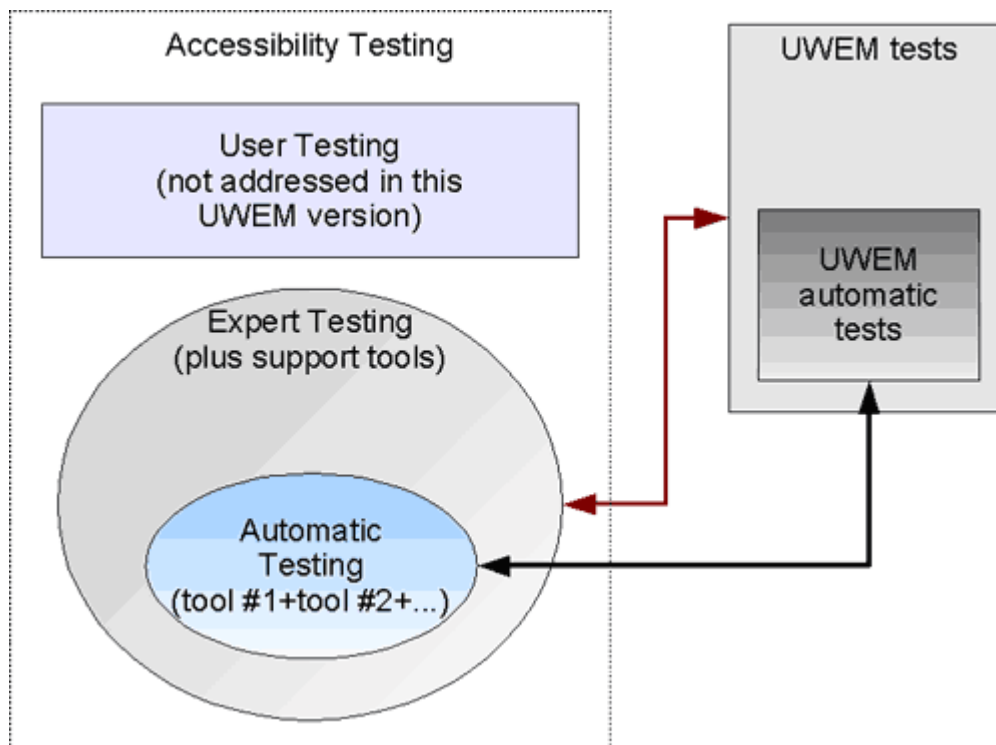


Figure 1: les différents types d'évaluation par UWEM (automatique, expert, utilisateur).

Les tests utilisateurs ne sont pas traités dans cette version de UWEM. Le site de WAI décrit comment impliquer les utilisateurs dans l'évaluation de contenus Web dans la suite sur l'évaluation [\[7\]](#).

Les principaux avantages des tests automatiques sont les suivants :

- La méthode est reproductible (bien que la nature dynamique de certaines technologies Web implique qu'une répétition complète est impossible [LEVENE99]).
- Elle peut être appliquée à un grand nombre de ressources sur un site Web et à des sites Web multiples.

3.2 Déclarations de conformité

Une déclaration de conformité détermine si un site Web répond aux normes d'accessibilité décrites à la section 5. Pour déclarer la conformité à UWEM, il est requis au minimum que :

1. La définition du site Web et l'échantillon des ressources évaluées respectent les conditions de la section 4.
2. Toutes les ressources de l'échantillon réussissent tous les tests applicables du niveau de conformité correspondant.

Les niveaux de conformité de UWEM sont la réplique de ceux des WCAG 1.0, c'est-à-dire :

- Niveau de conformité 1 : tous les tests relevant des points de contrôle de priorité 1 des WCAG 1.0 sont respectés.
- Niveau de conformité 2 : tous les tests relevant des points de contrôle de priorité 1 et de priorité 2 des WCAG 1.0 sont respectés.

[Plus d'informations sur la signification des priorités](#) sont disponibles en ligne.

Les déclarations de conformité à l'accessibilité selon la méthodologie UWEM doivent utiliser la forme suivante :

1. La version de UWEM et [l'identifiant URI de UWEM](#).
2. L'URI vers un document détaillant la définition du site Web et l'échantillon des ressources évaluées auxquels se réfère la déclaration.
3. Le niveau de conformité.

3.3 Conformité des outils

Les outils d'évaluation peuvent également permettre de déclarer une conformité à UWEM 1.0. De cette manière, les experts évaluant des sites Web selon UWEM 1.0 pourront se fier aux résultats de l'outil pour les tests de la méthodologie complètement automatisables.

Pour déclarer la conformité à UWEM 1.0, l'outil doit implémenter tous les tests complètement automatisables de la section 5, du niveau de conformité correspondant. Cette déclaration de conformité doit utiliser la forme suivante :

1. La version de UWEM et [l'identifiant URI de UWEM](#).
2. L'URI vers un document détaillant une série de fichiers de tests publics où cette déclaration de conformité a été vérifiée.
3. Le niveau de conformité.

4. Définition et échantillonnage d'un site Web

Sommaire:

- [4.1 Définitions](#)
- [4.2 Procédure pour définir un site Web](#)
- [4.3 Procédure pour générer des échantillons pour l'évaluation](#)
 - [4.3.1 Liste des ressources essentielles](#)
 - [4.3.2 Liste des ressources échantillonnées](#)
 - [4.3.3 Taille de l'échantillon dit manuel](#)

4.1 Définitions

Dans le cadre de UWEM, les déclarations de conformité doivent faire référence à une liste des ressources évaluées sur l'étendue du/des site(s) Web. Cette section donne les définitions de base pour les différents concepts utilisés.

- **Ressource:** un objet de données identifié par un URI [RFC3986]. Cette définition est adaptée à partir de la définition d'une ressource dans [RFC2616]. Ce concept concerne également les ressources non http comme par exemple, celles auxquelles on accède via le protocole FTP. Ce type de ressource doit être exprimé via une instance earl: WebContent Class (voir l'[annexe C](#) et [EARL10-Schema] pour plus de détails).
- **Ressource HTTP:** un objet de données identifié par une seule requête HTTP. Ce type de ressource doit être exprimé via une instance earl:WebContent Class (voir l'[annexe C](#) pour plus de détails) qui doit contenir des composants supplémentaires de la note du W3C [HTTP-RDF]. Cette distinction des ressources est due à la complexité sous-jacente du protocole HTTP [RFC2616] dans lequel la négociation de contenu peut causer différentes versions d'une ressource (comme par exemple des versions en différentes langues via l'en-tête Accept-Language HTTP).
- **Liste de ressources:** les déclarations de conformité dans UWEM sont liées à une liste donnée de ressources, exprimée comme une séquence RDF de ressources de tout type. L'[annexe C](#) décrit la syntaxe RDF utilisée pour exprimer une liste de ressources.

Selon les besoins des différentes applications de UWEM, cette liste de ressources peut être spécifiée par une variété de participants différents au processus d'évaluation, comme par exemple un propriétaire de site, un rédacteur de site, un organisme d'inspection etc. Ce document explique simplement comment une telle liste doit être exprimée de façon non équivoque.

4.2 Procédure pour définir un site Web

Pour la méthode UWEM, un site Web est défini comme un regroupement arbitraire de ressources Web en hyperliens, dont chacune est identifiée par la procédure décrite à la [section 4.1 \(Définitions\)](#).

Le but de UWEM est de garantir la répliquabilité des résultats. Par conséquent, il est d'une grande importance pour l'agrégation et la comparaison des résultats que la définition des ressources testées soit sans ambiguïté. Par conséquent, des déclarations de conformité du type "http://exemple.org/" est

conforme à la priorité 1 de WCAG 1.0" sans précision sur la définition du site exemple.org (cad, ses ressources) ne peut pas être acceptée.

Par ailleurs, en tenant compte du fait du nombre important de logiciels de sélection de ressources Web et du fait qu'ils utilisent des technologies différentes, ce n'est pas possible de vérifier l'exactitude de leurs sélections. De plus, les différents RFCs sur les Noms de Domaine laissent possible toutes les interprétations sur la définition des sous-domaines.

Par conséquent, en ce qui concerne les déclarations de conformité pour UWEM 1.0, la définition d'un site Web doit être exprimée par une liste de ressources (voir l'[annexe C](#)).

4.3 Procédure pour générer des échantillons pour l'évaluation

En général, il ne sera pas réalisable de tester toutes les ressources du site Web selon tous les critères d'évaluation. Par conséquent, après avoir déterminé et sélectionné une liste de ressources à évaluer et le niveau de conformité visé, nous proposons d'identifier un certain nombre de pages ou d'"échantillons".

Les ressources à sélectionner devraient inclure la liste des ressources essentielles complétée par une sélection de ressources arbitraires. Nous appelons cette dernière la liste des ressources échantillonnées.

4.3.1 Liste des ressources essentielles

La liste des ressources essentielles est une série de ressources génériques, probablement présentes sur la plupart des sites Web, et qui sont essentielles pour l'utilisation et l'évaluation de l'accessibilité d'un site. Par conséquent, la liste des ressources essentielles représente une série de ressources qui doivent être incluses dans toute évaluation de l'accessibilité du site. La liste des ressources essentielles ne peut généralement pas être identifiée de façon automatique mais nécessite un jugement humain pour sa sélection. Dans le cas de tests complètement automatiques, comme par exemple, dans un observatoire, la liste des ressources essentielles peut être déterminée par des méthodes heuristiques. La liste des ressources essentielles doit contenir toutes les ressources suivantes si elles sont applicables :

- **Page d'Accueil:** elle est définie comme la ressource identifiée par une URL seulement constituée d'un composant de protocole HTTP suivi par un nom de domaine. Le mapping de cette URL vers la ressource d'accueil peut être fait par l'intermédiaire d'une redirection côté serveur. Il est nécessaire que cette ressource soit présente sur le site (bien qu'elle puisse avoir un nom différent).
- **Page Contact:** la ressource sur les informations de contact si présente.
- **Page Aide:** la ressource générique pour "l'aide" (si présente).
- **Page Plan du site:** la ressource "plan du site", si présente.
- **Page Recherche:** les ressources comprenant le service de "recherche primaire dans le site" (si présentes). Celle-ci doit inclure au moins une ressource sur laquelle une recherche peut être effectuée et au moins une ressource montrant les résultats d'une recherche type.
- **Pages principales:** exemples de ressources qui sont représentatives de la nature primaire du site et de son utilisation principale (si elles sont identifiables).

- **Pages des scénarios principaux:** si le site propose des services qui impliquent qu'un utilisateur navigue systématiquement dans une suite de ressources (comme par exemple un formulaire ou une transaction sur plusieurs pages), alors les ressources représentatives auxquelles il a accès à chaque étape de chaque scénario clé doivent être incluses (si applicables et seulement si elles sont incluses dans la définition du site Web).
- **Pages avec un aspect différent:** les ressources représentatives de chaque catégorie de ressources ayant un aspect complètement différent (typiquement représentatifs de pages types clés du site) (si identifiables).
- **Pages avec des technologies différentes:** des ressources représentatives de chacune des différentes technologies Web suivantes (lorsqu'elles sont utilisées):
 - formulaires,
 - cadres,
 - tableaux de données,
 - scripts côté client,
 - applets, plug-ins, multimedia, etc.

Naturellement, chacune des ressources peut appartenir à plus d'une des catégories mentionnées ci-dessus : il est simplement nécessaire que toute la liste des ressources essentielles, dans la mesure du possible, traite collectivement tous les objectifs applicables de sélection. Toute ressource donnée doit apparaître seulement une fois dans la liste des ressources essentielles.

4.3.2 Liste des ressources échantillonnées

Une liste de ressources échantillonnées est un ensemble qui peut être généré par un outil automatique à partir d'une sélection donnée. Une liste de ressources échantillonnées pourrait être typiquement utilisée dans le contexte d'évaluations effectuées sur un grand nombre de sites (uniquement selon des critères automatiques) où il n'est pas faisable ou nécessaire d'évaluer la totalité des pages Web pour chaque site [8]. Si une méthode de sélection est utilisée, alors le résultat de la sélection doit être représentatif et sans ambiguïté, ce qui signifie qu'il doit être une sous-liste aléatoire du nombre total de ressources. La liste des ressources échantillonnées pour une évaluation automatique à grande échelle doit donc utiliser un algorithme de sélection qui sélectionne la série des ressources à l'aide d'un algorithme de sélection uniforme aléatoire ou presque uniforme aléatoire [HENZINGER00], ou une liste d'échantillons aléatoires issue de la liste complète des pages Web (à condition que la liste complète des pages Web soit disponible) [9]. La méthode d'agrégation de UWEM de la [section 6](#) s'applique au niveau d'une page Web, c'est pourquoi chaque unité d'échantillon doit ressembler à la série de ressources Web qui constituent ensemble un rendu de page Web.

La marge d'erreur de l'intervalle de confiance de 95% de la valeur moyenne des échantillons agrégés pour un site Web, à l'aide de la méthode d'agrégation UWEM de la [section 6](#), doit être clairement indiquée dans les résultats des tests. Le choix est laissé aux vendeurs d'outils de présenter la marge d'erreurs pour chaque site Web ou d'effectuer une sélection pour une marge d'erreurs donnée [10], afin de présenter une seule fois la marge d'erreurs maximale [11].

La marge d'erreurs m d'un intervalle de confiance est définie comme la valeur ajoutée ou soustraite au moyen de l'échantillon déterminant la longueur de l'intervalle :

$$m = z \frac{\sigma}{\sqrt{n}}$$

Figure 2: marge d'erreurs m d'un intervalle de confiance

Où $z=1.96$ pour un intervalle de confiance de 95% et sigma est la déviation standard des échantillons agrégés pour le site Web à l'aide de la méthode d'agrégation de la [section 6](#).

Il est à noter que l'algorithme de sélection utilisé ainsi que toute autre restriction limitant ou faussant le résultat, incluant, mais non limité à la liste de restrictions citées ci-dessous doit être indiqué explicitement dans tout rapport d'évaluation :

- Restriction à des pages sans interaction de formulaire.
- Restriction par type de contenu Web (comme par exemple pas de Javascript/Flash/PDF).
- Restriction par profondeur de lien de toute ressource essentielle.
- Restriction sur la ressource essentielle spécifique utilisée.
- Restreindre le nombre total de ressources dans l'échantillon.
- Restreindre le nombre total de données de ressources dans l'échantillon.

4.3.3 Taille de l'échantillon dit manuel

En alternative à la méthode décrite ci-dessus, et particulièrement appropriée pour les tests expert, nous autorisons une sélection manuelle du nombre minimum de ressources. Ce nombre minimum de ressources dans la liste de ressources échantillonnées dépend de la taille estimée du site Web.

La taille minimum de l'échantillon est de 30 ressources uniques (si disponibles) en ajoutant 2 ressources uniques pour 1000 ressources distinctes supplémentaires jusqu'à un maximum de 50 ressources dans la liste des ressources sélectionnée. Ce nombre est arbitraire. Des recommandations plus détaillées pour les tailles des échantillons seront ajoutées dans une version future de UWEM et se baseront largement sur les résultats des expériences au sein des projets EIAO et BenToWeb.

5. Liste des tests de conformité WCAG 1.0

Cette section présente la liste des tests UWEM qui sont automatisables par un logiciel (ie, tests automatiques) et ceux nécessitant un expert, sur les points de contrôle de priorité 1 et priorité 2 des [WCAG 1.0](#). Par conséquent, cette section fournit des tests pour une évaluation par un expert et/ou de manière automatique.

La structure des tests dans cette section est la suivante :

1. **Recommandation**

Citation de la recommandation WCAG 1.0 correspondante. Des éléments de clarification supplémentaires peuvent être ajoutés. Pour chaque point de contrôle, une série de un ou plusieurs tests est définie. Si aucun test automatique n'est défini pour une technologie donnée, cela signifie qu'il n'y a pas de test applicable pour une évaluation automatisée.

a) tests basés sur (X)HTML

Série de tests de conformité à effectuer sur des ressources (X)HTML. Chaque test se compose de :

- Un titre et un ID : un titre descriptif court (informatif) et un identifiant unique (normatif).
- Champ/critères d'application : les éléments, les attributs et leurs combinaisons, utilisés pour déterminer le champ d'application du test. Dans la mesure du possible, les critères sont présentés sous la forme d'expressions de type xpath, sinon, une description textuelle est proposée.
- Procédure de test : la description de la procédure du test, indépendamment de tout outil. La procédure peut consister en plusieurs étapes et elle est rédigée de façon à permettre, le cas échéant, l'évaluation par une machine.
- Résultats attendus : les éléments qui définissent les conditions de réussite ou d'échec, par rapport à une ou plusieurs étapes de la procédure de test. Les éléments ou le contenu spécifiés dans les critères d'accessibilité réussissent le test si le résultat n'est pas non validé.
- Entièrement automatisable : indication selon laquelle la procédure de test peut être entièrement automatisée (oui/non).

b) Tests basés sur CSS :

Série de tests de conformité à effectuer sur des ressources CSS. Chaque test se compose de :

- Un titre et un ID : un titre descriptif court (informatif) et un identifiant unique (normatif).
- Champ/critères d'application : les sélecteurs CSS, les propriétés et leurs combinaisons, utilisés pour déterminer (le champ)/les critères d'application du test.
- Procédure de test : la description de la procédure du test, indépendamment de tout outil. La procédure peut consister en plusieurs étapes et elle est rédigée de façon à permettre, le cas échéant, l'évaluation par une machine.
- Résultats attendus : les éléments qui définissent les conditions de réussite ou d'échec, par rapport à une ou plusieurs étapes de la procédure de test. Les éléments ou le contenu spécifiés dans les critères d'accessibilité réussissent le test si le résultat n'est pas non validé.

- Entièrement automatisable : indication selon laquelle la procédure de test peut être entièrement automatisée (oui/non).

c) Tests sur les objets externes :

Série de tests de conformité à effectuer sur des objets inclus ou intégrés dans les pages web, à l'aide d'éléments HTML ou par du contenu généré via les feuilles de style. Ceci comprend les applets, le flash, la vidéo et les composants audio. Chaque test se compose de :

- Un titre et un ID : un titre descriptif court (informatif) et un identifiant unique (normatif).
- Champ/critères d'application : les éléments, les attributs et leurs combinaisons, utilisés pour déterminer le champ d'application du test.
- Procédure de test : description de la procédure du test, indépendamment de tout outil. La procédure peut consister en plusieurs étapes et elle est rédigée de façon à permettre, le cas échéant, l'évaluation par une machine.
- Résultats attendus : les éléments qui définissent les conditions de réussite ou d'échec, par rapport à une ou plusieurs étapes de la procédure de test. Les éléments ou le contenu spécifiés dans les critères d'accessibilité réussissent le test si le résultat n'est pas non validé.
- Entièrement automatisable : indication selon laquelle la procédure de test peut être entièrement automatisée (oui/non).

2. (optionnel) Éléments supplémentaires de clarification, tels que des points de définition.

Cette section ne reproduit/reprend pas les informations disponibles dans les documents du w3c. En revanche, elle fournit des liens vers les pages correspondantes et, seulement lorsque cela s'avère nécessaire, complète les informations pour les tests définis.

Tests UWEM 1.0 pour la recommandation 1 des WCAG 1.0

["Provide equivalent alternatives to auditory and visual content."](#)

Cette recommandation donne des informations sur la façon de donner des alternatives textuelles complémentaires à du contenu audio et visuel.

Sommaire: [Point de contrôle 1.1](#) | [Point de contrôle 1.2](#) | [Point de contrôle 1.3](#) | [Point de contrôle 1.4](#)

Point de contrôle 1.1

Provide a text equivalent for every non-text element (e.g., via "alt", "longdesc", or in element content). This includes: images, graphical representations of text (including symbols), image map regions, animations (e.g., animated GIFs), applets and programmatic objects, art, frames, scripts, images used as list bullets, spacers, graphical buttons, sounds (played with or without user interaction), stand-alone audio files, audio tracks of video, and video. [Priority 1]

(Voir <http://www.w3.org/TR/WCAG10/#tech-text-equivalent> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-text-equivalent>)

Tests basés sur (X)HTML

Test 1.1_HTML_01

Le but de ce test est de vérifier que le contenu non textuel a un équivalent textuel.

- Critères d'application : tous les éléments non textuels qui supportent l'attribut alt.

```
//img
//area
//input[@type='image']
//applet
```

- Procédure de test :
 - vérifier que l'élément a un attribut alt.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : oui.

Test 1.1_HTML_02

Le but de ce test est d'analyser les éléments non textuels ayant une alternative textuelle vide.

- Critères d'application : les éléments non textuels ayant une alternative textuelle vide.

```
//img[@alt='']
//area[@alt='']
```

```
//input[@type='image'][@alt='']  
  
//applet[@alt=''][count(local-name(*)!='param')=0]  
  
//object[count(local-name(*)!='param')=0]
```

- Procédure de test :
 1. Vérifier que l'image/le contenu est purement décoratif.
 2. Si #1 est faux, vérifier qu'une alternative textuelle est adjacente au contenu non textuel.
- Résultats attendus : validé si #1 ou #2 est vrai. Non validé si #1 ou #2 est faux.
- Entièrement automatisable : non.

Test 1.1_HTML_03

Le but de ce test est d'analyser les éléments non textuels ayant une alternative textuelle non vide.

- Critères d'application : tous les éléments non textuels ayant une alternative textuelle non vide.

```
//img[@alt][@alt!='']  
  
//area[@alt][@alt!='']  
  
//input[@type='image'][@alt][@alt!='']  
  
//applet[@alt][@alt!='']  
  
//object[count(local-name(*)!='param')>0]
```

- Procédure de test :
 - vérifier que l'alternative textuelle [\[12\]](#) représente la fonction de l'élément non textuel de manière appropriée dans son contexte.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

Test 1.1_HTML_04

Le but de ce test est d'analyser les descriptions longues d'éléments media.

- Critères d'application : toutes les descriptions longues des images et des éléments media.

```
//img/@longdesc  
  
//object//a/@href
```

- Procédure de test :
 1. Vérifier que la description longue référencée par l'attribut longdesc ou href est présente.
 2. Vérifier qu'elle décrit l'élément de façon appropriée.
- Résultats attendus : non validé si #1 ou #2 est faux.
- Entièrement automatisable : non.

Test 1.1_HTML_05

Le but de ce test est de rechercher les images complexes et le contenu non textuel nécessitant une description longue.

- Critères d'application : tous les éléments `img` et `object`.

```
//img[not(@longdesc)]
```

```
//object[not(../a/@href)]
```

- Procédure de test :
 1. Vérifier qu'il ne nécessite pas une description longue.
 2. Si #1 est faux vérifier qu'une description longue est référencée soit par l'attribut `longdesc` pour les éléments `img` soit qu'elle est fournie via un élément fils pour les éléments `object` (y compris un élément a référençant une description longue). Noter que les éléments fils de l'élément `object` peuvent contenir un lien vers une description longue.
- Résultats attendus : Validé si #1 ou #2 est vrai. Non validé si #1 ou #2 est faux.
- Entièrement automatisable : non.

Test 1.1_HTML_06

Ce test concerne tout particulièrement le contenu non textuel intégré à l'aide de l'élément non standard `embed`.

Puisqu'il n'existe pas de méthode définie pour fournir des alternatives à `embed`, `embed` est en soi inaccessible.

- Critères d'application : tous les éléments `embed`.

```
//embed
```

- Procédure de test :
 - sélectionner les éléments.
- Résultats attendus : non validé si vrai.
- Entièrement automatisable : oui.

Test 1.1_HTML_07

Le but de ce test est de vérifier les alternatives textuelles d'un contenu non textuel chargé par un cadre en ligne.

- Critères d'application : tous les éléments `iframe`.

```
//iframe
```

- Procédure de test :
 1. Vérifier si l'élément charge un contenu non textuel.
 2. Si #1 est vrai, vérifier que l'élément contient une alternative textuelle ou un lien vers une alternative textuelle à l'élément non textuel.

Noter que le contenu de l'iframe ne peut pas être mis à jour lorsque le contenu chargé par l'iframe change suite à une interaction de l'utilisateur ou l'exécution d'un script.

- Résultats attendus : Validé si #2 est vrai. Non validé si #2 est faux.
- Entièrement automatisable : non.

Test 1.1_HTML_08

Le but de ce test est de vérifier les cadres chargeant directement du contenu non textuel.

- Critères d'application : tous les éléments frame.

```
//frame
```

- Procédure de test :
 - vérifier si l'élément charge directement du contenu non textuel.
- Résultats attendus : non validé si vrai.
- Entièrement automatisable : non.

Test 1.1_HTML_09

Le but de ce test est de rechercher les composants en audio seul intégrés ou en lien qui n'ont pas de transcription textuelle.

- Critères d'application : tous les composants en audio seul.

```
//object
```

```
//applet
```

```
//a
```

- Procédure de test :
 - vérifier s'il existe une transcription textuelle du composant en audio seul.
- Résultats attendus : validé si vrai.
- Entièrement automatisable : non.

Test 1.1_HTML_10

Le but de ce test est d'analyser les transcriptions textuelles des composants en audio seul intégrés ou en lien.

- Critères d'application : tous les composants en audio seul ayant une transcription.

```
//object
```

```
//applet
```

```
//a
```

- Procédure de test :

- vérifier que la transcription textuelle décrit complètement toutes les informations importantes des pistes audio du composant en audio seul incluant les mots prononcés, les sons non prononcés tels que les effets sonores.
- Résultats attendus : validé si vrai.
- Entièrement automatisable : non.

Tests sur les objets externes

Test 1.1_external_01

Le but de ce test est de rechercher les présentations multimédia intégrées ou en lien qui n'ont pas de sous-titres associés.

- Critères d'application : toutes les présentations multimédia ayant au moins une piste audio et au moins une piste vidéo.

//object

//applet

//a

- Procédure de test :
 - vérifier que tous les composants applicables ont un sous-titre qui leur est associé.
- Résultats attendus : validé si vrai.
- Entièrement automatisable : non.

Test 1.1_external_02

le but de ce test est d'analyser les sous-titres associées à des présentations multimedia.

- Critères d'application : toutes les présentations multimédia ayant au moins une piste audio et au moins une piste vidéo.

//object

//applet

//a

- Procédure de test :
 - vérifier que les sous-titres associés restituent entièrement toutes les informations importantes de la /des piste(s) audio de la présentation multimedia, y compris les mots prononcés, les sons non parlés tels que les effets sonores.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

Point de contrôle 1.2

Provide redundant text links for each active region of a server-side image map. [Priority 1]

(Voir <http://www.w3.org/TR/WCAG10/#tech-redundant-server-links> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-redundant-server-links>)

Tests basés sur (X)HTML

Test 1.2_HTML_01

Le but de ce test est de rechercher les zones actives d'une image map côté serveur n'ayant pas de liens textuels redondants.

- Critères d'application : toutes les images maps côté serveur.

```
//img[@ismap]
```

```
//input[@type='image'][@ismap]
```

- Procédure de test :
 1. Identifier toutes les zones actives de l'image map.
 2. Vérifier qu'il existe un lien textuel redondant pour chaque zone active.
- Résultats attendus : non validé si #2 est faux (pour au moins une zone active).
- Entièrement automatisable : non.

Point de contrôle 1.3

Until user agents can automatically read aloud the text equivalent of a visual track, provide an auditory description of the important information of the visual track of a multimedia presentation. [Priority 1]

(Voir <http://www.w3.org/TR/WCAG10/#tech-auditory-descriptions> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-auditory-descriptions>)

Tests sur les objets externes

Test 1.3_external_01

Le but de ce test est de rechercher les présentations multimedia n'ayant pas de description audio concernant les informations importantes de leur piste vidéo.

- Critères d'application : toutes les présentations multimedia ayant au moins une piste vidéo.

```
//object
```

```
//applet
```

```
//a
```

- Procédure de test :
 - vérifier qu'il existe une description audio pour les informations importantes de la piste vidéo.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

Test 1.3_external_02

Le but de ce test est d'analyser la description audio des présentations multimédia en lien et intégrées.

- Critères d'application : toutes les présentations multimedia ayant au moins une piste vidéo.

```
//object
```

```
//applet
```

```
//a
```

- Procédure de test :
 - vérifier que la description audio restitue effectivement tous les éléments visuels importants de la présentation y compris les informations sur les acteurs, les actions, la langue corporelle, les graphiques et les changements de scène.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

Point de contrôle 1.4

For any time-based multimedia presentation (e.g., a movie or animation), synchronize equivalent alternatives (e.g., captions or auditory descriptions of the visual track) with the presentation. [Priority 1]

(Voir <http://www.w3.org/TR/WCAG10/#tech-synchronize-equivalents> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-synchronize-equivalents>)

Tests sur les objets externes

Test 1.4_external_01

Ce test a pour but de vérifier la synchronisation d'alternatives équivalentes aux présentations multimedia.

- Critères d'application : toutes les présentations multimedia ayant des alternatives équivalentes.

```
//a
```

```
//applet
```

//object

- Procédure de test :
 - vérifier que l'alternative équivalente (sous-titre, description audio ou toute alternative équivalente, si applicable) est synchronisée avec la présentation.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

Tests UWEM 1.0 pour la recommandation 2 des WCAG 1.0

["Don't rely on color alone."](#)

Cette recommandation donne des informations sur la façon d'utiliser des couleurs de manière appropriée.

Sommaire: [Point de contrôle 2.1](#) | [Point de contrôle 2.2](#)

Point de contrôle 2.1

Ensure that all information conveyed with color is also available without color, for example from context or markup. [Priority 1]

(Voir <http://www.w3.org/TR/WCAG10/#tech-color-convey> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-color-convey>)

Tests basés sur (X)HTML

Test 2.1_HTML_01

Le but de ce test est de rechercher des expressions dans un texte qui font référence aux parties d'un document en mentionnant uniquement leur couleur.

- Critères d'application : tout le texte.

```
//body
```

- Procédure de test :
 1. Vérifier que le texte ne fait pas référence à des parties d'un document en mentionnant uniquement leur couleur.
 2. Si #1 est faux, vérifier que les références dans le texte par l'intermédiaire de la couleur sont redondantes.
- Résultats attendus : validé si #1 ou #2 est vrai. Non validé si #1 ou #2 sont faux.
- Entièrement automatisable : non.

Test 2.1_HTML_02

Le but de ce test est de rechercher des expressions dans un contenu non textuel qui font référence aux parties d'un document en mentionnant uniquement leur couleur.

- Critères d'application : tout le texte d'un contenu non textuel.

```
//img
```

```
//area
```

```
//input[@type='image']
```

```
//applet
```

```
//object
```

- Procédure de test :
 1. Vérifier que le texte ne fait pas référence à des parties d'un document en mentionnant uniquement leur couleur.
 2. Si #1 est faux, vérifier que les références dans le texte par l'intermédiaire de la couleur sont redondantes.
- Résultats attendus : validé si #1 ou #2 est vrai. Non validé si #1 ou #2 sont faux.
- Entièrement automatisable : non.

Test 2.1_HTML_03

Le but de ce test est de rechercher les éléments en couleur n'ayant pas de méthodes redondantes pour restituer l'information.

- Critères d'application : tous les éléments HTML en couleur.

```
/**/@color
```

```
/**/@bgcolor
```

```
/**/@text
```

- Procédure de test :
 - vérifier que l'information donnée par la couleur est redondante.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

Tests basés sur CSS

Test 2.1_CSS_01

Le but de ce test est de rechercher les éléments en couleur n'ayant pas de méthodes redondantes pour restituer l'information.

- Critères d'application : tout le contenu en couleur produit par une CSS.

```
color
```

```
background-color
```

```
background
```

```
border-color
```

```
border
```

```
outline-color
```

```
outline
```

- Procédure de test :
 - vérifier que l'information donnée par la couleur est redondante.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

Point de contrôle 2.2

Ensure that foreground and background color combinations provide sufficient contrast when viewed by someone having color deficits or when viewed on a black and white screen. [Priority 2 for images, Priority 3 for text].

(Voir <http://www.w3.org/TR/WCAG10/#tech-color-contrast> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-color-contrast>)

Tests basés sur (X)HTML

Test 2.2_HTML_01

Le but de ce test est de rechercher des images ayant des contrastes de couleurs insuffisants.

- Critères d'application : toutes les images :

```
//img
//area
//input[@type='image']
//object
```

- Procédure de test :
 - vérifier que le contraste entre la couleur de premier plan et d'arrière-plan est suffisant pour donner l'information.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : oui (si l'algorithme des contrastes de couleurs est disponible [\[13\]](#)).

Tests basés sur CSS

Test 2.2_CSS_01

Le but de ce test est de rechercher du texte ayant des contrastes de couleurs insuffisants.

- Critères d'application : les images référencées à partir de styles CSS.

```
background-image
background
```

content

cursor

list-style-image [\[14\]](#)

- Procédure de test :
 - vérifier que le contraste entre la couleur de premier plan et d'arrière-plan est suffisant pour donner l'information.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : oui (si l'algorithme des contrastes de couleurs est disponible).

Tests UWEM 1.0 pour la recommandation 3 des WCAG 1.0

["Use markup and style sheets and do so properly."](#)

Sommaire: [Point de contrôle 3.1](#) | [Point de contrôle 3.2](#) | [Point de contrôle 3.3](#) | [Point de contrôle 3.4](#) | [Point de contrôle 3.5](#) | [Point de contrôle 3.6](#) | [Point de contrôle 3.7](#)

Point de contrôle 3.1

When an appropriate markup language exists, use markup rather than images to convey information. [Priority 2]

(Voir <http://www.w3.org/TR/WCAG10/wai-pageauth.html#tech-use-markup> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-use-markup>)

Tests basés sur (X)HTML

Test 3.1_HTML_01

Le but de ce test est de vérifier qu'il n'y a aucune image contenant du texte qui pourrait être remplacée par du texte stylé.

- Critères d'application : toutes les images.

```
//img
```

```
//input[@type='image']
```

```
//object
```

- Procédure de test :
 1. Vérifier si l'image contient du texte.
 2. Si #1 est vrai, vérifier que l'image peut être remplacée par du texte stylé sans perdre les informations véhiculées par l'image.
- Résultats attendus : non validé si #2 est vrai.
- Entièrement automatisable : non.

Test 3.1_HTML_02

Le but de ce test est de vérifier qu'il n'y a pas d'images d'équations mathématiques qui pourraient être remplacées par du texte stylé.

- Critères d'application : toutes les images.

```
//img
```

```
//input[@type='image']
```

```
//object
```

- Procédure de test :

1. Vérifier si l'image contient une équation mathématique.
 2. Si oui, vérifier si l'image peut être remplacée par du texte stylé.
- Résultats attendus : non validé si #2 est vrai.
 - Entièrement automatisable : non.

Test 3.1_HTML_03

Le but de ce test est de vérifier qu'il n'y a pas d'images de type bitmap qui ne contiennent pas de texte ou d'équations mathématiques et peuvent être remplacées par du code.

- Critères d'application : toutes les images de type bitmap qui ne contiennent pas de texte ou d'équations mathématiques.

```
//img
```

```
//input[@type='image']
```

```
//object
```

- Procédure de test :
 - vérifier qu'il n'existe aucun langage de balises approprié capable de restituer les informations contenues dans l'image.
- Résultats attendus : validé si vrai.
- Entièrement automatisable : non.

Point de contrôle 3.2

Create documents that validate to published formal grammars. [Priority 2]

(Voir <http://www.w3.org/TR/wcag10/#tech-identify-grammar> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-identify-grammar>)

Tests basés sur (X)HTML

Test 3.2_HTML_01

Le but de ce test est de vérifier que le document contient une DTD valide.

Note : Le groupe assurance qualité du W3C édite un document intitulé "[Recommended DTDs to use in your Web document](#)" (traduction : les DTDs qu'il est recommandé d'utiliser dans vos documents Web).

- Critères d'application : contenu précédant l'élément HTML de tout document HTML 4.x ou XHTML 1.0.
- Procédure de test :
 - vérifier que la déclaration du doctype est valide [\[15\]](#).
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : oui.

Test 3.2_HTML_02

Le but de ce test est de rechercher les violations des schémas formels pour le HTML 4.x ou le XML 1.0.

- Critères d'application : tout document HTML 4.x ou XHTML 1.0.
- Procédure de test :
 - a) Pour le HTML vérifier que le document est valide selon le type de document spécifié à l'aide d'un outil de validation SGML.
 - b) Pour le XHTML, vérifier que le document est bien constitué et qu'il est valide selon le type de document spécifié à l'aide d'un outil de validation XML.
- Résultats attendus : non validé si faux.
- Entièrement automatisable : oui.

Tests basés sur CSS

Test 3.2_CSS_01

Le but de ce test est de vérifier les violations des grammaires formelles pour CSS 1.0 ou CSS 2.x.

- Critères d'application : toutes les règles de style CSS.
- Procédure de test :
 - a) Pour les règles de styles à l'intérieur de l'élément style ou dans les attributs style d'un fichier (X)HTML : vérifier qu'ils sont conformes à la grammaire formelle définie à <http://www.w3.org/TR/REC-CSS2/grammar.html> avec un outil de validation SAC.
 - b) pour les fichiers CSS : vérifier que valider chaque fichier CSS à l'aide d'un outil de validation SAC n'entraîne pas d'erreurs.

noter que le validateur CSS du W3C fait plus que vérifier les règles de CSS selon les grammaires formelles : il vérifie aussi les propriétés (non) définies ainsi que les valeurs qui ne sont pas incluses dans la grammaire. La grammaire ne définit pas le "vocabulaire" actuel des CSS.

- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : oui.

Point de contrôle 3.3

Use style sheets to control layout and presentation. [Priority 2]

(Voir <http://www.w3.org/TR/wcag10/#tech-style-sheets> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-style-sheets>)

Tests basés sur (X)HTML

Test 3.3_HTML_01

Le but de ce test est de rechercher les espaces utilisés pour contrôler l'espacement entre les caractères des mots.

Note : il n'y a pas de définition indépendante d'une langue de l'expression "mot", c'est pourquoi les évaluateurs doivent vérifier si l'expression "mot" est applicable à la langue du contenu qu'ils évaluent, et si c'est le cas, ils doivent s'assurer qu'ils comprennent ce que l'expression "mot" veut dire dans la langue du contenu qu'ils évaluent.

- Critères d'application : tout "mot" contenant un espace.

```
text()
```

- Procédure de test :
 - vérifier que l'espace n'est pas utilisé pour mettre en valeur.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

Test 3.3_HTML_02

Le but de ce test est de déterminer si la mise en forme ou la présentation d'un ou de plusieurs éléments a été effectuée par l'intermédiaire d'autres moyens que les CSS.

- Critères d'application : éléments et attributs pouvant être utilisés pour positionner ou influencer la présentation.

```
//img  
  
//font  
  
//td (dans un tableau de données)  
  
//th (dans un tableau de données)  
  
//center  
  
//u  
  
//b  
  
//i  
  
//blink  
  
//strong (à moins qu'il soit utilisé au niveau sémantique)  
  
//em (à moins qu'il soit utilisé au niveau sémantique)  
  
/**/@align  
  
/**/@border  
  
/**/@hspace  
  
/**/@vspace
```

```
/**/@bgcolor
```

- Procédure de test :
 - vérifier que le positionnement et/ou la présentation obtenues n'ont pas pu être effectuées à l'aide des feuilles de style.
- Résultats attendus : non validé si faux.
- Entièrement automatisable : non.

Point de contrôle 3.4

Use relative rather than absolute units in markup language attribute values and style sheet property values. [Priority 2]

(Voir <http://www.w3.org/TR/WCAG10/wai-pageauth.html#tech-relative-units> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-relative-units>).

Tests basés sur (X)HTML

Test 3.4_HTML_01

Le but de ce test est de vérifier les valeurs relatives des attributs (X)HTML de type %Length;

- Critères d'application : les attributs spécifiant la hauteur, la largeur, les marges de cellules, l'espacement entre les cellules, le dimensionnement des caractères en nombre de pixels ou en pourcentage.

```
//table/@cellpadding
```

```
//table/@cellspacing
```

```
//col/@charoff
```

```
//colgroup/@charoff
```

```
//tbody/@charoff
```

```
//td/@charoff
```

```
//tfoot/@charoff
```

```
//th/@charoff
```

```
//thead/@charoff
```

```
//tr/@charoff
```

```
//iframe/@height
```

```
//td/@height
```

```
//th/@height
```

```
//img/@height
//object/@height
//applet/@height
//hr/@width
//iframe/@width
//img/@width
//object/@width
//table/@width
//td/@width
//th/@width
//applet/@width
```

- Procédure de test :
 - vérifier que la valeur de l'attribut est une valeur en pourcentage (entier positif + '%') ou que la valeur absolue ne fait pas d'interférence avec la lisibilité d'autres éléments du texte.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : oui.

Test 3.4_HTML_02

Le but de ce test est de vérifier les valeurs relatives des attributs X(HTML) de type multi-length [\[16\]](#) ("%MultiLength;" dans la DTD HTML 4.01).

- Critères d'application : les attributs qui spécifient la largeur des colonnes ou de groupes de colonnes.

```
//col/@width
//colgroup/@width
```

- Procédure de test :
 - vérifier que la valeur de l'attribut est une valeur en pourcentage (entier positif + '%') ou une valeur * (astérisque) ou que la valeur absolue ne fait pas d'interférence avec la lisibilité d'autres éléments de texte.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

Test 3.4_HTML_03

Le but de ce test est de vérifier les valeurs relatives d'attributs X(HTML) de type multi-length-list [\[17\]](#) ou ("%MultiLengths;" dans la DTD HTML 4.01: une liste de MultiLength séparée par des virgules).

- Critères d'application : des attributs qui spécifient une liste de lengths en pixels, un pourcentage, ou une valeur relative.

```
//frameset/@cols
```

```
//frameset/@rows
```

- Procédure de test :
 - vérifier que chaque valeur listée dans l'attribut est une valeur en pourcentage (entier positif + '%') ou une valeur * (astérisque) ou que la valeur absolue ne fait pas d'interférence avec la lisibilité d'autres éléments du texte.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : oui.

Tests basés sur CSS

Test 3.4_CSS_01

Le but de ce test est de vérifier les unités en valeur relative des propriétés des feuilles de style pouvant contenir des valeurs <length>.

- Critères d'application : Les propriétés CSS spécifiant la longueur, la largeur, la hauteur, la taille, l'espacement ou les limites de positionnement en arrière-plan.

```
background-position
```

```
border-spacing
```

```
bottom
```

```
font-size
```

```
height
```

```
left
```

```
letter-spacing
```

```
line-height
```

```
marker-offset
```

```
max-height
```

```
max-width
```

```
min-height
```

```
min-width
```

```
right
```

```
size
```

```
text-indent
```

text-shadow
top
vertical-align
width
word-spacing

- Procédure de test :
 1. Vérifier que l'unité de valeur n'est pas cm, mm, in, pt, pc ou px [18]
 2. Vérifier que la valeur n'est pas xx-small, x-small, small, medium, large, x-large ou xx-large [19].
 3. Si une valeur absolue est utilisée, vérifier que la valeur absolue ne fait pas d'interférence avec la lisibilité de tout élément du texte.
- Résultats attendus : validé si #1 et #2 sont vrais ou #3 est vrai.
- Entièrement automatisable : oui.

Point de contrôle 3.5

Use header elements to convey document structure and use them according to specification.
[Priority 2]

(Voir <http://www.w3.org/TR/WCAG10/wai-pageauth.html#tech-logical-headings> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-logical-headings>)

Tests basés sur (X)HTML

Test 3.5_HTML_01

Le but de ce test est de rechercher des effets de style qui, dans leur concept, représentent des titres mais qui ne sont pas codés par des éléments hX.

- Critères d'application : le corps d'une page Web.

```
//body//*
```

- Procédure de test :
 1. Sélectionner des effets de style qui, dans leur concept, représentent des titres.
 2. Vérifier si les titres sont codés à l'aide des éléments hX.
Conseil : sont sujets à un codage insuffisant par exemple les combinaisons pour les changements de font-weight/font-style (les éléments HTML b et i ; les propriétés CSS font-weight, font-style), et les agrandissements des tailles de polices (les éléments HTML big, font ; la propriété CSS font-size). Cette liste n'est pas exhaustive.
- Résultats attendus : validé si vrai.
- Entièrement automatisable : non.

Test 3.5_HTML_02

Le but de ce test est de vérifier qu'il n'y a pas d'élément de titre dans la page ayant un niveau plus élevé que le premier titre.

- Critères d'application : tous les éléments de titre sauf h6[\[20\]](#).

```
//h1
```

```
//h2
```

```
//h3
```

```
//h4
```

```
//h5
```

- Procédure de test :
 - vérifier que l'élément de titre n'a pas un niveau plus élevé que le premier élément de titre du document.
- Résultats attendus : validé si vrai.
- Entièrement automatisable : oui.

Test 3.5_HTML_03

Le but de ce test est de vérifier qu'aucun niveau n'a été omis dans la hiérarchie des titres.

- Critères d'application : tous les éléments de titre sauf h1 et h2[\[21\]](#).

```
//h3
```

```
//h4
```

```
//h5
```

```
//h6
```

- Procédure de test :
 - vérifier que l'élément de titre examiné ne passe pas un ou plusieurs niveaux dans la structure. (c'est-à-dire vérifier que pour h5 l'élément de titre précédent est soit h4, h5 ou h6).
- Résultats attendus : validé si vrai.
- Entièrement automatisable : oui.

Test 3.5_HTML_04

Ce test permet de vérifier si des éléments de titre ont été utilisés (de façon incorrecte) pour le formatage de polices.

- Critères d'application : tous les éléments de titre (h1, ..., h6)

```
//h1
```

//h2

//h3

//h4

//h5

//h6

- Procédure de test :
 - vérifier que les titres n'ont pas été utilisés pour créer des effets de formatage de polices.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

Test 3.5_HTML_05

Le but de ce test est de vérifier la bonne hiérarchie des niveaux de titres.

- Critères d'application : tout le document.

//body

- Procédure de test :
 - vérifier que les éléments de titres restituent la structure logique du document.
- Résultats attendus : validé si vrai.
- Entièrement automatisable : non.

Point de contrôle 3.6

Mark up lists and list items properly. [Priority 2]

(Voir <http://www.w3.org/TR/WCAG10/#tech-list-structure> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-list-structure>)

Coder les structures en listes et les éléments de listes (UL, OL, DL, LI) de façon appropriée. Les éléments de liste HTML DL, UL, et OL (disponibles en HTML 3.2 et HTML 4.0) ne doivent être uniquement utilisés que pour créer des listes et non pour des effets de formatage tels que l'indentation. Si possible utiliser des listes ordonnées (numérotées) pour faciliter la navigation.

Tests basés sur (X)HTML

Test 3.6_HTML_01

Les auteurs peuvent désactiver le style de listes par défaut dans une liste ordonnée et non ordonnée et créer manuellement une numérotation à plusieurs niveaux (par exemple 1, 1.1, 1.2, 1.2.1). Le but de ce test est de vérifier qu'une numérotation de liste ajoutée manuellement fournit aux utilisateurs la profondeur de la liste.

- Critères d'application : toutes les listes ordonnées et non ordonnées imbriquées ayant une numérotation à plusieurs niveaux insérée manuellement.

```
//li/ol
```

```
//li/ul
```

- Procédure de test :
 - vérifier que la numérotation ne saute pas de niveaux ou des nombres.
- Résultats attendus : validé si vrai.
- Entièrement automatisable : non.

Test 3.6_HTML_02

Ce test a pour but de vérifier si des éléments de listes (li) sont appropriés au contexte du document, c'est-à-dire pour créer des listes et non pour le formatage comme l'indentation.

- Critères d'application : tous les éléments de listes y compris les définitions dans les listes de définitions.

```
//ul/li
```

```
//ol/li
```

```
//dl/dd
```

- Procédure de test :
 - vérifier que chaque élément li ou dd est utilisé pour coder des éléments de listes et non à des fins de formatage.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

Test 3.6_HTML_03

Le but de ce test est de rechercher les paragraphes, les retours à la ligne et les chiffres utilisés pour simuler des listes numérotées et qui peuvent être remplacés par l'élément ol.

- Critères d'application : tous les paragraphes commençant par un énumérateur (nombre ou caractère indiquant un ordre ou une séquence).

```
//p
```

```
//p//br
```

- Procédure de test :
 1. Vérifier que le document ne contient pas de séquences de paragraphes commençant par des énumérateurs afin de simuler des listes numérotées.
 2. Vérifier que le document ne contient pas de paragraphes avec des retours à la ligne suivis d'énumérateurs afin de simuler des listes numérotées.
- Résultats attendus : validé si #1 et #2 sont vrais.
- Entièrement automatisable : oui.

Test 3.6_HTML_04

Ce test a pour objectif de rechercher les paragraphes, les retours à la ligne et certains caractères tels que les astérisques et les tirets utilisés pour simuler des listes non ordonnées et qui peuvent être remplacés par l'élément ul.

- Critères d'application : tous les paragraphes commençant par des caractères pouvant être utilisés pour simuler des éléments de listes.

```
//p
```

```
//p//br
```

- Procédure de test :
 1. Vérifier que le document ne contient pas de séquences de paragraphes commençant par des caractères tels que astérisque ou tiret pour simuler des listes non ordonnées.
 2. Vérifier que le document ne contient pas de paragraphes avec des retours à la ligne suivis par des caractères tels que astérisque ou tiret pour simuler des listes non ordonnées.
- Résultats attendus : validé si #1 et #2 sont vrais. Non validé si #1 ou #2 sont faux.
- Entièrement automatisable : oui.

Test 3.6_HTML_05

Le but de ce test est de rechercher les paragraphes, les retours à la ligne et les images affichant des nombres utilisés pour simuler des listes ordonnées et pouvant être remplacés par l'élément ol et les CSS.

- Critères d'application : tous les paragraphes commençant par des images affichant un nombre ou d'autres types d'énumérateurs.

```
//p//img
```

```
//p//br/following-sibling::img
```

- Procédure de test :
 1. Vérifier que le document ne contient pas de séquences de paragraphes commençant par des images affichant des nombres ou autres types d'énumérateurs pour simuler des listes ordonnées.
 2. Vérifier que le document ne contient pas de paragraphes avec des retours à la ligne suivis par des images de chiffres consécutifs ou d'autres types d'énumérateurs pour simuler des listes ordonnées.
- Résultats attendus : validé si #1 et #2 sont vrais. Non validé si #1 ou #2 sont faux.18:11
21/08/2006
- Entièrement automatisable : non.

Test 3.6_HTML_06

Le but de ce test est de rechercher des paragraphes, des retours à la ligne et des images (en particulier les images de puces) utilisés pour simuler des listes non ordonnées et qui peuvent être remplacés par l'élément ul et des CSS.

- Critères d'application : tous les paragraphes commençant par des images de puces.

```
/p//img
```

```
//p//br/following-sibling::img
```

- Procédure de test :
 1. Vérifier que le document ne contient pas de séquences de paragraphes commençant par des images de puces pour simuler des listes non ordonnées.
 2. Vérifier que le document ne contient pas de paragraphes avec des retours à la ligne suivis par des images de puces pour simuler des listes non ordonnées.
- Résultats attendus : validé si #1 et #2 sont vrais. Non validé si #1 ou #2 sont faux.
- Entièrement automatisable : non.

Test 3.6_HTML_07

Le but de ce test est de rechercher les paragraphes, les retours à la ligne et les effets de formatage utilisés pour simuler des listes de définition et pouvant être remplacés par les éléments dt et dd.

- Critères d'application : tous les paragraphes commençant par un terme suivi d'une définition.

```
//p
```

```
//p//br
```

- Procédure de test :
 - vérifier que le document ne contient pas de paragraphes devant être remplacés par une liste de définition.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

Tests basés sur CSS

Test 3.6_CSS_01

Le but de ce test est de vérifier qu'un style de liste de secours est présent si les images sont utilisées en tant que puces de listes.

- Critères d'application : toutes les propriétés list-style.

```
*{list-style:...;},
```

```
*{list-style-image:url(...);},
```

```
*{list-style-type:...;}
```

- Procédure de test :

- vérifier qu'un style de puce de secours (comme par exemple 'disc') est spécifié dans le cas où l'image d'une puce ne peut pas être chargée.
- Résultats attendus : validé si vrai.
- Entièrement automatisable : oui.

Point de contrôle 3.7

Mark up quotations. Do not use quotation markup for formatting effects such as indentation.
[Priority 2]

(Voir <http://www.w3.org/TR/WCAG10/#tech-quotes> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-quotes>)

Tests basés sur (X)HTML

Test 3.7_HTML_01

L'objectif de ce test est de vérifier que les éléments de citation sont utilisés de manière appropriée pour coder des citations et non pour des effets de formatage et d'indentation.

- Critères d'application : tous les éléments blockquote.

```
//blockquote
```

- Procédure de test :
 - vérifier que blockquote est utilisé pour coder une citation.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

Test 3.7_HTML_02

Ce test a pour objectif de vérifier que les citations courtes (élément q) sont utilisées de manière appropriée pour des citation et non pour des raisons de mise en forme.

- Critères d'application : tous les éléments q.

```
//q
```

- Procédure de test :
 - vérifier que l'élément q est utilisé pour coder une citation.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

Test 3.7_HTML_03

Le but de ce test est de rechercher les citations qui n'ont pas été codées avec q ou blockquote.

- Critères d'application : tout le texte.

//p

- Procédure de test :
 1. Y a-t-il des citations dans les paragraphes sélectionnés ? C'est-à-dire des passages contenant des marques de citation dans le code ou qui ont des marques de citations générées par CSS ?
 2. Si oui, vérifier que les citations sont codées à l'aide de q ou blockquote.
- Résultats attendus : validé si #2 est vrai. Non validé si #2 est faux.
- Entièrement automatisable : non.

Test 3.7_HTML_04

Le but de ce test est de rechercher tout élément cite et address utilisés pour mettre du texte en italique.

- Critères d'application : tous les éléments cite et address.

//cite

//address

- Procédure de test :
 1. Sélectionner tous les éléments cite et address.
 2. Déterminer s'ils sont utilisés pour mettre du texte en italique au lieu de respectivement coder une citation ou donner des informations sur l'auteur du document.
- Résultats attendus : non validé si #2 est vrai.
- Entièrement automatisable : non.

Tests UWEM 1.0 pour la recommandation 4 des WCAG 1.0

"[Clarify natural language usage.](#)"

Cette recommandation donne des informations sur la manière de faciliter la prononciation ou l'interprétation de texte abrégé ou en langue étrangère.

Sommaire: [Point de contrôle 4.1](#)

Point de contrôle 4.1

Clearly identify changes in the natural language of a document's text and any text equivalents (e.g., captions). [Priority 1]

(Voir <http://www.w3.org/TR/wcag10/#tech-identify-changes> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-identify-changes>)

Tests basés sur (X)HTML

Test 4.1_HTML_01

Le but de ce test est d'identifier les changements dans la langue naturelle qui ne sont pas indiqués par le code.

- Critères d'application : tous les éléments contenant du texte.

```
//*[true(text())]
```

- Procédure de test :
 1. Sélectionner les éléments commençant au niveau le moins élevé (les "feuilles" dans la structure de l'arbre), et remonter vers les éléments parents en vérifiant chaque élément du document.
 2. Pour chaque segment de texte [\[22\]](#), de l'élément courant, déterminer la langue naturelle.
 3. Déterminer si le segment contient des noms propres, appelés noms d'emprunt, - mots et expressions qui sont devenus de facto internationaux, transcendant beaucoup de langues naturelles spécifiques - etc. qui pourraient être exemptés du point de contrôle 4.1.
 4. Vérifier que la langue naturelle trouvée dans #2 est la même pour le reste du contenu de l'élément courant.
 5. Déterminer la langue naturelle du texte de l'élément parent. (si l'élément parent est l'élément html, sa langue est définie soit par l'attribut lang ou à partir du contexte de la page).
 6. Si deux langues naturelles différentes sont trouvées dans #5, vérifier que l'élément (courant) a un attribut lang identifiant la langue naturelle à l'aide du code à deux lettres correspondant tel qu'il est défini dans ISO639[\[23\]](#).
- Résultats attendus : validé si #3 ou #4, #5 et #6 sont vrais. Non validé si #4, #5 ou #6 sont faux.
- Entièrement automatisable : non.

Test 4.1_HTML_02

Le but de ce test est de trouver les changements dans la langue naturelle qui ne sont pas indiqués par le code.

- Critères d'application : tous les attributs spécifiant une alternative textuelle, les informations obligatoires (l'attribut title), un résumé de tableau, un label dans un menu hiérarchique (l'attribut label), un texte en attente (l'attribut standby), ou tout autre contenu textuel.

```
//img/@alt  
  
//applet/@alt  
  
//area/@alt  
  
//input/@alt  
  
//meta/@content  
  
//option/@label  
  
//optgroup/@label  
  
//object/@standby  
  
//table/@summary  
  
//*/@title[24]  
  
//input[@type='text']/@value  
  
//input[@type='submit']/@value  
  
//frame/@name  
  
//iframe/@name
```

- Procédure de test :
 1. Déterminer la langue naturelle du texte dans l'attribut.
 2. Déterminer la langue naturelle du texte dans l'ancêtre le plus près de l'élément parent de l'attribut.
 3. Si les langues dans les étapes 1 et 2 sont différentes, vérifier que l'élément parent de l'attribut a un attribut lang identifiant la langue naturelle à l'aide du code à deux lettres défini dans ISO639.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

Test 4.1_HTML_03

Le but de ce test est de trouver les éléments de niveau bloc qui ont des changements de direction de texte (pour les langues naturelles) qui ne sont pas indiqués dans le code.

- Critères d'application : tous les éléments de niveau bloc contenant du texte[25].

```
/**[true(text())]
```

- Procédure de test :
 1. Déterminer la direction du texte dans l'élément (courant).
 2. Si l'élément courant est l'élément html ou body et la direction du texte va de la droite vers la gauche, vérifier que l'élément a un attribut dir avec la valeur "rtl".
 3. Si l'élément courant n'est pas l'élément html, déterminer la direction du texte dans l'élément parent.
 4. Si deux directions différentes ont été trouvées à l'étape 3, vérifier que l'élément courant a un attribut dir identifiant la direction du texte à l'aide de la valeur correspondante (rtl ou ltr).
- Résultats attendus : validé si #2 et #4 sont vrais. Non validé si #2 ou #4 est faux.
- Entièrement automatisable : non.

Test 4.1_HTML_04

Ce test a pour objectif de trouver les changements dans la direction du texte (pour les langues naturelles) qui ne sont pas indiqués par le code.

- Critères d'application : tous les attributs spécifiant une alternative textuelle, des informations obligatoires (l'attribut title), un résumé de tableau, un label dans un menu hiérarchique (l'attribut label), un texte en attente (l'attribut standby), ou tout autre contenu textuel.

```
//img/@alt
```

```
//applet/@alt
```

```
//area/@alt
```

```
//input/@alt
```

```
//meta/@content
```

```
//option/@label
```

```
//optgroup/@label
```

```
//object/@standby
```

```
//table/@summary
```

```
/**/@title[26]
```

```
//input[@type='text']/@value
```

```
//input[@type='submit']/@value
```

```
//frame/@name
```

```
//iframe/@name
```

- Procédure de test :
 1. Déterminer la direction du texte dans l'attribut.
 2. Vérifier que l'élément parent a un attribut dir identifiant la direction du texte à l'aide de la valeur correspondante (rtl ou ltr).

- Résultats attendus : validé si #2 est vrai. Non validé si #2 est faux.
- Entièrement automatisable : non.

Tests basés sur CSS

Test 4.1_CSS_01

Le but de ce test est de rechercher chaque style CSS générant du texte dans une langue naturelle différente que la langue de l'élément parent de l'élément ou des éléments pour lesquels le style a été défini.

Noter les informations suivantes issues de la spécification CSS 2.0 : comme l'indique leurs noms, les pseudo-éléments `:before` et `:after` indiquent l'emplacement du contenu avant et après le contenu de l'arbre du document de l'élément (mise en évidence)[\[27\]](#). C'est pourquoi la langue du contenu généré devrait être la même que la langue du parent de l'élément pour lequel le style CSS est défini.

- Critères d'application : chaque chaîne de texte générée par des styles CSS.

```
*:after {content: "...";}
```

```
*:before {content: "...";}
```

- Procédure de test :
 1. Déterminer la langue de la chaîne de caractères.
 2. Déterminer la langue définie ou héritée de l'élément parent de l'élément pour lequel le style CSS est défini.
 3. Vérifier que les langues trouvées dans #1 et #2 sont les mêmes.
- Résultats attendus : validé si #3 est vrai. Non validé si #3 est faux.
- Entièrement automatisable : non.

Test 4.1_CSS_02

Ce test a pour objectif de rechercher les styles CSS utilisés pour le sens de lecture du texte. Pour des informations complémentaires voir les techniques de conception pour "XHTML & HTML Internationalization: Handling Bidirectional Text 1.0" [\[28\]](#), "FAQ: CSS vs. markup for bidi support" [\[29\]](#) et la section "Text direction: the 'direction' and 'unicode-bidi' properties" dans la spécification CSS 2.0 [\[30\]](#).

- Critères d'application : les propriétés CSS `direction` et `unicode-bidi`.

```
* {direction: rtl; unicode-bidi: embed }
```

- Procédure de test :
 - Rechercher les styles CSS utilisant les propriétés `direction` et/ou `unicode-bidi`.
- Résultats attendus : non validé si vrai.
- Entièrement automatisable : oui.

Tests UWEM 1.0 pour la recommandation 5 des WCAG 1.0

"[Create tables that transform gracefully.](#)"

Cette recommandation donne des informations sur la façon d'identifier correctement les tableaux codés.

Sommaire: [Point de contrôle 5.1](#) | [Point de contrôle 5.2](#) | [Point de contrôle 5.3](#) | [Point de contrôle 5.4](#)

Point de contrôle 5.1

For data tables, identify row and column headers. [Priority 1]

(Voir <http://www.w3.org/TR/WCAG10/#tech-table-headers> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-table-headers>)

Tests basés sur (X)HTML

Test 5.1_HTML_01

Ce test a pour objectif de rechercher les tableaux de données qui n'ont pas d'en-têtes de lignes et de colonnes.

- Critères d'application : tous les tableaux de données.

```
//table
```

- Procédure de test :
 1. Sélectionner les cellules de données dans le tableau de données.
 2. Pour chaque cellule de données, vérifier qu'il y a une cellule d'en-tête de ligne et une cellule d'en-tête de colonne pouvant être identifiés par les sections "Algorithm to find heading information" ou "Associating header information with data cells" dans la norme HTML 4.01 [\[31\]](#).
- Résultats attendus : validé si #2 est vrai.
- Entièrement automatisable : non.

Test 5.1_HTML_02

Ce test a pour objectif d'identifier le texte préformaté utilisé pour afficher des informations en tableaux. Un texte préformaté ne possède pas de mécanismes permettant de spécifier les titres de lignes et de colonnes.

- Critères d'application : le texte préformaté.

```
//pre
```

- Procédure de test :
 - Déterminer si le texte préformaté est restitué visuellement sous forme de tableau.
- Résultats attendus : non validé si vrai.

- Entièrement automatisable : non.

Point de contrôle 5.2

For data tables that have two or more logical levels of row or column headers, use markup to associate data cells and header cells. [Priority 1]

(Voir <http://www.w3.org/TR/WCAG10/#tech-table-structure> et les techniques à <http://www.w3.org/TR/WCAG10-HTML-TECHS/#identifying-table-rows-columns>)

Tests basés sur (X)HTML

Test 5.2_HTML_01

Le but de ce test est d'identifier les tableaux ayant un ou plusieurs niveaux logiques de lignes et de colonnes qui ne sont pas codés correctement par l'utilisation du code pour les tableaux associant les lignes aux colonnes.

- Critères d'application : les tableaux de données où le contenu de chaque cellule de données a une relation avec au moins deux en-têtes de colonnes.

```
//table
```

- Procédure de test :
 1. Pour chaque cellule de données vérifier qu'au moins une des affirmations suivantes s'applique :
 - 1.a l'attribut headers contient une liste séparée par des espaces de toutes les valeurs des attributs id des cellules d'en-têtes avec lesquelles la cellule de données a une relation;
 - 1.b toutes les cellules d'en-tête de colonnes ont un attribut scope avec la valeur "col" et toutes les cellules d'en-tête de lignes ont un attribut scope avec la valeur "row".
- Résultats attendus : validé si #1.a ou #1.b est vrai.
- Entièrement automatisable : non.

Test 5.2_HTML_02

Le but de ce test est de déterminer si les cellules d'en-tête dans un titre avec deux ou plusieurs niveaux sont classées de manière cohérente. Ce test n'exige pas que axis soit toujours utilisé, mais que les catégories identifiées par l'attribut soient appropriées ou logiques.

- Critères d'application : en-têtes de tableaux avec deux ou plusieurs niveaux.

```
//table[count(descendant::tr[th]) > 1]
```

```
//table[count(descendant::tr[td[@scope]]) > 1]
```

```
//table[descendant::tr[count(th) > 1]]  
  
//table[descendant::tr[count(td[@scope]) > 1]]  
  
//table[descendant::td[boolean(substring-after(substring-  
after(normalize-space(@headers), ' '), ' '))]]
```

- Procédure de test :
 - pour chaque cellule d'en-tête avec un en-tête de tableau ayant deux ou plusieurs niveaux, vérifier que tout attribut axis indique de façon cohérente la catégorie à laquelle appartient la cellule d'en-tête.
Noter que la valeur de l'attribut axis est une étiquette qui peut être présentée à un utilisateur, au lieu d'être une classe ou un nom seulement lisible par une machine.
- Résultats attendus : non validé si faux.
- Entièrement automatisable : non.

Test 5.2_HTML_03

Le but de ce test est de rechercher le manque de cohérence dans la structure de tableaux. Ce test n'exige pas que colgroup, thead, tfoot ou tbody soient toujours utilisés mais que leur utilisation soit appropriée ou logique.

- Critères d'application : les tableaux qui définissent les groupes de colonnes, les titres de tableaux, les pieds de tableaux et les corps de tableaux.

```
//table[colgroup]  
  
//table[thead]  
  
//table[tfoot]  
  
//table[tbody]
```

- Procédure de test :
 - vérifier que chacun des éléments sélectionnés structure le tableau correctement.
- Résultats attendus : non validé si faux.
- Entièrement automatisable : non.

Point de contrôle 5.3

Do not use tables for layout unless the table makes sense when linearized. Otherwise, if the table does not make sense, provide an alternative equivalent (which may be a linearized version).
[Priority 2]

(Voir <http://www.w3.org/TR/WCAG10/#tech-avoid-table-for-layout> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-avoid-table-for-layout>)

Tests basés sur (X)HTML

Test 5.3_HTML_01

Le but de ce test est de rechercher les tableaux qui ne restituent pas la même information lorsqu'ils sont linéarisés.

- Critères d'application : les tableaux de mise en forme.

```
//table[not(@summary) and not(child::caption)]
```

```
//table
```

- Procédure de test :
 - vérifier que le tableau restitue les mêmes informations lorsqu'il est linéarisé [32].
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

Point de contrôle 5.4

If a table is used for layout, do not use any structural markup for the purpose of visual formatting.
[Priority 2]

(Voir <http://www.w3.org/TR/WCAG10/#tech-table-layout> et les techniques à <http://www.w3.org/TR/WCAG10-HTML-TECHS/#tech-table-layout>)

Tests basés sur (X)HTML

Test 5.4_HTML_01

Le but de ce test est de vérifier que les en-têtes de tableaux sont uniquement utilisés dans des tableaux de données.

- Critères d'application : les tableaux ayant des cellules d'en-tête.

```
//table[descendant::th]
```

```
//table[descendant::td[@scope]]
```

```
//table[descendant::td[@axis]]
```

- Procédure de test :
 - Vérifier si le tableau est un tableau de données.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

Test 5.4_HTML_02

Le but de ce test est de vérifier que les en-têtes et les pieds de tableaux sont uniquement utilisés dans des tableaux de données.

- Critères d'application : les tableaux ayant des en-têtes et des pieds.

```
//table[thead]
```

```
//table[tfoot]
```

- Procédure de test :
 - Vérifier si le tableau est un tableau de données.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

Test 5.4_HTML_03

Le but de ce test est de vérifier que les attributs id et headers sont utilisés uniquement dans des tableaux de données.

- Critères d'application : les tableaux ayant une ou plusieurs cellules de données avec un attribut headers et une ou plusieurs cellules d'en-tête avec l'attribut id.

```
//table[descendant::th[@id]]
```

```
//table[descendant::td[@id]]
```

```
//table[descendant::td[@header
```

- Procédure de test :
 - Vérifier si le tableau est un tableau de données.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

Test 5.4_HTML_04

Le but de ce test est de vérifier que les légendes sont uniquement utilisées pour les tableaux de données.

- Critères d'application : les tableaux avec une légende.

```
//table[descendant::caption]
```

- Procédure de test :
 - Vérifier si le tableau est un tableau de données.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

Test 5.4_HTML_05

Le but de ce test est de vérifier que les cellules sont classées par catégories uniquement dans les tableaux de données.

- Critères d'application : les tableaux dans lesquels les cellules sont classées au moyen de l'attribut axis.

```
//table[descendant::th[@axis]]
```

```
//table[descendant::td[@axis]]
```

- Procédure de test :
 - Vérifier si le tableau est un tableau de données.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

Tests UWEM 1.0 pour la recommandation 6 des WCAG 1.0

["Ensure that pages featuring new technologies transform gracefully."](#)

Cette recommandation fournit des informations qui permettent de s'assurer que les pages restent accessibles même lorsque de nouvelles technologies ne sont pas supportées ou qu'elles sont désactivées.

Sommaire: [Point de contrôle 6.1](#) | [Point de contrôle 6.2](#) | [Point de contrôle 6.3](#) | [Point de contrôle 6.4](#) | [Point de contrôle 6.5](#)

Point de contrôle 6.1

Organize documents so they may be read without style sheets. For example, when an HTML document is rendered without associated style sheets, it must still be possible to read the document. [Priority 1]

(Voir <http://www.w3.org/TR/WCAG10/#tech-order-style-sheets> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-order-style-sheets>)

Tests basés sur (X)HTML

Test 6.1_HTML_01

Ce test analyse les conséquences sur la lisibilité du document lorsque des styles sont appliqués via une feuille de styles séparée, une feuille de styles intégrée et des attributs de styles sur des éléments du document.

Note : la spécification CSS 2.0 définit une feuille de style comme "un ensemble d'instructions qui spécifient la présentation d'un document" [\[33\]](#). Ceci comprend les instructions dans les attributs de style.

- Critères d'application : les documents HTML 4.01 et XHTML 1.0 ayant une ou plusieurs feuilles de styles associées. Ceci inclut les feuilles de styles attachées aux documents par l'intermédiaire des en-têtes HTTP (voir la section "Linking to style sheets with HTTP headers" [\[34\]](#) dans HTML 4.01).

```
//link[@rel='stylesheet']  
  
//link[@rel='alternate stylesheet']  
  
//style  
  
/**/@style
```

- Procédure de test :
 1. Bloquer, supprimer ou désactiver toutes les feuilles de styles associées.
 2. Vérifier que le contenu ne devient pas invisible.
 3. Vérifier que le contenu n'est pas masqué par un autre contenu.

4. Vérifier que la signification n'est pas modifiée par des changements dans l'ordre de lecture, provoqués par l'étape 1.
- Résultats attendus : validé #si 2-4 vrais. Non validé si #2, #3 ou #4 faux.
 - Entièrement automatisable : non.

Test 6.1_HTML_02

Ce test analyse les conséquences sur la lisibilité du document lorsque des styles sont appliqués à l'aide d'éléments de programmation.

- Critères d'application : tout script ou gestionnaire d'événement qui modifie la présentation du contenu.

```
//script
//*/@onfocus
//*/@onblur
//*/@onkeypress
//*/@onkeydown
//*/@onkeyup
//*/@onsubmit
//*/@onreset
//*/@onselect
//*/@onchange
//*/@onload
//*/@onclick
//*/@ondblclick
//*/@onmousedown
//*/@onmouseup
//*/@onmouseover
//*/@onmousemove
//*/@onmouseout \[35\]
```

- Procédure de test :
 1. Désactiver tous les scripts des fichiers scripts appliqués au contenu, les scripts des blocs de script et les attributs de gestion d'événements.
 2. Vérifier que le contenu ne devient pas invisible.
 3. Vérifier que le contenu n'est pas masqué par un autre contenu.
 4. Vérifier que la signification n'est pas modifiée par des changements dans l'ordre de lecture, provoqués par l'étape 1.

5. Lorsque les gestionnaires d'événements sont bloqués, et en effectuant les actions susceptibles de déclencher un événement, vérifier que le contenu reste lisible - comme défini aux étapes 2 à 4.
- Résultats attendus : validé si #2-5 vrais. Non validé si #2, #3, #4 ou #5 faux.
 - Entièrement automatisable : non.

Point de contrôle 6.2

Ensure that equivalents for dynamic content are updated when the dynamic content changes.
[Priority 1]

(voir <http://www.w3.org/TR/WCAG10/#tech-dynamic-source> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-dynamic-source>)

Tests basés sur (X)HTML

Test 6.2_HTML_01

Ce test analyse l'équivalent textuel de tout contenu non textuel chargé dans un cadre.

- Critères d'application : tout contenu non textuel auquel l'attribut src de l'élément frame fait référence.

```
document(//frame/@src)
```

```
document(//iframe/@src)
```

- Procédure de test :
 - vérifier qu'il y a un équivalent textuel approprié à la version actuelle du contenu non textuel.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

Test 6.2_HTML_02

Ce test analyse l'équivalent textuel de tout contenu non textuel chargé dans un cadre, par le navigateur, résultant de l'activation d'un lien ou de l'exécution d'un script.

- Critères d'application : tout contenu non textuel chargé dans un cadre, résultant de l'activation d'un lien ou de l'exécution d'un script.

```
//script
```

```
//a/@href
```

```
//*/@onfocus
```

```
//*/@onblur
```

```
/**/@onkeypress
/**/@onkeydown
/**/@onkeyup
/**/@onsubmit
/**/@onreset
/**/@onselect
/**/@onchange
/**/@onload
/**/@onclick
/**/@ondblclick
/**/@onmousedown
/**/@onmouseup
/**/@onmouseover
/**/@onmousemove
/**/@onmouseout
```

- Procédure de test :
 - vérifier qu'il y a un équivalent textuel approprié à la version actuelle du contenu non textuel.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

Test 6.2_HTML_03

Ce test a pour objectif de vérifier qu'il y a des équivalents appropriés pour le contenu non textuel ajouté aux balises ou à l'arbre de document (DOM), par l'intermédiaire de scripts.

- Critères d'application : tout contenu non textuel ajouté aux balises ou à l'arbre de document (DOM), par l'intermédiaire de scripts.

```
//script
//a/@href
/**/@onfocus
/**/@onblur
/**/@onkeypress
/**/@onkeydown
```

```
//*/@onkeyup  
//*/@onsubmit  
//*/@onreset  
//*/@onselect  
//*/@onchange  
//*/@onload  
//*/@onclick  
//*/@ondblclick  
//*/@onmousedown  
//*/@onmouseup  
//*/@onmouseover  
//*/@onmousemove
```

- Procédure de test :
 - Vérifier qu'il y a un équivalent textuel approprié pour le contenu non textuel.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

Point de contrôle 6.3

Ensure that pages are usable when scripts, applets, or other programmatic objects are turned off or not supported. If this is not possible, provide equivalent information on an alternative accessible page. [Priority 1]

(Voir <http://www.w3.org/TR/WCAG10/#tech-scripts> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-scripts>)

Tests basés sur (X)HTML

Test 6.3_HTML_01

Ce test détermine si les informations ou les fonctionnalités fournies par du contenu intégré sont également disponibles sans ce contenu.

- Critères d'application : les applets (au sens large, non limité à java; ceci comprend le Flash).

```
//applet  
//object
```

- Procédure de test :

1. Désactiver le support des applets.
 2. Vérifier que la page est utilisable et que toutes les fonctionnalités sont toujours disponibles.
 3. Si la page n'est pas utilisable ou si certaines fonctionnalités ne sont plus disponibles, vérifier qu'il y a une page accessible alternative fournissant les informations équivalentes.
- Résultats attendus : validé si #2 ou #3 sont vrais. Non validé si #2 et #3 faux.
 - Entièrement automatisable : non.

Test 6.3_HTML_02

Ce test détermine si l'information ou la fonctionnalité fournie par un script est toujours disponible lorsque le script n'est pas exécuté.

- Critères d'application : (fonctionnalités fournies par) des scripts.

```
//script  
  
//a[starts-with(@href, 'javascript:')]  
  
/**/@onfocus  
  
/**/@onblur  
  
/**/@onkeypress  
  
/**/@onkeydown  
  
/**/@onkeyup  
  
/**/@onsubmit  
  
/**/@onreset  
  
/**/@onselect  
  
/**/@onchange  
  
/**/@onload  
  
/**/@onunload  
  
/**/@onclick  
  
/**/@ondblclick  
  
/**/@onmousedown  
  
/**/@onmouseup  
  
/**/@onmouseover  
  
/**/@onmousemove  
  
/**/@onmouseout
```

- Procédure de test :

1. Bloquer ou désactiver la prise en charge des scripts par le navigateur.
 2. Vérifier que la page est utilisable et que toutes les fonctionnalités sont encore disponibles.
 3. Si la page n'est plus utilisable ou si certaines fonctionnalités ne sont plus disponibles, vérifier qu'il y a une page alternative accessible avec les mêmes informations.
- Résultats attendus : validé si #2 ou #3 sont vrais. Non validé si #2 et #3 faux.
 - Entièrement automatisable : non.

Point de contrôle 6.4

For scripts and applets, ensure that event handlers are input device-independent. [Priority 2]

(Voir <http://www.w3.org/TR/WAI-WEBCONTENT/#tech-keyboard-operable-scripts> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-keyboard-operable-scripts>)

Tests basés sur (X)HTML

Test 6.4_HTML_01

Ce test a pour but de vérifier en particulier que les gestionnaire d'événement propres à la souris ont une version pour le clavier (ou indépendante du périphérique).

- Critères d'application : les éléments ayant des attributs de gestion d'événements liés à la souris, y compris les attributs d'événements implémentés par des scripts.

```

// *[@onclick]

// *[@onmousedown]

// *[@onmouseup]

// *[@onmouseout]

// *[@onmouseover]

```

- Procédure de test :
 - Vérifier que chacun de ces éléments possède un attribut de gestion d'événement de clavier qui déclenche exactement la même fonction ou les mêmes fonctions que l'attribut de gestion d'événement de la souris.

Le tableau suivant donne la correspondances entre les gestionnaires d'événement de périphériques :

Correspondance entre les attribut de gestion d'événements

Événements souris	Événements clavier
onmousedown	onkeydown
onmouseup	onkeyup
onclick [36]	onkeypress

onmouseover	onfocus
onmouseout	onblur

- Résultats attendus : validé si vrai.
- Entièrement automatisable : oui.

Test 6.4_HTML_02

Ce test a pour but de vérifier la présence de gestionnaires d'événements propres à la souris pour lesquels il n'y a pas de gestionnaires d'événements clavier ou indépendants du périphérique, équivalents définis dans la spécification HTML 4.

- Critères d'application :

```
/**[@ondblclick]
```

```
/**[@onmousemove]
```

- Procédure de test :
 1. Sélectionner tout élément possédant un attribut de gestion d'événement de souris pour lesquels il n'y a pas de gestionnaire d'événement de clavier permettant d'exécuter exactement la même fonction.
 2. Vérifier que les fonctions qui sont exécutées par les gestionnaires d'événements peuvent être également implémentées indépendamment de la souris.
- Résultats attendus : validé si #2 est vrai. Non validé si #2 faux.
- Entièrement automatisable : non.

Tests sur les objets externes

Test 6.4_external_01

Ce test vérifie en particulier si les gestionnaires d'événements dans les applets sont indépendants du périphérique.

- Critères d'application : les applets (au sens large, non limité à java; ceci comprend le Flash).

-

- Procédure de test :
 1. Sélectionner tout applet chargé dans la page.
 2. Vérifier que chaque fonction peut être déclenchée à l'aide d'une interface clavier.
- Résultats attendus : validé si #2 est vrai. Non validé si #2 faux.
- Entièrement automatisable : non.

Point de contrôle 6.5

Ensure that dynamic content is accessible or provide an alternative presentation or page. [Priority 2]

(Voir <http://www.w3.org/TR/WCAG10/#tech-fallback-page> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-fallback-page>)

Ce point de contrôle traite de l'accessibilité des contenus dynamiques. Il y a deux manières de créer du contenu dynamique : le contenu dynamique côté serveur, et le contenu dynamique côté client. Concernant les évaluations (semi) automatiques, seul le contenu dynamique côté client est pris en compte, car on ne peut clairement identifier la présence d'un contenu dynamique côté serveur.

Tests basés sur (X)HTML

Test 6.5_HTML_01

Ce test a pour but de détecter des présentations par cadres avec du contenu dynamique inaccessible et sans section noframes.

- Critères d'application : les présentations par cadres sans section noframes.

```
//frameset[not(descendant::noframes)]
```

- Procédure de test :
 - Vérifier que le contenu dynamique est accessible.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

Test 6.5_HTML_02

Ce test a pour but de détecter des présentations par cadres avec du contenu dynamique inaccessible et sans section noframes.

- Critères d'application : les présentations par cadres sans section noframes.

```
//frameset
```

- Procédure de test :
 1. Vérifier que le contenu dynamique est accessible.
 2. Si #1 est faux, vérifier que la présentation par cadres contient un élément noframes avec une présentation alternative ou un lien vers une présentation alternative.
- Résultats attendus : validé si #1 ou #2 sont vrais. Non validé si #1 et #2 faux.
- Entièrement automatisable : non.

Test 6.5_HTML_03

Ce test a pour but de détecter les liens qui utilisent javascript.

- Critères d'application : les liens avec le "pseudo-protocole javascript"

```
//a[starts-with(@href, 'javascript: ')]
```

- Procédure de test :
 1. Vérifier que l'URI n'utilise pas le "pseudo-protocole javascript".

2. Si l'étape 1 est fautive, vérifier qu'il existe une présentation alternative ou une page ayant le même contenu.
 - Résultats attendus : validé si #1 ou #2 est vrai. Non validé si #1 et #2 faux.
 - Entièrement automatisable : non.

Tests UWEM 1.0 pour la recommandation 7 des WCAG 1.0

["Ensure user control of time-sensitive content changes."](#)

Cette recommandation fournit des informations sur les objets ou les pages qui bougent, clignotent, défilent, se mettent automatiquement à jour, et qui rendent le contenu difficile, parfois même impossible d'accès ou à lire.

Sommaire: [Point de contrôle 7.1](#) | [Point de contrôle 7.2](#) | [Point de contrôle 7.3](#) | [Point de contrôle 7.4](#) | [Point de contrôle 7.5](#)

Point de contrôle 7.1

Until user agents allow users to control flickering, avoid causing the screen to flicker. [Priority 1]

Note : People with photosensitive epilepsy can have seizures triggered by flickering or flashing in the 4 to 59 flashes per second (Hertz) range with a peak sensitivity at 20 flashes per second as well as quick changes from dark to light (like strobe lights).

(Voir <http://www.w3.org/TR/WCAG10/#tech-avoid-flicker> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-avoid-flicker>)

Note : Concernant les points de contrôle avec la mention "until user agents", les wcag 1.0 font référence au document "User Agent Support for Accessibility" [37] donnant des informations sur le support des fonctionnalités d'accessibilité par les agents utilisateurs. La version actuelle de ce document (dernière mise à jour du 11 août 2005), indique que : "Netscape Navigator (versions, plateforme), Microsoft Internet Explorer (versions, plateforme), et Opera (versions, plateforme), permettent à l'utilisateur de désactiver le chargement des images, des scripts et des applets. Cette désactivation permettra aux utilisateurs d'éviter les changements brusques de luminosité provoqués par les images, les scripts et les applets. Pour les autres plug-ins, l'utilisateur peut choisir de ne pas charger ces plug-ins. Cependant, l'idéal serait que l'utilisateur puisse arrêter, mettre en pause, ou passer outre les animations, les scripts et autres contenus dynamiques qui provoquent un changement de luminosité, comme expliqué dans le point de contrôle UAAG 3.7 et le point de contrôle UAAG 3.10".

Tests basés sur (X)HTML

Test 7.1_HTML_01

Ce test a pour but de détecter les textes de type marquée qui provoquent un clignotement. Normalement, les marquées ne provoquent pas de clignotement, mais certaines combinaisons entre le taux de défilement, le délai de défilement, la taille des polices et les couleurs peuvent avoir pour effet de faire clignoter des parties de l'écran.

- Critères d'application : les éléments marquée.

```
//marquee
```

- Procédure de test :
 1. Vérifier que le marquee ne provoque pas de changement brusque de luminosité ou d'effet de flash dans la fourchette de 4 à 59 flashes par seconde (Hertz).
 2. Si faux, vérifier que l'effet de flash est un aspect incontournable de la présentation et que l'utilisateur en a le contrôle.
- Résultats attendus : validé si #1 ou #2 est vrai. Non validé si #1 et #2 faux.
- Entièrement automatisable : non.

Test 7.1_HTML_02

Ce test a pour but de détecter des fichiers d'animations gif qui provoquent un changement brusque de luminosité, (les autres types de fichiers images qui peuvent être inclus dans les pages HTML - JPEG et PNG - ne prennent pas en charge l'animation).

- Critères d'application : les fichiers de gif animés.

```
//img
```

```
//object \[38\]
```

- Procédure de test :
 1. Vérifier que la vitesse de rafraîchissement des cadres et que le contraste de couleurs entre les cadres successifs ne provoquent pas de changement brusque de luminosité ou d'effet de flash dans une fourchette de 4 à 59 flashes par seconde (Hertz).
 2. Si faux, vérifier que l'effet de flash est un aspect incontournable de la présentation et que l'utilisateur en a le contrôle.
- Résultats attendus : validé si #1 ou #2 est vrai. Non validé si #1 et #2 faux.
- Entièrement automatisable : non.

Test 7.1_HTML_03

Ce test a pour but de détecter des scripts côté client qui provoquent un changement brusque de luminosité ou un effet de flash.

- Critères d'application : les scripts côté client.

```
//script
```

```
/**/@onfocus
```

```
/**/@onblur
```

```
/**/@onkeypress
```

```
/**/@onkeydown
```

```
/**/@onkeyup
```

```
/**/@onsubmit  
/**/@onreset  
/**/@onselect  
/**/@onchange  
/**/@onload  
/**/@onclick  
/**/@ondblclick  
/**/@onmousedown  
/**/@onmouseup  
/**/@onmouseover  
/**/@onmousemove  
/**/@onmouseout
```

- Procédure de test :
 1. Vérifier que le script ne provoque pas de changement brusque de luminosité ou d'effet de flash dans une fourchette de 4 à 59 flashes par seconde (Hertz).
 2. Si faux, vérifier que l'effet de flash est un aspect incontournable de la présentation et que l'utilisateur en a le contrôle.
- Résultats attendus : validé si #1 ou #2 est vrai. Non validé si #1 et #2 faux.
- Entièrement automatisable : non.

Tests basés sur CSS

Test 7.1_CSS_01

Ce test a pour but de détecter du contenu généré par CSS et provoquant un changement brusque de luminosité ou un effet de flash.

- Critères d'application : les objets (images, vidéos, animations), intégrés aux pages Web par l'intermédiaire des CSS.

```
*:after {content: url(...);}  
*:before {content: url(...);}
```

- Procédure de test :
 1. Vérifier que l'objet ne provoque pas de changement brusque de luminosité ou d'effet de flash dans une fourchette de 4 à 59 flashes par seconde (Hertz).
 2. Si faux, vérifier que l'effet de flash est un aspect incontournable de la présentation et que l'utilisateur en a le contrôle.
- Résultats attendus : validé si #1 ou #2 est vrai. Non validé si #1 et #2 faux.
- Entièrement automatisable : non.

Tests sur les objets externes

Test 7.1_external_01

Ce test a pour but de détecter des applets java provoquant un changement brusque de luminosité ou un effet de flash.

- Critères d'application : les applets java.

```
//object[@codetype='application/java']  
  
//object[@codetype='application/java-archive']  
  
//object[starts-with(@codetype, 'application/x-java-applet')]  
  
//applet  
  
Tout contenu envoyé par HTTP avec les types MIME  
  
'application/java', 'application/java-archive', 'application/x-java-applet'
```

- Procédure de test :
 1. Vérifier que l'applet ne provoque pas de changement brusque de luminosité ou d'effet de flash dans une fourchette de 4 à 59 flashes par seconde (Hertz).
 2. Si faux, vérifier que l'effet de flash est un aspect incontournable de la présentation et que l'utilisateur en a le contrôle.
- Résultats attendus : validé si #1 ou #2 est vrai. Non validé si #1 et #2 faux.
- Entièrement automatisable : non.

Test 7.1_external_02

Ce test a pour but de détecter tout contenu vidéo provoquant un changement brusque de luminosité ou un effet de flash.

- Critères d'application : le contenu vidéo.
- ```
//object[starts-with(@type, 'video/')]
```
- Procédure de test :
  1. Vérifier que le contenu vidéo ne provoque pas de changement brusque de luminosité ou d'effet de flash dans une fourchette de 4 à 59 flashes par seconde (Hertz).
  2. Si faux, vérifier que l'effet de flash est un aspect incontournable de la présentation et que l'utilisateur en a le contrôle.
- Résultats attendus : validé si #1 ou #2 est vrai. Non validé si #1 et #2 faux.
- Entièrement automatisable : non.

## Point de contrôle 7.2

Until user agents allow users to control blinking, avoid causing content to blink (i.e., change presentation at a regular rate, such as turning on and off). [Priority 2]

(Voir <http://www.w3.org/TR/WCAG10/#tech-avoid-blinking> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-avoid-blinking>)

## Tests basés sur (X)HTML

### *Test 7.2\_HTML\_01*

Ce test a pour but de détecter tout élément blink.

- Critères d'application : les éléments blink

```
//blink
```

- Procédure de test :
  - Vérifier qu'un tel élément est présent.
- Résultats attendus : non validé si vrai.
- Entièrement automatisable : oui.

### *Test 7.2\_HTML\_02*

Ce test a pour but de détecter tout fichier de gif animé qui provoque un clignotement, (les autres types de fichiers images - JPEG et PNG - ne prennent pas en charge l'animation).

- Critères d'application : sélectionner une combinaison des éléments/attributs suivants :

```
//img
```

```
//object \[39\]
```

- Procédure de test :
  - Vérifier que l'image ne provoque pas de clignotement.
- Résultats attendus : validé si vrai.
- Entièrement automatisable : oui, si le seuil de contraste est connu.

### *Test 7.2\_HTML\_03*

Ce test a pour but de détecter des scripts qui provoquent un clignotement.

- Critères d'application : les scripts

```
script
```

```
/**/@onfocus
```

```
/**/@onblur
```

```
/**/@onkeypress
```

```
/**/@onkeydown
```

```
/**/@onkeyup
```

```
/**/@onsubmit
/**/@onreset
/**/@onselect
/**/@onchange
/**/@onload
/**/@onunload
/**/@onclick
/**/@ondblclick
/**/@onmousedown
/**/@onmouseup
/**/@onmouseover
/**/@onmousemove
/**/@onmouseout
```

- Procédure de test :
  - Vérifier que le script ne provoque pas de clignotement.
- Résultats attendus : validé si vrai. Non validé si faux. 18:47 21/08/2006
- Entièrement automatisable : non.

## Tests basés sur CSS

### *Test 7.2\_CSS\_01*

Ce test a pour but de détecter du contenu généré par CSS et qui provoque un clignotement.

- Critères d'application : les images, les vidéos et les animations générées par des styles CSS.

```
*:after {content: url(...);}
*:before {content: url(...);}
```

- Procédure de test :
  - Vérifier que le contenu ne provoque pas de clignotement.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

### *Test 7.2\_CSS\_02*

Ce test a pour but de détecter des règles CSS qui provoquent un clignotement du contenu.

- Critères d'application : les règles CSS avec text-decoration : blink

```
* { text-decoration: blink;}
```

- Procédure de test :
  - Vérifier qu'il n'y a pas de règle CSS avec `text-decoration: blink`.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : oui.

## Tests sur les objets externes

### *Test 7.2\_external\_01*

Ce test a pour but de détecter des applets java qui provoquent un clignotement.

- Critères d'application : les applets (Java, Flahs, ou autres).

```
//applet

//object \[40\]
```

- Procédure de test :
  - Vérifier que l'applet ne provoque pas de clignotement.
- Résultats attendus : validé si vrai.
- Entièrement automatisable : non.

### *Test 7.2\_external\_02*

Ce test a pour but de détecter tout contenu vidéo qui provoque un clignotement.

- Critères d'application : la vidéo.

```
//object \[41\]
```

- Procédure de test :
  - Vérifier que le contenu vidéo ne provoque pas de clignotement.
- Résultats attendus : validé si vrai.
- Entièrement automatisable : non.

## Point de contrôle 7.3

Until user agents allow users to freeze moving content, avoid movement in pages. [Priority 2]

(Voir <http://www.w3.org/TR/WCAG10/#tech-avoid-movement> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-avoid-movement>)

## Tests basés sur (X)HTML

### *Test 7.3\_HTML\_01*

Ce test a pour but de détecter les éléments marquée.

- Critères d'application : les éléments marquée.

```
//marquee
```

- Procédure de test :
  - Vérifier que de tels éléments sont présents.
- Résultats attendus : non validé si vrai.
- Entièrement automatisable : oui.

### ***Test 7.3\_HTML\_02***

Ce test a pour but de détecter des scripts qui provoquent un mouvement.

- Critères d'application : les scripts

```
//script
```

```
/**/@onfocus
```

```
/**/@onblur
```

```
/**/@onkeypress
```

```
/**/@onkeydown
```

```
/**/@onkeyup
```

```
/**/@onsubmit
```

```
/**/@onreset
```

```
/**/@onselect
```

```
/**/@onchange
```

```
/**/@onload
```

```
/**/@onunload
```

```
/**/@onclick
```

```
/**/@ondblclick
```

```
/**/@onmousedown
```

```
/**/@onmouseup
```

```
/**/@onmouseover
```

```
/**/@onmousemove
```

```
/**/@onmouseout
```

- Procédure de test :
  1. Vérifier que le script ne provoque pas de mouvement.
  2. Si faux, vérifier que le mouvement peut être arrêté.

- Résultats attendus : validé si #1 est vrai. Validé si #1 est faux et #2 est vrai. Non validé si #1 et #2 faux.
- Entièrement automatisable : non.

## Tests basés sur CSS

### *Test 7.3\_CSS\_01*

Ce test a pour but de détecter du contenu généré par CSS, qui provoque un mouvement.

- Critères d'application : le contenu généré par CSS.

```
*:after {content: url(...);}
```

```
*:before {content: url(...);}
```

- Procédure de test :
  - Vérifier que le contenu ne provoque pas de mouvement.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

## Tests sur les objets externes

### *Test 7.3\_external\_01*

Ce test a pour but de détecter des objets externes qui provoquent ou contiennent du mouvement.

- Critères d'application : les applets (Java, Flash, ou autres).

```
//applet
//object \[42\]
```

- Procédure de test :
  1. Vérifier que l'applet ne contient pas ou ne provoque pas de mouvement.
  2. Si faux, vérifier que le mouvement peut être arrêté.
- Résultats attendus : Validé si #1 est vrai. Validé si #1 est faux et #2 est vrai. Non validé si #1 et #2 faux.
- Entièrement automatisable : non.

### *Test 7.3\_external\_02*

Ce test a pour but de détecter des vidéos qui provoquent ou contiennent du mouvement.

- Critères d'application : les vidéos.

```
//object \[43\]
```

- Procédure de test :
  1. Vérifier que la vidéo ne contient pas ou ne provoque pas de mouvement.
  2. Si faux, vérifier que le mouvement peut être arrêté.

- Résultats attendus :  
Validé si #1 est vrai.  
Validé si #1 est faux et #2 est vrai.  
Non validé si #1 et #2 faux.
- Entièrement automatisable : non.

## Point de contrôle 7.4

Until user agents provide the ability to stop the refresh, do not create periodically auto-refreshing pages. [Priority 2]

(Voir <http://www.w3.org/TR/WCAG10/#tech-no-periodic-refresh> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-no-periodic-refresh>)

## Tests basés sur (X)HTML

### *Test 7.4\_HTML\_01*

Ce test a pour but de détecter des éléments qui peuvent provoquer le rafraîchissement de la page.

- Critères d'application : les éléments meta avec http-equiv="refresh".

```
//meta[@http-equiv='refresh']
```

- Procédure de test :
  1. Vérifier que l'élément meta contient un URI dans l'attribut content.
  2. Si vrai, vérifier que l'URI dans l'attribut content de l'élément meta n'est pas égal à l'URI de la ressource HTML qui contient l'élément meta.
- Résultats attendus : validé si #1 et #2 sont vrais. non validé si #1 faux. Non validé si #1 vrai et #2 faux.
- Entièrement automatisable : oui.

### *Test 7.4\_HTML\_02*

Ce test a pour but de détecter des objets scripts qui peuvent provoquer le rafraîchissement de la page.

- Critères d'application : les scripts.

```
//script
```

```
/**/@onfocus
```

```
/**/@onblur
```

```
/**/@onkeypress
```

```
/**/@onkeydown
```

```
//*/@onkeyup
//*/@onsubmit
//*/@onreset
//*/@onselect
//*/@onchange
//*/@onload
//*/@onunload
//*/@onclick
//*/@ondblclick
//*/@onmousedown
//*/@onmouseup
//*/@onmouseover
//*/@onmousemove
//*/@onmouseout
```

- Procédure de test :
  1. Vérifier que le script ne provoque pas un rafraîchissement de la page.
  2. Si faux, vérifier que le rafraîchissement peut être stoppé.
- Résultats attendus : validé si #1 ou #2 est vrai. Non validé si #1 et #2 faux.
- Entièrement automatisable : non.

## Tests sur les objets externes

### *Test 7.4\_external\_01*

Ce test a pour but de détecter des objets externes qui peuvent provoquer le rafraîchissement de la page.

- Critères d'application : les objets externes.

```
//applet
//object
```

- Procédure de test :
  1. Vérifier que l'objet ne provoque pas un rafraîchissement de la page.
  2. Si faux, vérifier que le rafraîchissement peut être stoppé.
- Résultats attendus : validé si #1 ou #2 est vrai. Non validé si #1 et #2 faux.
- Entièrement automatisable : non.

## Point de contrôle 7.5

Until user agents provide the ability to stop auto-redirect, do not use markup to redirect pages automatically. Instead, configure the server to perform redirects. [Priority 2]

(Voir <http://www.w3.org/TR/WCAG10/#tech-no-auto-forward> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-no-auto-forward>)

### Tests basés sur (X)HTML

#### *Test 7.5\_HTML\_01*

Ce test a pour but de détecter les éléments qui provoquent une redirection de la page.

- Critères d'application : les éléments meta avec http-equiv="refresh".

```
//meta[@http-equiv='refresh']
```

- Procédure de test :
  - Vérifier que l'URI dans l'attribut content de l'élément meta est égal à l'URI du document contenant l'élément meta.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : oui.

#### *Test 7.5\_HTML\_02*

Ce test a pour but de détecter les scripts qui provoquent une redirection sans fournir de mécanisme permettant de l'arrêter.

- Critères d'application : les scripts.

```
//script

//*/@onfocus

//*/@onblur

//*/@onkeypress

//*/@onkeydown

//*/@onkeyup

//*/@onsubmit

//*/@onreset

//*/@onselect

//*/@onchange

//*/@onload
```

```
/**/@onunload
/**/@onclick
/**/@ondblclick
/**/@onmousedown
/**/@onmouseup
/**/@onmouseover
/**/@onmousemove
/**/@onmouseout
```

- Procédure de test :
  1. Vérifier que le script ne provoque pas de redirection.
  2. Si faux, vérifier que la redirection peut être stoppée.
- Résultats attendus : Validé si #1 est vrai. Validé si #1 est faux et #2 est vrai. Non validé si #1 et #2 faux.
- Entièrement automatisable : non.

## Tests sur les objets externes

### *Test 7.5\_external\_01*

Ce test a pour but de détecter les objets externes qui provoquent une redirection sans fournir de mécanisme permettant de l'arrêter.

- Critères d'application : les objets externes.

```
//applet
//object
```

- Procédure de test :
  1. Vérifier que l'objet ne provoque pas de redirection.
  2. Si faux, vérifier que la redirection peut être stoppée.
- Résultats attendus : Validé si #1 est vrai. Validé si #1 est faux et #2 est vrai. Non validé si #1 et #2 faux.
- Entièrement automatisable : non.

## Tests UWEM 1.0 pour la recommandation 8 des WCAG 1.0

["Ensure direct accessibility of embedded user interfaces."](#)

Cette recommandation fournit des informations sur la manière de créer des interfaces utilisateurs intégrées accessibles.

Sommaire: [Point de contrôle 8.1](#)

### Point de contrôle 8.1

Make programmatic elements such as scripts and applets directly accessible or compatible with assistive technologies [Priority 1 if functionality is important and not presented elsewhere, otherwise Priority 2.]

(Voir <http://www.w3.org/TR/WCAG10/#tech-directly-accessible> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-directly-accessible>)

### Tests basés sur (X)HTML

#### *Test 8.1\_HTML\_01*

Ce test a pour but de détecter les scripts qui ne sont pas directement accessibles ou compatibles avec les aides techniques.

- Critères d'application : les scripts.

```
//script
/**/@onfocus
/**/@onblur
/**/@onkeypress
/**/@onkeydown
/**/@onkeyup
/**/@onsubmit
/**/@onreset
/**/@onselect
/**/@onchange
/**/@onload
/**/@onunload
/**/@onclick
```

```
/**/@ondblclick
```

```
/**/@onmousedown
```

```
/**/@onmouseup
```

```
/**/@onmouseover
```

```
/**/@onmousemove
```

```
/**/@onmouseout
```

- Procédure de test :
  1. Si le script est un agent utilisateur, vérifier qu'il est conforme aux UAAG.
  2. Si le script est un outil de création de contenu, vérifier qu'il est conforme aux ATAG.
  3. Dans les autres cas, vérifier que l'objet est directement accessible ou compatible avec les aides techniques, soit par l'intermédiaire d'une API d'accessibilité, si disponible, soit en s'assurant que toutes les fonctionnalités sont disponibles par la seule utilisation du clavier.
- Résultats attendus : validé si #1 ou #2 ou #3 est vrai. Non validé si #1, #2 et #3 faux.
- Entièrement automatisable : non.

## Tests sur les objets externes

### *Test 8.1\_external\_01*

Ce test a pour but de détecter les objets externes qui ne sont pas directement accessibles ou compatibles avec les aides techniques.

- Critères d'application : tous les objets externes.

```
//applet
```

```
//object
```

- Procédure de test :
  1. Si l'objet externe est un agent utilisateur, vérifier qu'il est conforme aux User Agent Accessibility Guidelines 1.0 [\[44\]](#) .
  2. Si l'objet externe est un outil de création de contenu, vérifier qu'il est conforme aux Authoring Tool Accessibility Guidelines 1.0 [\[45\]](#) .
  3. Dans les autres cas, vérifier que l'objet est directement accessible ou compatible avec les aides techniques, soit par l'intermédiaire d'une API d'accessibilité, si disponible, soit en s'assurant que toutes les fonctionnalités sont disponibles par la seule utilisation du clavier.
- Résultats attendus : validé si #1 ou #2 ou #3 est vrai. Non validé si #1, #2 et #3 faux.
- Entièrement automatisable : non.

## Tests UWEM 1.0 pour la recommandation 9 des WCAG 1.0

["Design for device-independence."](#)

Cette recommandation fournit des informations sur la manière de créer du contenu Web qui ne s'appuie pas sur un seul périphérique d'entrée ou de sortie spécifique.

Sommaire: [Point de contrôle 9.1](#) | [Point de contrôle 9.2](#) | [Point de contrôle 9.3](#)

### Point de contrôle 9.1

Provide client-side image maps instead of server-side image maps except where the regions cannot be defined with an available geometric shape. [Priority 1]

(Voir <http://www.w3.org/TR/WCAG10/#tech-client-side-maps> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-client-side-maps>)

### Tests basés sur (X)HTML

#### Test 9.1\_HTML\_01

Ce test a pour but de détecter les images map côté serveur.

- Critères d'application : les images map côté serveur.

```
//a//img[@ismap]
```

```
//input[@type='image']
```

- Procédure de test :
  - Vérifier si l'image map côté serveur ne peut être remplacée par une image map côté client.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : oui [\[46\]](#) .

### Point de contrôle 9.2

Ensure that any element that has its own interface can be operated in a device-independent manner. [Priority 2]

(Voir <http://www.w3.org/TR/WCAG10/#tech-keyboard-operable> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-keyboard-operable>)

### Tests sur les objets externes

#### Test 9.2\_external\_01

Ce test a pour but de contrôler l'indépendance de l'interface par rapport aux périphériques, sur les objets intégrés ou inclus.

- Critères d'application : tous les objets intégrés ou inclus.

```
//applet
```

```
//object
```

- Procédure de test :
  1. Vérifier que l'objet externe ne possède pas sa propre interface.
  2. Si faux, vérifier que toute opération sur l'interface peut s'effectuer indépendamment du périphérique en n'utilisant que le clavier ou une interface d'émulation clavier.
- Résultats attendus : validé si #1 ou #2 est vrai. Non validé si #1 et #2 faux.
- Entièrement automatisable : non.

### Point de contrôle 9.3

For scripts, specify logical event handlers rather than device-dependent event handlers. [Priority 2]

(Voir <http://www.w3.org/TR/WCAG10/#tech-device-independent-events> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-device-independent-events>)

### Tests basés sur (X)HTML

#### Test 9.3\_HTML\_01

Ce test a pour but de vérifier que des gestionnaires d'événements dépendants d'un périphérique peuvent être remplacés par des gestionnaires d'événements indépendants du périphérique.

- Critères d'application : les éléments ayant des attributs de gestion d'événements dépendants d'un périphérique.

```
/**/@onclick
```

```
/**/@ondblclick
```

```
/**/@onkeydown
```

```
/**/@onkeypress
```

```
/**/@onkeyup
```

```
/**/@onmousedown
```

```
/**/@onmousemove
```

```
/**/@onmouseout
```

```
/**/@onmouseover
```

//\*/@onmouseup

- Procédure de test :
  - Vérifier que les attributs de gestion d'événement dépendants d'un périphérique ne peuvent être remplacés par d'autres, indépendants du périphérique, comme @onblur, @onchange, @onfocus, @onload, @onreset, @onselect, @onsubmit et @onunload.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : oui.

# Tests UWEM 1.0 pour la recommandation 10 des WCAG 1.0

["Use interim solutions."](#)

Sommaire: [Point de contrôle 10.1](#) | [Point de contrôle 10.2](#)

## Point de contrôle 10.1

Until user agents allow users to turn off spawned windows, do not cause pop-ups or other windows to appear and do not change the current window without informing the user. [Priority 2]

(Voir <http://www.w3.org/TR/WCAG10/#tech-avoid-pop-ups> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-avoid-pop-ups>)

## Tests basés sur (X)HTML

### Test 10.1\_HTML\_01

Ce test a pour but de détecter les attributs target qui provoquent l'ouverture d'une nouvelle fenêtre, sans que l'utilisateur n'en soit averti.

- Critères d'application : les éléments a, area, form ou link ayant un attribut target, avec une valeur différente de "\_self", "\_parent" ou "\_top".

```
//a[@target!='_self' et @target!='_parent' et @target!='_top']
//area[@target!='_self' et @target!='_parent' et @target!='_top']
//form[@target!='_self' et @target!='_parent' et @target!='_top']
//link[@target!='_self' et @target!='_parent' et @target!='_top']
```

- Procédure de test :
  - Vérifier que le contenu avertit l'utilisateur que l'activation de l'élément déclenche l'ouverture d'une nouvelle fenêtre.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

### Test 10.1\_HTML\_02

Ce test a pour but de détecter les éléments base ayant un attribut target qui provoque l'ouverture d'une nouvelle fenêtre, sans que l'utilisateur n'en soit averti.

- Critères d'application : l'élément base ayant un attribut target, avec une valeur différente de "\_self", "\_parent" ou "\_top".

```
//base[@target!='_self' et @target!='_parent' et @target!='_top']
```

- Procédure de test :

- Vérifier que le contenu avertit l'utilisateur que l'activation d'un élément a, area, form ou link déclenche l'ouverture d'une nouvelle fenêtre.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

### ***Test 10.1\_HTML\_03***

Ce test a pour but de détecter les scripts qui déclenchent l'ouverture d'une nouvelle fenêtre sans que l'utilisateur n'en soit averti.

- Critères d'application : les scripts qui provoquent l'affichage du contenu dans une nouvelle fenêtre.

```
//script
/**/@onblur
/**/@onchange
/**/@onfocus
/**/@onload
/**/@onreset
/**/@onselect
/**/@onsubmit
/**/@onunload
//a[starts-with(@href, 'javascript: ')]
```

- Procédure de test :
  - Vérifier que le contenu avertit l'utilisateur que le script déclenche l'ouverture d'une nouvelle fenêtre.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

## **Point de contrôle 10.2**

Until user agents support explicit associations between labels and form controls, for all form controls with implicitly associated labels, ensure that the label is properly positioned. [Priority 2]

(Voir <http://www.w3.org/TR/WCAG10/#tech-unassociated-labels> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-unassociated-labels>)

## **Tests basés sur (X)HTML**

### ***Test 10.2\_HTML\_01***

Ce test a pour but de détecter des contrôles de formulaire ayant des labels implicitement associés et mal positionnés.

- Critères d'application : les contrôles de formulaire visibles, pouvant avoir un label.

```
//input[not(@type='hidden')][not(@type='submit')][not(@type='button')][not(@type='reset')]
```

```
//select
```

```
//textarea
```

- Procédure de test :
  - Vérifier que le label implicitement associé au contrôle est correctement positionné.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

## Tests UWEM 1.0 pour la recommandation 11 des WCAG 1.0

"[Use W3C technologies and guidelines.](#)"

Cette recommandation préconise d'utiliser les technologies w3c, et décrit ce qu'il faut faire lorsque d'autres technologies sont employées.

Sommaire: [Point de contrôle 11.1](#) | [Point de contrôle 11.2](#) | [Point de contrôle 11.4](#) |

### Point de contrôle 11.1

Use W3C technologies when they are available and appropriate for a task and use the latest versions when supported. [Priority 2]

(Voir <http://www.w3.org/TR/WCAG10/#tech-latest-w3c-specs> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-latest-w3c-specs>)

### Tests basés sur (X)HTML

#### *Test 11.1\_HTML\_01*

Ce test a pour but de vérifier que les dernières versions des technologies w3c pour le HTML et le XHTML ont été utilisées.

- Critères d'application : les en-têtes HTTP et les déclarations de type de document (DTD).  
-
- Procédure de test :
  1. Vérifier que le type de contenu HTTP est text/html ou application/xhtml+xml.
  2. Si le type de contenu HTTP est text/html, vérifier que le type de document est HTML 4.01 ou XHTML 1.0.
  3. Si le type de contenu HTTP est application/xhtml+xml, vérifier que le type de document est XHTML 1.1.
  4. Si le type de document est HTML 4.01, vérifier que le document est conforme à la DTD à l'aide d'un outil de validation SGML.
  5. Si le type de document est XHTML 1.0 ou XHTML 1.1, vérifier que le document est conforme à la DTD à l'aide d'un outil de validation XML.
- Résultats attendus : non validé si au moins une étape est fautive.
- Entièrement automatisable : oui.

### Point de contrôle 11.2

Avoid deprecated features of W3C technologies. [Priority 2]

(Voir <http://www.w3.org/TR/WCAG10/#tech-avoid-deprecated> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-avoid-deprecated>)

Noter que "déprécié" peut avoir une signification différente selon les spécifications. En HTML 4.01, cela signifie :

"Un élément ou un attribut déprécié le devient quand il a été déclassé par une nouvelle spécification. Les éléments dépréciés sont définis dans le manuel de référence, aux endroits appropriés, mais sont clairement signalés comme dépréciés. Les éléments dépréciés sont amenés à devenir obsolètes dans les versions futures de HTML. Pour des raisons de compatibilité descendante, les agents utilisateurs doivent continuer à supporter les éléments dépréciés."

Il est possible qu'une fonctionnalité dépréciée le soit au profit d'une fonctionnalité qui ne soit pas correctement supportée (ex' object est censé remplacer applet).

## Tests basés sur (X)HTML

### *Test 11.2\_HTML\_01*

Ce test a pour but de détecter les éléments HTML dépréciés.

- Critères d'application : les éléments dépréciés selon la spécification HTML 4.01.

```
//applet
//basefont
//center
//dir
//font
//isindex
//menu
//s
//strike
//u
```

- Procédure de test :
  - Vérifier que l'un de ces éléments est présent.
- Résultats attendus : validé si faux. Non validé si vrai.
- Entièrement automatisable : oui.

### *Test 11.2\_HTML\_02*

Ce test a pour but de détecter les attributs HTML dépréciés.

- Critères d'application : les attributs dépréciés selon la spécification HTML 4.01.

```
//body/@bgcolor
```

```
//body/@link
//body/@alink
//body/@vlink
//body/@background
//body/@text
//caption/@align
//iframe/@align
//img/@align
//img/@border
//img/@hspace
//img/@vspace
//object/@align
//object/@border
//object/@hspace
//object/@vspace
//input/@align
//legend/@align
//li/@value
//table/@align
//table/@bgcolor
//tr/@bgcolor
//th/@bgcolor
//th/@height
//th/@nowrap
//th/@width
//td/@bgcolor
//td/@height
//td/@nowrap
//td/@width
//hr/@align
```

```
//hr/@noshade
//hr/@size
//hr/@width
//p/@align
//div/@align
//h1/@align
//h2/@align
//h3/@align
//h4/@align
//h5/@align
//h6/@align
//br/@clear
//dl/@compact
//ol/@compact
//ol/@start
//ol/@value
//ul/@compact
//script/@language
```

XHTML 1.0 [\[47\]](#) déprécie formellement l'usage de :

```
//a/@name
//applet/@name
//form/@name
//frame/@name
//iframe/@name
//img/@name
```

- Procédure de test :
  - Vérifier que l'un de ces attributs est présent.
- Résultats attendus : validé si faux. Non validé si vrai.
- Entièrement automatisable : oui.

## Point de contrôle 11.4

If, after best efforts, you cannot create an accessible page, provide a link to an alternative page that uses W3C technologies, is accessible, has equivalent information (or functionality), and is updated as often as the inaccessible (original) page. [Priority 1]

(Voir <http://www.w3.org/TR/WCAG10/#tech-alt-pages> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-alt-pages>)

### Tests basés sur (X)HTML

#### *Test 11.4\_HTML\_01*

Ce test recherche les contenus alternatifs et vérifie s'ils sont équivalents.

- Critères d'application : la page alternative atteignable par un lien.

```
document(//a/href)
```

```
document(//link/@href)
```

- Procédure de test :
  1. Comparer le contenu des informations et des fonctionnalités des deux sources d'informations.
  2. Vérifier qu'elles sont équivalentes.
- Résultats attendus : validé si #2 est vrai. Non validé si #2 faux.
- Entièrement automatisable : non.

#### *Test 11.4\_HTML\_02*

Ce test recherche un contenu alternatif et vérifie qu'il est accessible.

- Critères d'application : la page alternative atteignable par un lien.

```
document(//a/href)
```

```
document(//link/@href)
```

- Procédure de test :
  1. Identifier le contenu alternatif.
  2. Vérifier qu'il est conforme aux points de contrôle de priorité 1 de UWEM, de cette section, autres que celui-ci, 11.4.
- Résultats attendus : validé si #2 est vrai. Non validé si #2 faux.
- Entièrement automatisable : non.

#### *Test 11.4\_HTML\_03*

Ce test recherche le contenu alternatif et vérifie si le contenu d'origine aurait pu être rendu accessible.

- Critères d'application : la page alternative atteignable par un lien.

`document(//a/href)`

`document(//link/@href)`

- Procédure de test :
  1. Identifier le contenu alternatif.
  2. Vérifier que le contenu d'origine n'aurait pu être rendu accessible sans un effort excessif.
- Résultats attendus : validé si #2 est vrai. Non validé si #2 faux.
- Entièrement automatisable : non.

# Tests UWEM 1.0 pour la recommandation 12 des WCAG 1.0

["Provide context and orientation information."](#)

Cette recommandation fournit des informations sur la manière de donner des informations contextuelles ou d'orientation afin d'aider l'utilisateur à comprendre des pages ou des éléments complexes.

Sommaire: [Point de contrôle 12.1](#) | [Point de contrôle 12.2](#) | [Point de contrôle 12.3](#) | [Point de contrôle 12.4](#)

## Point de contrôle 12.1

Title each frame to facilitate frame identification and navigation. [Priority 1]

(Voir <http://www.w3.org/TR/WCAG10/#tech-frame-titles> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-frame-titles>)

## Tests basés sur (X)HTML

### *Test 12.1\_HTML\_01*

Ce test a pour but de détecter les cadres sans description.

- Critères d'application : les éléments frames sans l'attribut title.

```
//frame[not(@title)]

//iframe[not(@title)]
```

- Procédure de test :
  - Vérifier que de tels cadres sont présents.
- Résultats attendus : non validé si vrai.
- Entièrement automatisable : oui.

### *Test 12.1\_HTML\_02*

Ce test a pour but de vérifier que l'attribut title identifie bien le cadre.

- Critères d'application : les éléments frames avec l'attribut title.

```
//frame[@title]

//iframe[@title]
```

- Procédure de test :
  1. Sélectionner les éléments.
  2. Vérifier que le titre identifie le cadre.
- Résultats attendus : validé si #2 est vrai. Non validé si #2 faux.
- Entièrement automatisable : non.

## Point de contrôle 12.2

Describe the purpose of frames and how frames relate to each other if it is not obvious by frame titles alone. [Priority 2]

(Voir <http://www.w3.org/TR/WCAG10/#tech-frame-longdesc> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-frame-longdesc>)

### Tests basés sur (X)HTML

#### *Test 12.2\_HTML\_01*

Ce test a pour but de vérifier que la description longue présente le contexte du cadre, si le titre du cadre n'est pas clair en soi.

- Critères d'application : les documents référencés par l'attribut longdesc des éléments frame, ou par l'attribut href des éléments a situés dans la balise noframes et dont le but est d'expliquer le contexte du cadre.

```
document (//frame/@longdesc)
```

```
document (//noframes//a/@href)
```

- Procédure de test :
  1. Sélectionner le document contenant la description longue auquel l'élément fait référence.
  2. Vérifier que le cadre et son contexte sont décrits par le texte du document, si cela ne peut pas être déduit à partir du titre du cadre lui-même.
- Résultats attendus : validé si #2 est vrai. Non validé si #2 faux.
- Entièrement automatisable : non.

## Point de contrôle 12.3

Divide large blocks of information into more manageable groups where natural and appropriate. [Priority 2]

(Voir <http://www.w3.org/TR/WCAG10/#tech-group-information> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-group-information>)

### Tests basés sur (X)HTML

#### *Test 12.3\_HTML\_01*

Ce test a pour but de détecter des regroupements de champ (fieldset) sans titre (legend).

- Critères d'application : les éléments fieldset sans l'élément fils legend.

```
//fieldset[not(legend)]
```

- Procédure de test :
  - Vérifier que de tels éléments fieldset sont présents.
- Résultats attendus : validé si faux. Non validé si vrai.
- Entièrement automatisable : oui.

### ***Test 12.3\_HTML\_02***

Ce test a pour but de vérifier que le titre (legend) décrit ce que représente le regroupement de champs (fieldset).

- Critères d'application : les éléments legend.

```
//legend
```

- Procédure de test :
  - Vérifier que le titre (legend) représente le contexte de l'élément parent fieldset.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

### ***Test 12.3\_HTML\_03***

Ce test a pour but de vérifier que les éléments sont regroupés de manière logique.

- Critères d'application : les éléments fieldset.

```
//fieldset
```

- Procédure de test :
  - Vérifier que les éléments de contrôles de formulaire dans le groupe de champ (fieldset) sont regroupés de manière logique.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

### ***Test 12.3\_HTML\_04***

Ce test a pour but de détecter les éléments optgroup sans attribut label.

- Critères d'application : les éléments optgroup sans attribut label.

```
//optgroup[not(@label)]
```

- Procédure de test :
  - Vérifier que de tels éléments optgroup sont présents.
- Résultats attendus : non validé si vrai.
- Entièrement automatisable : oui.

### ***Test 12.3\_HTML\_05***

Ce test a pour but de vérifier que le label donne bien la signification de l'élément optgroup.

- Critères d'application : les attributs label des éléments optgroup.

```
//optgroup/@label
```

- Procédure de test :
  - Vérifier que le contenu de l'attribut label représente le contexte de l'élément optgroup.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

### ***Test 12.3\_HTML\_06***

Ce test a pour but de vérifier que les éléments option sont groupés dans un ordre logique.

- Critères d'application : les éléments optgroup.

```
//optgroup
```

- Procédure de test :
  - Vérifier que les éléments sont regroupés de manière logique.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

### ***Test 12.3\_HTML\_07***

Ce test a pour but de détecter les tableaux sans titre.

- Critères d'application : les tableaux de données implémentés avec l'élément table et sans l'élément fils caption.

```
//table[not(caption)]
```

- Procédure de test :
  - Vérifier que de tels éléments table sont présents.
- Résultats attendus : validé si faux. Non validé si vrai.
- Entièrement automatisable : non.

### ***Test 12.3\_HTML\_08***

Ce test a pour but de vérifier que le titre donne la signification du tableau.

- Critères d'application : l'élément caption dans un tableau de données.

```
//caption
```

- Procédure de test :
  - Vérifier que le titre décrit la nature du tableau.
- Résultats attendus : validé si vrai. Non validé si faux.

- Entièrement automatisable : non.

### ***Test 12.3\_HTML\_09***

Ce test a pour but de vérifier que les éléments sont groupés dans un ordre logique.

- Critères d'application : les éléments `thead`, `tbody`, `tfoot`, et `colgroup`.

```
//thead

//tbody

//tfoot

//colgroup
```

- Procédure de test :
  - Vérifier que leurs éléments fils sont groupés dans un ordre logique.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

### ***Test 12.3\_HTML\_10***

Ce test a pour but de vérifier que les éléments sont groupés dans un ordre logique.

- Critères d'application : les éléments `ul`, `ol`, et `dl`.

```
//ul

//ol

//dl
```

- Procédure de test :
  - Vérifier que leurs éléments fils sont groupés dans un ordre logique.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

### ***Test 12.3\_HTML\_11***

Ce test a pour but de vérifier que les éléments sont structurés dans un ordre logique.

- Critères d'application : les paragraphes (éléments `p`).

```
//p
```

- Procédure de test :
  - Vérifier que le paragraphe est utilisé pour structurer le texte de manière logique.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

**Test 12.3\_HTML\_12**

Ce test a pour but de déterminer si les contrôles de formulaire nécessitent un regroupement.

- Critères d'application : l'élément form sans élément fieldset descendant.

```
//fieldset)]
```

- Procédure de test :
  - Vérifier que les contrôles de formulaire dans l'élément form ne nécessitent pas de regroupement.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

**Test 12.3\_HTML\_13**

Ce test a pour but de déterminer si les options nécessitent un regroupement.

- Critères d'application : les éléments select sans les éléments fils optgroup.

```
//select[not(optgroup)]
```

- Procédure de test :
  - Vérifier que les options dans l'éléments select ne nécessitent pas de regroupement.
- Résultats attendus : validé si vrai.
- Entièrement automatisable : non.

**Test 12.3\_HTML\_14**

Ce test a pour but de déterminer si les lignes d'un tableau nécessitent un regroupement.

- Critères d'application : les tableaux sans en-têtes ni pied.

```
//table[not(thead) or not(tfoot) or count(tbody)<2]
```

- Procédure de test :
  - Vérifier que les lignes du tableau ne nécessitent pas de regroupement.
- Résultats attendus : validé si vrai.
- Entièrement automatisable : non.

**Test 12.3\_HTML\_15**

Ce test a pour but de déterminer si le texte nécessite un regroupement à l'aide de titres et de paragraphes.

- Critères d'application : le corps du texte.

```
//body//text()
```

- Procédure de test :

- Vérifier que le texte est structuré de manière approprié à l'aide d'éléments de titres et de paragraphes.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

## Point de contrôle 12.4

Associate labels explicitly with their controls. [Priority 2]

(Voir <http://www.w3.org/TR/WCAG10/#tech-associate-labels> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-associate-labels>)

## Tests basés sur (X)HTML

### Test 12.4\_HTML\_01

Ce test a pour but de détecter des contrôles de formulaire sans identifiant.

- Critères d'application : les champs de saisie ou les sélecteurs dans un formulaire, sans l'attribut id.

```
//input[not(@type='hidden')][not(@type='submit')][not(@type='reset')][not(@type='button')][not(@type='image')][not(@id)]

//select[not(@id)]

//textarea[not(@id)]
```

- Procédure de test :
  - Vérifier qu'au moins un de ces éléments est présent.
- Résultats attendus : validé si faux. Non validé si vrai.
- Entièrement automatisable : oui.

### Test 12.4\_HTML\_02

Ce test a pour but de détecter des contrôles de formulaire sans l'élément label.

- Critères d'application : les champs de saisie ou les sélecteurs dans un formulaire sans élément label associé.

```
//input[not(@type='hidden')][not(@type='submit')][not(@type='reset')][not(@type='button')][not(@type='image')][@id]

//select[@id]
```

- Procédure de test :
  - Vérifier qu'il existe un élément label dans le document dont l'attribut for est égal à l'attribut id d'un élément de contrôle du formulaire.

- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : oui.

## Tests UWEM 1.0 pour la recommandation 13 des WCAG 1.0

["Provide clear navigation mechanisms."](#)

Cette recommandation fournit des informations sur la manière de donner des informations contextuelles ou d'orientation afin d'aider l'utilisateur à comprendre des pages ou des éléments complexes.

Sommaire: [Point de contrôle 13.1](#) | [Point de contrôle 13.2](#) | [Point de contrôle 13.3](#) | [Point de contrôle 13.4](#)

### Point de contrôle 13.1

Clearly identify the target of each link. [Priority 2]

(Voir <http://www.w3.org/TR/WCAG10/#tech-meaningful-links> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-meaningful-links>)

### Tests basés sur (X)HTML

#### *Test 13.1\_HTML\_01*

Ce test a pour but de détecter les éléments a, input et area ayant le même titre, le même intitulé et pointant vers des destinations différentes (href). S'il n'y a aucun attribut title, seul l'élément textuel est vérifié.

- Critères d'application : les éléments a, input et area ayant le même intitulé et le même attribut title, mais une destination différente.

```
//a
```

```
//area
```

```
//input
```

- Procédure de test :
  1. Sélectionner les éléments similaires.
  2. Vérifier que deux liens ayant le même intitulé (y compris les alternatives textuelles des éléments non textuels), et le même attribut title, (s'il existe), pointent vers la même ressource.
- Résultats attendus : validé si #2 est vrai. Non validé si #2 faux.
- Entièrement automatisable : oui.

#### *Test 13.1\_HTML\_02*

Ce test a pour but de détecter les éléments a, input et area dont l'intitulé ne permet pas d'identifier clairement la destination.

- Critères d'application : les éléments a, input et area ayant le même intitulé et le même attribut title, mais pointant vers des destinations différentes.

//a

//area

- Procédure de test :
  - Vérifier que l'intitulé du lien (y compris les alternatives textuelles des éléments non textuels) décrit clairement l'effet provoqué par l'activation de l'élément.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

## Point de contrôle 13.2

Provide metadata to add semantic information to pages and sites. [Priority 2]

(Voir <http://www.w3.org/TR/WCAG10/#tech-use-metadata> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-use-metadata>)

### Tests basés sur (X)HTML

#### *Test 13.2\_HTML\_01*

Ce test a pour but de détecter l'élément title d'un document dans une ressource web.

- Critères d'application : les documents.
- Procédure de test :
  1. Vérifier qu'il existe un élément title.
  2. Vérifier que ce titre n'est pas vide.
  3. Vérifier que l'élément title fournit un titre approprié à la ressource.
- Résultats attendus : validé si vrai. Non validé si faux.
- Entièrement automatisable : non.

## Point de contrôle 13.3

Provide information about the general layout of a site (e.g., a site map or table of contents). [Priority 2]

(Voir <http://www.w3.org/TR/WCAG10/#tech-site-description> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-site-description>)

### Tests basés sur (X)HTML

#### *Test 13.3\_HTML\_01*

Ce test a pour but de détecter un site web sans information générale sur l'architecture du site.

- Critères d'application : tout le site web.

- Procédure de test :
  1. Vérifier que le site web comporte un document qui décrit l'architecture générale du site. Un plan du site ou une table des matières répondent à ces exigences.
  2. Vérifier que ce document peut être atteint à partir de la page d'accueil.
- Résultats attendus : validé si #1 et #2 sont vrais. Non validé si #1 ou #2 faux.
- Entièrement automatisable : non.

## **Point de contrôle 13.4**

Title each frame to facilitate frame identification and navigation. [Priority 1]

(Voir <http://www.w3.org/TR/WCAG10/#tech-clear-nav-mechanism> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-clear-nav-mechanism>)

## **Tests basés sur (X)HTML**

### ***Test 13.4\_HTML\_01***

Ce test a pour but de vérifier que les mécanismes de navigation sont utilisés de manière cohérente.

- Critères d'application : les menus et les barres de navigation.
- Procédure de test :
  1. Sélectionner les éléments de navigation sur plusieurs pages.
  2. Vérifier que les modalités de navigation sont similaires dans leur présentation (emplacement sur le document final, emplacement dans le contenu sans les feuilles de styles, couleurs, police). et dans leur comportement.
- Résultats attendus : validé si #2 vrai. Non validé si #2 faux.
- Entièrement automatisable : non.

## Tests UWEM 1.0 pour la recommandation 14 des WCAG 1.0

Ensure that documents are clear and simple.

(Voir <http://www.w3.org/TR/WCAG10/#gl-facilite-comprehension>)

Cette recommandation fournit des informations sur la manière de créer des document clairs et simples.

Sommaire: [Point de contrôle 14.1](#)

### Point de contrôle 14.1

Use the clearest and simplest language appropriate for a site's content. [Priority 1]

(Voir <http://www.w3.org/TR/WCAG10/#tech-simple-and-straightforward> et les techniques à <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-simple-and-straightforward>)

### Tests basés sur (X)HTML

#### *Test 14.1\_HTML\_01*

Ce test a pour but d'analyser la lisibilité du contenu.

- Critères d'application : tout le document.
- Procédure de test :
  1. Lire le texte.
  2. Vérifier que le langage est simple.
  3. Sinon, vérifier que le langage utilisé dans le document est nécessaire pour fournir l'information.
- Résultats attendus : validé si #2 ou #3 sont vrais. Non validé si #2 et #3 faux.
- Entièrement automatisable : non.

## 6. Agrégation des résultats des tests

Cette section décrit un modèle mathématique et statistique pour calculer la probabilité liée à la notion de "barrière d'accessibilité" pour une page Web ou un site Web complet. Le contenu de cette section n'est consultable que par des personnes avec une compétence mathématique spécifique et ne contient pas d'informations relatives à la méthodologie d'évaluation. Par conséquent, il n'a pas été traduit en français. Cependant, la section 6 est consultable en anglais dans la [version officielle de UWEM 1.0 \(anglais\)](#).

## 7. Rapport des résultats des tests

Dans UWEM, le rapport des résultats des tests doit être défini de façon non ambiguë. A cet effet, le langage le plus approprié est le langage EARL [EARL10-Schema], qui deviendra bientôt une recommandation candidate du W3C. Cependant, il faut reconnaître que ce format n'est pas le plus approprié pour l'évaluation expert, en particulier lorsqu'aucun outil supportant EARL n'est disponible ou que l'expert n'a pas de connaissances en RDF.

Par conséquent, UWEM donne la possibilité d'un format alternatif à EARL suivant une méthode descriptive selon le modèle de l'[annexe B](#). Ce modèle est basé sur celui de la [suite sur l'évaluation de W3C/WAI \(anglais\)](#) qui sert de modèle pour les rapports d'évaluation de l'accessibilité.

Le niveau de détail minimum proposé pour le rapport d'une évaluation expert est par point de contrôle pour l'échantillon entier. Un rapport plus détaillé (tel qu'un rapport par test et/ou un rapport pour chaque ressource échantillonnée) est optionnel.

Le langage EARL est un cadre pour exprimer les résultats des tests. Le concept de "test" dans EARL est pris au sens plus large et peut inclure des rapports d'erreurs, des tests pour des unités de logiciels, des évaluations de suites de tests, et des déclarations de conformité, entre autres. EARL est basé sur RDF.

## 8. Rapport d'analyse évolutive

Cette section 8 décrit une méthode de rapport spécialement conçue pour les décideurs politiques à l'aide d'une méthode basée sur des scorecard. Le principe général de cette méthode est de permettre la comparaison de résultats obtenus par des évaluations d'un même site Web suivant une méthodologie d'évaluation de même type. Ainsi, il devient possible d'observer l'évolution du niveau d'accessibilité d'un site Web.

Le contenu de cette section fait référence en particulier au modèle mathématique de "barrière d'accessibilité" défini dans la section 6 et ne contient pas d'informations relatives à la méthodologie d'évaluation. Par conséquent, il n'a pas été traduit en français. Cependant, la section 8 est consultable en anglais dans la [version officielle de UWEM 1.0 \(anglais\)](#).

## 9. Glossaire

Le glossaire en anglais n'a pas été traduit en français car cette traduction n'était pas nécessaire pour une bonne compréhension du contenu de cette traduction. Cependant, la section 9 est consultable en anglais dans la [version officielle de UWEM 1.0 \(anglais\)](#).

## 10. Références

[BRIN98] Brin S, Page L (1998). The anatomy of a large-scale hypertextual Web search engine. In: Enslow P H, Ellis A (eds), Proceedings of the Seventh international Conference on World Wide Web 7 (Brisbane, Australia), pp. 107—117. Amsterdam: Elsevier Science Publishers B. V. DOI=[http://dx.doi.org/10.1016/S0169-7552\(98\)00110-X](http://dx.doi.org/10.1016/S0169-7552(98)00110-X)

[EARL10-Schema] McCathieNevile C, Abou-Zahra S (eds) (2006). Evaluation and Report Language (EARL) 1.0 Schema. W3C Editor's Working Draft. Available at: <http://www.w3.org/WAI/ER/EARL10/WD-EARL10-Schema-20060101>

[KAPLAN96] Kaplan R S, Norton D P (1996). The Balanced Scorecard: Translating Strategy into Action. Boston, MA: Harvard Business School Press.

[HENZINGER00] Monika R. Henzinger, Allan Heydon, Michael Mitzenmacher, Marc Najork (2000). On Near-Uniform URL Sampling. In Proc. of the 9th Int. World Wide Web Conference, May 2000.

[HTTP-RDF] Koch J, Velasco C A (2006). HTTP in RDF. W3C Note 14 February 2006. Available at: <http://www.w3.org/WAI/ER/HTTP/WD-HTTP-in-RDF-20060214>

[KRUG00] Krug S (2000). Don't Make Me Think: A Common Sense Approach to Web Usability. New Riders Press (2nd edition).

[LEVENE99] Levene M, Loizou G (1999). Navigation in Hypertext is easy only sometimes. SIAM Journal on Computing, 29, pp. 728—760.

[RDFS] Brickley D, Guha R V (2004). RDF Vocabulary Description Language 1.0: RDF Schema - W3C Recommendation, 10 February 2004. Available at: <http://www.w3.org/TR/rdf-schema/>

[RFC2119] Bradner S (ed) (1997). Key words for use in RFCs to Indicate Requirement Levels. Request for Comments: 2119. IETF. Available at: <http://www.ietf.org/rfc/rfc2119.txt>

[RFC2616] Fielding R, Gettys J, Mogul J, Frystik H, Masinter L, Berners-Lee T (eds) (1999). Hypertext Transfer Protocol – HTTP/1.1. Request for Comments: 2616. IETF. Available at: <http://www.ietf.org/rfc/rfc2616.txt>

[RFC3986] Berners-Lee T, Fielding R, Masinter L (eds) (2005). Uniform Resource Identifier (URI): Generic Syntax. Request for Comments: 3986. IETF. Available at: <http://www.ietf.org/rfc/rfc3986.txt>

[WCAG10] Chisholm W, Vanderheiden G, Jacobs I (eds) (1999). Web Content Accessibility Guidelines 1.0, W3C Recommendation 5-May-1999. World Wide Web Consortium. Available at: <http://www.w3.org/TR/WCAG10/>

[WCAG10-TECHS] Chisholm W, Vanderheiden G, Jacobs I (eds) (1999). Techniques for Web Content Accessibility Guidelines 1.0, W3C Note 6 November 2000. World Wide Web Consortium. Available at: <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/>

[WCAG20] Caldwell B, Chisholm W, Slatin J, Vanderheiden G, White J (eds) (2005). Web Content Accessibility Guidelines 2.0, W3C Working Draft 30 June 2005. World Wide Web Consortium. Available at: <http://www.w3.org/TR/WCAG20/>

[ZENG04] Zeng X (2004). Evaluation and Enhancement of Web Content Accessibility for Persons with Disabilities. Ph.D. Thesis. University of of Pittsburgh. Available at: [http://etd.library.pitt.edu/ETD/available/etd-04192004-155229/unrestricted/XiaomingZeng\\_April2004.pdf](http://etd.library.pitt.edu/ETD/available/etd-04192004-155229/unrestricted/XiaomingZeng_April2004.pdf)

## 11. Annexe A : licence du document

Copyright © 2005 Commission Européenne et membres du [WAB Cluster](#).

[Licence basée sur la licence des documents du W3C [49]. **Note** : cette licence s'applique spécifiquement aux documents UWEM et du WAB Cluster. Aucun des documents référencés dans ce document émanant du W3C ou de son initiative WAI ne sont sujets aux conditions de cette licence.]

En utilisant et/ou copiant ce document (Unified Web Evaluation Methodology), ou le document depuis lequel cette citation est liée, vous (la personne qui utilise un document sous cette licence) déclarez avoir lu, compris et accepté de vous conformer aux termes et conditions suivants :

Permission de copier, et distribuer les contenus de ce document ou du document depuis lequel cette citation est liée, sous toute forme, pour toute cause et sans qu'aucune rémunération ou droit ne soit accordé à condition que vous incluiez les informations suivantes sur **toutes** les copies du document (ou les portions de celui-ci) que vous avez utilisées :

- Un lien ou une URL vers le document original.
- La notice de droit d'auteur pré-existante du document original ou, si elle n'existe pas, la notice (un lien hypertexte est préférable mais une représentation textuelle est permise) de la forme "Copyright © 2005 Commission Européenne et membres du WAB Cluster. Tous droits Réservés."
- Le statut du document.

Si l'espace est suffisant, l'inclusion du texte intégral de cette **notice** doit être faite. Nous demandons que la référence au nom de l'auteur soit inscrite pour chaque logiciel, document, ou autre article ou produit que vous créerez à partir de l'implémentation des contenus de ce document ou de toute portion de celui-ci.

Cette licence autorise l'utilisation, la modification, et l'extension de ce document à toute organisation sans droits d'auteur, dans les conditions exprimées ci-dessus. Dans le cas de modifications en dehors de l'organisme de normalisation sélectionné ou de l'entité équivalente par le détenteur des droits, ni l'expression Unified Web Evaluation Methodology ni le sigle "UWEM" ne peuvent être utilisés pour dénommer le travail effectué.

**Ce document est fourni "tel quel", et les détenteurs de droits d'auteurs n'assurent aucune garantie, explicite ou implicite, comprenant mais non limitée à des garanties propres aux règles commerciales, aux aptitudes pour un but particulier, non atteinte, ou à des règles de propriété; que les contenus de ce document sont appropriés pour toute cause ; ni que l'implémentation de tels contenus n'enfreindra pas de brevets, droits d'auteurs, marques ou tous autres droits faits pour des tiers.**

**Les détenteurs des droits ne seront pas responsables de tous dommages directs, indirects, spéciaux ou causés suite à l'utilisation de ce document ou de l'exécution ou l'implémentation des contenus de celui-ci.**

Les noms et les marques des détenteurs des droits ne doivent pas être utilisés pour faire la promotion ou la publicité concernant ce document ou ses contenus sans permission préalable

spécifique écrite. Le titre au copyright de ce document restera toujours la propriété des détenteurs des droits.

## 12. Annexe B : modèle de rapport (expert)

Sommaire:

- [12.1 Introduction](#)
- [12.2 Résumé](#)
- [12.3 Contexte de l'évaluation](#)
  - [12.3.1 Site Web évalué](#)
  - [12.3.2 les évaluateurs](#)
  - [12.3.3 Procédure d'évaluation](#)
- [12.4 Liste des ressources](#)
- [12.5 Résultats des vérifications](#)
- [12.6 Références](#)
- [12.7 Annexes](#)

Cette annexe présente un modèle de rapport à utiliser pour l'évaluation expert permettant de présenter les résultats d'une évaluation faite suivant UWEM. Ce modèle est basé sur le modèle de rapport d'évaluations de WAI [50], avec des changements mineurs pour des ajouts exigés par la méthodologie UWEM. Il est à noter que certains points ne peuvent être utilisés que s'ils sont applicables.

[Début du modèle de rapport proposé]

### 12.1 Introduction

Ce rapport d'évaluation décrit l'évaluation du site Web : <http://www.example.org>, selon la méthodologie UWEM 1.0. La liste des ressources dont la conformité a été testée se trouve à <http://www.example.org/tests/test1.rdf>.

### 12.2 Résumé

Ce rapport décrit la conformité du site Web <http://www.example.org> à UWEM 1.0. Les résultats de l'évaluation sont décrits ci-dessous et se basent sur la méthodologie UWEM 1.0 ([version officielle \(anglais\)](#) et [version traduite en français](#)) fournie par le [WAB Cluster](#).

Sur la base de cette évaluation à l'aide des tests UWEM 1.0 pour chaque point de contrôle WCAG 1.0 de priorité 1 et 2, le site Web <http://www.example.org> [répond/ne répond pas] au niveau 1 de conformité UWEM 1.0 (identique au niveau A de WCAG) et/ou niveau 2 (identique au niveau double A de WCAG). Les résultats détaillés de l'évaluation sont disponibles à la [section 12.5](#). Les ressources du site Web qui ont été évaluées (échantillon représentatif du site Web) sont listées à la [section 12.6](#). Tous les commentaires sur cette évaluation sont les bienvenus.

### 12.3 Contexte de l'évaluation

L'évaluation de conformité de l'accessibilité du Web peut être une combinaison d'une évaluation par des outils et d'une évaluation manuelle par un évaluateur expérimenté. Les résultats de l'évaluation contenus dans ce rapport se base sur une évaluation effectuée aux dates suivantes :

\_\_\_\_\_ et à l'aide du/des outil(s) suivant(s): \_\_\_\_\_. Il se

peut que le site Web ait été modifié depuis cette date. Des informations supplémentaires sur la procédure d'évaluation UWEM 1.0 sont disponibles en ligne sur le site du [WAB Cluster](#).

### 12.3.1 Le site Web évalué

- [Nom du site Web]  
[et but du site, si approprié]
- [URL de base du site]
- [Etendue de l'évaluation, en mentionnant les exclusions de façon explicite]
- [Lien vers la liste des ressources échantillonnées]
- [Date ou liste des dates exactes auxquelles l'évaluation a été effectuée]
- [Langue(s) naturelle(s) du site Web]

### 12.3.2 Les évaluateurs

- [Nom de l'évaluateur, ou de l'équipe des évaluateurs, sauf s'ils sont anonymes]
- [Organisme auquel le/les évaluateur(s) est/sont affilié(s), si approprié et s'ils ne sont pas anonymes]
- [Information Contact du/des évaluateur(s) ou de l'organisme du/des évaluateur(s), sauf s'ils sont anonymes]

### 12.3.3 Procédure d'évaluation

- [Niveau de conformité WCAG 1.0 testé à l'aide de UWEM 1.0, c'est-à-dire WCAG 1.0 niveau A, double A.]
- [Procédure UWEM 1.0]

## 12.4 Liste des ressources

Description de l'échantillon des ressources spécifiques pour ce site Web et échantillonnées selon UWEM 1.0.

La liste des ressources échantillonnées se trouve à l'annexe [YY/URL].

## 12.5 Résultats des vérifications

- [Résumé interprétant les résultats de l'évaluation.]
  - [par exemple ce site Web [répond/ne répond pas] au [niveau de priorité A, double A] des WCAG 1.0]
  - [les caractéristiques d'accessibilité respectées par ce site Web incluent \_\_\_\_\_]
- [Résultats détaillés, structurés selon la liste de vérification des WCAG 1.0]
  - [Inclure des liens vers les points de contrôles et les techniques WCAG 1.0 pour tous les points non conformes]
  - [Donner des liens vers des exemples de non conformité]
  - [Joindre ou donner un lien vers des rapports spécifiques dans des annexes comme, par exemple, les résultats de validateurs et d'outils d'évaluation]
  - [Donner des liens vers des recommandations permettant de traiter les points de contrôle non conformes]
- [Si disponible, joindre des résultats de tests plus détaillés en annexe à cette évaluation]

- [Décrire ou mettre un lien vers un programme proposant une surveillance continue de l'accessibilité du site Web, la réévaluation des outils de rédaction, etc...]

## 12.6 Références

- [Web Content Accessibility Guidelines 1.0](#)
- [WAB Cluster](#)
- [Projet EIAO](#)
- [Projet support-EAM](#)
- [Projet BenToWeb](#)
- [Checklist for Web Content Accessibility Guidelines 1.0](#)
- [Techniques for Web Content Accessibility Guidelines 1.0](#)
- [Evaluating Web Sites for Accessibility](#)
- [Evaluation, Repair, and Transformation Tools for Web Content Accessibility](#)
- [Selecting and Using Authoring Tools for Web Accessibility](#) [ébauche]
- [Review Teams for Evaluating Web Site Accessibility](#) [ébauche]

## 12.7 Annexes

Joindre toute annexe nécessaire ici.

**[Fin du rapport proposé]**

## **13. Annexe C: Schémas RDF**

Cette annexe décrit des schémas RDF pour des listes de ressources Web. Elle n'a pas été traduite en français. Cependant, l'annexe C est consultable en anglais dans la [version officielle de UWEM 1.0 \(anglais\)](#).

## 14. Annexe D: Contributeurs

- Eric Velleman
- Mikael Snaprud
- Dominique Burger
- Carlos A Velasco
- Pierre Guillou
- Sylvie Duchateau
- Denis Boulay
- Tanguy Loheac
- Jesse Tan
- Colin Meerveld
- Barry McMullin
- Johanna Bolton
- Nils Ulltveit-Moe
- Johannes Koch
- Annika Nietzio
- Helen Petrie
- Jenny Craven
- Alan Chuter
- Gerhard Weber
- Christophe Strobbe
- Sandor Herramhof
- Evangelos Vlachogiannis
- Alistair Garrison
- Olaf Perlick
- Yehya Mohamad
- Henrike Gappa
- Gabriele Nordbrock
- Dirk Stegemann
- Finn Aslaksen
- Torben Bach Pedersen
- Patrizia Bertini
- Terje Gjørseter
- Morten Goodwin Olsen
- Ole-Christoffer Granmo
- Andreas Prinz
- Agata Sawicka
- Edith Helene Unander

## 15. Annexe E : Licence des documents du W3C

### [\[Lire le texte original de la licence pour les documents du W3C \(anglais\)\]](#)

Les documents publics sur le site du W3C sont fournis par les détenteurs de droits sous la licence suivante. En utilisant, et/ou copiant ce document, ou le document W3C depuis lequel cette citation est liée, vous (la personne qui utilise un document sous cette licence) déclarez avoir lu, compris et accepté de vous conformer aux termes et conditions suivants :

La permission de copier, et distribuer les contenus de ce document ou du document W3C depuis lequel cette citation est liée, sous toute forme, pour toute cause et sans rémunération ou droit est accordée à condition que vous incluiez les informations suivantes sur **toutes** les copies du document ou les portions de celui-ci que vous utilisez :

- Un lien ou une URL vers le document W3C original.
- La notice de droit d'auteur pré-existante du document original ou, si elle n'existe pas, la notice (un lien hypertexte est préférable mais une représentation textuelle est permise) de la forme : "Copyright © [date du document] World Wide Web Consortium, (Massachusetts Institute of Technology, European Research Consortium for Informatics and Mathematics, Keio University). Tous Droits Réservés.  
<http://www.w3.org/Consortium/Legal/2002/copyright-documents-20021231>"
- S'il existe, le **statut** du document W3C.

Si l'espace est suffisant, l'inclusion du texte intégral de cette **notice** doit être faite. Nous demandons que la référence au nom de l'auteur soit donnée pour chaque logiciel, document, ou autre article ou produit que vous créerez selon l'implémentation des contenus de ce document ou de toute portion de celui-ci.

Selon cette licence, aucun droit de créer des modifications ou dérivés des documents W3C n'est accordé. Néanmoins, si des exigences supplémentaires (documentées dans le [Copyright FAQ](#) sur les droits d'auteurs) sont remplies, le droit de créer des modifications ou dérivatifs est parfois accordé par le W3C aux individus qui se conforment à ces exigences.

**Ce document est fourni "tel quel", et les détenteurs de droits d'auteurs n'assurent aucune garantie, explicite ou implicite, comprenant mais non limitée à des garanties propres aux règles commerciales, aux aptitudes pour un but particulier, non atteinte, ou à des règles de propriété; que les contenus de ce document sont appropriés pour toute cause ; ni que l'implémentation de tels contenus n'enfreindra pas de brevets, droits d'auteurs, marques ou tous autres droits faits pour des tiers.**

**Les détenteurs des droits ne seront pas responsables de tous dommages directs, indirects, spéciaux ou causés suite à l'utilisation de ce document ou de l'exécution ou l'implémentation des contenus de celui-ci.**

Les noms et les marques des détenteurs des droits ne doivent pas être utilisés pour faire la promotion ou la publicité concernant ce document ou ses contenus sans permission préalable spécifique écrite. Le titre au copyright de ce document restera toujours la propriété des détenteurs des droits.

Cette formulation de la notice et de la licence du W3C est devenue active le 31 décembre 2002. Cette version supprime la notice sur la détention des droits d'auteur de telle sorte que cette licence peut être utilisée pour d'autres documents que ceux qui sont la propriété du W3C, déplace les informations sur les feuilles de style, les DTD et schémas vers le FAQ sur les droits d'auteur, indique que ERCIM est maintenant un représentant du W3C, inclut des références vers cette version spécifique datée de la licence, et supprime le droit d'"utilisation" ambigu. Voir l'ancienne formulation pour la politique antérieure à cette date. Veuillez consulter notre Copyright FAQ pour les questions communes sur l'utilisation des documents issus de notre site, telles que les spécifications sur la traduction ou l'annotation. Toutes autres questions sur cette notice peuvent être adressées à : [site-policy@w3.org](mailto:site-policy@w3.org).

Joseph Reagle [site-policy@w3.org](mailto:site-policy@w3.org).

Last revised \$Id: copyright-documents-20021231.html,v 1.6 2004/07/06 16:02:49 slesch Exp \$

## 16. Notes

1. L'évaluation de ressources Web via les tests utilisateurs sera traitée dans de futures versions de UWEM.
2. <http://www.w3.org/Consortium/Legal/IPR-FAQ-20000620>
3. <http://www.w3.org/Consortium/Legal/IPR-FAQ-20000620#annotate>
4. <http://www.w3.org/WAI/eval/>
5. <http://www.w3.org/WAI/eval/reviewteams.html>
6. <http://www.w3.org/WAI/eval/selectingtools.html>
7. <http://www.w3.org/WAI/eval/users.html>
8. Noter que si un petit site est évalué entièrement, alors la valeur moyenne des échantillons agrégés peut être exactement calculée. Pour des sites Web plus importants, il est tolérable de sélectionner une sous liste aléatoire de pages Web, tant que la marge d'erreurs de l'intervalle de sûreté de 95% est indiquée.
9. Tant que l'algorithme utilisé sélectionne une série impartiale aléatoire de pages Web, alors l'échantillon est valide et devrait fournir les mêmes résultats dans la marge d'erreurs calculée pour un intervalle de sûreté de 95%.
10. Jusqu'à ce que la marge d'erreurs soit atteinte l'échantillon est basé sur le fait qu'une augmentation de la taille de l'échantillon diminuera la longueur de l'intervalle de sûreté sans réduire le niveau de confiance. C'est parce que la déviation standard du moyen d'échantillon diminue lorsque n augmente.
11. Noter qu'il est aussi possible de déterminer un nombre minimum d'échantillons qui donneront des résultats étant dans une marge d'erreurs donnée même dans le pire des cas avec une variation de 0,5. Néanmoins, cela nécessitera plus d'échantillons que strictement nécessaire pour tous les sites Web qui sont meilleurs (moins de variance) que le pire des cas. Gardant cela en tête, nous pensons qu'il vaut mieux avoir une exigence pour dire la marge d'erreurs du résultat, qu'une exigence sur le nombre d'échantillons.
12. Si du contenu textuel est adjacent à l'élément non textuel l'alternative textuelle peut être ce contenu textuel associé à la valeur de l'attribut alt des éléments non textuels.
13. [L'exemple d'un tel algorithme est disponible dans WCAG 2.0](http://www.w3.org/TR/2006/WD-WCAG20-20060427/appendixA.html#luminosity-contrastdef) à l'adresse <http://www.w3.org/TR/2006/WD-WCAG20-20060427/appendixA.html#luminosity-contrastdef>.
14. Ces propriétés de CSS peuvent indiquer l'URI d'une image à utiliser comme contenu ou arrière-plan.
15. Dans les documents de type XHTML 1.0 l'identifiant public de la déclaration du doctype doit faire référence à l'une des trois DTDs en utilisant l'identifiant public formel respectif. Voir 3.1.1 (les documents strictement conformes) du XHTML 1.0. [Extensible HyperText Markup Language \(Second Edition\)](#) - Reformulation of HTML 4 in XML 1.0 recommandation du W3C du 26 janvier 2000 révisée le 1er août 2002.
16. <http://www.w3.org/TR/1999/REC-html401-19991224/types.html#type-multi-length>
17. <http://www.w3.org/TR/1999/REC-html401-19991224/types.html#type-multi-length>
18. La spécification CSS 2.0 liste 'px' (pixel) comme une unité relative. Néanmoins, la taille d'un pixel est relative à l'affichage de l'ordinateur et non à toute propriété définie dans le contenu Web. La spécification CSS 2.0 définit également un "pixel de référence" avec une taille absolue.
19. Celles-ci sont définies comme des valeurs absolues à <http://www.w3.org/TR/1998/REC-CSS2-19980512/fonts.html#value-def-absolute-size>.
20. Les éléments H6 n'ont pas besoin d'être vérifiés car ils représentent le niveau de titre le plus bas.

21. Il n'est pas nécessaire de vérifier H1 et h2 parce qu'il n'y a pas de niveau d'en-tête qui soit deux niveau supérieur.
22. Dans beaucoup de langues un segment de texte correspond à des mots isolés ou à des groupes de mots ; dans certaines langues, en particulier, celles ayant un système d'écriture par idéogrammes, un segment de texte peut même correspondre à un seul caractère.
23. L'en-tête http 'Content-Language' n'est pas pris en compte ici car cet en-tête ne devrait pas être utilisé pour le traitement de texte. Voir pour plus d'informations <http://www.w3.org/TR/i18n-html-tech-lang/#ri20040808.110827800>.
24. Tous les éléments excepté base, basefont, head, html, meta, param, script et title.
25. Noter les informations suivantes issues de la section "[Inheritance of text direction information](#)" de la norme HTML 4.01 à <http://www.w3.org/TR/html401/struct/dirlang.html#blocklevel-bidi> : "The Unicode bidirectional algorithm requires a base text direction for text blocks. To specify the base direction of a block-level element, set the element's dir attribute. The default value of the dir attribute is "ltr" (left-to-right text). When the dir attribute is set for a block-level element, it remains in effect for the duration of the element and any nested block-level elements. Setting the dir attribute on a nested element overrides the inherited value. (...) Inline elements, on the other hand, do not inherit the dir attribute. This means that an inline element without a dir attribute does not open an additional level of embedding with respect to the bidirectional algorithm. (Here, an element is considered to be block-level or inline based on its default presentation. Note that the INS and DEL elements can be block-level or inline depending on their context.)" Noter également les informations ci-dessous issues de la section "[The effect of style sheets on bidirectionality](#)" dans la norme HTML 4.01: "In general, using style sheets to change an element's visual rendering from block-level to inline or vice-versa is straightforward. However, because the bidirectional algorithm relies on the inline/block-level distinction, special care must be taken during the transformation. When an inline element that does not have a dir attribute is transformed to the style of a block-level element by a style sheet, it inherits the dir attribute from its closest parent block element to define the base direction of the block. When a block element that does not have a dir attribute is transformed to the style of an inline element by a style sheet, the resulting presentation should be equivalent, in terms of bidirectional formatting, to the formatting obtained by explicitly adding a dir attribute (assigned the inherited value) to the transformed element."
26. Tous les éléments excepté base, basefont, head, html, meta, param, script et title.
27. <http://www.w3.org/TR/1998/REC-CSS2-19980512/generate.html#before-after-content>
28. <http://www.w3.org/TR/i18n-html-tech-bidi/#ri20030728.092130948>
29. <http://www.w3.org/International/questions/qa-bidi-css-markup.html>
30. <http://www.w3.org/TR/REC-CSS2/visuren.html#direction>
31. <http://www.w3.org/TR/1999/REC-html401-19991224/struct/tables.html#h-11.4>
32. Le W3C a un [service pour la linéarisation de tableaux](#).
33. <http://www.w3.org/TR/1998/REC-CSS2-19980512/conform.html#x10>
34. <http://www.w3.org/TR/1999/REC-html401-19991224/present/styles.html#h-14.6>
35. Tous les attributs sur les événements intrinsèques, sauf onload. Voir les définitions des attributs dans la section "[Intrinsic events](#)" de html 4.01.
36. La plupart des navigateurs déclencheront l'événement onclick à l'appui sur la touche "entrée". Il n'est donc pas requis d'implémenté onclick lorsque onkeypress est spécifié.
37. <http://www.w3.org/WAI/Resources/WAI-UA-Support>
38. Le sélecteur //object[@type='image/gif'] ignorera certains fichiers gif si l'attribut type de l'élément object n'est pas défini ou incorrect.

39. Le sélecteur //object[@type='image/gif'] ignorera certains fichiers gif si l'attribut type de l'élément object n'est pas défini ou incorrect.
40. Les éléments object pour les applets Java peuvent avoir un attribut codetype avec les valeurs 'application/java', 'application/java-archive' ou 'application/x-java-applet'. Le test s'applique également aux contenus envoyés par HTTP avec les types MIME 'application/java', 'application/java-archive' ou 'application/x-java-applet'.
41. L'élément object pour le contenu vidéo peut avoir un attribut type dont la valeur commence par 'video/'. Le test s'applique également à tout contenu envoyé par HTTP avec le type MIME commençant par 'video/'.
42. Les éléments object pour les applets Java peuvent avoir un attribut codetype avec les valeurs 'application/java', 'application/java-archive' ou 'application/x-java-applet'. Le test s'applique également aux contenus envoyés par HTTP avec les types MIME 'application/java', 'application/java-archive' ou 'application/x-java-applet'.
43. L'élément object pour le contenu vidéo peut avoir un attribut type dont la valeur commence par 'video/'. Le test s'applique également à tout contenu envoyé par HTTP avec le type MIME commençant par 'video/'.
44. <http://www.w3.org/TR/2002/REC-UAAG10-20021217/>
45. <http://www.w3.org/TR/2000/REC-ATAG10-20000203/>
46. Ce test est automatisable car il produit toujours le résultat "non validé", puisque toute figure géométrique peut être définie par un polygone et tout pixel peut être défini comme un cercle de rayon 0.
47. [Section 4.10 de la spécification XHTML 1.0](http://www.w3.org/TR/xhtml1/#h-4.10) : "les éléments avec l'attribut id et name", (<http://www.w3.org/TR/xhtml1/#h-4.10>).
48. C'est la [définition des WCAG 2.0](#).
49. <http://www.w3.org/Consortium/Legal/2002/copyright-documents-20021231>. Notre licence autorise des extensions et des modifications de UWEM, tant que les références vers le document original sont données et des copies de cette licence sont fournies.
50. <http://www.w3.org/WAI/eval/template.html>

## 17. Téléchargement

La [version officielle de UWEM 1.0 \(anglais\)](#) est consultable en ligne et téléchargeable sur le site du WAB Cluster.

La [traduction en français de UWEM 1.0 est consultable en ligne](#) en html [<http://www.accessiweb.org/fr/uwem/index.html>].

Elle est également téléchargeable en format Word, Open Office et en PDF:

- [Fichier Word de UWEM 1.0 en français \(804.0 ko\)](#).
- [Fichier Open Office zippé de UWEM 1.0 en français \(120.2 ko\)](#).
- [Fichier PDF de UWEM 1.0 en français \(315.3 ko\)](#).

## 18. Remerciements

L'[association BrailleNet](#) tient à remercier tous les partenaires du projet [Support EAM](#) pour leur travail entre octobre 2004 et juillet 2006 sur la création d'une méthodologie d'évaluation de l'accessibilité du Web harmonisée au niveau européen, étape nécessaire à la création du futur label européen.

En particulier, l'association BrailleNet remercie toute l'équipe [AccessiWeb](#) pour tous ses apports (les travaux méthodologiques ont été menés par Denis Boulay et Pierre Guillou). Enfin, un remerciement appuyé à Tanguy Lohéac et tout particulièrement à Sylvie Duchateau pour leur travail de traduction en français de ce document UWEM 1.0.