



Accessibilité & Handicap

Objectif

L'objectif du TP est de vous familiariser avec l'évaluation de ressources numériques accessibles d'une part et aux problèmes liés à l'écriture de code « inclusif » d'autre part.

Documents de référence utilisés

Il existe de nombreuses ressources en ligne permettant de comprendre les enjeux de l'accessibilité et les différentes solutions proposées

- **Accessibility Cheatsheet** : <https://moritzgiessmann.de/accessibility-cheatsheet>
- **Getting Started with Website Accessibility** : <https://medium.com/statuscode/getting-started-with-website-accessibility-5586c7febc92>
- **ARIA (Accessible Rich Internet Application)** : <https://www.w3.org/TR/using-aria>

Si la plupart des solutions exposées semblent l'être pour des mal ou non-voyants, le handicap ne se limite pas à cette problématique tant les obstacles à l'accessibilité et les situations sont nombreux. Non-voyants, mal-voyants, daltoniens, dyslexiques, ... pensez l'accessibilité, c'est aussi penser à des usages où chacun peut être en situation de handicap (mains, yeux occupés à une autre tâche, bruit ambiant, etc.).

Comprendre la situation, c'est déjà commencer à répondre à la problématique en utilisant le plus souvent des modalités alternatives.

Nous allons donc explorer quelques situations, problématiques et proposer quelques voies de solution.

Préambule - solutions techniques : l'exemple des lecteurs d'écran

Essayez un lecteur d'écran (utilisé par les non-voyants). Vous pouvez par exemple utiliser *Narrateur* sous windows, *VoiceOver* sous MacOS ou télécharger NVDA (Non Visual Desktop Access) ici → <https://www.nvda-fr.org> (pour une version portable, télécharger la version ici : <https://www.softpedia.com/get/PORTABLE-SOFTWARE/System/System-Enhancements/Portable-NVDA.shtml>)

Utilisez un **lecteur d'écran les yeux fermés** et tentez de naviguer sur le site de l'**Université Toulouse 3** (<https://www.univ-tlse3.fr>) pour y trouver le lien d'accès à l'**intranet ENT étudiant** ((en **n'utilisant que les feedbacks vocaux**)).

Toujours dans le même contexte (retour audio seul), ouvrez ensuite un éditeur de texte et prenez des notes sur votre expérience vécue 😊.

- Quels sont les problèmes que vous avez identifiés ?
- Comment pensez-vous y remédier (solutions matérielles ou logicielles) ?

Un site web à évaluer

- Evaluer le site web de l'Université Toulouse 3 avec les outils suivants :
 - <https://wave.webaim.org>
 - <https://color.ally.com/?wc3>
- Quelles sont les erreurs les plus fréquentes (listez-les)
- Comment y remédier ?

Une mini application accessible à développer

p5.js

Téléchargez l'archive zip **p5js.zip** à l'adresse

<https://github.com/truillet/upssitech/blob/master/SRI/3A/FH/TP/Outils/p5js.zip>

p5.js (<https://p5js.org/get-started>) est la version « javascript » de Processing qui permet le prototypage rapide d'applications interactives pour le web.

- Dézippez l'archive
- Testez et comprenez le code écrit dans **Accessibilite.html**
- Lancer **ReconnaissanceVocale_Simple.html** dans un navigateur web Google Chrome (à cause du système de reconnaissance vocale et de synthèse utilisés).
- Faites de même avec **ReconnaissanceVocale_Continue.html**.

Analysez le code et **modifiez-le** de manière à pouvoir aider l'utilisateur dans ses actions (utilisation de différents feedbacks, ...).

Nota : Vous pouvez utiliser les outils d'accessibilité d'Adobe Color

(<https://color.adobe.com/fr/create/color-contrast-analyzer>) afin de vérifier les contrastes ou créer des palettes adaptées à la vision daltonienne

Développez enfin une mini-application interactive capable de compléter un début de dessin (à l'aide de réseaux de neurones récurrents) **ET accessible** à au moins un handicap (par exemple : handicap moteur, daltonisme, dyslexie, ...).

Cette application est à produire avec le framework **p5.js** (<https://p5js.org>) et les modules **p5.accessibility**, **p5.speech** et **m5.js** voire d'autres disponibles ici :

- <https://p5js.org/libraries>
- ou là pour la réalité augmentée (TopCodes – Réalité Augmentée) :
<https://github.com/sehmon/TIDAL-topcodes-demo>.



Vous pouvez télécharger l'archive zip pour démarrer **dessin.zip** à l'adresse

<https://github.com/truillet/upssitech/blob/master/SRI/3A/FH/TP/Outils/dessin.zip>

L'utilisateur devra être capable de changer de « modèle de dessin » à la voix pour au moins les modèles suivants : ananas (*pineapple*), chat (*cat*), lobster (*homard*), oiseau (*bird*) et trombone.

Une description de la situation devra être proposée sur demande de l'utilisateur (en utilisant le module accessibilité).

Évaluez enfin votre production à l'aide des **WCAG 2.2** - <https://www.w3.org/TR/WCAG22> et produisez un mini-rapport d'évaluation. Itérez sur votre solution en prenant en compte les problèmes identifiés.

Les notes prises durant la séance sont à envoyer par mél