

Accessibilité



Objectif

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

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 : 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.

Vous pouvez essayer les applications :

- **Eye-View** (<https://aveuglesdefrance.org/eyeview-lapplication-de-la-federation-des-aveugles-de-france>) pour des simulations de dégénérescence
- **Chromatic Vision Simulator**
(<https://play.google.com/store/apps/details?id=asada0.android.cvsimulator>) pour une simulation du daltonisme (voir aussi : <https://www.fr.colorlitelens.com/images/tesztek/simulator>)

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

Avant de démarrer, utilisez le logiciel « The vOICe » (<https://www.seeingwithsound.com>), une solution de substitution sensorielle. Parcourez l'espace

Essayez maintenant 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> (pour une version portable, télécharger la version ici : <https://portableapps.com/apps/accessibility/nvda-portable>)

Utilisez NVDA les yeux fermés et tentez de naviguer sur le site de l'**Université Toulouse 2** (<https://www.univ-tlse2.fr>) pour y trouver le lien d'accès aux informations relatives à la CVEC (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 2** 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 ?

Un framework de développement à tester

p5.js

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

<https://github.com/truillet/uftmip/blob/master/m2ice/TP/code/accessibilite.zip>

P5.js (<https://p5js.org>) est la version « javascript » de Processing qui permet le prototypage rapide d'applications interactives pour le web.

Le framework **p5.js** (<https://p5js.org>) comprend un module interne d'accessibilité (<https://p5js.org/tutorials/writing-accessible-canvas-descriptions>), **p5.speech** et bien d'autres disponibles sur le site.

- Dézippez l'archive et lancez **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**. Regardez le code et modifiez-le de manière à aider l'utilisateur dans ses actions de déplacement avec un feedback vocal (TTS) par exemple.
- Testez et comprenez enfin le code écrit dans **accessibilite.html** (à tester notamment avec un lecteur d'écran)