

## Checklist accessibilité

### Rapide checklist pour une pull request

Vérifier :

- Images et icônes ont une alternative textuelle
- L'orientation de l'écran est supportée
- Les champs de saisie sont labellisés
- Les textes peuvent être agrandis
- L'information ne s'appuie pas uniquement sur la couleur
- Les contrastes sont respectés
- Les zones interactives ont une taille minimale
- Il n'y a pas de flash lumineux (épilepsie)

### Checklist complète avant mise en production

Avant de mettre en production votre application, il vous est conseillé de faire, pour chaque écran, une vérification de l'ensemble des critères indiqués sur cette fiche. Vous trouverez aussi une liste de critères plus exhaustives sur le site des recommandations accessibilité Orange, ainsi que sur la va11ydette :

<https://la-va11ydette.orange.com/?list=wcag-ios&lang=fr>

## Options supplémentaires

### Masquer des éléments à l'accessibilité

Il est possible de masquer des éléments aux outils d'assistance (Voiceover par exemple).

```
AccessibilityElement  
AccessibilityElementIsHidden  
AccessibilityViewIsModal
```

### Déclencher une vocalisation

Il est possible de faire parler le lecteur d'écran (lorsque celui-ci a été activé par l'utilisateur).

```
UIAccessibilityAnnouncementNotification  
UIAccessibilityPostNotification
```

### Connaître l'état du lecteur d'écran

Dans certains cas il peut être intéressant de savoir si l'utilisateur navigue à l'aide du lecteur d'écran.

```
UIAccessibilityIsVoiceOverRunning
```

## Gestes de base de Voice Over

### Activer le lecteur d'écran

iOS : Réglages > Général > Accessibilité > VoiceOver

### Déplacement avec un doigt sur l'écran



Pour explorer l'écran et entendre la vocalisation de l'élément qui est touché.

### Double-cliquer n'importe où sur l'écran



Pour ouvrir ou activer l'élément qui a été touché (vocalisé) en dernier.

### Slide vers la droite en utilisant un doigt



Pour déplacer le focus sur l'élément suivant.

### Slide vers la gauche en utilisant un doigt



Pour déplacer le focus sur l'élément précédent.

Photo credit: openexhibits (CC Share Alike 2.0.).

## Outil de test automatique

### Accessibiliy Inspector

Application disponible sur MAC OS par défaut permettant d'effectuer des tests d'accessibilité sur votre application.



Critères d'accessibilité à vérifier dès la phase de conception



# Mémo accessibilité Mobile iOS

E-Accessibility  
Solutions for  
Everyone

## Table des matières

Checklist accessibilité  
Options supplémentaires  
Gestes de base de Voice Over  
Outil de test automatique  
Critères accessibilité



Accessibilité Numérique Orange  
<http://a11y-guidelines.orange.com>



## Critères accessibilité 1/3

### Fournir des alternatives textuelles

Pour les images porteuses d'information et tout élément qui nécessite une description complémentaire.

À noter : pas d'alternative pour les images décoratives.

```
accessibilityLabel, accessibilityValue,  
accessibilityTrait, accessibilityHint
```

### Ne pas s'appuyer seulement sur la couleur

Utiliser un autre moyen que la couleur pour passer l'information.

### Vérifier le contraste des couleurs

Utiliser l'outil **Show Color Contrast Calculator** :  
Disponible sous **Windows > Show Color Contrast Calculator**  
(sous l'application Accessibility Inspector)

Plus d'infos dans la vidéo suivante à partir de 7:28min :  
<https://www.developer.apple.com/videos/play/wwdc2019/257/?time=94>

### Définir un titre pour chaque écran

Chaque écran doit avoir un titre qui lui est propre pour se repérer dans la navigation.

### Indiquer l'état des éléments

Tout élément dont le statut est modifié durant l'utilisation de l'application doit restituer son état à travers son alternative textuelle (élément sélectionné, section dépliée, onglet sélectionné...).

### Utiliser des composants standards

Utiliser au maximum les composants natifs en ajustant leur apparence. Si aucun composant standard ne permet de répondre au besoin, créer un composant dédié basé sur un composant standard en veillant à conserver la cohérence de navigation et l'accessibilité.

### Définir des zones de clic suffisamment grandes

Se référer aux préconisations de la plateforme.

```
largeur/hauteur + de marge autour de l'élément suffisante
```

## Critères accessibilité 2/3

### Laisser le contrôle à l'utilisateur

Tout contenu en mouvement, mis à jour, clignotant, doit pouvoir être stoppé, caché ou mis en pause par l'utilisateur. Celui-ci doit rester maître de l'application (pas de vidéo qui se lance automatiquement, etc.).

### Prévenir l'utilisateur des changements de contenu

Déclencher la vocalisation d'un message par le lecteur d'écran en cas de changement dynamique du contenu de l'écran (rafraîchissement de liste, compteur de temps, etc.).

```
UIAccessibilityPostNotification  
UIAccessibilityLayoutChangedNotification  
UIAccessibilityScreenChangedNotification
```

### Indiquer les défilements horizontaux

Donner une indication visuelle aux utilisateurs lors de la présence de « scroll » horizontaux (présence de flèches ou de plusieurs points indiquant une pagination). De plus pour les utilisateurs malvoyants : compléter l'alternative textuelle.

```
UIPageControl, accessibilityIncrement,  
accessibilityDecrement
```

### Labelliser les champs de formulaire

Lier les champs de formulaires avec leurs labels.

```
accessibilityLabel
```

### Ordre de lecture

Par défaut, les lecteurs d'écran lisent les éléments dans un ordre logique. Si besoin, il est possible de modifier cet ordre.

```
Protocole UIAccessibilityContainer  
shouldGroupAccessibilityElement
```

### Langue

La vocalisation donnée par VoiceOver s'effectue par défaut dans la langue du système. La modification de la langue se fait depuis les paramètres du téléphone :  
Réglages > Général > Langue et région

## Critères accessibilité 3/3

### Proposer des textes lisibles

Les polices d'écriture avec empâtement, les tailles de texte trop petites, ainsi que l'italique sont à éviter.

### Proposer les deux orientations d'écran possible

L'accès au contenu d'une application ne doit pas dépendre de l'orientation de l'écran (portrait et paysage), sauf si une orientation particulière est essentielle pour la compréhension ou l'utilisation du contenu.

### Proposer une alternative aux actions gestuelles

Les actions déclenchées par un geste de l'utilisateur ou un mouvement de l'appareil doivent disposer d'une alternative.

### Permettre d'adapter les limites de temps

Lorsque le contexte change après une certaine limite de temps (ex: overlay qui disparaît sur un lecteur vidéo après plusieurs secondes), il faut permettre à l'utilisateur d'adapter ce délai ou proposer une alternative à la limite de temps.

### Identifier les erreurs de saisie

Lorsqu'un utilisateur se trompe lors de la saisie de différents champs, il est important de lui indiquer clairement quelles sont ses erreurs, afin qu'il puisse corriger les champs en question sans difficulté (indication visuelle du champ en erreur, suggestion de correction, message d'erreur explicite).

### Gérer la navigation par entêtes

Pour que Voice Over puisse reconnaître les entêtes et ainsi naviguer plus rapidement entre les différentes sections, il faut bien implémenter la barre de navigation (LeftBarItem, Title, RightBarItem)

```
accessibilityLabel
```

## Options supplémentaires

### Taille des textes

Afin de permettre à l'option « grands caractères » d'interagir correctement avec l'application.

```
Dynamic Type API  
Gérer les débordements de contenu (contraintes)
```