



## Introduction à l'accessibilité sur iOS, pour les développeurs

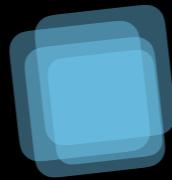
*Une introduction aux problématiques de l'accessibilité dans le domaine du développement mobile.*

Cette présentation a pour objectif de vous présenter ce qu'est l'accessibilité, tout particulièrement dans le numérique et sur la plateforme mobile iOS.

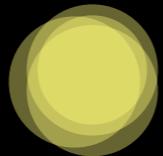
Nous verrons ainsi quels sont les principaux outils mis en œuvre par iOS pour répondre à cette problématique, et surtout pourquoi les développeurs mobiles doivent y attacher de l'importance.

Cette présentation est particulièrement orientée vers iOS/watchOS/tvOS/macOS.

Cependant, les grands principes restent les mêmes sur toutes les plateformes, y compris le web et les ordinateurs de bureau.



## Accessibilité ?



## Outils



## Mise en œuvre

- **L'accessibilité :**
  - c'est quoi ?
  - c'est pour qui ?
  - pourquoi s'en préoccuper ?
- **Les outils** : ce que propose iOS pour les utilisateurs.
- La **mise en œuvre** : entre ne rien faire et un minimum d'effort on peut obtenir de grandes différences.

# Accessibilité ?



#a11y

Une première partie en forme d'introduction pour essayer de **définir** l'accessibilité, le **public** des utilisateurs concernés ainsi les **motivations** à sa mise en œuvre.

## Accessibilité Numérique

- **adaptation** des systèmes numériques,  
*dont les sites web*, aux différents types de handicap ;
- développement d'**outils spécifiques**

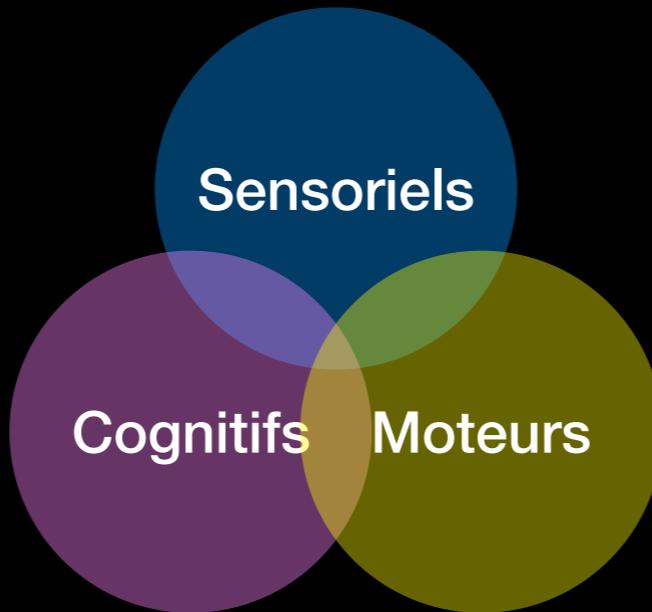
Une définition directement extraite de la page Wikipédia :

<https://fr.wikipedia.org/wiki/Accessibilit%C3%A9>

# Quels Publics ?

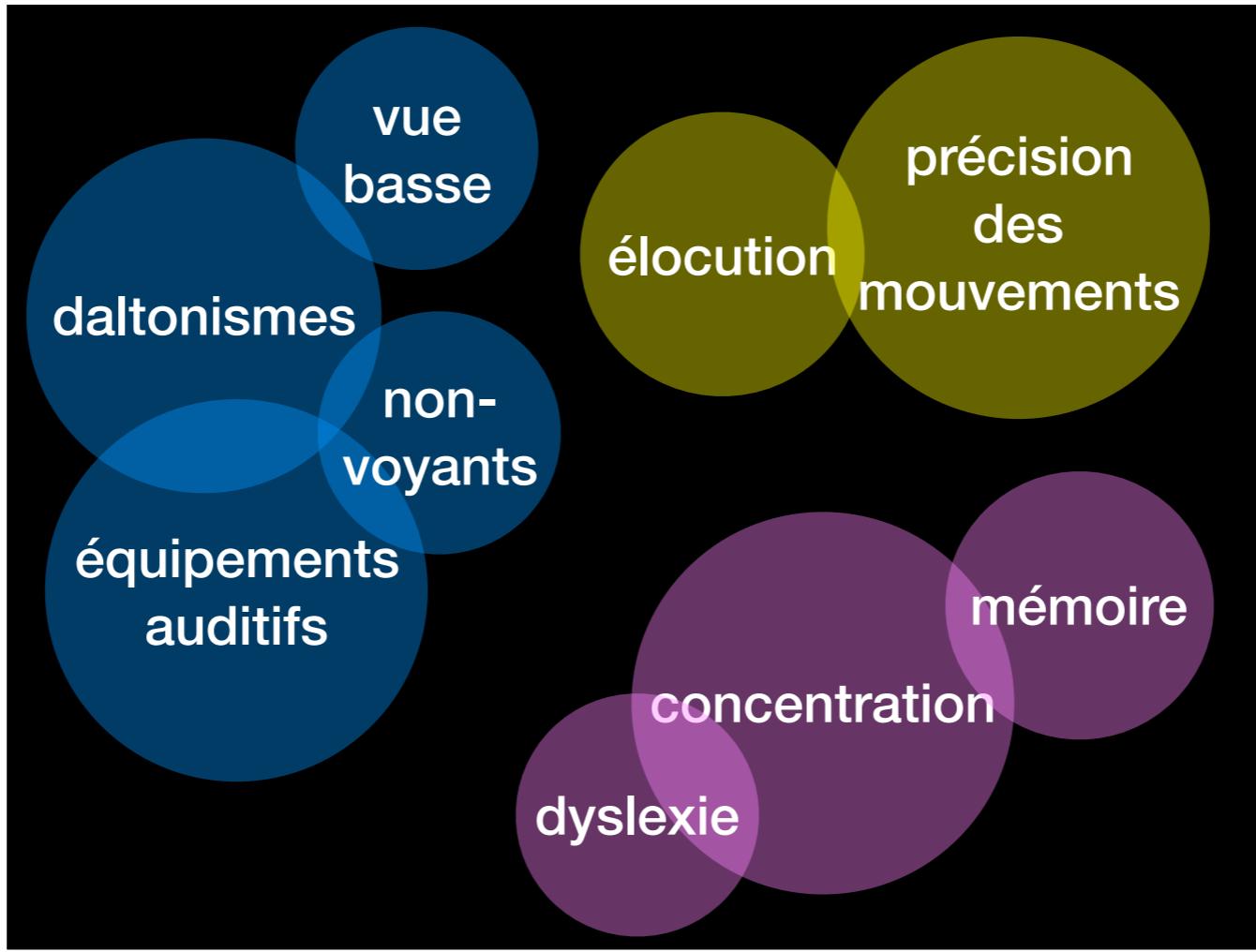
Le public ne se limite certainement pas aux seules populations considérées comme *handicapés*.

# Handicaps



Quels sont les principaux types de handicaps ?

- **Sensoriels** : la vue, l'ouïe
- **Moteurs** : capacités motrices, précision
- **Cognitifs** : troubles de l'attention, lecture



### Troubles sensoriels

- *Non-voyants* : VoiceOver
- *Mal-voyants* : VoiceOver, texte dynamique, les réglages de contraste, les animations...
- *Daltoniens* : la palette de couleurs, les contrastes
- *Malentendants* : habillage sonore

### Troubles moteurs

- *Précision du mouvement* : utilisation des gestes
- *Élocution* : à prendre en compte si vous envisagez d'utiliser le contrôle vocal.

### Troubles cognitifs

- *Dyslexie* : attention aux polices, peut-être aux icônes.
- *Concentration* : ergonomie générale, flux et enchaînements des actions.

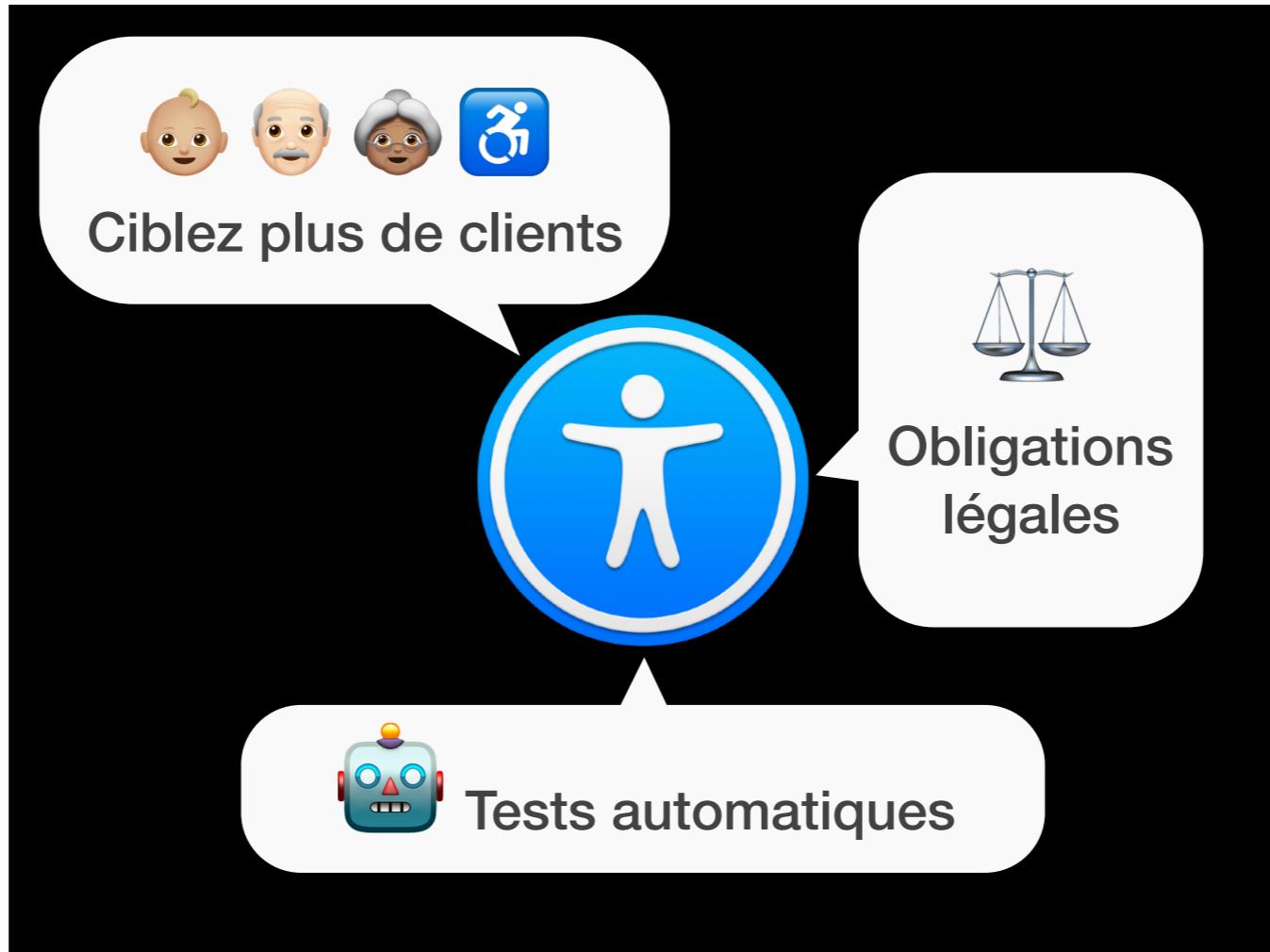
# Pourquoi ?

Nous avons vu ce qu'est l'accessibilité.

Nous avons identifié une partie du public concerné.

Mais pourquoi, si cela ne vous est pas évident, faut-il augmenter le coût de votre application pour prendre en charge l'accessibilité?

***Spoil:*** introduire l'accessibilité après la fin du projet vous coutera *encore plus cher*.



#### **Adressez plus d'utilisateurs**

- une population vieillissante (1/3 de la population française à plus de 50 ans) ;
- les communautés handicapées sont actives et feront volontiers la promotion des applications les mieux adaptées à leurs besoins ;
- le daltonisme, par exemple, touche environ 8% de la population française, 10% aux USA...
- pas seulement pour les handicaps, mais aussi pour certains cadres de travail où l'usage des mains n'est pas possible. Dans ce cas, le pilotage par des boutons de contrôle et des mouvements de tête est une solution.

#### **La réglementation**

- Le DOT impose, depuis fin 2016, aux compagnies aériennes de proposer un site accessible pour les usagers.
- Chaque appel d'offres pour le gouvernement US doit respecter des critères d'accessibilité.
- L'Europe impose progressivement l'accessibilité. Depuis mai 2016, les services/sites/application des services publics doivent être accessibles.

#### **Les tests automatiques**

- l'accessibilité est la base pour créer des tests automatiques de non-régression pour vos applications.
- Vos applications n'en seront que meilleures et plus stables.

**Bonus:** votre karma pour rendre le monde meilleur !

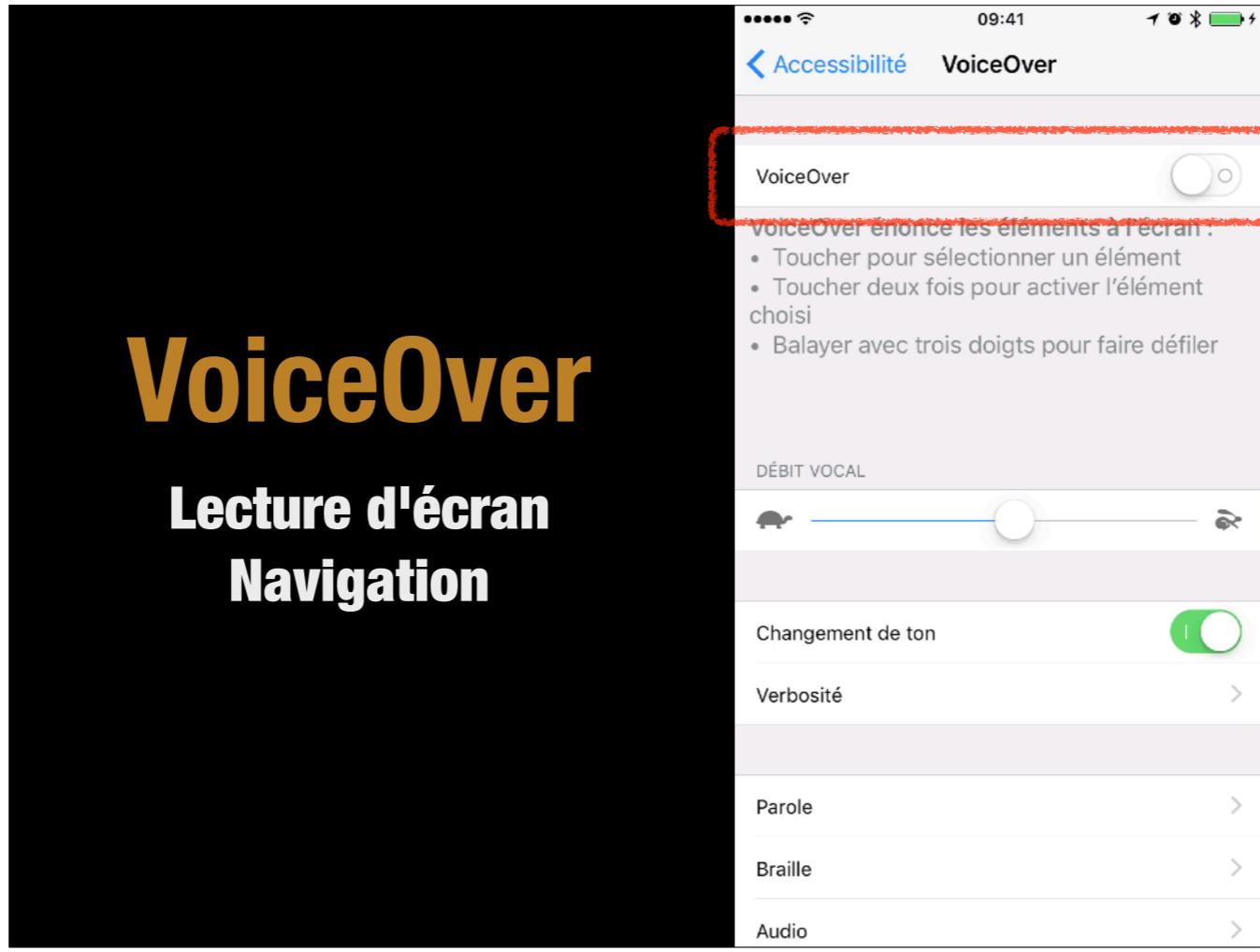
# Les Outils



Dans cette seconde partie, nous faire un rapide tour d'horizon des principales technologies offertes par iOS.

Ce n'est pas un inventaire complet, loin de là.

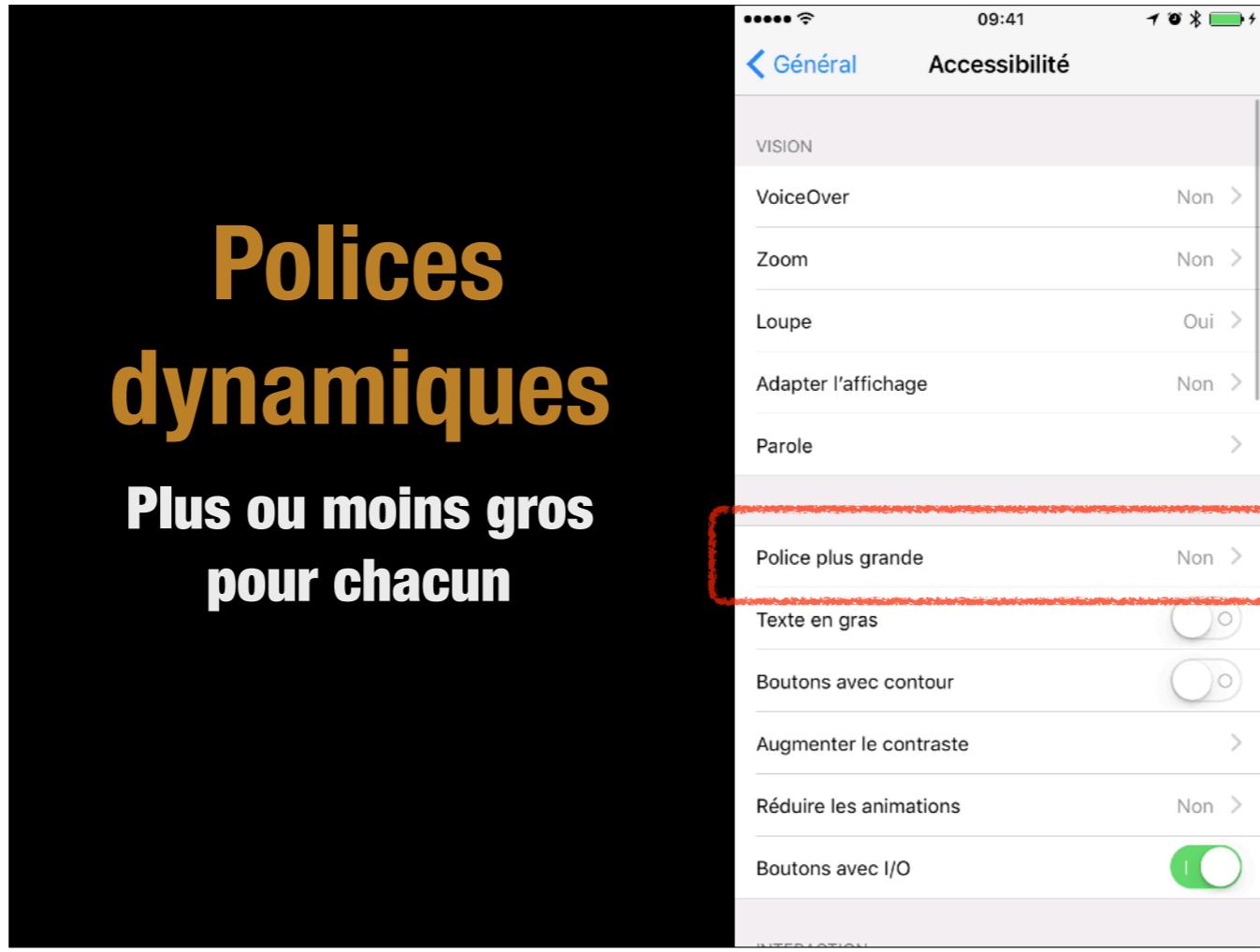
Je vais rapidement vous présenter uniquement celles qu'il est important de prendre en compte pour que vos applications soient de bonnes citoyennes dans l'écosystème de vos utilisateurs.



**VoiceOver** est la mise en œuvre d'une combinaison d'outils, dont le plus visite est la *lecture de l'écran*.

Principalement destiné aux non-voyants, et mal voyants, VoiceOver est intégré très largement dans le système.  
Sa mise en œuvre demande peu d'efforts.

C'est également relativement simple à utiliser et donc de valider votre implémentation.



Les **polices dynamiques** donnent un *contrôle plus fin à l'utilisateur* sur le rendu visuel du texte.

Directement dans les réglages il est ainsi possible de choisir la taille des caractères ainsi que leur graisse.

La mise en œuvre de cette fonction impose un effort plus important et peut avoir un impact direct sur le design de votre application.

En règle générale, cela signifie que vous devrez utiliser la police du système et utiliser des styles de textes plutôt que définir vous même la taille des caractères.



**AssistiveTouch** permet de réaliser des actions et des gestes *de façon indirecte* en utilisant un menu contextuel.

Utilisez autant que possible des gestes standard d'un système permettra à votre application d'être utilisable directement avec cet outil.

# Mise en œuvre



Dans cette dernière partie, nous allons rapidement voir comment prendre en charge l'accessibilité pour VoiceOver.

Nous allons procéder en trois étapes :

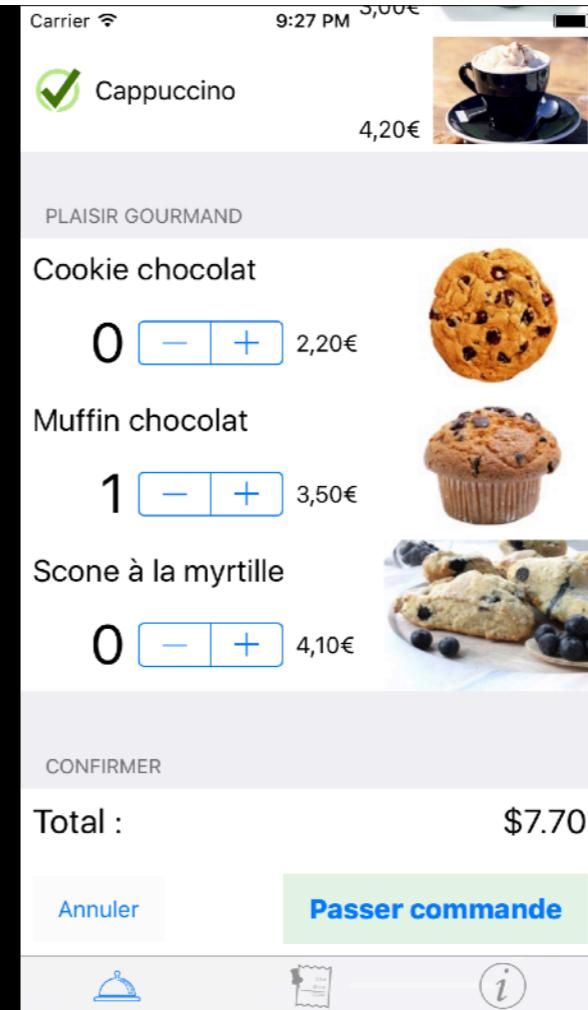
- ce qui est proposé en standard, sans rien faire ;
- comment ajuster le support de VoiceOver ;
- comment en profiter pour automatiser des tests.

**Le projet sur Github:**

<https://github.com/sylvaingml/AllDayJava>

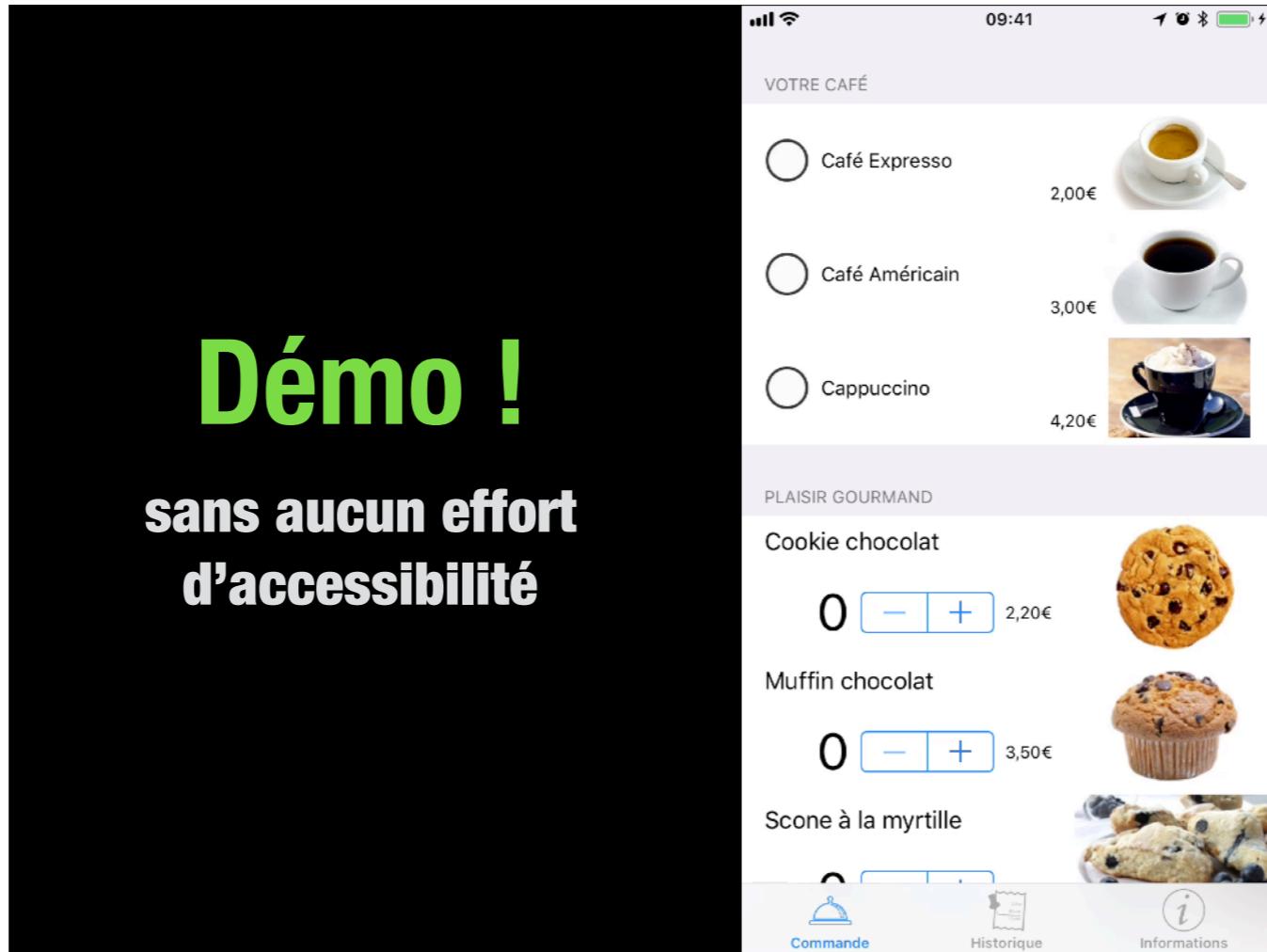
# Notre Application

- navigation par onglets
- boutons
- des champs
- des images
- dialogues



L'application exemple est très simple et permet de passer une commande virtuelle d'un café et de pâtisseries.

- la sélection du café avec des boutons à 2 états
- le choix des gâteaux avec :
  - un champ pour saisir la quantité
  - une paire de boutons pour augmenter/réduire cette quantité
- Le total
- Les actions de confirmation/annulation de la commande.



Tout fonctionne relativement bien sans trop d'efforts. Cependant:

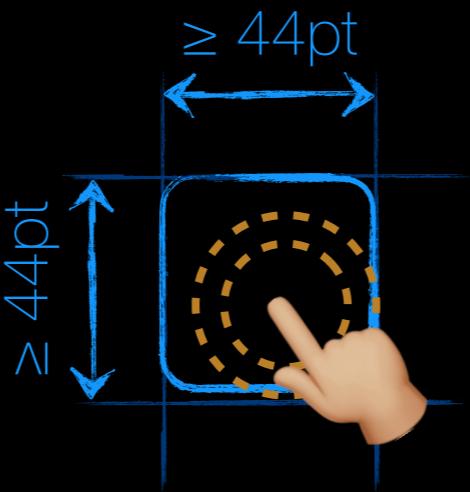
- l'ordre de lecture des cellules pour la sélection du café par VoiceOver n'est pas conforme à l'ordre de lecture naturel.
- les images sont invisibles.
- il n'est pas évident d'associer le champ de quantité avec la pâtisserie correspondante.

# Ajustements

- Taille des **zones sensibles**
- **Décrire** les illustrations
- L'**ordre** de lecture dans le tableau

- Zone sensible souhaitable :  
*on ajustera la mise en forme des boutons d'annulation/confirmation*
- Les illustrations peuvent porter des informations, vous pouvez les préciser.
- L'ordre de lecture est correct avec les types standards de cellules. Il devient plus difficile si vous utilisez des cellules avec une structure personnalisée.

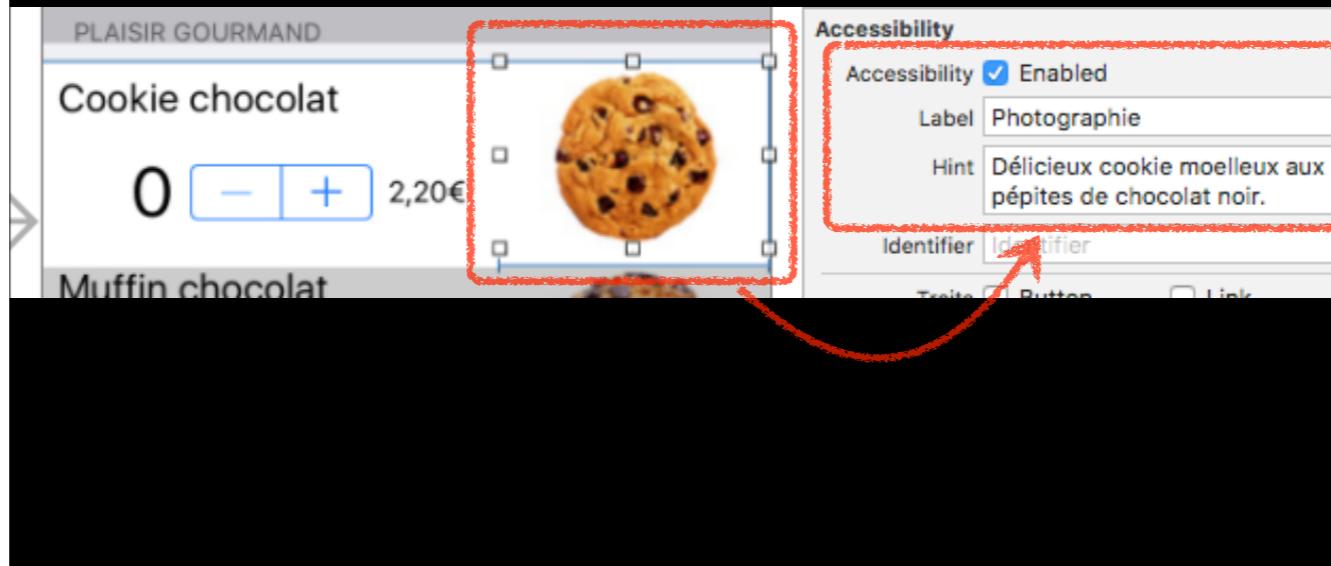
# Cible tactile



Cible « *habituelle* » : 44pt pour un écran de téléphone

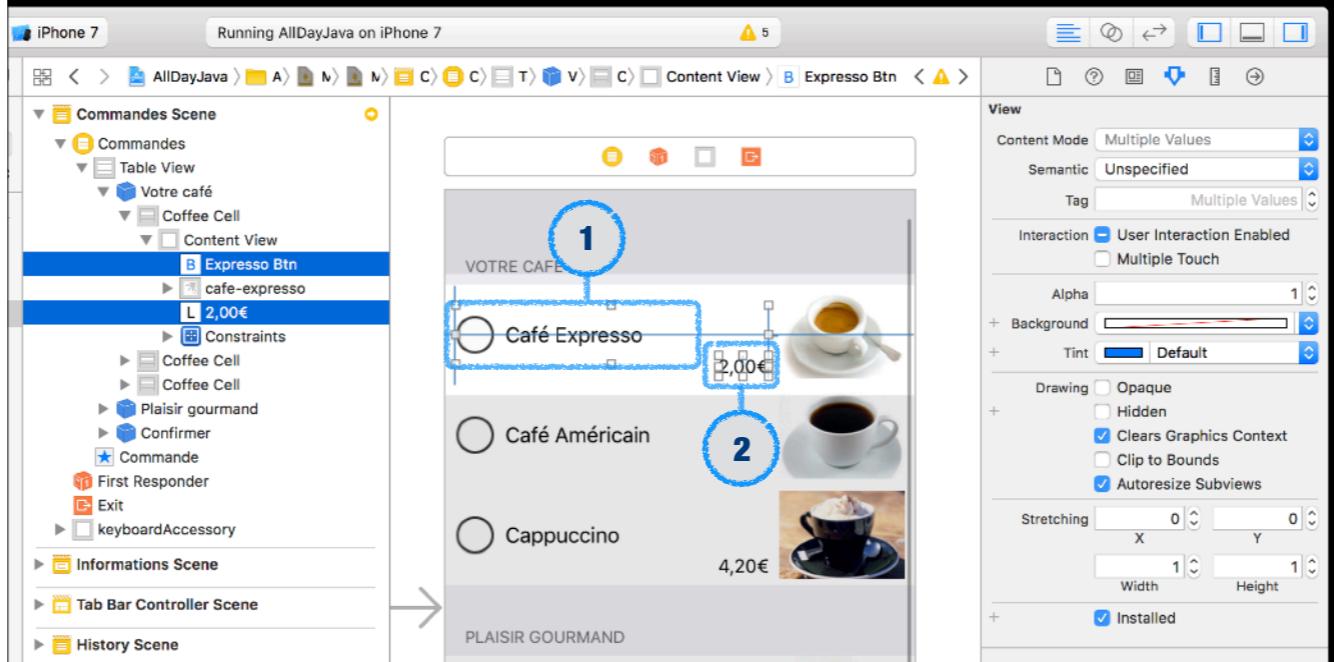
Si vous ciblez une population avec des difficultés motrices, ajustez cette taille que l'on considère comme un minimum.

# Décrire les images



Dans Xcode, cela se fait dans Interface Builder (*l'éditeur d'interfaces*) en renseignant les attributs d'accessibilité: dans l'inspecteur, onglet « identité ».

# Ordonner



## Comment régler le problème d'ordre de lecture ?

Chaque élément est associé à un **identifiant numérique**, le **tag**.

Cet identifiant pourra ensuite être utilisé pour indiquer l'ordre de lecture.

# Ordonner

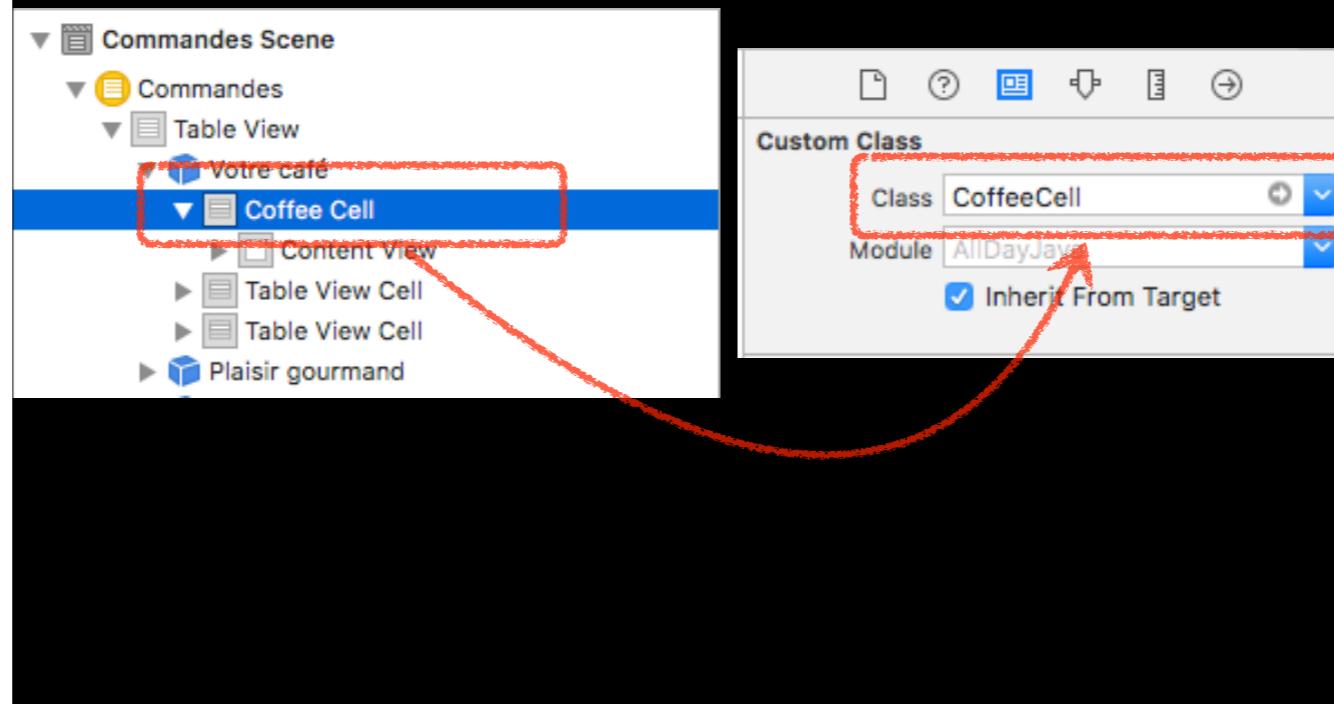
```
class CoffeeCell: UITableViewCell {  
    override var accessibilityElements: [Any]? {  
        get {  
            return [  
                viewWithTag(1) as Any,  
                viewWithTag(2) as Any  
            ]  
        }  
        set {  
            super.accessibilityElements = newValue  
        }  
    }  
}
```

Pour assurer un ordre de lecture spécifique, vous pouvez créer votre propre valeur pour le label.

Mais il est tout aussi simple de spécifier votre propre ordre :

1. créer une classe pour les cellules de table ;
2. dans cette classe, surcharger la propriété « **accessibilityElements** » pour retourner la liste des vues ou textes à annoncer par VoiceOver.
3. utilisez cette classe dans votre storyboard.

# Classe des cellules



## Comment associer la classe au composant de l'interface ?

Pour ceux qui ne connaissent pas Xcode, l'association se fait dans Interface Builder (*l'éditeur d'interfaces*) en indiquant simplement le nom de la classe pour l'objet cellule (UITableViewCell).



Pour bien voir les différences une fois l'accessibilité légèrement améliorée.

# Tests automatiques

## Créer et exécuter un test

Comment utiliser les propriétés d'accessibilité pour écrire vos tests automatiques d'interface ?

# Comment ?

- en écrivant du code
- avec l'enregistrement  
(qui va écrire le code pour vous...)

Les tests sont écrits en **Swift**.

*Note:* le framework construit sur JavaScript n'est plus supporté.

Vous pouvez commencer à écrire votre test en enregistrant vos actions. Le résultat sera un début de code.

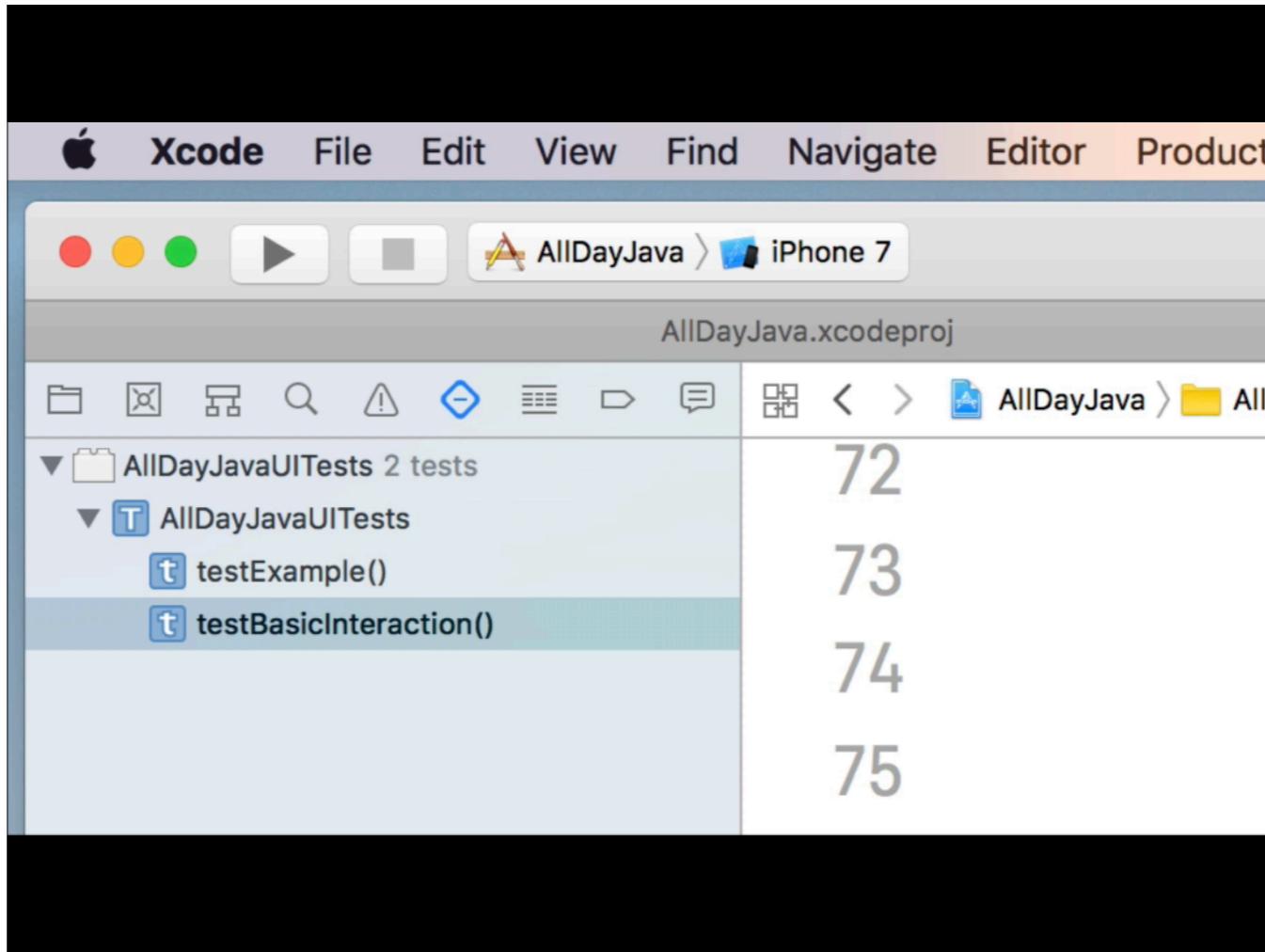
Il est largement préférable de structurer vous même ce code en l'écrivant vous même.

# Démo !

**Créer un test**

Demos:

- création d'un test
- enregistrement



On peut commencer un test en enregistrant les actions dans le simulateur ou directement sur l'iPhone.

On crée un nouveau test, puis on lance l'enregistrement.

Chaque action est répercutée dans le test sous la forme d'instructions.

# Exemple

```
func testExample() {
    let app = XCUIApplication()

    // No identifier for a11y 😞
    app.tabBars.buttons["Commande"].tap()

    let buttons = app.tables.buttons

    // Pick a coffee and change you mind! ☕
    buttons.matching(identifier: "cofee-americano").element.tap()
    buttons.matching(identifier: "cofee-expreso").element.tap()

    // Add some 🍰
    let bakeFld =
        app.tables.textFields.matching(
            identifier: "scone-fld").element
    ...
}
```

- « XCUIApplication » : l'objet pour manipuler l'application
- « tabbars.buttons » renvoie la liste des boutons dans la barre d'onglets (en bas)
- « buttons.matching(identifier: "<ID>") » permet de construire une requête pour rechercher un élément spécifique.
- l'attribut « element » résout cette requête et retourne l'objet correspondant.

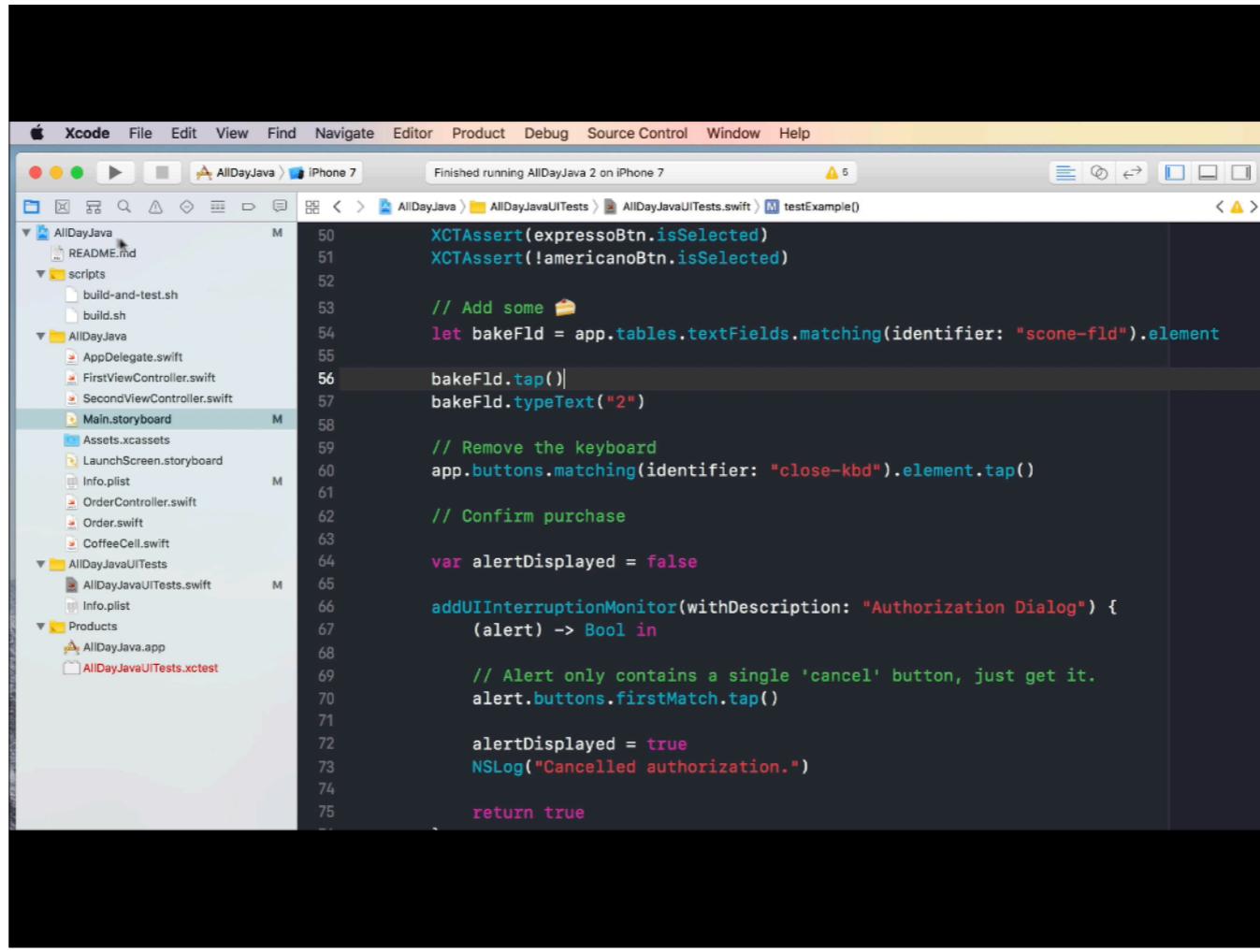
Un vrai test utilisera des vérifications pour s'assurer de l'existence et de la validité des objets.

# Démo !

**Dérouler un test automatique**

Demos:

- exécution

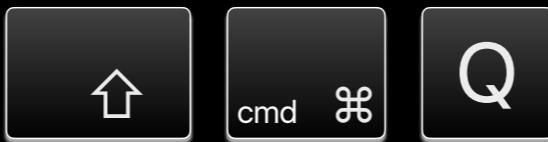


The screenshot shows the Xcode interface with the project 'AllDayJava' open. The left sidebar displays the project structure, including 'AllDayJava' (with files like AppDelegate.swift, FirstViewController.swift, SecondViewController.swift, and Main.storyboard), 'scripts' (with build-and-test.sh and build.sh), 'AllDayJavaUITests' (with AllDayJavaUITests.swift and Info.plist), and 'Products' (with AllDayJava.app and AllDayJavaUITests.xctest). The main editor window shows a Swift code file named 'testExample()' which contains UI test code for an application. The code uses XCUIApplication to interact with UI elements like buttons and text fields, and handles alerts.

```
50     XCTAssert(expressoBtn.isSelected)
51     XCTAssert(!americanoBtn.isSelected)
52
53     // Add some 🍰
54     let bakeFld = app.tables.textFields.matching(identifier: "scone-fld").element
55
56     bakeFld.tap()
57     bakeFld.typeText("2")
58
59     // Remove the keyboard
60     app.buttons.matching(identifier: "close-kbd").element.tap()
61
62     // Confirm purchase
63
64     var alertDisplayed = false
65
66     addUIInterruptionMonitor(withDescription: "Authorization Dialog") {
67         (alert) -> Bool in
68
69         // Alert only contains a single 'cancel' button, just get it.
70         alert.buttons.firstMatch.tap()
71
72         alertDisplayed = true
73         NSLog("Cancelled authorization.")
74
75         return true
76     }
77 }
```

Les tests d'interface peuvent être exécutés sur simulateur ou directement sur le terminal.

# Pour conclure...



Nous n'avons certainement pas fait le tour de la question...

# Check-list Minimale

- cibles assez grandes
- assez de contrastes
- ordre de lecture naturel
- respect des polices dynamiques
- chaque élément à une description
- structurez vos documents :
  - titres sur du texte
  - titres de sections dans les tables/...
- Testez vos applications
- utilisez des identifiants pour vos tests

Cet inventaire est loin d'être complet. Mais c'est un bon point de départ.

IBM propose sa propre check-list :

[http://www-03.ibm.com/able/guidelines/ci162/accessibility\\_checklist.html](http://www-03.ibm.com/able/guidelines/ci162/accessibility_checklist.html)

L'accessibilité n'est pas en contradiction avec l'expérience utilisateur :

<http://blend2017.access42.net>

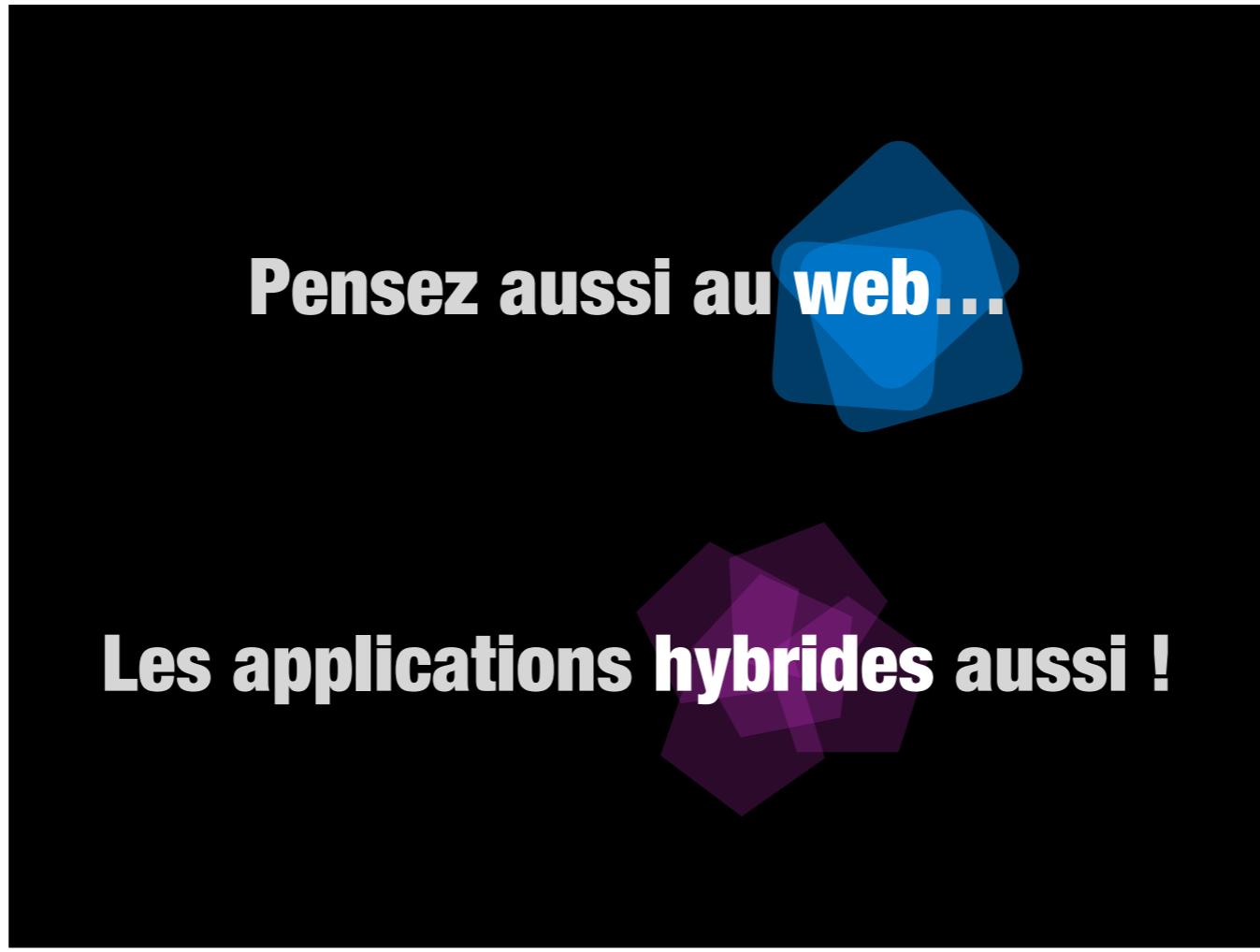
Creating an Accessibility Engineering Practice

<http://blog.danielna.com/2017/09/14/creating-an-accessibility-engineering-practice.html>

## Outils:

- Colour Contrast Analyser: <https://developer.paciellogroup.com/resources/contrastanalyser/>

-



## Web

Votre site web est tout aussi important puisque c'est le plus souvent lui qui vous amènera des clients ou servira à leur support.

- Les check-lists d'**opquast**:
  - <http://checklists.opquast.com/fr/accessibility-first-step/>
  - <http://checklists.opquast.com/fr/accessibility-second-step/>
- Guides du w3c (ARIA)
- Guide du développeur RGAA 3  
<https://disic.github.io/guide-developpeur/4-aria.html>

## Applications hybrides

« *Making Your Xamarin.Forms Apps Accessible* » <https://blog.xamarin.com/accessibility-xamarin-forms/>

Quels plugins/techniques pour Cordova?

Android... un système de lecture d'écran est également disponible.

# Ressources

- <https://www.apple.com/fr/accessibility/>
- <https://developer.apple.com/accessibility/>
- <https://www.raywenderlich.com/142058/ios-accessibility-tutorial>
- <https://www.oxytude.org>

Quelques liens pour obtenir plus d'informations.

## Autres resources:

Access 42 : <https://access42.net>

Mobile Accessibility Testing Guide for Android and iOS  
<https://developer.paciologroup.com/blog/2017/11/mobile-accessibility-testing-guide-for-android-and-ios-free/>

DISIC - Direction Interministérielle des Systèmes d'Information et de Communication  
<https://github.com/DISIC/>

Défauts D'accessibilité : Impacts Sur Les Utilisateurs :  
[https://disic.github.io/guide-impacts\\_utilisateurs/](https://disic.github.io/guide-impacts_utilisateurs/)

« VIDEO. Les non-voyants ont leur système révolutionnaire pour jouer aux cartes »  
<https://t.co/hQGV6opSkw> (Nice Matin); un système pour permettre aux non-voyants de jouer au bridge.

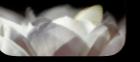
Le guide des contrastes de couleurs :  
<https://www.webdesignerdepot.com/2017/09/the-designers-guide-to-color-contrast/>



# À suivre



@IBMAccess



@AdobeAccess



@MSFTEnable



@fbaccess



@IAAPOrg

(International Association of  
Accessibility Professionals)



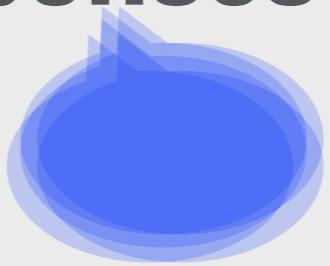
@UserWayOrg

Twitter :

@UserWayOrg ; @IAAPOrg (International Association of Accessibility Professionals) @FreedomSci (JAWS) ;



# Questions Réponses



## INFORMATIONS COMPLÉMENTAIRES

### Plus d'utilisateurs potentiels

<http://webzine.okeenea.com/16-des-francais-touches-par-un-handicap-mental-psychique-ou-cognitif-quelles-sont-les-regles-daccessibilite/>

Par exemple, le site Oxytude lance une campagne pour mal noter les applications qui ne sont pas accessibles :

<https://www.oxytude.org/molotov-pour-ios-inaugure-la-rubrique-sale-note-pour-les-applications-non-accessibles/>

Les personnes handicapées sont aussi des clients du jeu vidéo.

<https://www.gamekult.com/actualite/joueurs-et-handicapes-ils-vivent-le-jeu-video-en-mode-hard-3050798791.html>

### La réglementation

<http://www.fondshs.fr/technologies/internet-europe-impose-regles-pour-accessibilite-services-publics-en-ligne>

Accessibilité au handicap : un des enjeux de l'Europe

<https://informations.handicap.fr/art-acte-accessibilite-europe-1027-10169.php>



Développement Mobile Côte d'Azur  
**2017**

**Sylvain GAMEL**

<http://SylvainGamel.fr/>



@sgamel

Merci de vos retours.