

TD n°2 - Développement mobile sous iOS

Exercice 1 - Première application

1. En vous aidant des diapositives du cours si nécessaire, créez une application iOS simple avec Xcode.
2. Ouvrez le storyboard principal de l'application et placez un label au milieu de l'écran contenant le texte que vous souhaitez.
3. Enregistrez et lancez l'application sur le simulateur de votre choix.

Exercice 2 - Formulaire de connexion

Le but de cet exercice est de vous amener pas à pas à construire une application qui va proposer à l'utilisateur de remplir un faux formulaire de connexion.

PREMIÈRE PARTIE : CRÉATION DU FORMULAIRE

1. Créez une nouvelle application iOS
2. Dans le storyboard principal de cette application, ajoutez un label « Login », un champ texte, un label « Mot de passe », un autre champ texte et un bouton « Connexion ».

Pour changer le texte d'un label ou d'un bouton, vous pouvez double-cliquer dessus.

3. Cliquez ensuite sur le champ associé au mot de passe et dans le panneau droit, ouvrez l'inspecteur d'attributs.

L'inspecteur d'attributs est au niveau de la 5ème icône.

C'est dans cet inspecteur que vous pouvez personnaliser beaucoup de paramètres liés à l'apparence, au positionnement et à l'accessibilité de vos vues.

Descendez ensuite jusqu'à la section « Text Input Traits » et changez « Content Type » en « Password ». Tout en bas de cette section, cochez la case « Secure Text Input ».

4. En utilisant l'*Interface Builder*, connectez les éléments du storyboard avec le contrôleur associé. Vous aurez :

1. Un attribut `login` représentant le champ texte du login
2. Un attribut `password` représentant le champ texte du mot de passe
3. Un attribut `button` représentant le bouton de connexion

SECONDE PARTIE : GESTION DE L'ENVOI

Vous allez maintenant gérer le click sur le bouton de connexion.

1. Créez une méthode `connexion()` ayant l'attribut `@objc`.
2. Dans le corps de cette méthode, vous allez récupérer les valeurs des champs de login et de mot de passe et vérifier leur validité avec un login et un mot de passe que vous définirez (e.g. login « michel » et mot de passe « s3cr3t »).

Pour récupérer la valeur d'éléments `UITextField`, il suffit d'accéder à leur attribut `text`.

Note : cet attribut est un optionnel ; vous pouvez utiliser l'opérateur `??` par exemple, vu dans le cours n°5.

3. Selon si le login et le mot de passe fournis correspondent ou non, vous afficherez une alerte.

Pour afficher une alerte, vous pouvez utiliser le code suivant :

```
let alert = UIAlertController(title: "", message: "", preferredStyle: .alert)
alert.addAction(UIAlertAction(title: "Ok", style: .default))
self.present(alert, animated: true)
```

4. En vous aidant du cours, connectez ensuite le bouton avec la méthode `connexion()` en utilisant `addTarget` afin de déclencher l'appel à cette méthode lors du clique sur le bouton.

Exercice 3 - Application de conversion d'unités

Le but de cet exercice est de mettre en application le modèle MVC en créant une application permettant de convertir une unité vers une autre et inversement. Les unités utilisées ici seront les degrés Celsius et les degrés Fahrenheit.

PREMIÈRE PARTIE : CONVERSION (MODÈLE)

1. Créez une nouvelle application iOS
2. Ajoutez une classe Swift à votre projet (en allant dans le menu `File > New > File` et en sélectionnant « Swift File »). Vous pouvez la nommer `Temperature` par exemple.
3. Ajoutez deux attributs `_celsius` et `_fahrenheit` à votre classe étant de type `Double`. (Le *underscore* devant le nom de ces attributs a son importance pour ne pas entrer en conflit avec la suite de l'exercice).
4. Implémentez ensuite deux constructeurs pour cette classe : un prenant un paramètre `celsius` de type `Double` et un autre prenant un paramètre `fahrenheit` de type `Double` également.

Ces constructeurs définiront la valeur de l'attribut associé, tout en définissant la valeur de l'autre attribut à `Double.infinity`. On aurait pu utiliser 0 comme valeur mais nous ne saurions pas déterminer s'il s'agit d'une valeur définie par nos soins ou renseignée par l'utilisateur.

5. Vous allez finalement utiliser les « propriétés évaluées » de Swift (*computed properties* en anglais) pour calculer la valeur dans une unité par rapport à l'autre si elle n'est pas définie.

Note : pour convertir de degrés Celsius vers Fahrenheit, il faut multiplier par 1.8 et ajouter 32.

Par exemple, voilà comment est définie la propriété `celsius` :

```
var celsius : Double {
    get {
        if _celsius != Double.infinity {
            return _celsius
        } else {
            return _fahrenheit / 1.8 - 32
        }
    }
}
```

Implémentez de la même manière la propriété `fahrenheit`.

DEUXIÈME PARTIE : CRÉATION DE L'INTERFACE (VUE)

1. Dans le storyboard principal ajoutez :
 1. Un label « Celsius »
 2. Un champ texte
 3. Un label « Fahrenheit »
 4. Un champ texte
2. Essayez d'aligner correctement les éléments. Pour cela, vous pouvez ajouter des contraintes en utilisant la troisième icône située à droite du panneau bas dans le storyboard. (c.f. cours n°4)

Vous pouvez par exemple ajouter des contraintes d'espacement en haut et à gauche.

TROISIÈME PARTIE : TRAITEMENT DES ÉVÈNEMENTS (CONTRÔLEUR)

1. En utilisant l'Interface Builder, connectez les deux champs textes précédemment créés au contrôleur.
2. Avec l'Interface Builder toujours, connectez une méthode appelée lors de l'édition de chacun de ces champs.

Vous pouvez appeler ces méthodes `celsiusFieldChanged` et `fahrenheitFieldChanged` par exemple.

Veillez à sélectionner l'évènement « Editing changed » dans la liste déroulante correspondant à l'évènement qui déclenche l'appel de ces méthodes et sélectionner le bon type (`UITextField`) dans la liste déroulante associée.

3. Finalement, implémentez le code des deux méthodes gérant les changements sur les champs. Vous devez :
 1. Créer une instance de la classe `Temperature` à partir de la valeur du champ. Puisque le constructeur requiert un `Double`, vous devez convertir la chaîne de caractère en `Double`. Pour réaliser cette opération, vous pouvez passer la chaîne au constructeur de la classe `Double`, par exemple : `Double("4.2")`.
 2. Obtenir la température convertie vers l'autre unité et convertir le résultat en chaîne de caractère. Pour convertir un `double` en chaîne de caractères, il suffit de le passer au constructeur de la classe `String`, par exemple : `String(4.2)`.
 3. Finalement, associer la valeur de la chaîne de caractère à la valeur de l'autre champ texte.
4. Lancez le simulateur pour tester l'application.