

## **DEVELOPPEMENT APPLICATION CALCULATRICE**

<b>DEVELOPPEMENT APPLICATION CALCULATRICE</b>	<b>1</b>
I. INTRODUCTION	3
II. XCODE/SWIFT	3
1. Découverte de Xcode	3
2. Développement de la Calculatrice	5

## I. INTRODUCTION

---

Pour débiter le projet, nous devons développer une application de calculatrice sur iOS et Android dans le but de s'initier au développement d'application.

Possédant un appareil sur MacOS et étant plus a l'aise avec les appareils iOS, j'ai préféré commencer avec un développement iOS.

## II. XCODE/SWIFT

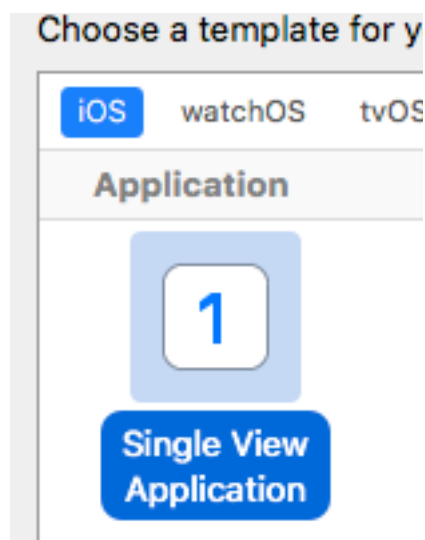
---

Xcode est le logiciel de développement de MacOS, il permet de développer pour iOS sur iPad iPhone mais aussi pour l'Apple TV, l'Apple Watch, etc..

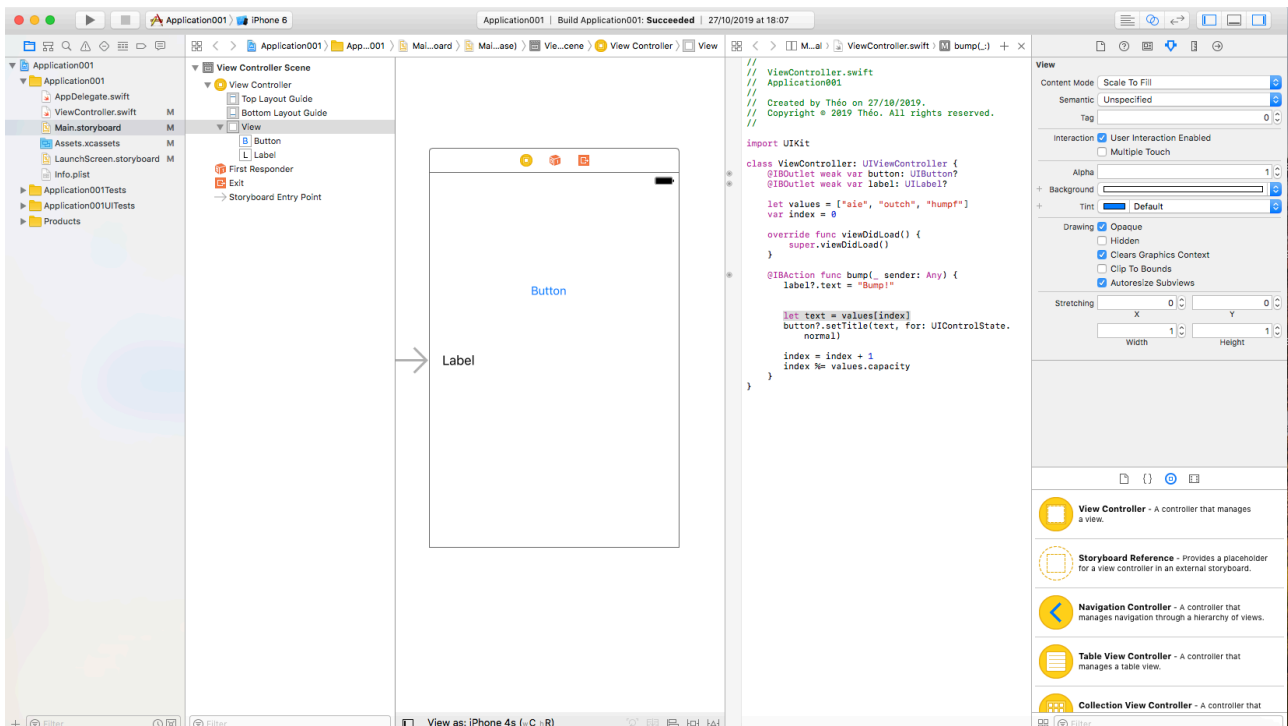
Pour développer sur iOS, nous utilisons le langage objective-c car l'année passé nous avons déjà travaillé sur le langage C.

### 1. Découverte de Xcode

Pour developper une calculatrice, une application simple a une seule vue suffis, donc j'ai sélectionné un projet en « Single View » .



Voici l'interface de Xcode :



On peut voir que sur l'interface, j'ai intégré deux fenêtre, une pour le contrôleur (à gauche) une pour le code (à droite).

Avec le menu déroulant sur le bas de la droite j'ai placé un bouton et un label, que qui sont configurable sur le menu du haut de la droite.

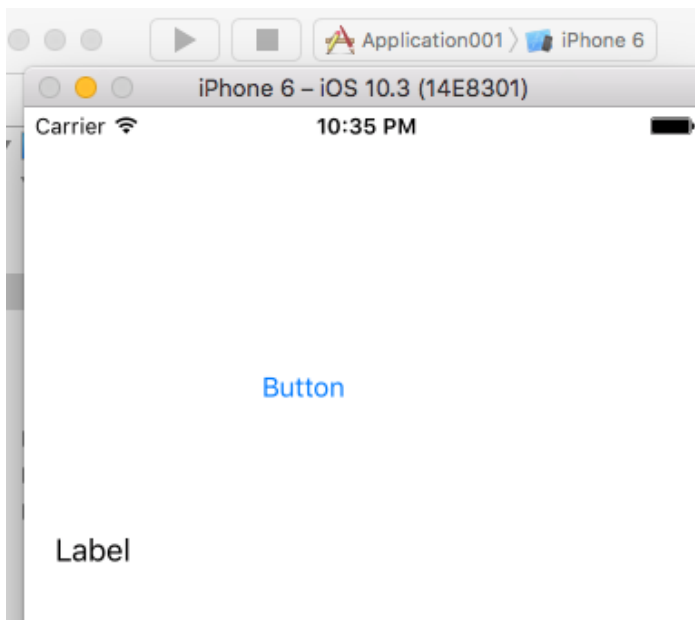
Puis avec le bouton ctrl et le clique gauche, avec la souris j'ai créé une liaison du bouton vers le code.

```
@property (weak, nonatomic) IBOutlet UITextField *insert;  
@property (weak, nonatomic) IBOutlet UIButton *button;  
@property (weak, nonatomic) IBOutlet UILabel *resultat;
```

On crée donc des outlets qui vont permettre d'intégrer ces objets dans le code sous forme de variable.

```
- (IBAction)calcul:(id)sender {
```

On peut utiliser la même manipulation pour intégrer l'action d'un objet sous forme de fonction, pour enclencher un code.



Nous pouvons aussi simuler un iPhone avec Xcode pour tester notre application.

---

## 2. Développement de la Calculatrice

Nous avons à la base développé une calculatrice, semblable aux calculatrices intégrées dans les OS des téléphones, mais celles-ci étant trop compliquées nous avons revu nos objectifs et nous nous sommes concentrés sur une calculatrice plus simplifiée.

Nous avons tout d'abord commencé avec une calculatrice la plus simplifiée possible, avec seulement des opérations à 1 chiffre.



On place tout d'abord un bouton pour faire le calcul et un textfield pour y écrire le calcul.

---

```
@property (weak, nonatomic) IBOutlet UITextField *insert;  
@property (weak, nonatomic) IBOutlet UIButton *button;
```

Puis on les lie au code.

```
- (IBAction)calcul:(id)sender {
```

On lie donc notre action du bouton comme fonction.

---

```
NSString *get = self.insert.text;
```

On va ensuite récupérer le calcul entré.

---

```
NSString *char1 = [NSString stringWithFormat:@"%c", [get characterAtIndex:0]];
NSLog(@"%@", char1);

NSString *op = [NSString stringWithFormat:@"%c", [get characterAtIndex:1]];
NSLog(@"%@", op);

NSString *char2 = [NSString stringWithFormat:@"%c", [get characterAtIndex:2]];
NSLog(@"%@", char2);
```

On sépare les différents caractères (on travail sur des chiffres a un nombre donc a/+/b) et on les stocke dans de différentes variables.

Les NSLog sont la pour afficher dans la console de XCode (non visible sur l'application) que le déroulement de l'application fonctionne Correctement, ils peuvent aussi aider a situer l'erreur dans le cas ou il y en a.

---

```
NSInteger Num1 = [char1 integerValue];
NSInteger Num2 = [char2 integerValue];
```

On insère les caractères chiffres dans des variable Integer.

---

```
if([op isEqualToString:@"+"]){
    printf("\nOn veut additionner");
    res = Num1 + Num2;
}
else if([op isEqualToString:@"-"]){
    printf("\nOn veut soustraire");
    res = Num1 - Num2;
}
else if([op isEqualToString:@"*"]){
    printf("\nOn veut multiplier");
    res = Num1 * Num2;
}
else if([op isEqualToString:@"/"]){
    printf("\nOn veut diviser");
    res = Num1 / Num2;
}
```

On peut ainsi faire le calcul en fonction de l'opérateur.

```
NSString *result = [NSString stringWithFormat:@"%zd", res];  
self.insert.text = result;
```

Il nous reste plus qu'à insérer le résultat dans une variable String et de l'afficher.

---

2+4

Button

On insère le calcul.

---

6

Button

Et voila le résultat.

---

```
2019-11-15 00:49:14.911 ebauche[2173:322453] 2  
2019-11-15 00:49:14.911 ebauche[2173:322453] +  
2019-11-15 00:49:14.912 ebauche[2173:322453] 4  
  
Le premier chiffre : 2  
Le deuxieme chiffre : 4  
On veut additionner  
Le resultat est : 6
```

All Output ↕

Filter



La console nous a permis de voir que l'action s'est bien déroulée.

Nous allons maintenant passer à une calculatrice plus complexe, mais restant simpliste, elle sera pareil mais permettra de calculer des chiffres de plus d'un nombre.

Inserez le calcul
Resultat
Calcule !

Nous avons, aussi au passage amélioré l'affichage en rajoutant un label pour afficher le résultat.

---

On a commencé par effacer tout le code excepté les liaisons bouton, et la ligne permettant de récupérer le calcul.

Prenons exemple de l'addition :

```
if([get containsString:@"+"]){
```

Nous avons ajouté un if pour chaque opérateur, pour déterminer quel opération fallait t'il faire, avec une fonction permettant de déterminer la présence d'un caractère précis dans la variable « get ».

---

```
NSArray * array = [get componentsSeparatedByString:@"+"];
```

Puis un array est créé en récupérant dans différents plusieurs blocs de caractères séparé par un symbole +.

Ex : si on écrit « 234+3ea+ze », il créera un array avec « 234 » dans la première case, « 3ea » dans la deuxième, « ze » dans la troisième.

---

```
NSString * num1 = [array objectAtIndex:0];  
NSString * num2 = [array objectAtIndex:1];
```

Nous allons ensuite insérer les deux chiffres séparés par l'ARRAY dans deux STRING différentes.

---

```
result = [num1 integerValue] + [num2 integerValue];
```

Nous pouvons donc faire le calcul en veillant a ce que les variables soit bien utilisé en tant que valeurs INTEGER.



```
////////////////////////////////////  
NSString * string = [NSString stringWithFormat:@"%ld", result];  
self.resultat.text = string;
```

Nous affichons donc le résultat après l'avoir inséré dans un variable STRING

---

Nous avons donc recopié ce bout de code en changeant l'opérateur pour les différentes opérations.

Nous testons donc le code :

---

12+122

---

134

Calcule !

12+122 fais bien 134

---

```
2019-11-15 01:07:00.126 CalSp001[2335:369500]  
Le premier numero est : 12  
2019-11-15 01:07:00.126 CalSp001[2335:369500]  
Le deuxieme numero est : 122  
2019-11-15 01:07:00.126 CalSp001[2335:369500]  
Il s'agit d'une addition  
2019-11-15 01:07:00.126 CalSp001[2335:369500]  
Le resultat est : 134
```

La console nous montre que cela s'est bien déroulé.