1

LES BASES DU LANGAGE SWIFT

03/02/2025 Abdelali.Lasfar@est.um5.ac.ma

1

Plan

2

- 1- les bases du langage Swift
  - Les variables et opérateurs
  - Les conditions
  - Les boucles
  - Les tableaux et les dictionnaires
  - Les fonctions et les closures.
- 2- la POO en Swift
- 3- Utilisation de l'interface graphique iOS
- 4- utilisation de l'interface graphique OS X

2

Swift

3

- Swift est un langage de programmation crée par Apple.
- Annoncé le 02 juin 2014 lors de la conférence mondiale des développeurs d'applications pour l'écosystème Apple (WWDC : World Wide Developers Conference)
- Swift permet la création des applications sous les systèmes iOS, OS X et WatchOS

3

L'IDE

4

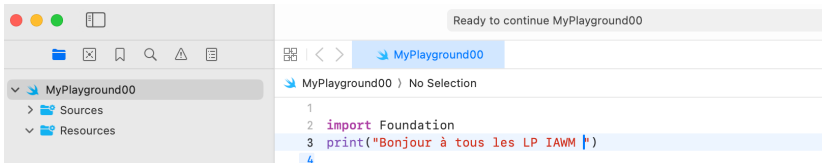
Un seul choix: Xcode

Un seul système: OS X

4

5

- Xcode est un environnement de développement pour macOS, watchOS, tvOS et iOS. Il est connu pour sa simplicité et surtout pour ses astuces proposées pour développer plus rapidement.



5

6

Les variables et les constantes

6

Les variables

7

- En Swift, les variables sont utilisées pour stocker des données qui peuvent être modifiées au cours de l'exécution du programme.
- Pour déclarer une variable, on utilise le mot-clé **var**

```
var nomDeLaVariable: Type = valeur
```


7

Les variables

8

- Pour déclarer une variable, on utilise le mot-clé **var**

```
var nomDeLaVariable: Type = valeur
```

- var : mot-clé pour déclarer une variable.
- nomDeLaVariable : le nom que vous donnez à la variable.
- Type : le type de données que la variable va stocker (optionnel si Swift peut inférer le type).
- valeur : la valeur initiale de la variable.

8

Les variables

9

- Exemples
  - var age: Int = 25
  - var nom: String = " LP IAWM"
  - var estEtudiant: Bool = true
  - var temperature: Double = 23.5

03/02/2025

9

Les variables

10

- **Inférence de type**
- Swift peut inférer le type de la variable à partir de la valeur initiale, donc vous n'êtes pas obligé de spécifier le type explicitement :

- var age = 25 // Swift infère que `age` est de type Int
- var nom = "Jean" // Swift infère que `nom` est de type String
- var estEtudiant = true // Swift infère que `estEtudiant` est de type Bool

03/02/2025

10

Les variables

11

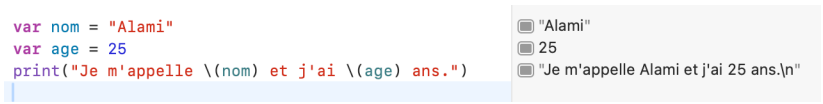
- Si vous déclarez une variable, les prochaines affectations à cette même variable doit rester du même type. Par exemple, si je déclare une variable de type entier. Je ne pourrais modifier la variable que par des nombres entiers à l'avenir.

Les variables

12

- **Concaténation de variables**
- Vous pouvez utiliser des variables dans des chaînes de caractères en utilisant l'interpolation de chaîne :

```
var nom = "Alami"
var age = 25
print("Je m'appelle \(nom) et j'ai \(age) ans.")
```



Les constantes

13

- Pour déclarer une constante en Swift :
  - let pi = 3.1415

13

Types courants de variables

14

- Int : entier
- Double : nombre à virgule flottante (précision double)
- Float : nombre à virgule flottante (précision simple)
- String : chaîne de caractères
- Bool : booléen (true ou false)

14

Opérateurs arithmétiques

15

Opérateur	Description	Exemple
+	Addition	3 + 2 = 5
-	Soustraction	5 - 3 = 2
*	Multiplication	4 * 3 = 12
/	Division	10 / 2 = 5
%	Modulo (reste de la division)	10 % 3 = 1

15

Opérateurs de comparaison

16

Opérateur	Description	Exemple
==	Égal à	5 == 5 → true
!=	Différent de	5 != 3 → true
>	Supérieur à	5 > 3 → true
<	Inférieur à	5 < 3 → false
>=	Supérieur ou égal à	5 >= 5 → true
<=	Inférieur ou égal à	5 <= 3 → false

16

Opérateurs logiques

17

Opérateur	Description
&&	ET logique
||	OU logique
!	NON logique (inverse)

```
let estMajeur = true
let aUnPermis = false

if estMajeur && aUnPermis { print("Vous pouvez conduire.") }
else { print("Vous ne pouvez pas conduire.") }
if estMajeur || aUnPermis { print("Vous avez au moins une condition remplie.") }
if !estMajeur { print("Vous êtes mineur.") }
```


17

Opérateurs d'affectation composés

18

Opérateur	Description	Exemple
+=	Ajoute et affecte	a += 3 → a = a + 3
-=	Soustrait et affecte	a -= 3 → a = a - 3
*=	Multiplie et affecte	a *= 3 → a = a * 3
/=	Divise et affecte	a /= 3 → a = a / 3
%=	Modulo et affecte	a %= 3 → a = a % 3

18

Opérateurs de plage (Range Operators)

19

Opérateur	Description	Exemple
...	Plage fermée	1...5 → 1, 2, 3, 4, 5
..<	Plage semi-ouverte	1..<5 → 1, 2, 3, 4

19

20

Les structures conditionnelles

1. **if** : Exécute un bloc de code si une condition est vraie.
2. **if-else** : Exécute un bloc de code si une condition est vraie, et un autre bloc si elle est fausse.
3. **else-if** : Permet de chaîner plusieurs conditions.
4. **switch** : Permet de gérer plusieurs cas de figure en fonction de la valeur d'une variable.

20

La condition if

21

- La structure if permet d'exécuter un bloc de code uniquement si une condition est vraie.
- **Syntaxe :**

```
if condition {
// Code à exécuter si la condition est vraie
}
```

03/02/2025

21

La condition if

22

- La structure if permet d'exécuter un bloc de code uniquement si une condition est vraie.
- **Exemple :**

```
let age = 18
if age >= 18 {
print("Vous êtes majeur.")
}
```

03/02/2025

22

La condition if-else

23

- La structure if-else permet d'exécuter un bloc de code si la condition est vraie, et un autre bloc si elle est fausse.
- **Syntaxe :**

```
if condition {
// Code à exécuter si la condition est vraie }
else {
// Code à exécuter si la condition est fausse }
```

03/02/2025

23

La condition if-else

24

- La structure if-else permet d'exécuter un bloc de code si la condition est vraie, et un autre bloc si elle est fausse.
- **Exemple :**

```
let temperature = 30
if temperature > 25 {
   print("Il fait chaud.") }
else {
   print("Il fait frais.") }
```

03/02/2025

24

La condition else-if

25

- La structure else-if permet de chaîner plusieurs conditions. Si la première condition est fausse, Swift vérifie la suivante, et ainsi de suite.

- **Syntaxe :**
- if condition1 { // Code à exécuter si condition1 est vraie }
  else if condition2{ // Code à exécuter si condition2 est vraie }
  else { // Code à exécuter si aucune condition n'est vraie }

La condition else-if

26

- Exemple :

```
let note = 18.4
if note >= 16
        { print("Excellent !") }
else if note >= 14
        { print("Bien !") }
else if note >= 10
        { print("Passable.") }
else { print("  Insuffisant.") }
```

La structure switch

27

- La structure switch permet de gérer plusieurs cas de figure en fonction de la valeur d'une variable. Elle est souvent plus lisible qu'une série de if-else lorsqu'il y a de nombreux cas à traiter.
- **Syntaxe :**

```
switch valeur {
case cas1: // Code à exécuter si valeur == cas1
case cas2, cas3: // Code à exécuter si valeur == cas2 ou valeur == cas3
...
default: // Code à exécuter si aucun cas ne correspond
}
```


La structure switch

28

- Exemple :

```
let jour = "Lundi"
switch jour {
case "Lundi": print("C'est le début de la semaine.")
case "Mardi", "Mercredi", "Jeudi": print("On est en plein dans la semaine.")
case "Vendredi": print("C'est bientôt le week-end !")
case "Samedi", "Dimanche": print("Enfin le week-end !")
default: print("Jour non reconnu.") }
```

Les tuples

29

- un tuple peut contenir plusieurs valeurs de n'importe quel type. Une variable ou une constante peut être un tuple.
  - let tuple = (valeur1, valeur2, ...)
  - var tuple = (valeur1, valeur2, ...)

- let JoursSemaine= ("lundi", "Mardi", ...)
- var tuple = (2.5, "bonjour",2, ...)

Switch case

30

```
let note=12
var mention="Pas de mention"
switch note {
case 0..<10     : mention="Pas de mention"
case 10..<12    : mention="Passable"
case 12..<14    : mention="A.Bien"
case 14..<16    : mention="Bien"
case 16...20     : mention="T.Bien"
default : print(" La note doit être entre 0 et 20")
}
print(" note =\(note) mention =\(mention)")
```

31

Les structures répétitives

1. **for-in** : Pour itérer sur une séquence (comme un tableau, une plage, etc.).
2. **while** : Pour répéter un bloc de code tant qu'une condition est vraie.
3. **repeat-while** : Similaire à while, mais la condition est vérifiée après l'exécution du bloc de code.

31

for - in

32

- La boucle for-in est utilisée pour itérer sur une séquence, comme un tableau, une plage de nombres, ou une collection.
- **Syntaxe :**

```
for élément in séquence {
// Code à exécuter pour chaque élément }
```

- **Exemples :**

```
for i in 1...5 { print(i) // Affiche 1, 2, 3, 4, 5 }
```


32

for - in

33

- **Exemples :**

```
for  in 1...5 { print(i) // Affiche 1, 2, 3, 4, 5 }

let fruits = ["Pomme", "Banane", "Orange"]
for fruit in fruits {
        print(fruit)
                // Affiche "Pomme", "Banane", "Orange"
}
```


for - in

34

- **Exemples (Itérer sur un dictionnaire):**

```
let ages = ["Aalmi": 25, "Mahdaoui": 30, "Youssfi": 22]
for (nom, age) in ages
{
        print("\(nom) a \(age) ans.")
}
```

for - in

35

- **Itérer avec un pas (stride) :**
- Si vous voulez itérer avec un pas spécifique, utilisez stride :

```
for   in stride(from: 0, to: 10, by: 2) {
print( ) // Affiche 0, 2, 4, 6, 8 }
```


while

36

- La boucle while exécute un bloc de code **tant qu'une condition est vraie**. La condition est vérifiée avant chaque itération.

- **Syntaxe :**

```
while condition {
// Code à exécuter tant que la condition est vraie }
```

- **Exemple :**

```
var compteur = 0
while compteur < 5 {
print("Compteur : \(compteur)")
compteur += 1 }
```

Repeat - while

37

- La boucle repeat-while est similaire à while, mais la condition est vérifiée après l'exécution du bloc de code. Cela garantit que le bloc est exécuté au moins une fois.
- **Syntaxe :**

```
repeat condition {
// Code à exécuter tant que la condition est vraie } while condition
```

- **Exemple :**

```
var compteur = 0
repeat {
print("Compteur : \(compteur)")
compteur += 1 } while compteur < 5
```

03/02/2025

37

Contrôles des boucles

38

- Swift propose des mots-clés pour contrôler l'exécution des boucles :
  - **break** : Arrête immédiatement l'exécution de la boucle.
  - **continue** : Saute le reste de l'itération en cours et passe à la suivante.
- **Exemples utilisation de break :**

```
for   in 1...10 { if   == 5
break // Arrête la boucle lorsque i == 5
print( ) // Affiche 1, 2, 3, 4 }
```

03/02/2025

38