

Desenvolvimento iOS

Objetivo da Aula

- Variáveis/Constantes
- If/else
- Funções
- Array
- Dictionaries
- Loops



Variáveis / Constantes

`var`

`let`



Variáveis / Constantes

var

Muda

let

Não muda

If - Else

Comparações

== -> Igual

!= -> Diferente

> -> Maior

< -> Menor

>= -> Maior igual

<= -> Menor igual

Operadores Lógicos

&& -> E

|| -> Ou

! -> Não





Let's code!



Funções



Uma função é um bloco de código que realiza uma tarefa em específico. Permite encapsular código e reutilizá-lo diversas vezes.

```
func retorneEsseNumero (numero: Int) -> Int { return numero }
```


Função void (sem retorno)



```
func imprimeOla() {  
    print("Olá")  
}
```

Função com retorno



```
func entendeu() -> Bool {  
    return true  
}
```

Parâmetros na função



```
func imprime(texto: String) {  
    print(texto)  
}
```



Let's code!

Exercícios



1. Definir três métodos que imprimam saudações diferentes em tela e sejam executados da seguinte forma:

`cumprimentarA(pessoa: "João")`

`cumprimentar(a: "João")`

`cumprimentarA("João")`

2. Escrever o método ***eMenor(oPrimeiro: Int, oSegundo: Int) -> Bool*** deve usar dois números inteiros como parâmetros e retornar ***true*** se o primeiro número for menor que o segundo número ou ***false*** , caso contrário.

- Invocar o método com os números 3 e 5 e imprimir na tela o resultado.
- Invocar o método com os números 7 e 5 e imprimir na tela o resultado.
- Invocar o método com os números 10 e 10; e imprimir na tela o resultado.

Exercícios



1. Escrever o método ***elmparMaiorQueDez(umNumero: Int) -> Bool*** que deve analisar um número inteiro como parâmetro e retornar ***true*** se o número for ímpar e maior do que dez e ***false***, caso contrário.
 - Invocar o método com o número 3 e imprimir na tela o resultado.
 - Invocar o método com o número 4 e imprimir na tela o resultado.
 - Invocar o método com o número 13 e imprimir na tela o resultado.
 - Invocar o método com o número 14 e imprimir na tela o resultado.

Ajuda: o operador % calcula o resto da divisão entre dois números inteiros.
(Ex.: “5 % 2” dá 1 como resultado; enquanto “4 % 2” dá 0 como resultado).



Array

Array

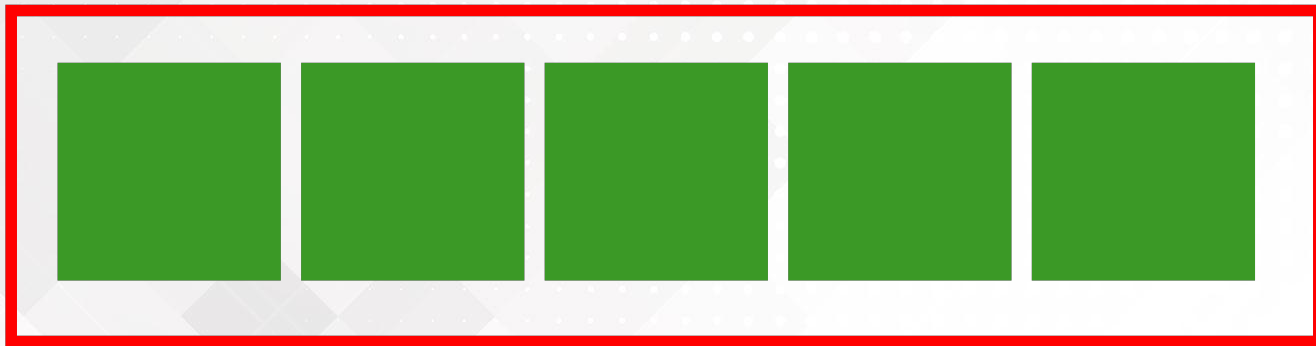
Mapa de Dados

- Estrutura de dados que é identificado por um índice;
- O índice sempre começa em 0;
- É como uma planilha excel;
- Pode ser uma ou mais colunas, mas vamos tratar inicialmente com uma só coluna.

	A
1	A
2	B
3	Z
4	J
5	C

Array

Como funciona?



Array

Como funciona?



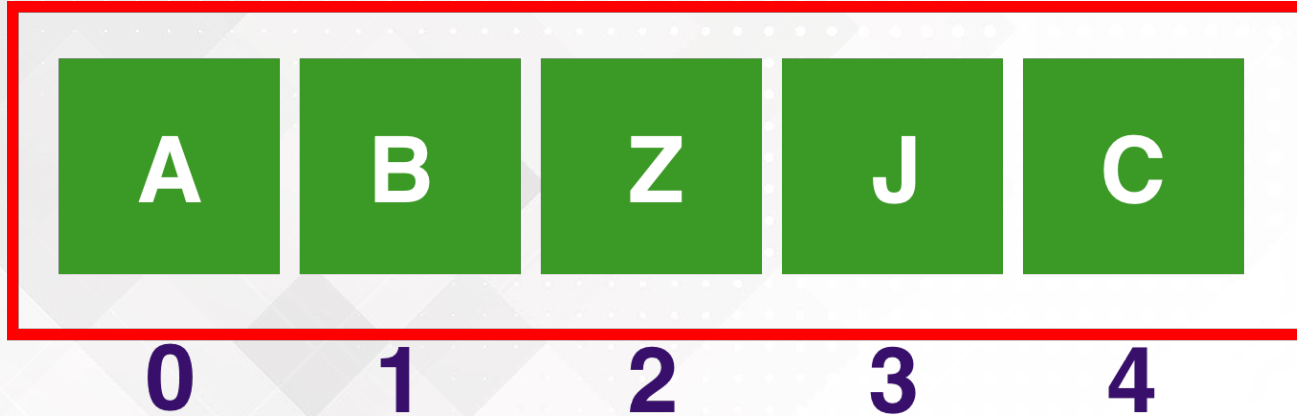
Array

Como funciona?



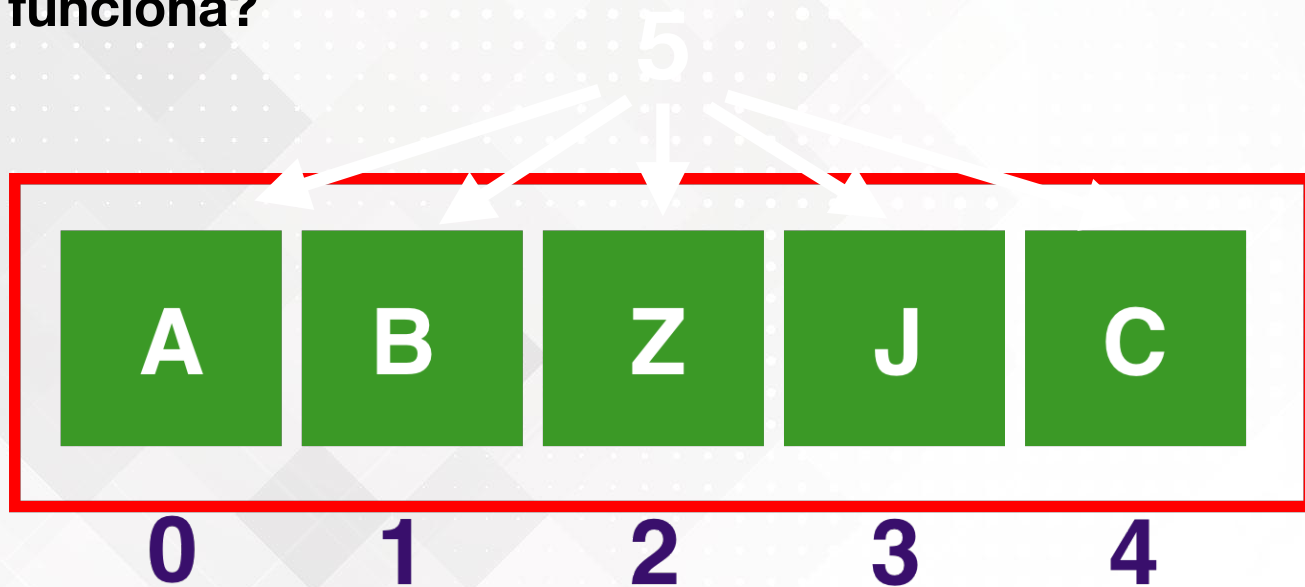
Array

Como funciona?



Array

Como funciona?



Array

Mapa de Dados

- Lembre sempre do excel, com a diferença que o array sempre começa em ZERO!

A	B	Z	J	C
0	1	2	3	4

	A
1	A
2	B
3	Z
4	J
5	C

The background features a large, faint, light-gray geometric pattern of overlapping squares and diamonds. In the top-left and bottom-left corners, there are clusters of small triangles in red, white, and dark blue. A similar cluster is in the top-right corner, and a smaller one is in the bottom-left corner.

Em Swift...

Array

Declaração

```
// Definindo Tipo
var arrayNames: [String]

// Inicializando com array vazio
var array2Names = [String]()

// Inicializando com objetos
var array3Names = ["Narlei", "Américo"]
```


Vamos usar o modelo 2

```
// Inicializando com array vazio  
var array2Names = [String]()
```

Array

Adicionando informações

```
var array2Names = [String]()  
var total = array2Names.count
```

[]

0

```
array2Names.append("Narlei")  
total = array2Names.count
```

["Narlei"]

1

```
array2Names.append("Américo")  
total = array2Names.count
```

["Narlei", "Américo"]

2

Array

Acessando informações

```
var name1 = array2Names[0]  
var name2 = array2Names[1]
```

"Narlei"

"Américo"

Array

Alterando informações

```
var name1 = array2Names[0]
```

```
var name2 = array2Names[1]
```

```
array2Names[0] = "Américo"
```

```
array2Names[1] = "Narlei"
```

```
name1 = array2Names[0]
```

```
name2 = array2Names[1]
```

"Narlei"

"Américo"

"Américo"

"Narlei"

"Américo"

"Narlei"



Let's code!



Dictionary

Dictionary

Dicionário de Dados

- Estrutura de dados que é identificado por uma chave;
- O índice pode ser de vários tipos. (Int, String)
- Como um dicionário, tem a chave e valor.



Dictionary

Declaração

```
// Inicializando Vazio  
var dictionary1 = [String: String]()  
  
// Declarando tipo  
var dictionary2: [String: String]  
  
// Inicializando com objetos  
var dictionary3 = ["narlei": "Professor", "turma": "iOS"]
```


Vamos usar o modelo 1

```
// Inicializando Vazio  
var dictionary1 = [String: String]()
```

Dictionary

Adicionando informações

```
var dictionary1 = [String: String]()  
dictionary1["nome"] = "Narlei"  
dictionary1["materia"] = "iOS"
```

Dictionary

Acessando informações

```
var nome = dictionary1["nome"]  
var text2 = dictionary1["materia"]
```



Let's code!



Loops

While - (Enquanto)



```
var index:Int = 1
while index <= 10 {
    print("index: \(index)")
    index = index + 1
}
```

Repeat - (Repita enquanto)



```
var index = 1
repeat {
    print("index: \(index)")
    index = index + 1
} while index <= 10
```

For - (para)



```
for index in 1...4 {  
    print("index: \(index)")  
}  
  
for index in 1..  
4 {  
  
for index in ["A", "B", "C", "D", "E"] {
```


For - (para)



```
for (index, element) in ["A", "B",  
"C", "D", "E"].enumerated() {
```

Exercícios



1. Imprima na tela a soma dos primeiros dez números naturais ÍMPARES elevados ao quadrado

2. Imprimir números aleatórios entre 0 e 5, até que se imprima um 3.
Ajuda: A função `arc4random_uniform(_ n: Int)` (definida na biblioteca Foundation) retorna um número aleatório entre 0 e o parâmetro, exemplo.

```
var numero = arc4random_uniform(5)
```

Links

Cookbook - Hello World

Cookbook - Functions

Cookbook - Array