

# Módulo 02 - JavaScript

## Aula 04 – Estruturas de Controle

- Estruturas de Controle
- If / else
- Switch
- For
- While / Do While
- Iteração de Arrays e Objetos

# Objetivos

- Entender o que são as Estruturas de Controle.
- Saber definir o que é um looping de repetição.
- Saber definir o que é uma variável de controle.
- Fazer iterações por Arrays e Objetos.

# *1. Estruturas de Controle*

## ***Estruturas de Controle***

# *1. Estruturas de Controle*

## **Estruturas de Controle**

- Componentes principais de um algoritmo.
- São divididos em :

### **Desvios Condicionais:**

- Alterações no fluxo lógico do código, separam quais caminhos o algoritmo pode seguir.

### **Laços de Repetição: (Loopings)**

- Definem um bloco de códigos a ser repetido.

# *1. Estruturas de Controle*

## **Estruturas de Controle:**

- **Desvios Condicionais**
  - If / else
  - Switch / Case
- **Laços de Repetição**
  - For
  - While / Do While
  - For In / For Of

## *2. IF / ELSE*

***If / Else***

## 2. IF / ELSE

### **If / Else**

- Mais recorrente estrutura de controle
- Bifurcação no fluxo de processamento do código
- Recebe um argumento e possui dois comportamentos, um para caso o argumento seja verdadeiro outro para caso o argumento seja falso
- Pode ser usado de forma encadeada, apesar de não ser recomendado

## 2. IF / ELSE

If / Else – Aplicação simples

```
if (valor){  
    // Código caso o valor seja verdade  
}  
  
else {  
    // Código caso o valor seja falso  
}
```



## 2. IF / ELSE

If / Else – Aplicação composta

```
if (valor = opção 01){  
    // Código caso o valor seja igual a opção 01  
}  
else if (valor = opção 02){  
    // Código caso o valor seja igual a opção 02  
}  
else{  
    // Caso não seja nenhuma opção  
}
```

### 3. *SWITCH / CASE*

## ***Switch / Case***

### 3. SWITCH / CASE

#### Switch case

- Bifurcação múltipla no fluxo de processamento
- Compara seu argumento com uma série de possibilidades
- Mais recomendado do que usar “ifs” encadeados
- Guarda dentro de si vários blocos de código, diferente do setup do if que só comporta dois blocos

### 3. SWITCH / CASE

Switch case

```
switch (valor){  
    case 'opt 1        //Bloco de código opção 01  
    break;  
    case 'opt 2        //Bloco de código opção 01  
    break;  
    default:  
        //Bloco de código para nenhuma opção  
}
```

## 4. *FOR*

***For***

## 4. FOR

### For

- Estrutura de controle mais complexa, pois seu argumento é construído de 3 parâmetros separados por ;
- Primeiro dos muitos loopings de repetição das estruturas de controle
- Argumentos
  - Declaração de uma variável com papel de contador
  - Condição que fará o looping ser quebrado
  - Modificação no contador

## 4. FOR

For

```
for (DECLARAÇÃO ; CONDIÇÃO ; MODIFICAÇÃO) {  
    // Código a se repetir X vezes  
}
```

**DECLARAÇÃO** : Uma variável simbólica, que terá seu valor alterado toda vez que o looping rodar

**CONDIÇÃO** : Gatilho que fará o looping ser finalizado

**MODIFICAÇÃO** : Qual modificação será feita na variável a cada vez que o looping for processado

## 4. FOR

**For**

```
for (var i = 0 ; i < 5 ; i++){  
    // Código a se repetir 5 vezes  
}
```

**Leitura :**

*i* possui valor *x*

*i* é menor que 5 ?

código é processado

*i* = *i* + 1

repete



## *5. WHILE / DO WHILE*

### ***While / Do While***

## 5. *WHILE / DO WHILE*

### **While**

- Estrutura de repetição mais simples que o “For”, apesar de possuir um conceito semelhante
- É passado um único argumento que enquanto for verdade repetirá o código dentro das {}
- Enquanto o For possui uma formula que especifica quantas vezes o looping deve ser processado, o While pode se repetir indefinidas vezes
- Pode ser perigoso se não usado com cautela, pois pode gerar um processamento infinito no script

## 5. WHILE / DO WHILE

### While

```
while (argumento){  
    // Código que se repete enquanto o argumento for  
    verdadeiro  
    // Não esqueça de alterar a variavel do argumento  
    ao decorrer do loop  
}
```

## 5. WHILE / DO WHILE

### Do While

- Muito semelhante ao laço de repetição while, mas enquanto o while faz a comparação depois é processado, o do while primeiro realiza um processamento depois é feita uma averiguação se seu argumento continua sendo verdadeiro

**do{**

**//Código que se repete**

**} while(argumento)**

## *5. Iteração de Arrays e Objetos*

# ***Iteração de Arrays e Objetos***

# 5. Iteração de Arrays e Objetos

## Iteração em Arrays / Objetos e sua importância

- *Muitas vezes na programação surge a necessidade de aplicar um fragmento de script em cada um dos elementos de um array (ou propriedades de um objeto, que veremos mais pra frente no curso)*
- *Iteração é o nome que se dá ao ato de percorrer todos os elementos de uma lista realizando alguma operação*
- As duas mais importantes iterações em variáveis:
  - Modificar todos os elementos de um array com um único script
  - Procurar algo em específico no meio de todos elementos de um array

## 5. Iteração de Arrays e Objetos

### For in

- Poderoso looping de iteração que se repete para cada elemento de um array, que é passado como parâmetro
- Assim como no For possuímos uma variável que muda seu valor toda vez que o loop é processado, no For in temos uma variável que assume o **ÍNDICE** de cada elemento de um array percorrido
- `for(var indexQueSeAltera in nomeDoArray) { //  
Código }`

## 5. Iteração de Arrays e Objetos

### For of

- Muito semelhante ao For In, mas sua principal diferença está na primeira variável, que ao invés de assumir o **ÍNDICE** do array ela irá assumir o **VALOR** de cada um dos elementos do array ao decorrer da iteração
- `for(var valorQueSeAltera in nomeDoArray) { // Código }`



# Questionário

1. Crie uma função que receba um valor de ping de como parâmetro e que mostre no console uma mensagem dizendo se a conexão está boa ou se ela está lenta.
2. Crie um array com uma lista de menus, depois crie uma função que receba um elemento do array como parâmetro e mostre no console uma mensagem qual menu deve ser aberto.
3. Crie uma função que receba um array como parâmetro e que adicione 20 números consecutivos neste array, mostrando no console todas as vezes que esta lista for alterada.
4. Crie uma função que delete todos elementos de um grande array até que só restem 3 elementos.
5. Crie uma função que percorra todos os nomes de uma lista de clientes online e que gere uma nova lista com mensagens dizendo “Fulano está online”, depois mostre no console todas essas mensagens.