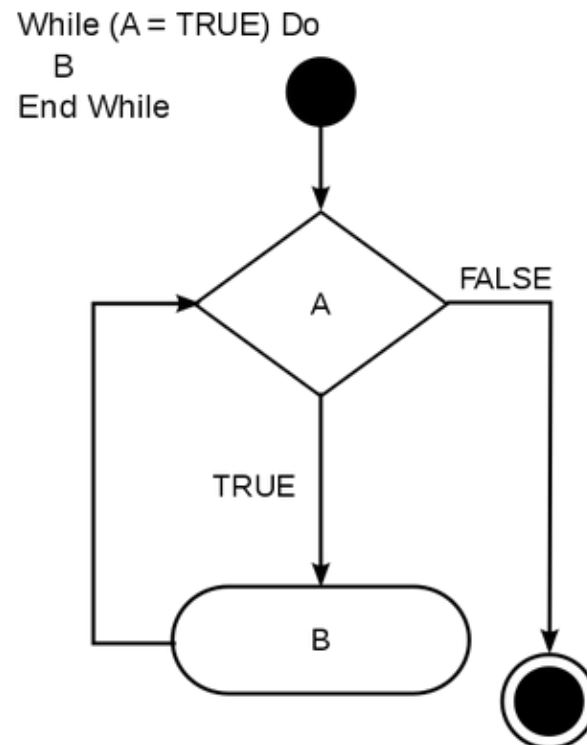


Estrutura de Repetições WHILE

O **while** é usado quando queremos que o bloco de código seja repetido até que uma condição seja satisfeita. Ou seja, é necessário que uma expressão booleana dada seja verdadeira. Assim que ela se tornar falsa, o **while** para.



Estruturas de Repetições WHILE

- ▶ O while é usado quando queremos que o bloco de código seja repetido até que uma condição seja satisfeita. Ou seja, é necessário que uma expressão booleana dada seja verdadeira. Assim que ela se tornar falsa, o while para.

<script>

```
let numero = 0
```

```
while (numero <= 10) {
```

```
  alert(numero)
```

```
  numero = numero + 1
```

```
}
```

```
</script>
```

Estrutura de Repetições WHILE

```
numero = 0
while numero <= 10:
    print(numero)
    numero = numero + 1
```

```
numero = 0
while numero <= 10:
    print(numero)
    numero = numero + 1
else:
    print("Acabou o Laço While")
```

Estrutura de Repetições WHILE

Utilizando o break para tirar loop infinitos:

Javascript

```
numero = 0
while numero <= 10 {
  print(numero)
  break
}
```

Python

```
numero = 0
while numero <= 10:
  print(numero)
  break
```

Estruturas de Repetições WHILE

Nesse exemplo calculamos a Tabuada do valor que inserimos em Javascript

```
let numtabuada = Number(prompt("Insira o Numero para Calcular Tabuada"));  
    let multiplicador = 1;  
    while (multiplicador <= 10){  
        alert( numtabuada * multiplicador )  
        multiplicador = multiplicador + 1  
    }  
</script>
```

Estrutura de Repetições WHILE

Nesse exemplo calculamos a Tabuada do valor que inserimos em Python

```
tabuada = int(input("Insira o numero da tabuada"))  
multiplicador = 1  
while (multiplicador <= 10):  
    print(tabuada * multiplicador )  
    multiplicador = multiplicador + 1  
else:  
    print("Fim da Tabuada de ",tabuada)
```

Exercícios WHILE

Escreva em Javascript e Python

EX1:

Nesse exercício vamos mostrar os anos que ocorreram as olimpíadas. Onde iremos passar a data inicial.

Da olimpíada e a data limite da olimpíada. O Programa irá imprimir a saída em tela de todas os anos que ocorreram olimpíadas nesse intervalo de tempo definido.

ex: 1940 há 2020. Irá imprimir em tela todas as datas que ocorreram as olimpíadas em um período de 4 em 4 anos.

EX2:

Escrever um programa que faça o incremento de 500 reais no valor final do salário sendo que o incremento será de 100 reais até totalizar os 500 reais.

E você deve entrar com o valor do salário e o mesmo deve imprimir em tela os aumentos de salário com os incrementos de valores.

Estrutura de Repetição FOR em Javascript

Vamos trabalhar agora com estrutura **for**. A estrutura for trabalha com 3 espaço. 1) Condição inicial da variável. 2) as vezes que irá repetir a ação. 3) alteração do estado da variável incrementando ou diminuindo.

for(espaco 1; espaco 2; espaco 3)

Vamos utilizar o exemplo da contagem de números que utilizamos com o comando while:

```
<script>
```

```
for(let numero =0 ; numero <= 10 ; numero = numero +1 ) {
```

```
    alert(numero)
```

```
}
```

```
</script>
```


Estrutura de Repetição FOR em Javascript

Vamos usar o exemplo da tabuada escrita com o comando **while**, agora com o comando **for**.

```
<script>
```

```
let numtabuada = Number(prompt("Insira o Numero para Calcular Tabuada"));
```

```
let multiplicador = 1;
```

```
for( let multiplicador =1 ; multiplicador <= 10 ; multiplicador++ ) {
```

```
document.write( numtabuada * multiplicador )
```

```
}
```

```
</script>
```

Exercícios FOR JavaScript

Escreva em Javascript

EX1:

Nesse exercício vamos mostrar os anos que ocorreram as olimpíadas. Onde iremos passar a data inicial.

Da olimpíada e a data limite da olimpíada. O Programa irá imprimir a saída em tela de todas os anos que ocorreram olimpíadas nesse intervalo de tempo definido.

ex: 1940 há 2020. Irá imprimir em tela todas as datas que ocorreram as olimpíadas em um período de 4 em 4 anos.

EX2:

Escrever um programa que faça o incremento de 500 reais no valor final do salário sendo que o incremento será de 100 reais até totalizar os 500 reais.

E você deve entrar com o valor do salário e o mesmo deve imprimir em tela os aumentos de salário com os incrementos de valores.

Estrutura de Repetições FOR em Python

O for é usado quando se quer iterar sobre um bloco de código um número determinado de vezes.

O laço **for** nos permite percorrer os itens de uma coleção e, para cada um deles, executar o bloco de código declarado no loop.

Mas para antes trabalharmos com for vamos aprender a nossa primeira estrutura de dados que se chama lista.

Em **Python**, uma **lista** é representada como uma **sequência de objetos** separados por vírgula e dentro de colchetes []. A lista ela é mutável e pode sofrer alteração de valores dentro da mesma.

Exemplo de Lista em Python:

```
frutas = ["Manga", "Pera", "Uva", "Morango"]
```

Estrutura de Repetições FOR em Python

Vamos fazer o laço for percorrer a lista de frutas.

```
frutas = ['Manga', 'Pera', 'Uva', 'Morango']  
for f in frutas:  
    print(f)
```

Podemos percorrer uma lista de numero com o range

```
for intervalonumeros in range(10,20):  
    print(intervalonumeros)
```

```
nome = "Python"  
for letras in nome:  
    print(letras)
```

Estrutura de Repetições FOR em Python

Vamos enumerar os índices da nossa lista:

```
frutas = ['Manga', 'Pera', 'Uva', 'Morango']  
for f, frutas in enumerate(frutas):  
    print(f, frutas)
```

```
frutas = ['Manga', 'Pera', 'Uva', 'Morango']  
for f in frutas:  
    print(frutas[2])
```

```
frutas = ['Manga', 'Pera', 'Uva', 'Morango']  
for f in frutas:  
    print(f[0:4])
```

```
for multiplicacao in [1,2,3,4,5]:  
    print(multiplicacao*10)
```

Exercícios FOR em Python

Crie um programa que imprima na tela numero que começam de 0 até 100.

Crie um algoritmo onde o mesmo percorre uma lista de nome de carros e imprima em tela os elementos da lista criada.

Crie uma lista com nome de Cidades onde contenha a cidade Florianópolis de imprima em tela somente a cidade de Florianópolis da lista criada.

Crie um algoritmo que faça a soma de 5 numero de uma lista com ele mesmo.

Funções em Javascript

Funções são utilizadas para otimizar o código ao invés de digitar repetidamente o mesmo código podemos utilizar uma função. Afim de deixar prático e menor tamanho dos códigos e não repetir a digitação. A mesma executa a linha de código quando é chamada(executada).

Exemplo:

`<script>`

```
function criaespaço() {  
    document.write("<br>");  
}  
    let ano = 2019;  
    document.write("Eu nasci em : " + (ano - 33)) ;  
    criaespaço();  
    document.write("Adriano nasceu em : " + (ano - 26));  
    criaespaço();  
    document.write("Paulo nasceu em : " + (ano - 32)) ;  
    criaespaço();  
</script>
```

Funções em Javascript

Podemos criar as funções passando parâmetros dentro da mesma.

```
<script>
```

```
function soma(a,b) {
```

```
    return a+b
```

```
}
```

```
var s = soma(5,5)
```

```
alert(s)
```

```
</script>
```


Funções em Javascript

Podemos criar as funções chamando outras funções otimizando ainda mais o código.

Nesse exemplo criamos uma função para imprimir uma frase na tela e chamando a variável que pula para outra linha na mesmo código da função.

```
<script>
function criaespaço() {
    document.write("<br>");
}

function mostra(frase) {
    document.write(frase);
    criaespaço()
}

mostra("Mostra Texto na Variável")
mostra("Segunda frase com espaço")
</script>
```

Exercício de Fixação

Escrever uma função que faça uma multiplicação onde serão passados os números nos parâmetros. E saída do resultado da multiplicação seja impresso em tela e em console.

Crie uma segunda função que escrever um valor em tela com 2 funções ou seja função dentro de outra função e escreva em console.

Funções em Python

Declarações de função são feitas usando o comando def :

```
def oi():  
    print("Olá Estou dando Oi")
```

```
oi()
```

Podemos trabalhar com Funções com parametros:

```
def soma(a, b):  
    return a + b
```

```
c= soma(1,3)  
print(c)
```

Funções em Python

```
def identificacao(meu_nome,idade):  
    print('Olá',meu_nome,'Sua idade é:',idade)  
  
identificacao("João",34)  
  
salario = float(input("Digite seu Salário"))  
  
def salario_descontado_imposto(salario, imposto):  
    return salario - (salario * imposto * 0.01)  
  
desconto = salario_descontado_imposto(salario)  
print(desconto)
```

Funções em Python

Parâmetros Nomeados ou argumentos nomeados são onde definimos nomes e valores e sequencia de valores das nossas funções.

Vamos usar os mesmo exemplos das funções acima:

```
def identificacao(meu_nome,idade):  
    print('Olá',meu_nome,'Sua idade é:',idade)
```

```
identificacao(idade=34, meu_nome="João")
```

```
salario = float(input("Digite seu Salário"))
```

```
def salario_descontado_imposto(salario, imposto=27.0):  
    return salario - (salario * imposto * 0.01)
```

```
desconto = salario_descontado_imposto(salario, imposto=10.0)  
print(desconto)
```

Funções em Python

Funções de Parâmetros padrão.

São funções que você pode passar argumentos ou não. Os argumentos são opcionais que a função irá executar da mesma maneira.

```
print(" Argumento")  
print()
```

Funções em Python

Funções com Condicionais:

```
def validausuario(nome_usuario,senha):  
    if nome_usuario == "admin" and senha == "python" :  
        return "Usuário e senha Corretos Bem Vindos"  
    elif nome_usuario == "admin":  
        return "Usuário Correto"  
    elif senha == "python":  
        return "senha correta"  
    else:  
        return "Usuário e Senha Incorretos"  
  
print(validausuario("", ""))
```

Funções em Python

Podemos passar uma função dentro de outra função

```
def soma(a,b):  
    return a + b
```

```
def calculo (num1,num2, fun=soma):  
    return fun(num1,num2)
```

```
print(calculo(3,2))
```


Exercícios Função

Crie uma função que faça a soma de 3 números e imprima em tela o resultado da soma.

Crie uma algoritmo utilizando funções onde calcule o valor hora ganho de um funcionário. Sendo que o valor hora devera ser um padrão nomeado. E o numero de horas trabalha deverá ser inserido pelo usuário. E imprima em tela o total ganho desse funcionário baseado nas horas trabalhadas.

Crie um programa onde a função irá validar o nome e o CPF da pessoa estão corretos baseados nos dados apresentados. Valide com retornos em tela se o CPF está correto se o nome está correto.

Crie uma função dentro de outra função onde a mesma irá somar e multiplicar os valores informados na função.