

# Linguagens e suas funções

## PYTHON

### IDE

- A IDE ajuda o(a) programador(a) a criar os seus códigos;
- Para o Python, utilizarei o Visual Studio Code.

### CRIAÇÃO DE VARIÁVEIS

- Não é preciso definir o tipo da variável.

```
>>> nome = "Interface up"
>>>
>>> █
```

### COMANDOS ÚTEIS

- O comando print retorna uma informação na tela;

```
>>> print(nome)
Interface up
>>> █
```

Obs.: Essas informações foram adicionadas no terminal do Python.

- O comando type verifica qual é o tipo da informação desejada.

```
>>> type(nome)
<class 'str'>
>>> █
```

Obs.: str significa que a variável é do tipo string.

### ESTRUTURA CONDICIONAL

- A estrutura condicional refere-se ao if;
- Se (if) a condição adicionada seja verdadeira, então processará todas as informações para verdadeiro;
- Se (if) a condição adicionada seja falsa, então processará todas as informações para falso;
- Para a estrutura condicional funcionar corretamente, todas as informações devem seguir a indentação acertadamente.

```
nome = "interface"
if nome != "interface up":
    print("O nome não se encontra correto! =[ ")
else:
    print("O nome está correto =] ")
```

Explicação: Se (if) nome for diferente (!=) de interface up, então o nome estará incorreto. Se o nome não for diferente, ou seja, se ele for igual a interface up, então o nome estará correto.

Saída:

```
O nome não se encontra correto! =[
```

### ESTRUTURA DE REPETIÇÃO OU LOOP

- A estrutura de repetição ou loop refere-se ao while e for;
- Enquanto determinada informação for verdadeira,

## Jacqueline Rodrigues - Interface Up!

ocorrerá quantas vezes for programada;

- Importante: para todo while precisa ter um contador, geralmente, identificado pela letra i. Esse contador realizará o inter rompimento conforme programado;
- Para a estrutura de repetição ou loop funcione corretamente, todas as informações devem seguir a indentação acertadamente.

While:

```
i = 0

while i <= 10:
    print(i)
    i += 1
```

Explicação: Enquanto (while) o contador i for menor ou igual a 10 ( $\leq 10$ ) mostre pra mim o contador na tela - print(i). Para que o contador aumente seu valor a cada vez que é rodado, somamos mais 1, até que ele chegue a 10.

Saída:

```
0
1
2
3
4
5
6
7
8
9
10
```

For:

```
nome = "Interface up"
for letra in nome:
    print(letra)
```

Explicação: Para cada letra do nome, imprimir letra por letra.

Saída:

```
I
n
t
e
r
f
a
c
e

u
p
```

### JAVA

#### IDE

- A IDE ajuda o(a) programador(a) a criar os seus códigos;
- Para o Java, utilizarei o Eclipse.

#### PARTE INICIAL DO JAVA

- Todo programa do Java utiliza a seguinte situação:

```
public class PrimeiroPrograma {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
    }
}
```

- O public informa como poderá ser acessado, neste caso, de forma pública (modificador de acesso);
- O static informa que o método é estático, ou seja, essa classe não

precisa ser chamada para chamar o método;

- O void é um tipo de retorno de um método, neste caso, esse método não traz nenhum retorno;
- O main refere-se ao nome do método;
- O String[] args informa que o método pode receber algum parâmetro.

### CLASSE, ATRIBUTO, OBJETOS E MÉTODOS

- Classe refere-se ao projeto do objeto;
- Objeto refere-se à entidade, podendo ser concreto (vejo e consigo tocar) ou abstrato (vejo, mas não toco);
- Atributo refere-se às características do objeto;
- Métodos refere-se ao que o objeto pode realizar, quais serão os seus comportamentos.

### CRIAÇÃO DE VARIÁVEIS

- Sempre precisa realizar a declaração de variáveis;

```
String meuNome = "Interface up";
```

### COMANDOS ÚTEIS

- O comando System.out.println() mostra a informação na tela;
- Esse comando tem o intuito de apresentar a informação na tela de qualquer dispositivo.

```
String meuNome = "Interface up";  
System.out.println(meuNome);
```

### ESTRUTURA CONDICIONAL

- A estrutura condicional refere-se ao if;
- Se (if) a condição adicionada seja verdadeira, então processará todas as informações para verdadeiro;
- Se (if) a condição adicionada seja falsa, então processará todas as informações para falso;
- Deve sempre ser adicionado entre chaves.

```
String meuNome = "Interface";  
  
if (meuNome != "Interface up") {  
    System.out.println("O nome encontra-se errado");  
} else {  
    System.out.println("O nome encontra-se correto");  
}
```

### ESTRUTURA DE REPETIÇÃO OU LOOP

- A estrutura de repetição ou loop refere-se ao while e for;
- Enquanto determinada informação for verdadeira, ocorrerá quantas vezes for programada;
- Importante: para todo while e for precisa ter um contador, geralmente, identificado pela letra i. Esse contador realizará o interrompimento conforme programado;
- Para o funcionamento, precisa ser adicionado as chaves

```
int i = 0;  
  
while (i <= 10) {  
    System.out.println(i);  
    i++;  
}
```

## Jacqueline Rodrigues - Interface Up!

```
for (int i = 0; i <= 10; ++i) {  
    System.out.println(i);  
}
```

Explicação: Enquanto (while ou for) o contador i for menor ou igual a 10 (<= 10) mostre pra mim o contador na tela – System.out.println(i). Para que o contador aumente seu valor a cada vez que é rodado, somamos mais 1, até que ele chegue a 10.

Saída: as saídas serão iguais para as duas formas.

```
0  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

### SHELL SCRIPT

#### IDE

- A IDE ajuda o(a) programador(a) a criar os seus códigos;
- Para o Shell Script, utilizarei o Visual Studio Code;
- Além disso, podemos utilizar o terminal do próprio sistema operacional, como é o caso do Linux.

### CRIAÇÃO DE VARIÁVEIS

- Existem vários modos de variáveis. No entanto, para se adicionar alguma informação, utilizamos o exemplo abaixo:

```
NOME="Interface up"
```

Obs.: O exemplo acima foi escrito no Visual Studio Code.

### COMANDOS ÚTEIS

- Sempre deve-se adicionar a direção do bash, com o comando: `#!/bin/bash` (identificado com o .sh);
- O comando echo imprimirá na tela as informações adicionadas;

```
echo $NOME
```

- O comando read lê uma informação adicionada pelo usuário.

```
echo "Informe o seu nome: " NOME  
read $NOME
```

### ESTRUTURA CONDICIONAL

- A estrutura condicional refere-se ao if;
- Se (if) a condição adicionada seja verdadeira, então processará todas as informações para verdadeiro;
- Se (if) a condição adicionada seja falsa, então processará todas as informações para falso;
- Deve-se sempre abrir e fechar a estrutura condicional.

```
#!/bin/bash  
  
NOME="interface"  
  
if [ $NOME == "interface" ]  
then  
    echo nome correto  
else  
    echo nome incorreto  
fi
```

Obs.: código criado pelo terminal do sistema operacional Linux.

Saída:

```
nome correto
```

### ESTRUTURA DE REPETIÇÃO OU LOOP

- A estrutura de repetição ou loop refere-se ao while e for;
- Enquanto determinada informação for verdadeira, ocorrerá quantas vezes for programada;
- Importante: para todo while precisa ter um contador, geralmente, identificado pela letra i. Esse contador realizará o interrompimento conforme programado;
- Ambos precisam ser abertos e fechados;

Utilização do while:

```
#!/bin/bash

I=0

while [[ $I -le 10 ]]
do
    echo $I
    I=$((I + 1))
done
```

Saída:

```
0
1
2
3
4
5
6
7
8
9
10
```

Utilização do for:

```
#!/bin/bash

for i in $(seq 10)
do
    echo $i
done
```

Saída:

```
1
2
3
4
5
6
7
8
9
10
```

### JAVA SCRIPT

#### IDE

- A IDE ajuda o(a) programador(a) a criar os seus códigos;
- Para o Java Script, utilizarei o Visual Studio Code.

### COMANDOS ÚTEIS

- O comando var informa que aquela determinada informação é uma variável;

```
var a = "interface up"
```

```
> a
'interface up'
```

Obs.: Essas informações foram adicionadas no terminal do node.js

- O comando function cria uma função para realizar, por

exemplo, eventos (ações) no site;

```
function clicar() {  
    // informações para que ocorra o evento clicar  
}
```

Explicação: ao definir o evento como clicar, podemos, por meio do site, clicar na parte específica do site, na qual determinada ação adicionada no Java Script ocorrerá (se clicar na parte específica, por exemplo, pode mudar a cor, etc.)

## ESTRUTURA CONDICIONAL

- Para que o Java Script funcione no HTML, precisa adicioná-lo dentro da tag script;
- A estrutura condicional refere-se ao if;
- Se (if) a condição adicionada seja verdadeira, então processará todas as informações para verdadeiro;
- Se (if) a condição adicionada seja falsa, então processará todas as informações para falso;
- Sempre deve ser adicionado as chaves para o seu funcionamento de modo correto.

```
<script>  
    var nome = "Interface"  
  
    if (nome == "Interface up") {  
        window.alert("O nome está correto")  
    } else {  
        window.alert("O nome está incorreto")  
    }  
</script>
```

Saída:



## ESTRUTURA DE REPETIÇÃO OU LOOP

- Para que o Java Script funcione no HTML, precisa adicioná-lo dentro da tag script;
- A estrutura de repetição ou loop refere-se ao while;
- Enquanto determinada informação for verdadeira, ocorrerá quantas vezes for programada;
- Importante: para todo while precisa ter um contador, geralmente, identificado pela letra i. Esse contador realizará o interrompimento conforme programado;
- Sempre deve ser adicionado as chaves para o seu funcionamento de modo correto.

```
<script>  
    var i = 0  
  
    while (i <= 10) {  
        window.document.write(i + "\n")  
        i++  
    }  
</script>
```

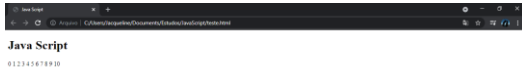
Obs. 1: O window.document.write escreve na tela a informação adicionada entre parênteses.

Obs. 2: O Java Script realiza diversos eventos. Com o uso do if e do while,

# Jacqueline Rodrigues - Interface Up!

podemos programar como tal e tal evento ocorrerá, além de suas características (de cada evento).

Saída:



## PHP

### IDE

- A IDE ajuda o(a) programador(a) a criar os seus códigos;
- Para o PHP, utilizarei o Visual Studio Code.

### CRIAÇÃO DE VARIÁVEIS

- Em PHP as variáveis são criadas com o uso do \$.

```
$nome = "interface up";
```

### COMANDOS ÚTEIS

- O comando echo retorna à informação na tela;

```
echo "Interface up";
```

- Ao realizar a ligação com o HTML podemos receber os valores do HTML, utilizando o comando \$\_GET (sempre vindo de um formulário).

```
$nome = $_GET["ligação_com_o_html"]
```

Explicação: com o \$\_GET, o usuário preencherá um campo vindo do HTML e esse campo será adicionado na variável nome, por exemplo.

## ESTRUTURA CONDICIONAL

- A estrutura condicional refere-se ao if;
- Se (if) a condição adicionada seja verdadeira, então processará todas as informações para verdadeiro;
- Se (if) a condição adicionada seja falsa, então processará todas as informações para falso;
- Sempre deve ser adicionado as chaves para o seu funcionamento de modo correto;
- Receberemos a determinada informação do HTML e, através do PHP, realizaremos a condição.

Formulário HTML:

Nome:

Estrutura condicional em PHP:

```
<?php
$nome = $_GET["nome"]; // a variável nome veio do HTML
echo "Você informou que o seu nome é: $nome <br>";

if ($nome == "Interface up") {
    echo "Porém, o seu nome está correto";
} else {
    echo "Porém, o seu nome está incorreto";
}
?>
```

Saída:

Você informou que o seu nome é: Interface  
Porém, o seu nome está incorreto

## ESTRUTURA DE REPETIÇÃO OU LOOP

- A estrutura de repetição ou loop refere-se ao while;
- Enquanto determinada informação for verdadeira,

ocorrerá quantas vezes for programada;

- Sempre deve ser adicionado as chaves para o seu funcionamento de modo correto;
- Importante: para todo while precisa ter um contador, geralmente, identificado pela letra i. Esse contador realizará o interrompimento conforme programado;
- Receberemos a determinada informação do HTML e, através do PHP, realizaremos a repetição.

Formulário HTML:

Número:

Estrutura de repetição em PHP:

```
<?php

$numero = $_GET["num"];

while ($numero <= 10) {
    echo $numero;
    echo "<br>";
    $numero++;
}

?>
```

Explicação: Por meio do HTML, com o formulário, o usuário poderá escolher qualquer número. No entanto, se o número escolhido for menor ou igual a 10, ele rodará na repetição. Caso o número seja maior, não passará pela repetição. Para a saída abaixo, o número escolhido foi 3.

Saída:

3  
4  
5  
6  
7  
8  
9  
10

## HTML E CSS

### IDE

- A IDE ajuda o(a) programador(a) a criar os seus códigos;
- Para o HTML e o CSS, utilizarei o Visual Studio Code.

### RELEMBRANDO

- O HTML é uma linguagem de marcação e composto por tags;
- No HTML adicionamos o conteúdo;
- Toda tag, no HTML que for aberta, precisa ser fechada;
- O CSS é o designer da página;
- Para o HTML saber onde está o CSS, precisa adicioná-lo dentro da tag style.

### ALGUMAS TAGS DO HTML

- O !DOCTYPE html informa ao navegador qual a versão do HTML;

```
<!DOCTYPE html>
```

- A tag html lang="pt-br" informa a linguagem que o navegador trabalhará. Neste caso, em Português – Brasil;

```
<html lang="pt-br">
```



## Jacqueline Rodrigues - Interface Up!

- A tag head é a parte do HTML que não é apresentada pelo navegador, porém, é no head que adicionamos o CSS. Aqui também adicionamos o título da página;

```
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Nome da página</title>
</head>
```

- A tag body é a parte do HTML que será apresentada pelo navegador, ou seja, é onde fica o conteúdo do site;

```
<body>
  <!-- adiciona o conteúdo -->
</body>
```

- A tag h1 refere-se ao título de nível 1. Podendo ir do nível 1 ao 6;

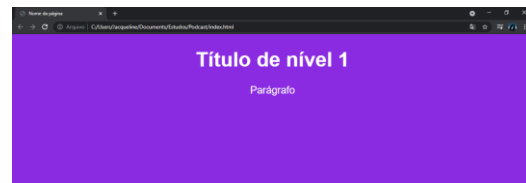
```
<body>
  <h1>Título de nível 1</h1>
</body>
```

- A tag p refere-se ao parágrafo que será adicionado no site.

```
<body>
  <h1>Título de nível 1</h1>
  <p>Parágrafo</p>
</body>
```

```
<style>
  body {
    background: blueviolet;
    font: normal 20pt Arial;
    color: white;
    text-align: center;
  }
</style>
```

O site no final:



### O CSS

- Com o CSS podemos adicionar cores, alterar tamanhos dos parágrafos, títulos, etc. Algumas formas de realizar é:
- Dentro do body podemos adicionar cor de fundo da página, alterar a font, mudar a cor da letra e deixar o texto centralizado.

*Jacqueline Rodrigues - Interface Up!*

*Obrigada!!*

*Interface up!*