

# Preparando o Ambiente

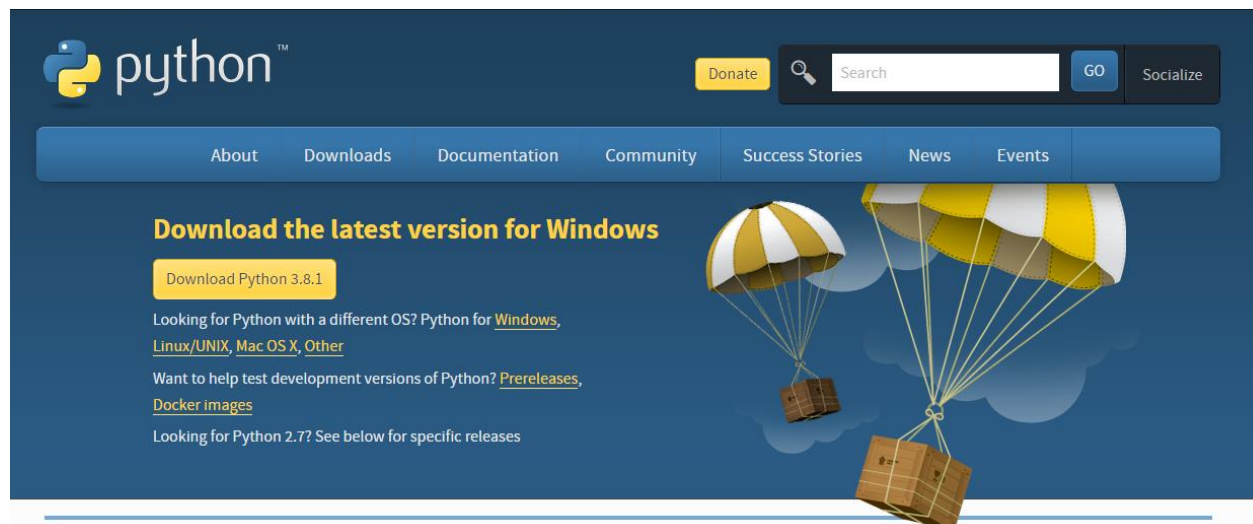
- Para começarmos a programar precisamos instalar as bibliotecas de linguagem do python.

A mesma pode ser baixada no endereço:

<https://www.python.org/downloads/>

Baixe o Python para sua versão de Sistema Operacional.

Ex: Windows, Linux, MacOS

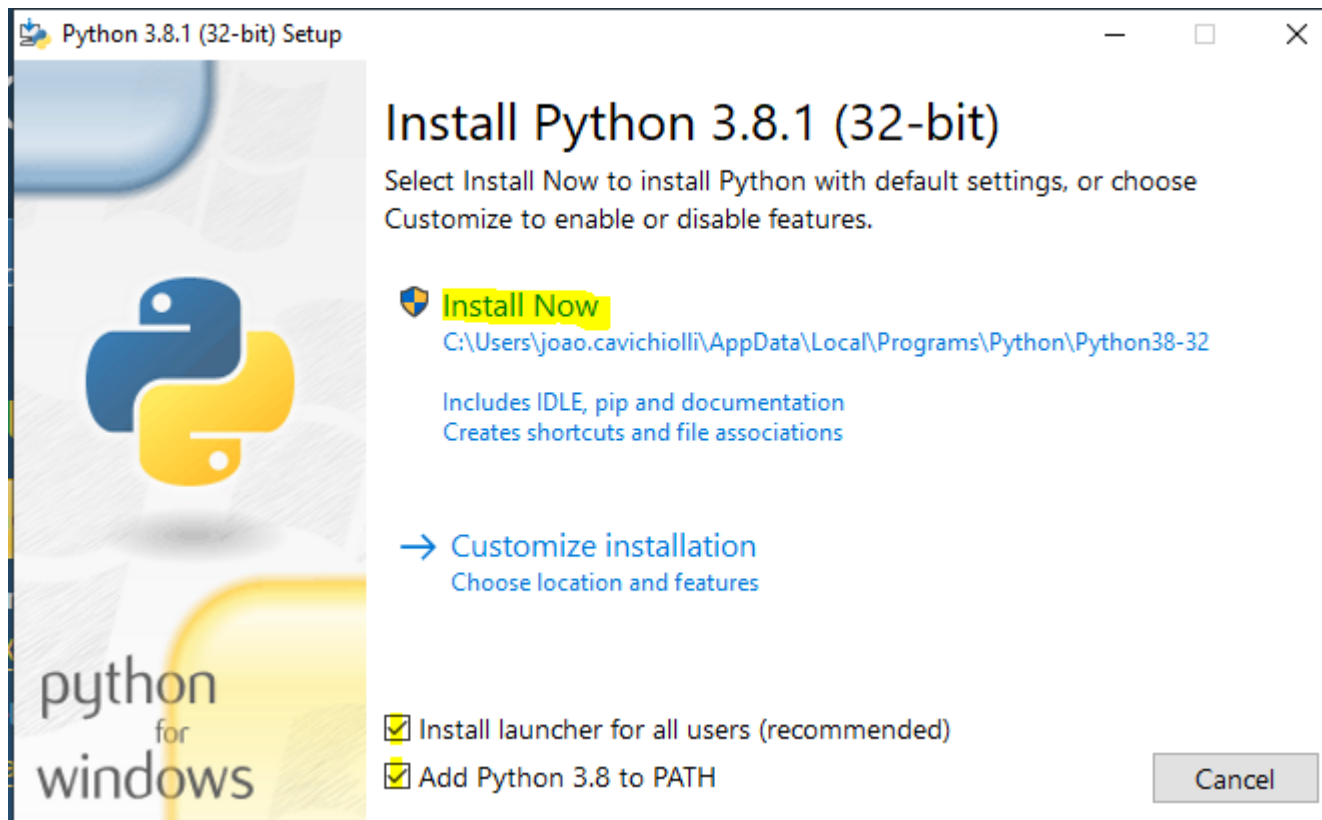


# Preparando o Ambiente

Instalar o Python marcando as opções:

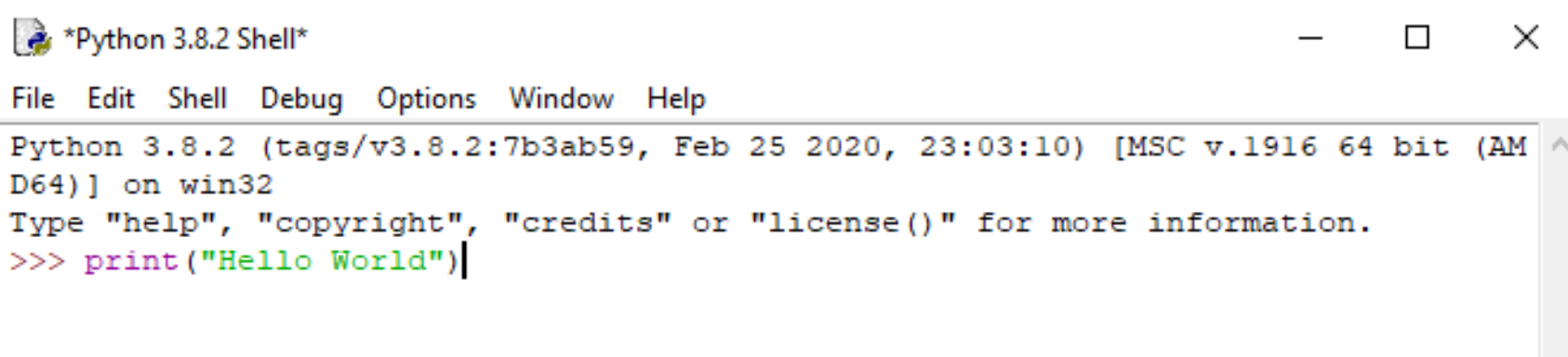
**Install launcher for all users**

**Add Python 3.8 to PATH** e depois clicar em **Install Now**



# Iniciando os primeiros Códigos

- ▶ Vamos utilizar a IDLE do Python para inserirmos nossos primeiros códigos e testar como o Python irá funcionar.



```
*Python 3.8.2 Shell*  
File Edit Shell Debug Options Window Help  
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>> print("Hello World")
```

# Iniciando os primeiros Códigos

Digite os primeiros Códigos:

```
print("Hello World")
```

```
3 + 2
```

```
help()
```

```
print("seu nome")
```

```
21 == 3* 7
```

# Iniciando os primeiros Códigos

- ▶ Vamos utilizar o bloco de notas para inserirmos nossos primeiros códigos e testar como o Javascript irá funcionar.

**<h3> Primeiro Programa</h3>**

Criar um evento com saída com uma mensagem de texto. E um pop-up.

```
<script>
```

```
alert("Bem Vindo a Programação");
```

```
</script>
```

# Iniciando os primeiros Códigos

Utilizaremos o Visual Studio Code para escrever nosso códigos

Para criarmos o primeiro programa vamos criar um código HTML e salvar o arquivo em **.html** para utilizarmos com o Javascript.

<> primeiroprograma.html X


C: > Users > joao.cavichioli > Desktop > <> primeiroprograma.html > ...


```
1  <h3> Primeiro Programa</h3>
2
3  |
4  <script>
5  alert("Bem Vindo a Programação");
6  </script>
7
```

# Iniciando os primeiros Códigos

Utilizaremos o Visual Studio Code para escrever nosso códigos

Para criarmos o primeiro programa vamos criar um código Python e salvar o arquivo em **.py** para utilizarmos com o Python.

 primeiroprograma.py ✕

C: > Users > joao.cavichioli > Desktop >  primeiroprograma.py

```
1  print("Olá Mundo")
```

Se percebemos que a mensagem saiu com erros de acentuação no nosso código JavaScript. Para corrigirmos vamos inserir no nosso código o padrão UTF-8.

**<meta charset="UTF8">**

Salvar o código e atualizar a pagina e verificar se a mensagem está correta.

```
C: > Users > joao.cavichioli > Desktop > <> primeiroprograma.html > ...
```

```
● 1  <meta charset="UTF8">
   2
   3  <h3> Primeiro Programa</h3>
   4  |
   5
   6  <script>
   7  alert("Bem Vindo a Programação");
   8  </script>
   9
```



# Meu Código não funcionou: Como Testar ?

Quando não usamos uma IDE de desenvolvimento como o caso do VS Code e nosso código não funciona como sabemos se temos erros no nosso código ?

Vamos inserir uma linha adicional ao nosso código conforme abaixo:

```
<script>
```

```
alert("Bem Vindo a Programação");
```

```
Alert"Meu Primeiro Programa";
```

```
</script>
```

Quando salvamos e tentamos executar a pagina não irá exibir o popup. Para vermos onde está o erro utilizamos a opção **Inspecionar** dos navegadores

# Comentários de Códigos

Dentro do nosso código podemos querer deixar algumas informações sobre o código escrito.

Ou então até mesmo não executar parte do código sem ter a necessidade de excluí-lo e redigitar novamente. Para isso utilizamos o comentário

Segue o exemplo de como usar um comentário dentro do código de programação:

Javascript:

```
// para comentários de uma linha

/*
    para comentarios de várias linhas
*/
```

Python:

```
# Comentário de 1 linha

"""
    Comentário de Varias Linhas
"""
```

# Escrevendo Saída em tela para o Usuário

Em JavaScript podemos escrever para o usuário de outras formas além de um pop-up para evitar de ficar clicando em OK em repetidas vezes. Podemos utilizar o **document.write**.

```
<script>  
document.write("Bem Vindo a Programação");  
document.write("Meu Segundo Programa");  
</script>
```

O mesmo ficou um do lado do outro sem quebra de linha. Para executar a quebra de linha inserimos a TAG **<br>**

# Escrevendo para o Usuário

```
<script>  
document.write("Bem Vindo a Programação <br>");  
document.write("Meu Segundo Programa");  
</script>
```

Ele efetuou a quebra de linhas dos 2 itens de códigos escritos no document.write

# Escrevendo Saída em tela para o Usuário

Em python podemos escrever para o usuário com a Função **print**

```
print("Mensagem Saida para Usuário")
```

# Tipos de Dados

Nas Liguagens de Programação trabalhamos com inúmeros tipos de dados. Porém os mais comuns são:

String  
Integer  
Float  
Booleano

Exemplo de Integer

```
>>> 1    >>> 2    >>> 45
```

Exemplos de Float

```
>>> 2.3    >>>45.25    >>> 10.20
```

# Tipos de Dados

Exemplo de String

```
>>> "Texto" >>> 'Texto'
```

Exemplo de Booleanos

```
>>> True >>> False
```

# Tipos de Dados

O Tipo de dados em String(texto) em uma linguagem de alto nível tipo o Python e o Javascript possuem diversas formas de ser declaradas e formatadas

Dados tipo string pode ser escritas:

Print("Olá Mundo") ou print('Olá Mundo') pode ser utilizado “ ou ‘ aspas.

Em Python ainda podemos utilizar 3 aspas para fazer a quebra de linhas de uma string.

```
print(''' Ola'  
Mundo  
''')
```



# Tipos de Dados

Tratando String e uma sequencia de posições na linguagem Python:  
Exemplo:

```
>>> fruta = "laranja"
>>> fruta[1]
'a'
>>> fruta[0:4]
'lara'
```

Podemos trazer o tamanho de caracteres de uma string:

```
>>> len(fruta)
7
```

Podemos utilizar algumas formatações como COUNT e REPLACE

```
>>> fruta.count('a')
3
>>> fruta.replace("laranja","abacaxi")
'abacaxi'
```

# Tipos de Dados

Para testarmos o tipos de dados float ou Integer basta executar operações aritméticas no Python.

```
>>> 3 +2           >>> 34 - 2.34
```

Para sabermos se o valor dos dados executamos no comando type.

```
>>> type(20)
```

O retorno se dará `<class 'int'>` dizendo que esse numero é inteiro

O mesmo irá acontecer com o tipo float

```
>>> type(45.2)
```

```
<class 'float'>
```

# Tipos de Dados

Vamos escrever um código em Javascript e vamos entender como os tipos de dados funcionam:

```
<script>
```

```
document.write("Minha idade é: ");
```

```
document.write(33);
```

```
</script>
```

Notem que a segunda linha foi junto com a primeira linha e o valor ficou entre as aspas. Adicione as aspas e analise o comportamento do código. O mesmo ficou da mesma forma. Então qual a diferença de escrever o numero com “” ou sem “” ?

# Tipos de Dados

Vamos escrever o mesmo código com as aspas e fazendo uma soma e depois o mesmo numero sem as aspas e analise a diferença.

```
<script>
```

```
document.write("Minha idade é: ");
```

```
document.write("33"+"33"); ou (33+33);
```

```
</script>
```

Após executarmos o código detectamos que os números com aspas concatenaram(se juntaram) e sem as aspas efetuaram a soma dos valores. Todo o item escrito sobre “” aspas é tratado como string(texto). E os números escritos sem aspas são reconhecidos como valores que podem ser calculados.

# Tipos de Dados

Podemos trabalhar com qualquer função matemática: adição, subtração, multiplicação e divisão.

```
<script>
```

```
document.write("A média das nossas idades é: ");
```

```
document.write(25 + 33 + 27 / 3);
```

```
</script>
```

Podemos escrever o código misturando strings e números

```
document.write("Minha idade é: " + 33 + " <br>");
```

```
document.write("A média das nossas idades é: " + ((25 + 33 + 27) / 3) );
```

# Trabalhando Código e Números

Juntando todos os cálculos em único comando.

```
<script>  
document.write("Minha idade é: " + 33 + "<br>")  
document.write("A Soma das nossas idades é:" );  
document.write(25 + 33 + 27);  
document.write("<br>")  
document.write("A média das nossas idades é: " + ((25 + 32 + 26) / 3));  
</script>
```

# Trabalhando Código e Números

Juntando todos os cálculos em único comando utilizando Python.

```
print("Minha Idade é :", 34)
```

```
print("A soma das Nossas Idades é :", (25+ 33+ 27))
```

```
print("A média das Nossas Idades é: ", (25+ 33 + 37) /3)
```

# Exercício de Fixação

## Execute em Javascript

Crie 2 arquivos HTML sendo 1 do tipo que gera um Pop-UP e outro que escreva na tela.

O programa deve conter o valor de 3 Produtos faça a soma do valor desses produtos. E depois faça a média do valor desses produtos.

No programa que escreve a saída em tela não esqueça do comando de quebra de linhas.

## Execute em Python

- ▶ Execute um programa que teste se uma valor é INT ou Float.
- ▶ Crie um programa que faça a soma de 2 valores e jogue em tela o resultado.
- ▶ Crie um programa deve conter o valor de 3 Produtos faça a soma do valor desses produtos. E depois faça a média do valor desses produtos.



# Variáveis

- ❑ Variáveis são espaços alocados em memória
- ❑ Os nomes de variáveis não podem:
  - ❑ Conter espaços
  - ❑ Começar por número
  - ❑ Conter caracteres especiais, como +, -, \*, /, %, (, ), {, }, !, @, #
  - ❑ Utilizar nomes de palavras reservadas da linguagem, como *function*, *var*, *new*, *for* ou *return*
- ❑ CamelCase

Exemplo em Python:

```
>>> Cidade = "Floripa"  
>>> print(Cidade)
```

```
>>> num1 = 2  
>>> num2 = 3  
>>> num3 = 4  
>>> print(num1 + num2 + num3)
```

# Variáveis

Exemplo em Javascript:

Em Javascript antes do nome da variável temos que invocar a função  
**let**

```
>>> let Cidade = "Floripa"  
>>> alert(Cidade)
```

```
>>> let num1 = 2  
>>> let num2 = 3  
>>> let num3 = 4  
>>> alert(num1 + num2 + num3)  
9
```

# Exercício de Fixação

## Execute em Javascript e em Python

Crie um programa que passe 2 variáveis e faça a multiplicação dessas variáveis e imprima o resultado em tela. (Pode ser em POP-UP no caso do Javascript)

Rescreva o mesmo programa do exercício anterior onde os produtos serão declarados através de variáveis.

Segue o Algoritmo:

Crie um programa que deve conter o valor de 3 Produtos faça a soma do valor desses produtos. E depois faça a média do valor desses produtos.