

# ECOi2222 – Fundamentos de Lógica de Programação

Prof: Rafael Francisco dos Santos

E-mail: [rsantos@unifei.edu.br](mailto:rsantos@unifei.edu.br)



# Linguagem de Programação

## Python



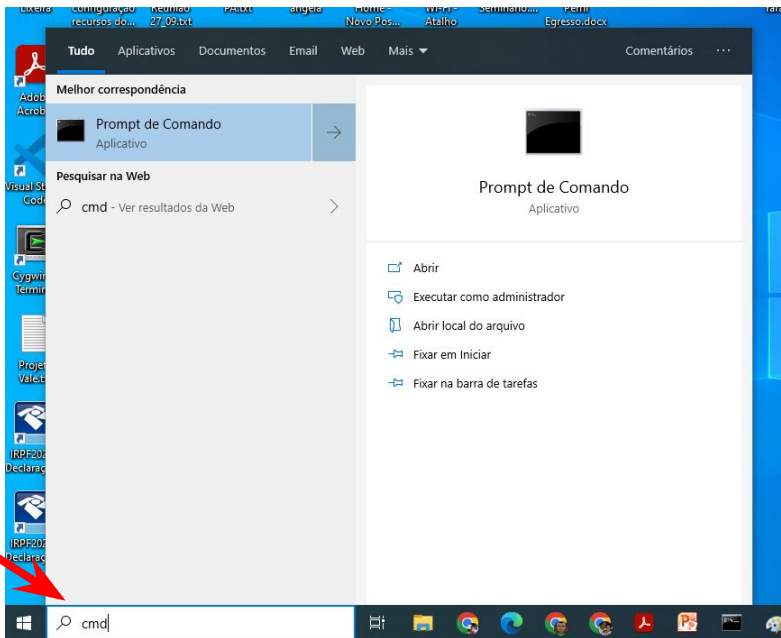
# Python

- Python foi criada em 1990 por Guido Van Rossum no Centro de Matemática Stichting na Holanda.
- Guido é lembrado como o principal autor de Python, mas outros programadores ajudaram com muitas contribuições..



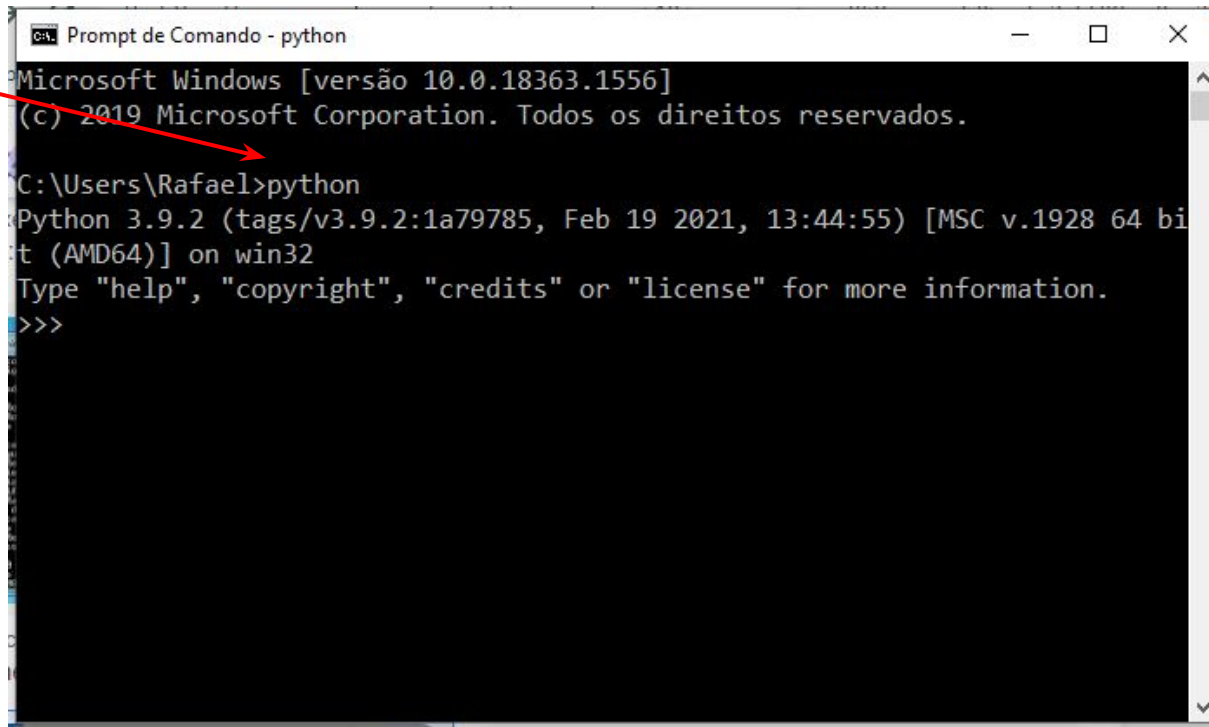
# Python

- **Modo interativo**
  - Digite `cmd` no menu pesquisa do windows



# Python

- **Modo interativo**
  - Digite `python` e `<ENTER>` para entrar



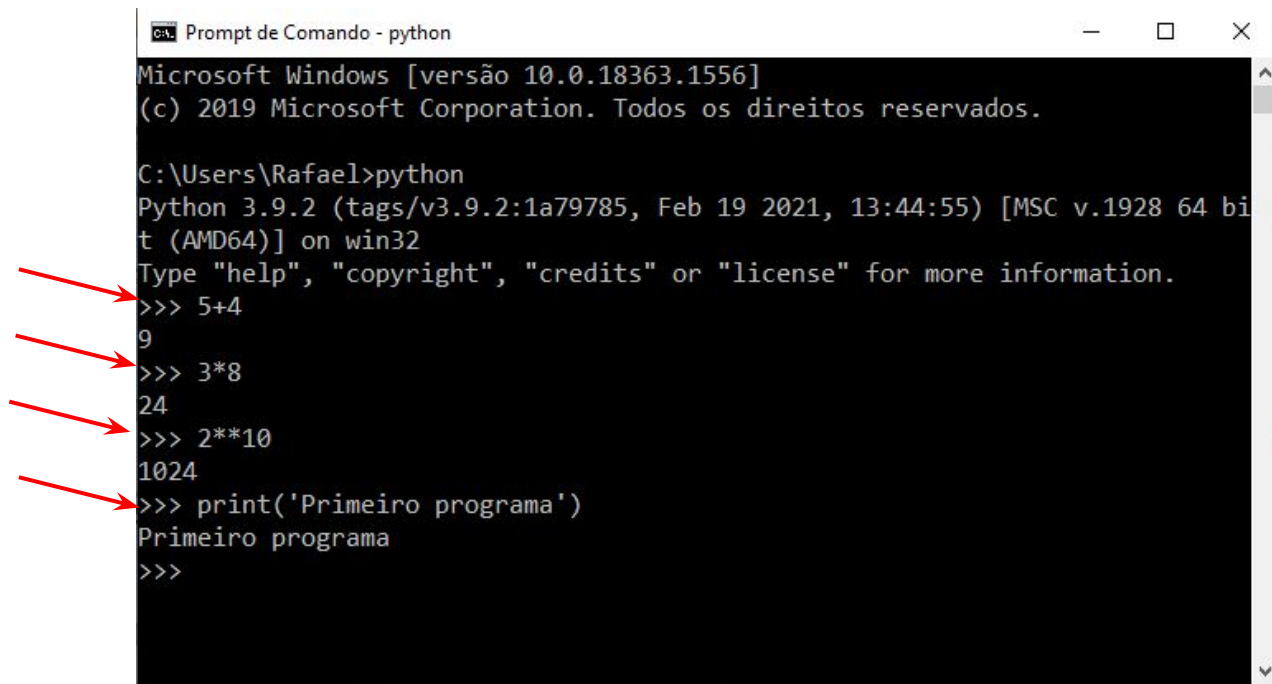
```
Microsoft Windows [versão 10.0.18363.1556]
(c) 2019 Microsoft Corporation. Todos os direitos reservados.

C:\Users\Rafael>python
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

# Python

- **Modo interativo**

- Agora basta digitar as instruções e apertar <ENTER> que elas serão executadas.
- Para sair digite <CTRL + Z> e <ENTER>



```
Prompt de Comando - python
Microsoft Windows [versão 10.0.18363.1556]
(c) 2019 Microsoft Corporation. Todos os direitos reservados.

C:\Users\Rafael>python
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bi
t (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> 5+4
9
>>> 3*8
24
>>> 2**10
1024
>>> print('Primeiro programa')
Primeiro programa
>>>
```



# Python

- **Modo script**

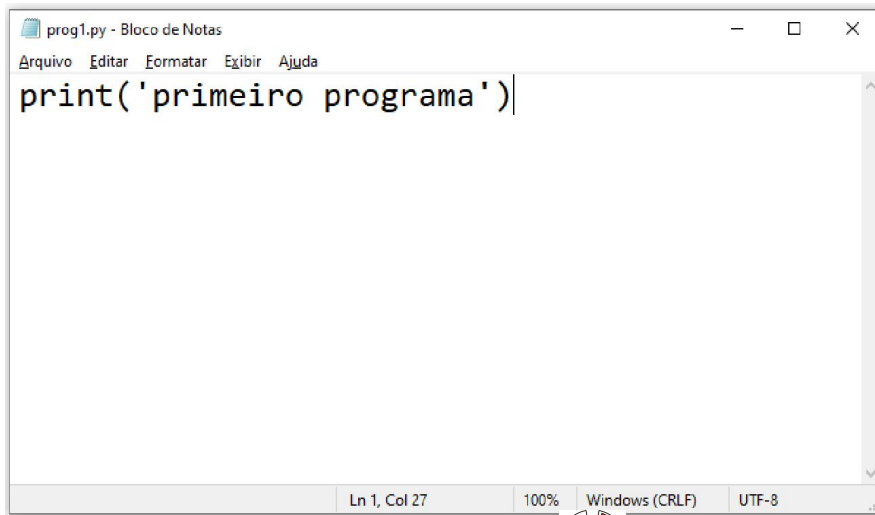
- O modo interativo é mais utilizado para testes, enquanto que o **modo script** é mais comumente utilizado na hora de desenvolver.
- No modo script, colocamos o código Python em um arquivo com extensão **.py**.



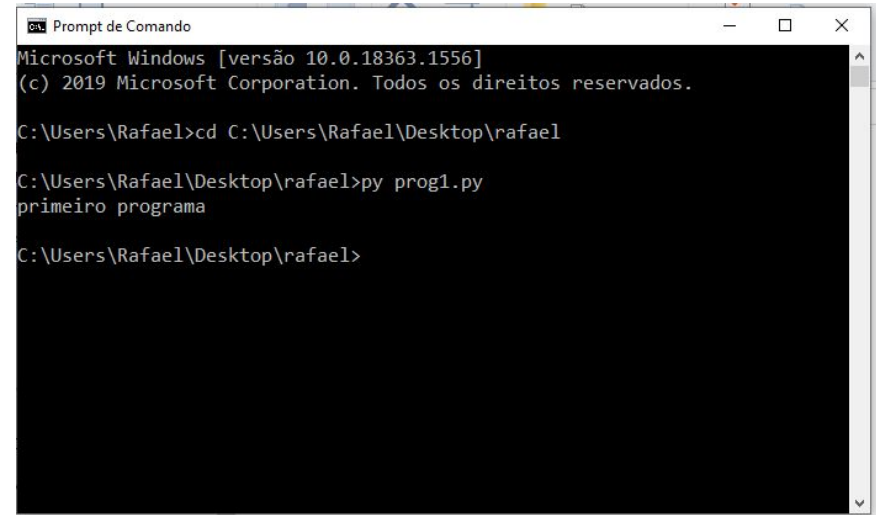
# Python

- **Modo script**

- Abra um editor de texto, exemplo “[notepad](#)”.
- Salve o arquivo como [prog1.py](#).
- Para executá-lo, abra o terminal, navegue até o diretório onde se encontra o arquivo [prog1.py](#) e digite: [py prog1.py](#)



```
print('primeiro programa')
```



```
Microsoft Windows [versão 10.0.18363.1556]
(c) 2019 Microsoft Corporation. Todos os direitos reservados.

C:\Users\Rafael>cd C:\Users\Rafael\Desktop\rafael

C:\Users\Rafael\Desktop\rafael>py prog1.py
primeiro programa

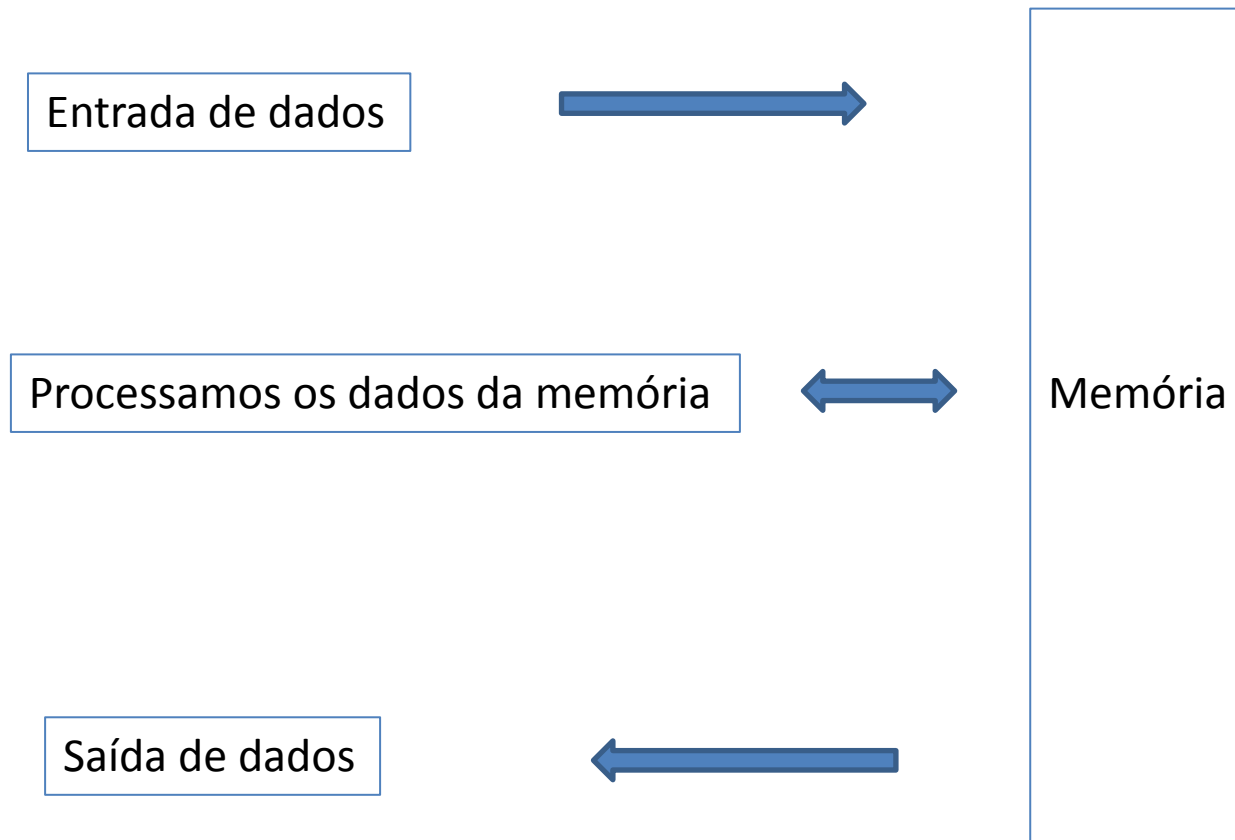
C:\Users\Rafael\Desktop\rafael>
```





# Python

## Estrutura básica de um programa



# Python

## Como utilizamos a memória?

Todos os dados utilizados no computador são armazenados em memórias.

Memória do Computador  
como uma planilha

	A	B	C	D
1				
2		5		
3				
4				
5			Casa	

Como guardar estes endereços?

Não é necessário. Utilizamos para identificar os endereços **variáveis** e **constantes**.

Logo, uma **variável** ou **constante** representa uma posição na memória.



# Python

## Variáveis e Constantes

- As variáveis e constantes só podem assumir um valor a cada instante.
- O conteúdo das variáveis podem mudar durante a execução do programa.
- O conteúdo da constante não podem ser modificados.
- Toda variável e constante possui um **nome** e um **tipo**.

## Os tipos de dados mais utilizados

### Numéricos

#### inteiros :

34

-15

#### reais:

23.45

-14.03

### Lógicos

verdadeiro (true)

falso (false)

### Literais

#### Cadeia de caracteres:

“aluno”

“0.34”

‘casa’

‘legal’



# Python

## Tipos de dados da biblioteca padrão (*built-ins*)

**Números inteiros ( int )**

Exemplos: 5, -5, 6

**Números reais ( float )**

Exemplos: 2.32 , -5.0

**Sequência de caracteres ( str )**

Exemplos: “casa” , ‘joão pedro’

**Lógico ( bool )**

Exemplos: true, false



# Linguagem de Programação

## Identificadores

- São utilizados para dar nomes a **variáveis, constantes, funções**, etc.

### Regras básicas para criar os identificadores:

- Podemos utilizar números, letras maiúsculas e minúsculas e o caractere sublinhado.
- O primeiro caractere deve ser uma letra ou sublinhado.
- Não são permitidos espaços em branco e caracteres especiais (#, @, %, +, ...)
- Não podemos utilizar palavras reservadas pela linguagem.

#### Identificadores válidos

Nota  
N1  
n2  
\_casa  
Nota\_1

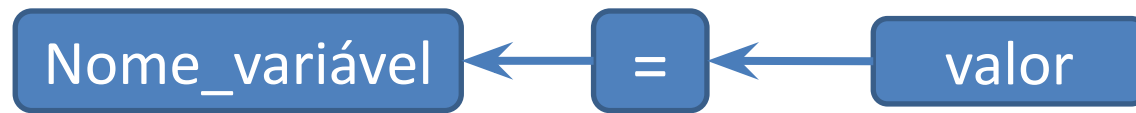
#### Identificadores inválidos

4a - começa com número  
N 1 - contém espaço  
n(2) - contém o caractere especial ( )  
if - palavra reservada  
from - palavra reservada



# Python

## Criando variáveis



## Exemplo

```
>>> x = 323
>>> nome='rafael santos'
>>> Numero=5
```

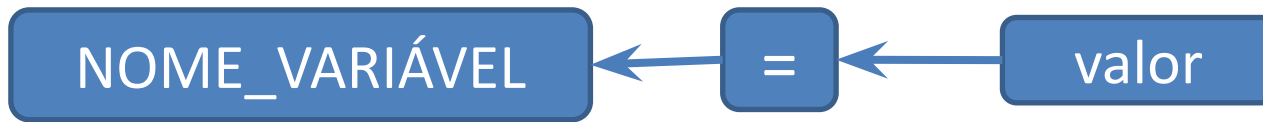
Numero recebe 5

# Python

## Criando constantes

Em Python não tem como criar constantes de forma nativa na linguagem.

A comunidade Python segue uma convenção para indicar que um identificador se refere a uma constante, usando letras maiúsculas e sublinhados para separar palavras.



### Exemplo

```
>>> PI = 3.1415
```

```
>>> MAX_VALUE = 10
```



# Python

## Verificando tipo

Para verificar o tipo da variável utilizasse a função

`type (variável)`

### Exemplo

```
>>> pi = 3.1415
>>> type(pi)
<class 'float'>

>>> nome='rafael santos'
>>> type(nome)
<class 'str'>

>>> n1=5
>>> type(n1)
<class 'int'>
```





# Python

## Comando de entrada de dados

É utilizado para armazenar os dados digitados pelo usuário em variáveis na memória.

### Comando input

#### Sintaxe

Variável = input()

Variável = input(str)

### Exemplo

```
>>> x=input()
```

```
5
```

```
>>> nome=input('Digite seu nome: ')
```

```
Digite seu nome: Rafael
```

```
>>> n1=input('Digite um número: ')
```

```
Digite um número: 45
```



# Python

## Comando de entrada de dados

A função input ao ler o dado sempre o armazena em uma “str” (cadeia de caracteres).

### Exemplo

```
>>> x=input()  
15  
>>> x  
'15'  
>>> type(x)  
<class 'str'>
```



# Python

## Convertendo string para outros tipos

Utilizamos as funções

`int()` `float()`

### Exemplo

```
>>> x=input()
```

```
5
```

```
>>> x
```

```
'5'
```

```
>>> int(x)
```

```
5
```

```
>>> x=int(x)
```

```
>>> x
```

```
5
```

```
>>> n=input('Digite um número: ')
```

```
Digite um número: 3.45
```

```
>>> n
```

```
'3.45'
```

```
>>> type(n)
```

```
<class 'str'>
```

```
>>> n=float(n)
```

```
>>> n
```

```
3.45
```

```
>>> type(n)
```

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



# Python

## Comando de entrada de dados

### Entrada de dados + converter para o tipo

`float(input(str))` ou `int(input(str))`

### Exemplo

```
>>> n=int(input('Digite um número: '))
Digite um número: 45
>>> n
45
>>> type(n)
<class 'int'>
>>> pi=float(input('Digite o valor de PI: '))
Digite o valor de pi: 3.1415
>>> type(pi)
<class 'float'>
```



# Python

## Comando de saída de dados

É utilizado para mostrar informações na tela  
(valores das variáveis e textos no formato de string (str)).

### Comando print

#### Sintaxe

```
print (str, variável, variável, ...)
```

#### Exemplo

```
print('Meu nome é Rafael')  
print ("Também funciona")
```

```
idade =5
```

```
print('Meu nome é Rafael e tenho ', idade, ' anos')
```

#### Saída na tela

```
Meu nome é Rafael  
Também funciona  
Meu nome é Rafael e tenho 5 anos
```



# Python

## Comando de saída de dados

### Comando print

#### Sintaxe

```
print (str, variável, variável, ...)
```

#### Exemplo Concatenado

```
idade =5
```

```
print('Meu nome é Rafael e tenho '+ str(idade)+ ' anos')
```

#### Saída na tela

Meu nome é Rafael e tenho 5 anos



# Python

## Comando de saída de dados

### Comando print

#### Sintaxe

```
print (f 'um texto com variáveis entre chaves')
```

O **f** antes da str (cadeia de caracteres) é para informar ao python para substituir as marcações **{nome\_variável}** pelo valor da variável na memória.

#### Exemplo

```
idade =5
```

```
nome='Rafael'
```

```
print(f 'Meu nome é {nome} e tenho {idade} anos')
```

#### Saída na tela

Meu nome é **Rafael** e tenho **5** anos



# Python

## Comentários

São textos que podem ser inseridos nos programas com o objetivo de documentá-lo.

### **Comentário de uma única linha**

# comentário

### **Comentário de várias linhas**

'''

linhas de comentário

linhas de comentário

'''





# Python

## Exemplo de entrada e saída

**Problema:** Implemente um programa que leia o nome e a idade de uma pessoa e apresente na tela a mensagem: “Bem vindo **nome** você tem **idade** anos”.

### Algoritmo

- 1 – Solicitar o nome e a idade de uma pessoa
- 2 – Ler o nome e a idade
- 3 – Apresentar a mensagem informando o nome e a idade.

```
nome = input('Digite o seu nome: ')
idade = int(input('Digite a sua idade: '))

print(f'Bem vindo {nome} você tem {idade} anos')
```



# Python

## Exemplo de entrada e saída

**Problema:** Implemente um programa que leia o nome e a idade de uma pessoa e apresente na tela a mensagem: “Bem vindo **nome** você tem **idade** anos”.

```
nome = input('Digite o seu nome: ')
idade = int( input('Digite a sua idade: ') )

print(f 'Bem vindo {nome} você tem {idade} anos')
```

memória

	idade		
	23		
		nome	
		Maria	

console

Digite o seu nome: Maria

Digite a sua idade : 23

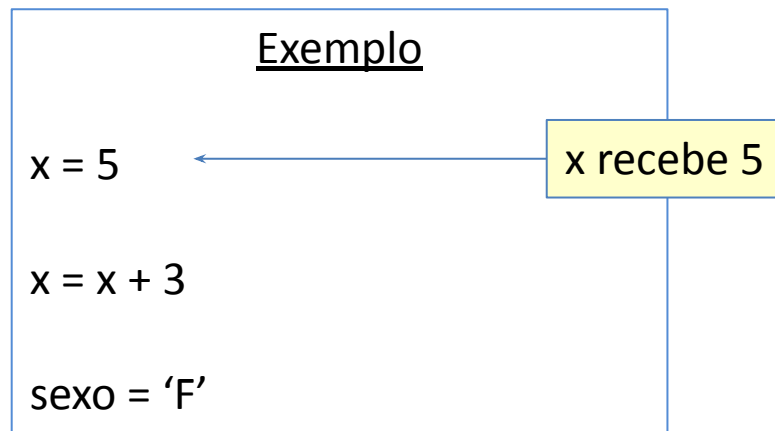
Bem vindo Maria você tem 23 anos



# Python

## Operador de atribuição

Atribui valores a variáveis. É representado pelo sinal de igualdade =



# Python

## Operadores Aritméticos

Aritméticos	
+	Adição
-	Subtração
*	Multiplicação
/	Divisão
//	Divisão inteira
%	Módulo (Resto da divisão)
**	Exponenciação

### Exemplos

```
x = x + 1
x = 5 * x - 4
x = (3 * x) / 5
x = 3**2
x = y**0.5
```

### Divisão de inteiros

```
x = 5/2
y = 5//2
```

Valor de x é 2.5

Valor de y é 2



# Python

## Operadores Aritméticos

Binários	
+	Adição
-	Subtração
*	Multiplicação
/	Divisão
//	Divisão inteira
%	Resto da divisão inteira

### Resto da divisão inteira

$x = 17/5$   
 $y = 17\%5$

$$\begin{array}{r} \text{— } 17 \\ \text{— } 15 \\ \hline 2 \end{array} \quad \begin{array}{r} 5 \\ \hline 3 \end{array}$$

Resto da divisão inteira

Divisão de inteiros



# Python

## Precedência entre os operadores aritméticos

Prioridade	Operador(es)
1ª	- (Menos unário), parênteses mais internos
2ª	*, /, %
3ª	+, -

Os operadores que possuem a mesma prioridade devem ser executados da esquerda para a direita. Para alterar a prioridade da tabela devem-se utilizar parênteses mais internos.

### Exemplo:

$$x = 3 + x/4*y \longrightarrow x = 3 + \frac{x}{4} * y$$

$$x = (3 + x)/4*y \longrightarrow x = \frac{3+x}{4} * y$$

$$x = (3 + x)/(4*y) \longrightarrow x = \frac{3+x}{4*y}$$



# Python

## Operadores de atribuição

São operadores que combinam as operações aritméticas com atribuição

Operador Aritmético de Atribuição	Operadores equivalentes
<code>i += 2</code>	<code>i = i + 2</code>
<code>x *= y + 1</code>	<code>x = x * (y+1)</code>
<code>t /= 2.5</code>	<code>t = t / 2.5</code>
<code>p %= 5</code>	<code>p = p % 5</code>
<code>d -= 3</code>	<code>d = d - 3</code>

# Python - Exemplo

**Problema:** Implemente um programa que leia 3 notas de um aluno e apresente a média aritmética calculada.

## Algoritmo

- 1 – Solicitar as 3 notas
- 2 – Ler as 3 notas
- 3 – Calcular a média
- 4 – Apresentar a mensagem informando a média calculada.

```
print('Digite 3 notas:')
n1=float(input('Digite a nota 1: '))
n2=float(input('Digite a nota 2: '))
n3=float(input('Digite a nota 3: '))

media = ( n1 + n2 + n3)/3

print ('A media das notas e: ', media)
```





# Python - Exemplo

**Problema:** Implemente um programa que leia 3 notas de um aluno e apresente a média aritmética calculada.

```
print('Digite 3 notas:')
n1=float(input('Digite a nota 1: '))
n2=float(input('Digite a nota 2: '))
n3=float(input('Digite a nota 3: '))

media = ( n1 + n2 + n3)/3

print ('A media das notas e: ', media)
```

memória

	n1 6		
		n3 7	
n2 8			
		media 7	

console

```
Digite as 3 notas
6 8 7
A media das notas e: 7
```



# Python - Exemplo

**Problema:** Implemente um programa que leia 2 catetos de um triângulo retângulo e calcule a sua hipotenusa.

## Algoritmo

- 1 – Solicitar os 2 catetos
- 2 – Ler os 2 catetos
- 3 – Calcular a hipotenusa
- 4 – Apresentar a mensagem informando o valor da hipotenusa.

```
c1=float(input('Digite o primeiro cateto: '))
c2=float(input('Digite o segundo cateto: '))

aux = (c1**2) + (c2**2)
hipotenusa = aux**0.5

print(f'O valor da hipotenusa e {hipotenusa}')
```



# Python - Exemplo

**Problema:** Implemente um programa que leia 2 catetos de um triângulo retângulo e calcule a sua hipotenusa.

```
c1=float(input('Digite o primeiro cateto: '))
c2=float(input('Digite o segundo cateto: '))

aux = (c1**2) + (c2**2)
hipotenusa = aux**0.5

print(f'O valor da hipotenusa e {hipotenusa}')
```

memória

c1 9	aux 225		
hipotenusa 15		c2 12	

console

```
Digite os 2 catetos
9 12
O valor da hipotenusa e: 15
```



# Python - Exercícios

**Problema 1:** Uma empresa decide dar um aumento de 30% aos funcionários. Faça um programa em python que receba o salário do funcionário e mostre o valor do salário reajustado.

**Problema 2:** Sabe-se que para iluminar de maneira correta os cômodos de uma casa, para cada  $m^2$ , deve se usar 18 W de potência. Faça um programa que receba as duas dimensões de um cômodo (em metros), calcule e mostre a sua área (em  $m^2$ ) e a potência de iluminação que deverá ser utilizada.



# Python- Exercícios

**Problema:** Implemente um programa que receba um inteiro N de quatro dígitos e apresente os 4 dígitos separados, cada um em uma variável.  
Por exemplo, se N=2354, as variáveis d1=2, d2=3, d3=5 e d4=4.

## Como resolver?

Divisão de inteiros

$$y = 2548 // 1000$$

Resto da divisão inteira

$$y = 2548 \% 1000$$

$$\begin{array}{r} 2548 \\ - 2000 \\ \hline 548 \end{array} \quad \begin{array}{r} 1000 \\ \hline 2 \end{array}$$

Resto da divisão inteira

Divisão de inteiros



# Python Exercícios

**Problema:** Implemente um programa que receba um inteiro N de quatro dígitos e apresente os 4 dígitos separados, cada um em uma variável.

Por exemplo, se  $N=2354$ , as variáveis  $d1=2, d2=3, d3=5$  e  $d4=4$ .

**Problema:** Implemente um programa que receba um inteiro representando segundos e imprima a conversão deste tempo em horas minutos e segundos.



# Python - Exercícios

**Problema:** Faça um programa que receba 2 pontos A e B no plano cartesiano e calcule a distância entre eles. Abaixo a fórmula da distância.

Distância entre dois pontos A e B:  $d_{AB} = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$

