



# Programação de Computadores

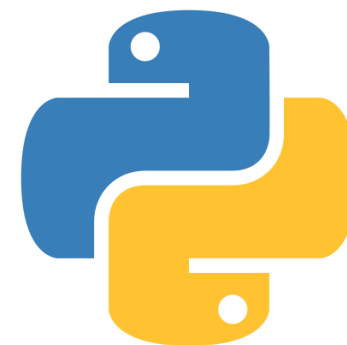
- ✓ Introdução ao Python
- ✓ Atribuição e entrada e saída

# Introdução ao Python

---

## O que é Python?

“ **python** é uma linguagem de programação de propósito geral, de alto nível, interpretada, multiparadigma, de tipagem dinâmica e forte. ”



# Introdução ao Python



- ✓ Foi concebida no final da década de 1980 pelo programador **Guido van Rossum**;
- ✓ Desenvolvida no CWI (Centrum Wiskunde & Informatica), em Amsterdã, Holanda;
- ✓ Projeto de desenvolvimento de software de código-fonte aberto, quem decide como a linguagem irá evoluir é o Guido;
- ✓ Projetada para tornar os programas bastante legíveis (código simples).



\*\* [https://pt.wikipedia.org/wiki/Guido\\_van\\_Rossum](https://pt.wikipedia.org/wiki/Guido_van_Rossum)

# Introdução ao Python

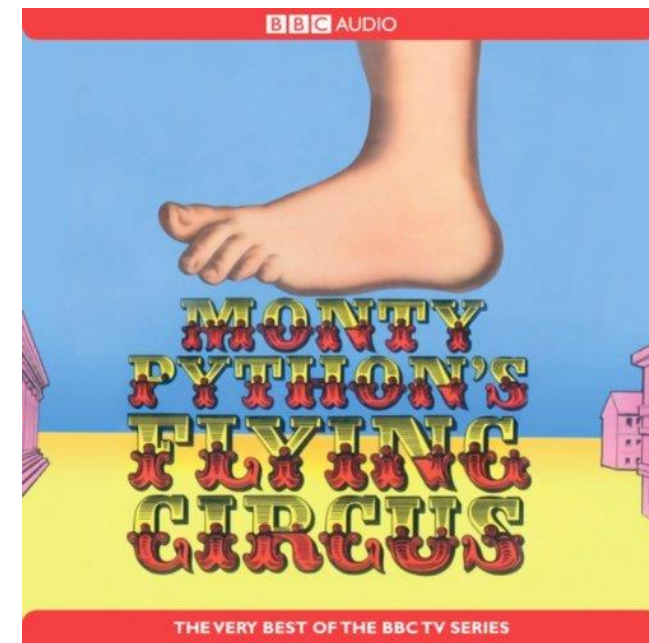


- ✓ Rica biblioteca => criar aplicações sofisticadas.
- ✓ Tornou-se uma linguagem de desenvolvimento de aplicações popular.
- ✓ Preferência como “primeira” linguagem de programação.



O nome *Python* teve a sua origem no grupo humorístico britânico Monty Python, embora muitas pessoas façam associação com o réptil do mesmo nome\*.

\* <https://pt.wikipedia.org/wiki/Python>



# Introdução ao Python

## Por que Python?

- ✓ Simplicidade e clareza.
- ✓ Linguagem poderosa, pode ser usada para administrar sistemas e desenvolver grandes projetos.
- ✓ Linguagem clara e objetiva.



# Introdução ao Python



Web application



Software development



Machine Learning



Data analysis



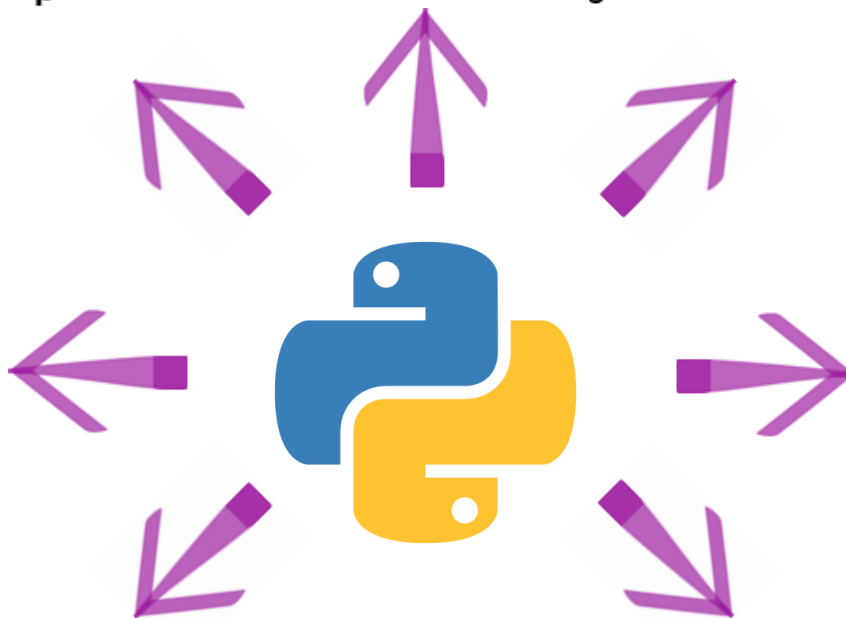
Automation



System Scripting



Mathematics



Web development

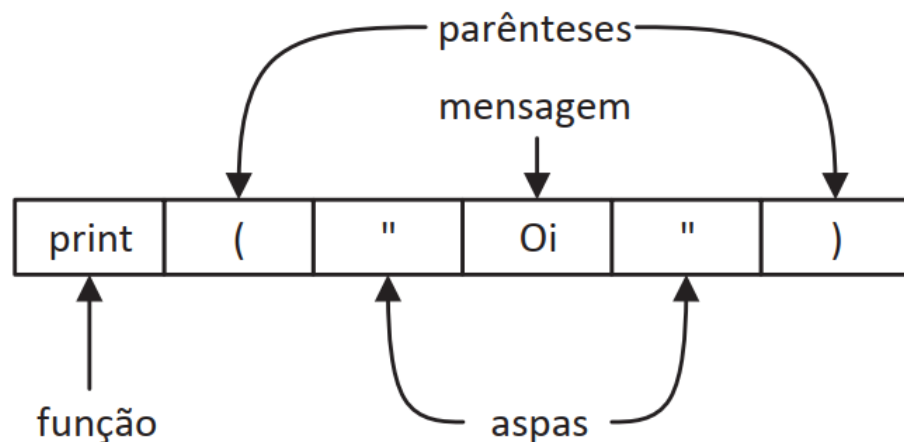


Artificial Intelligence

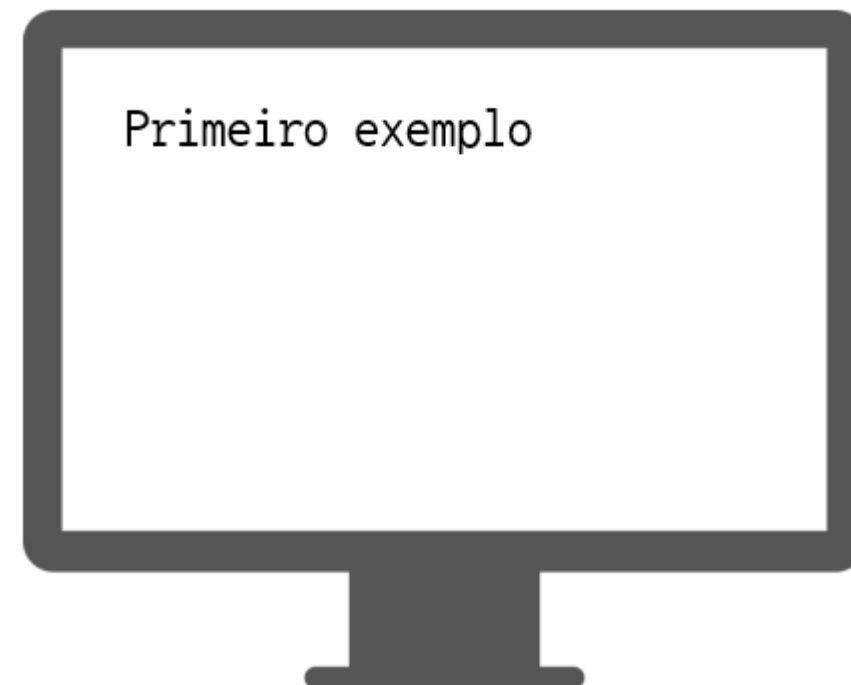
Para que Python?

# Comandos de saída de dados em Python

Função `print()`: é utilizada para exibir mensagens na tela ou enviá-las para outro dispositivo, como imprimir dentro de arquivos de texto.



```
main.py
1 print('Primeiro exemplo')
2
```



# Variáveis

---

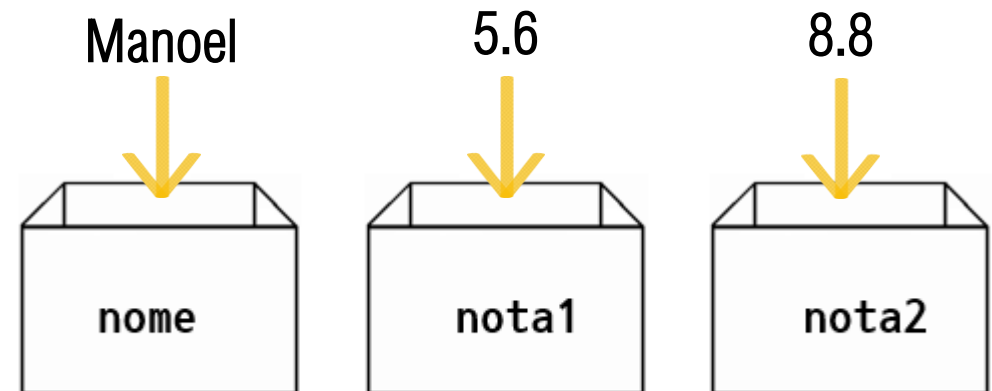
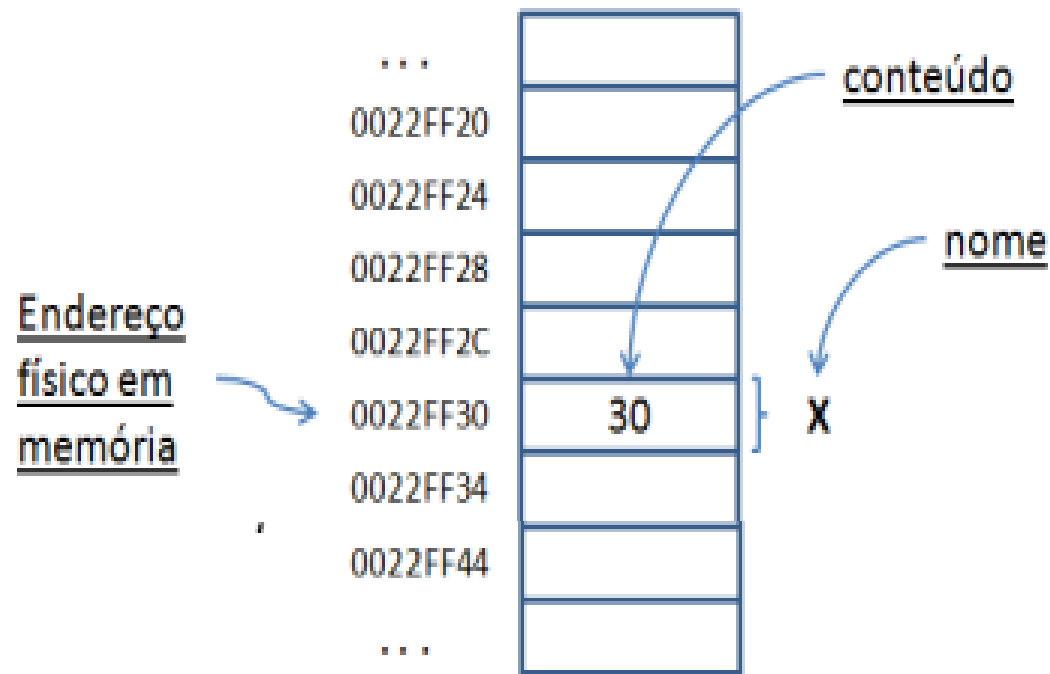
- ✓ Quando desejamos guardar temporariamente uma informação no computador, seja uma frase, um nome, dados numéricos ou até mesmo o resultado de um cálculo, fazemos isto na memória.
- ✓ Mas para isto precisamos identificar em que parte dela estamos guardando para depois poder recuperar a informação.
- ✓ Quando fazemos esta identificação, damos um nome para esta parte, isto é criamos uma variável.





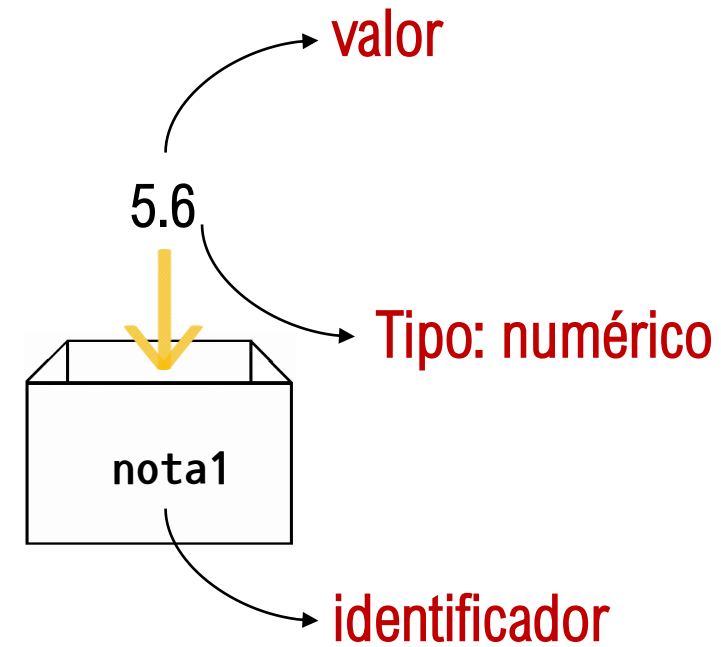
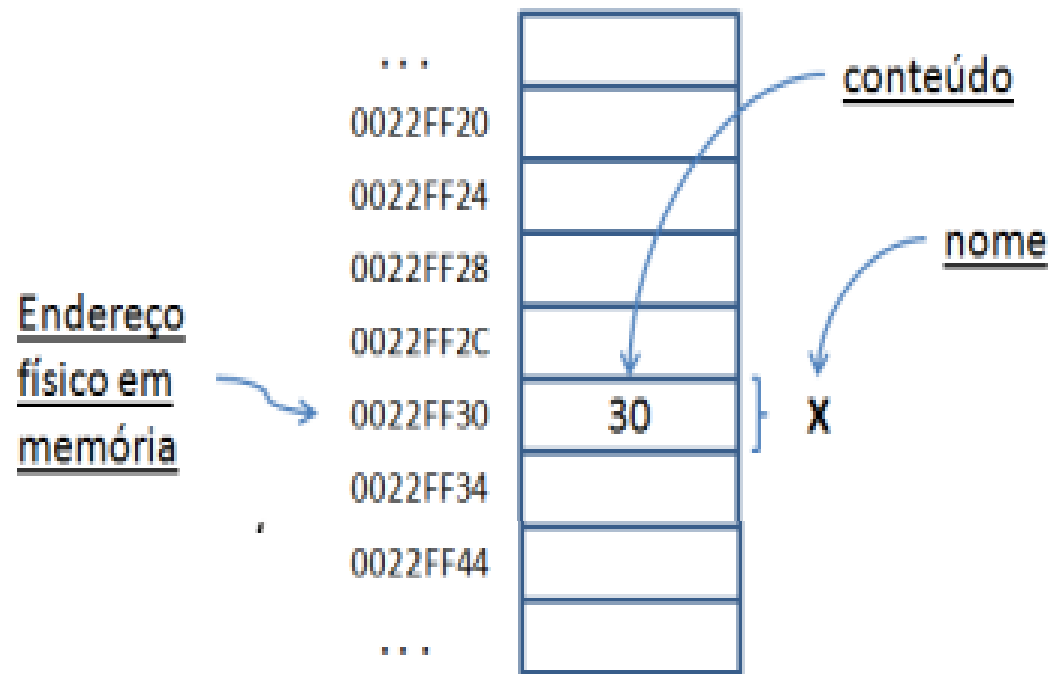
# Variáveis

- ✔ Portanto, podemos definir variável como o espaço de memória que pode receber um valor e sofrer alteração no decorrer do algoritmo/tempo.
- ✔ Toda variável tem um nome único que a identifica (identificador), um valor e o tipo correspondente à informação a ela atribuída.



# Variáveis

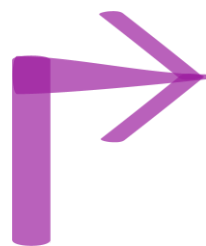
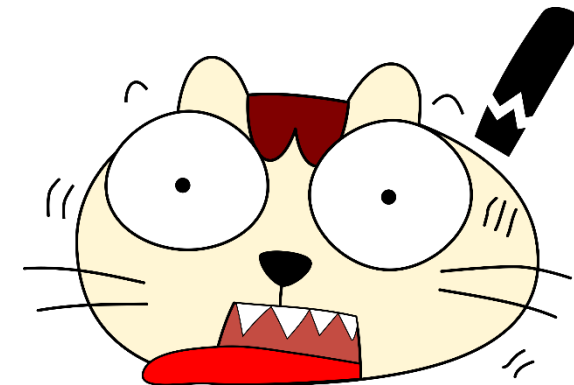
- ✔ Portanto, podemos definir variável como o espaço de memória que pode receber um valor e sofrer alteração no decorrer do algoritmo/tempo.
- ✔ Toda variável tem um nome único que a identifica (identificador), um valor e o tipo correspondente à informação a ela atribuída.



# Identificador (nome)

Regras para os nomes das variáveis:

- ✓ Devem começar com letra minúscula ou maiúscula;
- ✓ Podem conter o caractere sublinhado (\_);
- ✓ Não podem começar com números;
- ✓ Não podem conter pontuação nem espaços;
- ✓ Não podem ser utilizadas palavras reservadas da linguagem.



**33 Palavras reservadas  
da linguagem.**

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	

# Como imprimir variáveis em Python

Há várias formas para se imprimir variáveis. Nos exemplos seguintes, vamos adotar duas variáveis: **nome** e **nota**

```
>>> nome = 'Joao'  
>>> nota = 10
```

- ✔ A forma mais simples é passar as variáveis como parâmetros:

```
>>> print('A nota de', nome, 'foi', nota)  
A nota de Joao foi 10
```



# Como imprimir variáveis em Python

Há várias formas para se imprimir variáveis. Nos exemplos seguintes, vamos adotar duas variáveis: **nome** e **nota**

```
>>> nome = 'Joao'
>>> nota = 10
```

- ✓ A segunda forma é passando uma estrutura de dados como parâmetro. Observe que a marcação indicada por %s será substituída pelos valores das variáveis.

```
>>> print('A nota de %s foi %s' % (nome, nota))
A nota de Joao foi 10
```

**%s** indica que a saída é uma String!

Estrutura de dados  
composta por **nome** e **nota**



# Como imprimir variáveis em Python

Há várias formas para se imprimir variáveis. Nos exemplos seguintes, vamos adotar duas variáveis: **nome** e **nota**

```
>>> nome = 'Joao'  
>>> nota = 10
```

- ✓ Podemos usar também o **f-strings**. Neste modo, utilizaremos **f**, antes da string a ser impressa e colocaremos as variáveis a serem impressas entre chaves (Exemplo: **{nome}**)

```
>>> print(f'A nota de {nome} foi {nota}')
```

A nota de Joao foi 10



**Observação:** O modo f-strings só está disponível a partir da versão 3.6

# Como imprimir variáveis em Python

Há várias formas para se imprimir variáveis. Nos exemplos seguintes, vamos adotar duas variáveis: **nome** e **nota**

```
>>> nome = 'Joao'
>>> nota = 10
```

- ✔ O método **String format ()** permite inserir valores dentro de espaços reservados definidos entre chaves (Exemplo: **{nome}**)

```
>>> txt = 'Tudo por apenas {preco:.2f} reais'
>>> print(txt.format(preco=1.99))
Tudo por apenas 1.99 reais
```



# Definição de Variáveis em Python

## Exemplo: Definição de variável

1- Vamos fazer um programa em Python que calcule a média de um aluno, a partir de duas notas armazenadas em variáveis, e imprima o nome do aluno, a disciplina, as notas e, finalmente a média desse aluno.

```
>_ Console x Shell x +  
Nome do aluno Marcelo  
Disciplina: Programação de computadores  
Primeira nota: 6.45  
Segunda nota: 8.78  
Média: 7.62  
✦
```





# Definição de Variáveis em Python

```
main.py
1  #Exemplo 1 - Definição de variáveis
2  nome = 'Marcelo'
3  disciplina = 'Programação de computadores'
4  nota1 = 6.45
5  nota2 = 8.78
6  media = (nota1 + nota2)/2
7  print('Nome do aluno %s ' %(nome))
8  print(f'Disciplina: {disciplina}')
9  print('Primeira nota: %.2f \nSegunda nota: %.2f' %(nota1,nota2))
10 print(f'Média: {media:.2f}')
```

Os comentários são ignorados durante a execução!!



→

```
>_ Console x  Shell x  +
Nome do aluno Marcelo
Disciplina: Programação de computadores
Primeira nota: 6.45
Segunda nota: 8.78
Média: 7.62
>
```

# Entrada de dados em Python

Função `input()`: Recebe o valor que o usuário fornece através do teclado e armazena em uma variável.

```
variável = input('Mensagem')
```



se  
Liga!  
Ai

A mensagem impressa na tela permite que o usuário saiba que o programa está esperando a entrada de um valor!!

# Entrada de dados em Python

Função `input()`: Recebe o valor que o usuário fornece através do teclado e armazena em uma variável.

```
variável = input('Mensagem')
```



se  
Liga  
Aí

O valor fornecido pelo usuário é sempre um texto, nunca um número.

# Exemplo de aplicação

2- Vamos criar um programa que pergunte o nome e a idade de uma pessoa e depois apresente estas informações na tela. O algoritmo em pseudocódigo ficará assim:

```
algoritmo exemplo2
  inicio
    literal nome
    inteiro idade
    escreva ("Digite seu nome")
    leia (nome)
    escreva ("Digite sua idade")
    leia (idade)
    escreva (nome, " você tem ", idade, " anos")
  fim
```



# Exemplo de aplicação

```
main.py x +
main.py
1 #Exemplo 2 - entrada de dados
2 nome = input('Digite o seu nome: ')
3 idade = input('Digite a sua idade: ')
4
5 print(nome, 'tem', idade, 'anos de idade')
```



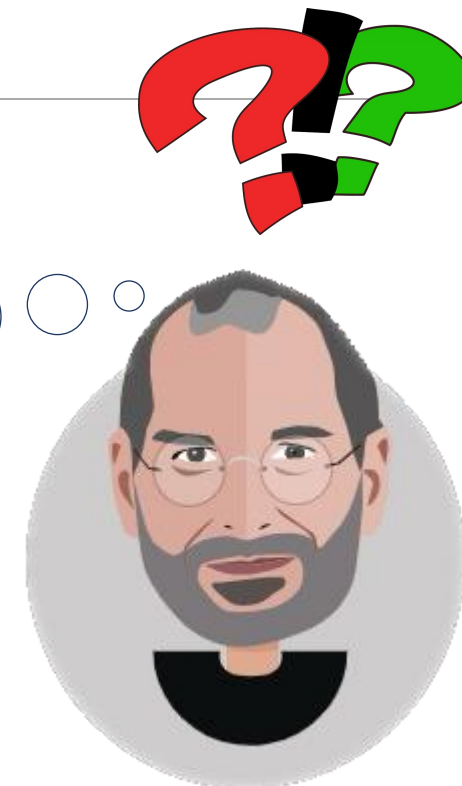
```
>_ Console x Shell x +
Digite o seu nome: Alexandre
Digite a sua idade: 18
Alexandre tem 18 anos de idade
>
```



# Entrada de dados em Python



Mas, e se fosse necessário realizar uma operação aritmética com uma variável???



A entrada de dados por meio da função `input()` será sempre uma **String**. Caso o tipo desejado seja número, você deverá fazer a coerção de dados.

# Tipos de dados

Classificação	Tipos	Descrição
numérico	<code>int</code>	Números inteiros. <b>Ex:</b> idade, ano, quantidade de filhos
	<code>float</code>	Números de ponto flutuante (reais). <b>Ex:</b> salário, peso, altura
texto	<code>string</code>	São sequências (cadeias) de caracteres. <b>Ex:</b> nome
lógico	<code>boolean</code>	forma de se armazenar um valor lógico, ou seja, verdadeiro ou falso. <b>Ex:</b> formado, solteiro



# Conversões de tipos em Python

- ✓ De string para um número inteiro:

```
v1 = input('Digite um número inteiro: ')\nv1 = int(v1)
```

v1 é algo como "12" ou "-7", ou ocorrerá um erro.

- ✓ De string para um número com casas decimais:

```
v2 = input('Digite um número real: ')\nv2 = float(v2)
```

v2 algo como "10.5" ou "-5.5", ou ocorrerá um erro.





# Conversões de tipos em Python

- ✓ Avalia uma String como uma expressão:

```
v3 = input('Digite uma expressão matemática ')\nv3 = eval(v3)
```

v3 é algo como “2+3”, ou ocorrerá um erro

- ✓ Podemos simplificar, compondo qualquer função de conversão com o input():

```
a = int(input('Digite o primeiro número: '))\nb = int(input('Digite o segundo número: '))\nprint(a + b)
```



# Operadores aritméticos e prioridade (Python)

Operador	Operação	Prioridade	Exemplo
+	Soma	4º	$2 + 3 = 5$
-	Subtração	4º	$3 - 2 = 1$
*	Multiplicação	3º	$2 * 3 = 6$
/	Divisão	3º	$5 / 4 = 1.25$
//	Parte Inteira	3º	$5 // 4 = 1$
%	Resto de uma divisão inteira	2º	$10 \% 3 = 1$
**	Exponenciação	2º	$2^{**}3 = 8$
+	Manutenção de sinal	1º	$+(-2) = -2$
-	Inversão de sinal	1º	$-(-2) = 2$

## Observações:

- ✓ Em uma expressão com operadores da mesma prioridade, as operações serão executadas de esquerda a direita.
- ✓ Nas linguagens com operadores para potência, eles teriam prioridade maior que + - / \*



# Exemplo de aplicação

3- Faça uma calculadora em Python que solicite ao usuário dois valores e, em seguida imprima as operações matemáticas de soma, subtração, multiplicação, divisão e o resto da divisão desses números.

exemplo3.py

```
1 # Exemplo 3 - Calculadora
2 a = int(input('Digite o 1º valor: '))
3 b = int(input('Digite o 2º valor: '))
4 print('A soma é ', (a+b))
5 print('A subtração é ', (a-b))
6 print('A multiplicação é ', (a*b))
7 print('A divisão é ', (a/b))
8 print('O resto da divisão é ', (a%b))
9 # print(f'{a} + {b} = {a+b}')
```



# Exemplo de aplicação

4- Faça um programa em Python que solicite ao usuário dois valores e, em seguida, calcule a média desses valores e imprima na tela.

exemplo4.py

```
1 # Exemplo 4 - Cálculo da média de dois valores
2 # com um programa na linguagem de
3 # programação Python
4
5 a = float(input('Digite o 1º valor: '))
6 b = float(input('Digite o 2º valor: '))
7 media = (a + b) / 2
8 print(f'A média é {media:.2f}')
```



# Alguma dúvida????

---





# Exercícios de aplicação

---



# Mas, antes...

---

## Vamos corrigir os exercícios da última aula?



# Correção dos exercícios da aula 01

---

- 1- Faça um programa em Python que escreva a mensagem “Ciências da Computação - Unicsul” na tela.
- 2- Faça um programa em Python que solicite ao usuário a sua profissão e mostre na tela a informação digitada.
- 3- Faça um programa em Python que solicite ao usuário a sua idade e apresente a informação na tela.





# Correção dos exercícios da aula 01

---

- 4- Faça um programa em Python que solicite o último sobrenome do usuário e mostre na tela a mensagem: “Família” e na sequência o sobrenome digitado.
- 5- Faça um programa em Python que solicite o esporte favorito de uma pessoa e apresente-o na tela.



# Então, agora é pra valer... Exercícios de aplicação

---



# Observações sobre exercícios

- ✓ Todos os exercícios devem ser resolvidos em Python.
- ✓ O código Python pode ser feito no IDLE, no Repl.it, ou na ferramenta que você ache mais adequada e deve ser salvo um arquivo por exercício com a extensão .py
- ✓ Após finalizar todos os exercícios da aula, coloque-os em uma pasta com o nome da aula, compacte a pasta e envie no Blackboard.



<https://youtu.be/BuxuUbgKwCg>





# Exercícios de aplicação

- 1- Desenvolva um programa em Python que solicite ao usuário os valores dos lados de um retângulo e calcule e mostre seu perímetro e sua área.
- 2- Escreva um programa em Python que solicite ao usuário o salário atual e mostre o salário acrescido de 5% de comissão.
- 3- Escreva um programa em Python que solicite ao usuário a distância entre duas cidades e o tempo de viagem. O programa deverá calcular e exibir a velocidade média de um carro que vai de uma cidade para outra. Utilize a fórmula:

$$v_m = \frac{\text{distancia}}{\text{tempo}}$$



# Exercícios de aplicação

4- Escreva um programa em Python que calcule as duas raízes de uma equação de 2º grau  $ax^2+bx+c$ , conhecendo os valores dos coeficientes da mesma (a, b, c). Suponha que as raízes são reais. Lembre-se que para calcular as duas raízes:

$$x_{1,2} = \frac{-b \pm \sqrt{\Delta}}{2a} \quad \text{com} \quad \Delta = b^2 - 4.a.c$$

5- Escreva um programa em Python que leia a cotação do dólar (taxa de conversão), leia um valor em dólares e converta e mostre o valor equivalente em Reais.



# Exercícios de aplicação

6- Escreva um programa em Python que leia um valor representado por um cliente do restaurante ComaBem e visualize o valor total, considerando os 10% do garçom.

7- Escreva um programa em Python que obtenha uma temperatura em graus Celsius, calcule e mostre a respectiva temperatura nas escalas Fahrenheit e Kelvin. Use as fórmulas abaixo:

$$t_F = 1,8 * t_C + 32$$

$$T_K = t_C + 273$$



# Créditos

Esta aula foi elaborada com base no material produzido e cedido gentilmente pelos **Professores Alcides, Lédon, Amilton e Cristiane.**







*That's all Folks!*