



# TÉCNICAS DE PROGRAMAÇÃO II

PROFº LUIZ CLÁUDIO



Variáveis e Entrada de Dados

Condições

Repetições



Lista

Funções

Arquivos



Gui(Interface Gráfica do Usuário)

Integração com Banco de Dados



CONTEÚDO PROGRAMÁTICO

# FORMAS DE AVALIAÇÃO



25%

Avaliação  
Integradora

+



Atividades  
15%

**N1**

Atividades  
15%

**N2**

+



25%

Projeto Interdisciplinar

PO



Portfólio OnLine 10%

+

+



10%

MRT – Maratona de Prog.

**MÉDIA FINAL** – MEDIAPARCIAL + QTE

1,3 pontos na Média Final (MF)

**EXAME** – somente aos alunos que obtiveram Média Final < 6

# INTRODUÇÃO A LINGUAGEM PYTHON

- Python é uma linguagem de altíssimo nível orientada a objeto, de tipagem dinâmica e forte, interpretada e interativa.
- Obs. **Tipagem dinâmica** é uma característica de determinadas linguagens de programação, que não exigem declarações de tipos de dados, pois são capazes de escolher que tipo utilizar dinamicamente para cada variável, podendo alterá-lo durante a compilação ou a execução do programa.
- **Tipagem forte** costuma ser a característica que não permite um mesmo dado ser tratado como se fosse de outro tipo.



# CARACTERÍSTICAS DA LINGUAGEM PYTHON

- A linguagem suporta programação modular e funcional, além da orientação a objetos.
- Mesmo os tipos básicos no Python são objetos.
- A linguagem é interpretada através de bytecode pela máquina virtual Python, tornando o código portátil.
- Com isso é possível compilar aplicações em uma plataforma e rodar em outros sistemas ou executar direto do código fonte.



# CARACTERÍSTICAS DA LINGUAGEM PYTHON

- Python é um software de código aberto (com licença compatível com a General Public License (GPL), porém menos restritiva, permitindo que o Python seja inclusive incorporado em produtos proprietários).
- A especificação da linguagem é mantida pela Python Software Foundation (PSF).



# CARACTERÍSTICAS DA LINGUAGEM PYTHON

- O Python também é muito utilizado como linguagem script em vários softwares, permitindo automatizar tarefas e adicionar novas funcionalidades, entre eles: BrOffice.org, PostgreSQL, Blender, GIMP e Inkscape.



# HISTÓRICO DA LINGUAGEM PYTHON

- A linguagem foi criada em 1990 por Guido van Rossum, no Instituto Nacional de Pesquisa para Matemática e Ciência da Computação da Holanda e tinha originalmente foco em usuários como físicos e engenheiros. O Python foi concebido a partir de outra linguagem existente na época, chamada ABC. Hoje, a linguagem é bem aceita na indústria por empresas de alta tecnologia,





# PYTHON

- A implementação oficial do Python é mantida pela PSF e escrita em C, e por isso, é também conhecida como CPython. A versão mais recente está disponível para download no endereço: <http://www.python.org/download/>
- Para a plataforma Windows, basta executar o instalador. Para outras plataformas, como em sistemas Linux, geralmente o Python já faz parte do sistema, porém em alguns casos pode ser necessário compilar e instalar o interpretador a partir dos arquivos fonte.



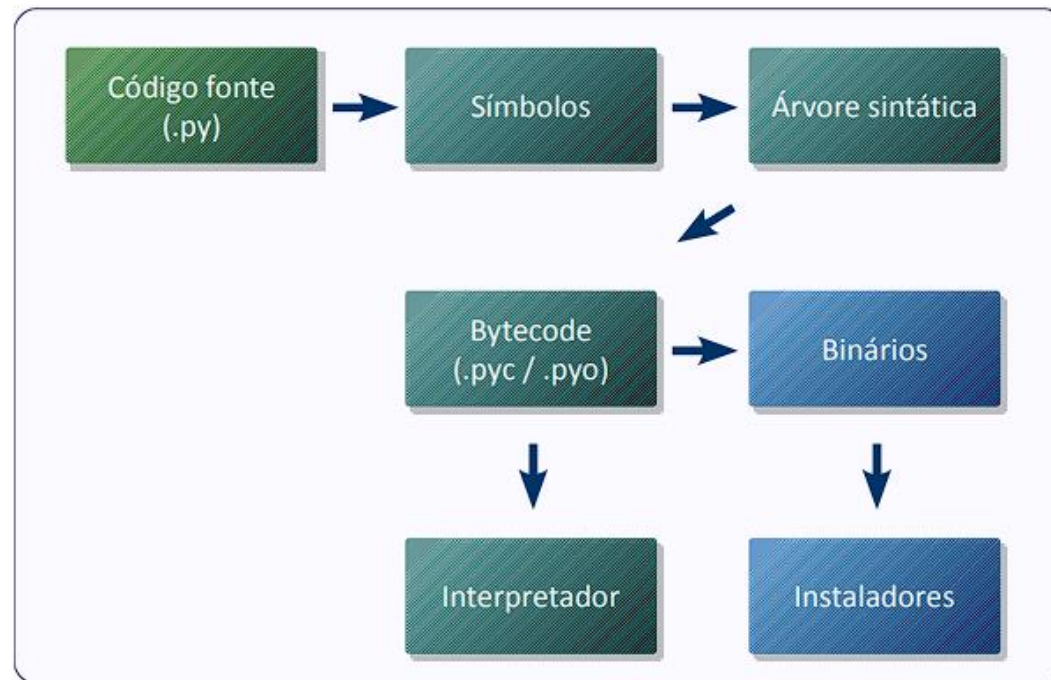
## TIPAGEM DINÂMICA

- Python utiliza tipagem dinâmica, o que significa que o tipo de uma variável é inferido pelo interpretador em tempo de execução. No momento em que uma variável é criada através de atribuição, o interpretador define um tipo para a variável, com as operações que podem ser aplicadas.
- A tipagem forte, ou seja, o interpretador verifica se as operações são válidas e não faz introduções automáticas entre tipos incompatíveis. Para realizar a operação entre tipos não compatíveis, é necessário converter explicitamente o tipo da variável ou variáveis antes da operação



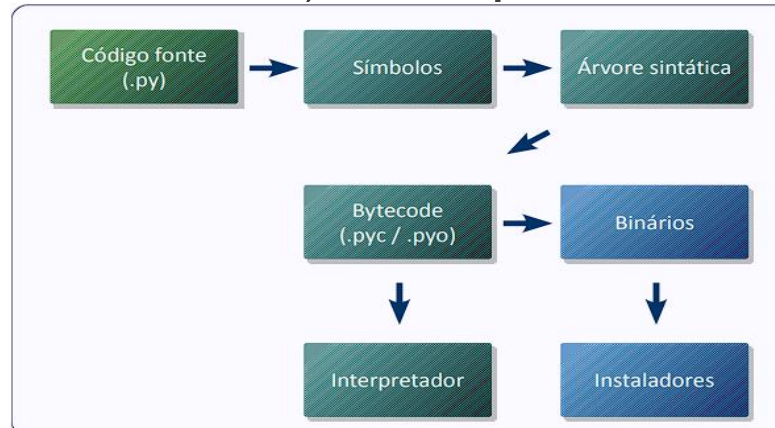
# COMPILAÇÃO E INTERPRETAÇÃO

- O código fonte é traduzido pelo Python para bytecode, que é um formato binário com instruções para o interpretador. O bytecode é multiplataforma e pode ser distribuído e executado sem fonte original.



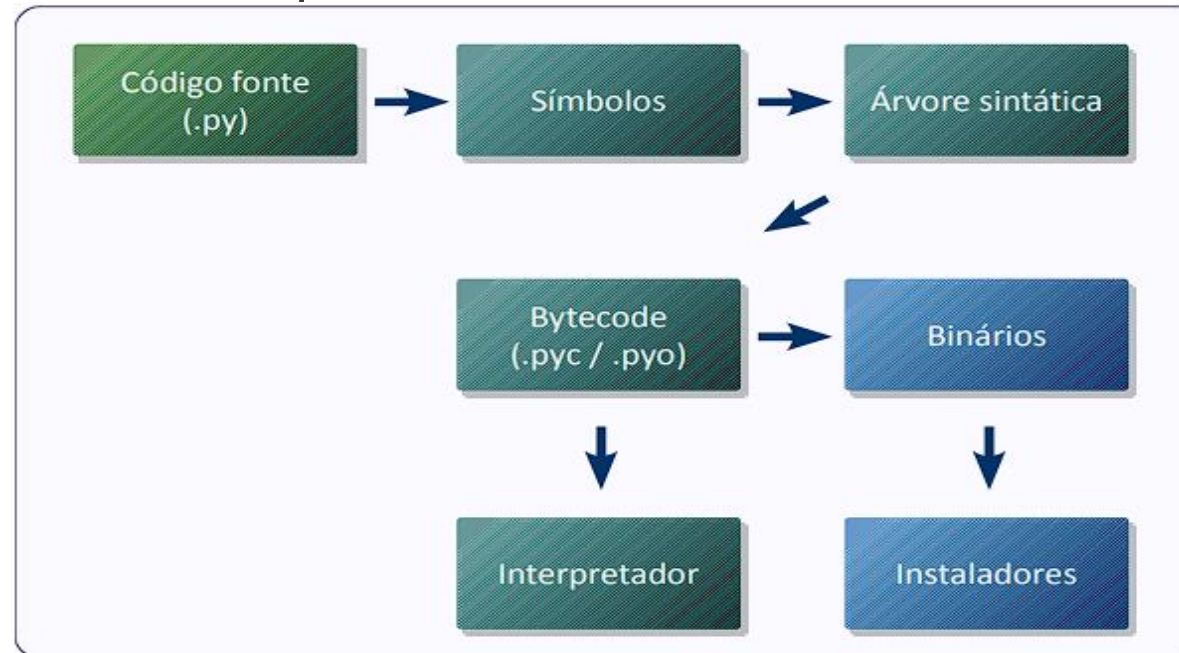
# COMPILAÇÃO E INTERPRETAÇÃO

- O interpretador compila o código e armazena o bytecode em disco, para que a próxima vez que o executar, não precise compilar novamente o programa, reduzindo o tempo de carga na execução.
- Se os arquivos fontes forem alterados, o interpretador se encarregará de regenerar o bytecode automaticamente, mesmo utilizando o shell interativo.
- Quando um programa ou um módulo é evocado, o interpretador realiza a análise do código, converte para símbolos, compila



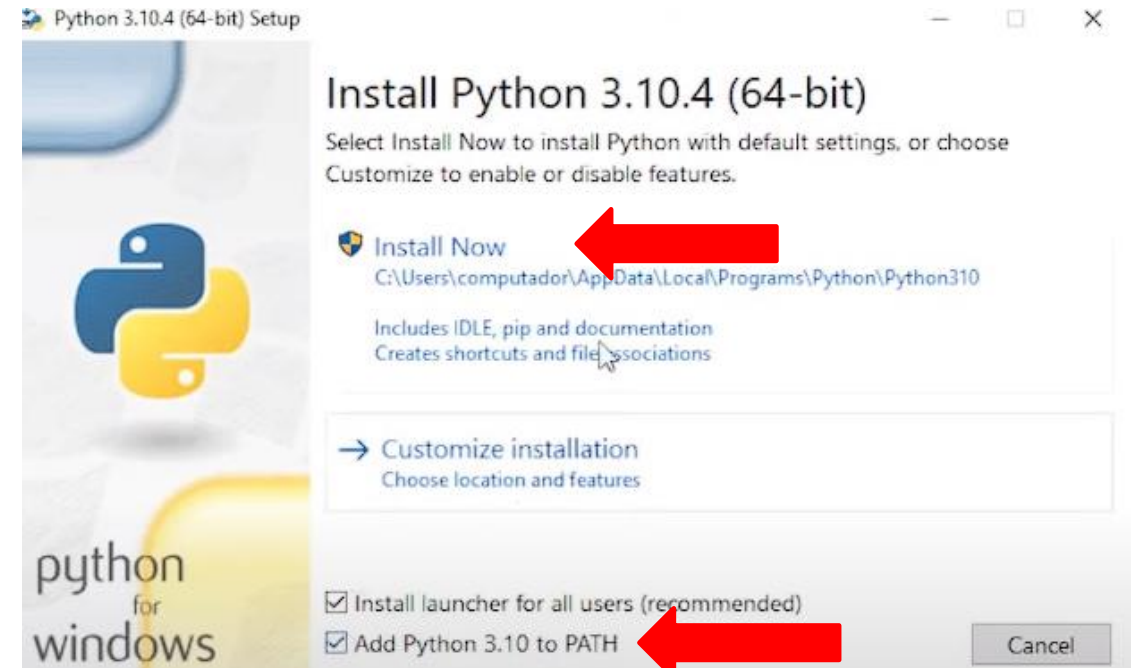
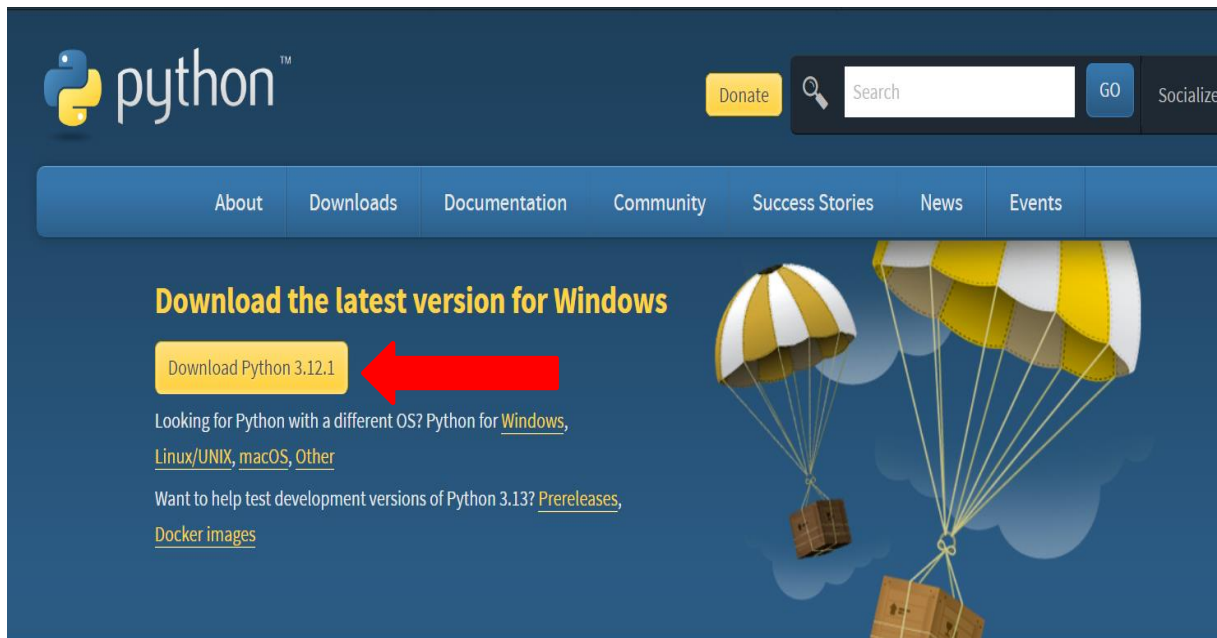
# COMPILAÇÃO E INTERPRETAÇÃO

- O bytecode é armazenado em arquivos com extensão “.pyc” (bytecode normal) ou “.pyo” (bytecode otimizado). O bytecode também pode ser empacotado junto com o interpretador em um executável, para facilitar a distribuição da aplicação, eliminando a necessidade de instalar Python em cada computador.



## PARA INSTALAR O PYTHON

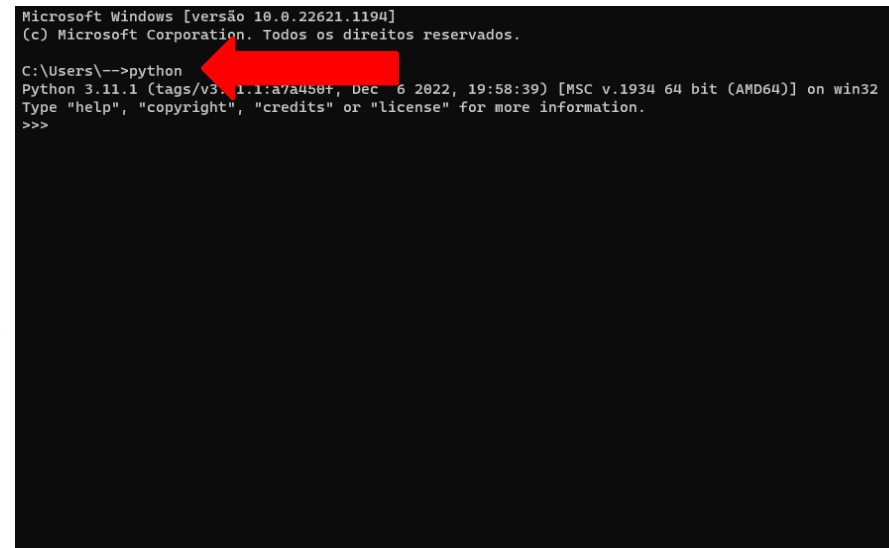
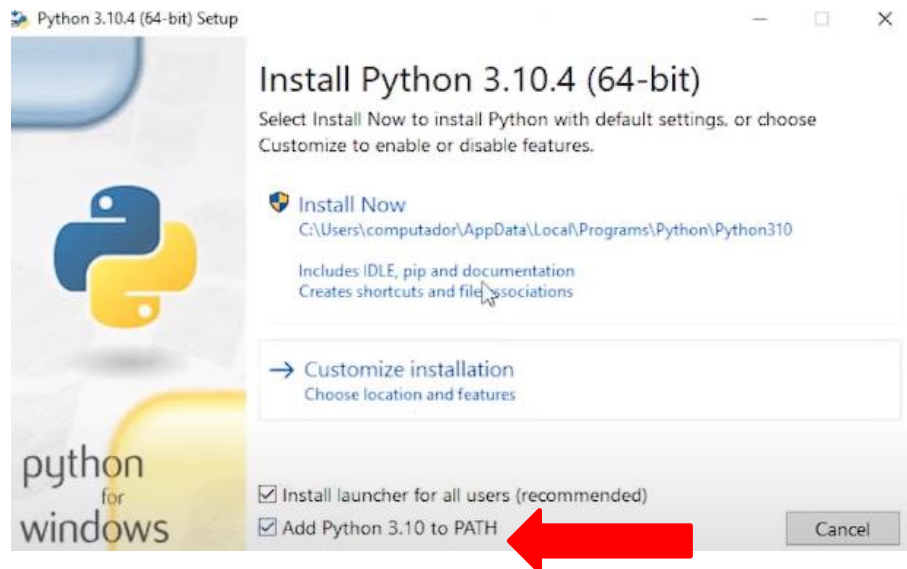
- Para instalar o python , basta baixar o interpretador do python no site <https://www.python.org/downloads/>, depois de baixar coloque para instalar e marque a opção add Python e clique na opção install Now





## MODO INTERATIVO

- O interpretador Python pode ser usado de forma interativa, na qual as linhas de código são digitadas em um prompt (linha de comando) semelhante ao shell do sistema operacional. Para evocar o modo interativo basta executar o interpretador (se ele estiver no path):

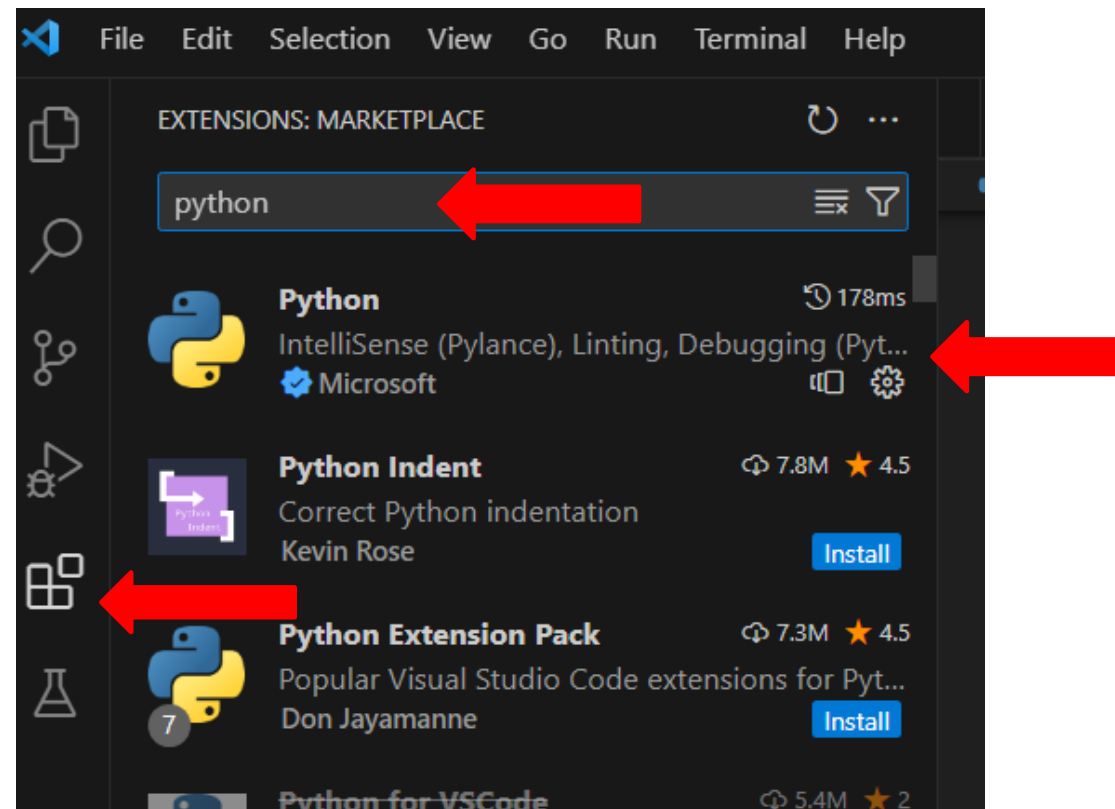


Para executar o modo interativo basta executar o interpretador

No cmd do Windows digite o comando python

## PARA INSTALAR O PYTHON

- Baixe o Visual Code e instale a extensão necessária para utilizar o python no VsCode,
- Clique em extensões em seguida na barra de pesquisa digite python e clique na primeira opção Python





# TIPOS DE DADOS

- As Linguagens de programação, em geral, usam o termo “tipo primitivo” para representar a informação em sua forma mais elementar, tais como inteiro, real, lógico ou caractere.
- Diferentemente de outras linguagens de programação como C, C++ e Java, Python possui tipagem dinâmica.
- Uma linguagem de programação que possui tipagem dinâmica como Python, PHP ou Perl não exige que o programador declare, explicitamente, o tipo de dado que será armazenado por cada variável.
- Essa característica permite que, ao longo da execução de um programa, uma mesma variável armazene valores de tipos distintos.

```
nome= "luiz"  
altura= 1.75  
peso=75.0
```



# PALAVRAS RESERVADAS

- Além disso, deve-se observar também que as palavras reservadas da linguagem Python não podem ser usadas como nomes de variáveis. Portanto, não é permitido declarar variáveis com os seguintes nomes.

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



# OPERADORES ARITMÉTICOS

- Python oferece diversos conjuntos de operadores que podem ser utilizados em um programa. Para realizar operações matemáticas, podemos utilizar os operadores aritméticos.

OPERADOR	DESCRIÇÃO	EXEMPLO DE APLICAÇÃO
+	Adição	<code>print(4 + 2)</code> #resulta em 6
-	Subtração	<code>print(4 - 2)</code> #resulta em 2
*	Multiplicação	<code>print(4 * 2)</code> #resulta em 8
/	Divisão	<code>print(4 / 3)</code> #resulta em 1.3333
//	Quociente inteiro da divisão	<code>print(4 // 3)</code> #resulta em 1



# OPERADORES ARITMÉTICOS

- Python oferece outros operadores aritméticos como para cálculo do resto da divisão e potenciação

OPERADOR	DESCRIÇÃO	EXEMPLO DE APLICAÇÃO
%	Resto da divisão inteira	<pre>print(4 % 2) #resulta em 0</pre>
**	Potenciação	<pre>print(4 ** 2) #resulta em 16</pre>



# OPERADORES DE ATRIBUIÇÃO

- O operador de atribuição, desde aquele mais simples e direto – como o sinal de igual (=), gostamos de chamar de “recebe” – até as abreviações envolvendo operadores aritméticos e de atribuição.

OPERADOR	DESCRIÇÃO	EXEMPLO DE APLICAÇÃO
=	Atribuição simples	<code>x = 2</code> #x recebe 2
+=	Atribuição de adição	<code>x += 2</code> #equivale a <code>x = x + 2</code>
-=	Atribuição de subtração	<code>x -= 2</code> #equivale a <code>x = x - 2</code>
*=	Atribuição de multiplicação	<code>x *= 2</code> #equivale a <code>x = x * 2</code>
/=	Atribuição de divisão	<code>x /= 2</code> #equivale a <code>x = x / 2</code>
%=	Atribuição de resto inteiro da divisão	<code>x %= 2</code> #equivale a <code>x = x % 2</code>
**=	Atribuição de potência	<code>x **= 2</code> #equivale a <code>x = x ** 2</code>



# TIPOS DE DADOS EM PYTHON PARA CONVERSÃO DE VALORES

- int – armazena valores numéricos inteiros
- float – armazena valores numéricos com ponto flutuante
- bool – armazena valores lógicos (True ou False).
- str – armazena cadeias de caracteres



# ENTRADA E SAÍDA DE DADOS EM PYTHON

- Para entrada de dados geralmente o usuário interage com o programa informando dados de entrada. Em Python, utiliza-se a função `input()` e para mostrar os dados usamos o comando `print()`

```
primeiro_valor = input("Digite o primeiro valor: ")  
segundo_valor = input("Digite o segundo valor: ")  
print(primeiro_valor + "\n" + segundo_valor)
```



## EXEMPLO CONVERSÃO DE VALORES

- Para fazer um cálculo em Python precisamos ler os valores e converter para o tipo de dados no qual será realizado o cálculo

```
primeiro_valor = int(input("Digite o primeiro valor: "))  
segundo_valor = int(input("Digite o segundo valor: "))  
soma = primeiro_valor + segundo_valor  
print(soma)
```





## EXEMPLO CONVERSÃO DE VALORES

- Ao ser utilizado os valores lidos em cálculos , temos que converter os valores no tipo no qual será realizado o cálculo , neste caso está sendo convertido para real(float)

```
base = float(input("Digite a base "))  
altura = float(input ("Digite a altura"))  
area = base * altura  
print (area)
```



# CONCATENAÇÃO EM PYTHON (F-STRING)

- As f-strings vão servir para que você consiga colocar uma variável dentro de um texto, e isso é feito utilizando a letra “f” antes do texto e colocando a sua variável dentro de {} chaves.

```
valor = float(input("Digite o valor da compra"))  
valortotal = valor * 10  
print(f" O valor total é {valortotal}")
```

Através da letra f é possível realizar concatenação entre a frase e a variável

A variável deve ser colocada entre {} chaves



# ARREDONDAMENTO DE VALORES EM PYTHON

- Para limitar as casas decimais em Python é usando o método `format()`. incluindo o número de casas decimais desejado. Utilizamos o padrão “`{:. 2f}`” para formatar o número com duas casas decimais.

```
print(f"Área = {area:.2f} cm2")
```



Formata a saída do valor , com duas casas decimais utilizamos `:.2f`



# FUNÇÃO MATH.POW NO PYTHON PARA POTENCIAÇÃO

- Caso tenhamos números elevados a potências muito grandes podemos utilizar a função da biblioteca math que já está incluída no python , basta apenas utilizar o comando `pow(variável,expoente)`.

```
potencia = pow(numero,2)
```

**A função pow calcula a  
potenciação do numero**

**Colocamos a variável,  
e depois da vírgula o expoente**



# EXERCÍCIOS

1. Construa um programa no qual um usuário informe a sua estatura em metros e o programa converta-a para centímetros. ( $\text{cm} = \text{m} * 100$ )
2. Construa um programa que receba do usuário a variação do deslocamento de um objeto (em metros) e a variação do tempo percorrido (em segundo). Ao fim, o programa deve calcular a velocidade média, em m/s, do objeto. ( $v_m = d / t$ )
3. Construa um programa para calcular a área de convivência de uma escola cujo formato é circular. Para isso, o usuário deve informar o valor do raio. ( $\text{área} = 3.14 * r^2$ )
4. Um aluno iniciou seus estudos em geometria plana e, para validar se suas respostas estão corretas, solicitou sua ajuda. Sabendo que  $\text{área} = (\text{base} * \text{altura}) / 2$ , construa um programa para auxiliar esse aluno

## EXERCÍCIOS

5. Ler dois valores , efetuar e mostrar o resultado das operações de adição, subtração, multiplicação e divisão de um número pelo outro.

6. Escreva um algoritmo para ler o salário mensal atual de um funcionário e o percentual de reajuste. Calcular e escrever o valor do novo salário.

7. Faça um algoritmo que leia nome a idade de uma pessoa, e escreva quantos dias essa pessoa já viveu. Considerar ano com 365 dias. Atenção mostre o nome e quantidade de dias que a pessoa viveu

8. Receba três números , faça a soma deles e calcule o valor da soma elevado ao quadrado.



## EXERCÍCIOS

9. Leia o número de votos brancos, nulos e válidos. Faça a soma do número total de eleitores. Calcular e escrever o percentual que cada um representa, em relação ao total de eleitores.  $\text{percbrancos} \leftarrow (\text{votosbrancos} * 100) / \text{totaleleitores}$

10. Calcular e apresentar o valor do volume de um cilindro, utilizando a fórmula:  $\text{VOLUME} \leftarrow \pi * \text{raio}^2 * \text{ALTURA}$

11. Leia a descrição do produto (nome), a quantidade comprada e o preço unitário. Calcular e escrever o total e a descrição do produto.

12. Calcule quantos azulejos são necessários para azulejar uma parede. É necessário conhecer a altura da parede, a sua largura, e a altura do azulejo e sua largura. Leia os dados em seguida calcule a área da parede, e do azulejo, em seguida calcule a quantidade de azulejos necessários.



OBRIGADO

LUIZ.BARRETO2@FATEC.SP.GOV.BR