**Introdução**

Esse material foi escrito com o objetivo de organizar o estudo para alunos da disciplina *Introdução à Computação* em Python.

O objetivo central de *Introdução à Computação* é desenvolver um [raciocínio aplicado na formulação e resolução de problemas computacionais](http://en.wikipedia.org/wiki/Computational_thinking).

Ferramentas tais como funções nativas e módulos existentes no Python evitam que reinventemos soluções disponíveis e façamos tarefas presumidamente desnecessárias. O domínio dessas ferramentas é fundamental para um programador efetivo. Apesar disso, esta disciplina utiliza um subconjunto *bem limitado* dos recursos de Python. Não por esses recursos não serem importantes, mas simplesmente porque eles não são o foco em uma disciplina cujo objetivo é devenvolver um [raciocínio aplicado na formulação e resolução de problemas computacionais](http://en.wikipedia.org/wiki/Computational_thinking).

O andamento da disciplina é orientado pelos problemas propostos. Cada problema introduz novos desafios que, para serem transpostos, nos levam a novas ideias e práticas de programação.

No que diz respeito à programação em Python, veremos:

* funções de entrada e saída: input() e print();
* tipos nativos de variáveis: int, float, str, e bool;
* funções de conversão de tipos: int() e float() ;
* operadores aritméticos: +, -, \*, /, //, %,...;
* operadores relacionais: <, >, <=, >=, == e !=;
* operadores lógicos ou booleanos: and, or e not;
* expressões aritméticas, relacionais e lógicas;
* execução condicional, alternativa e em cadeia: if, if-else, if-elif-else;
* comando de repetição: while;
* indicadores de passagem (*flags*);
* funções;
* listas;
* strings; e
* objetos.

# Variáveis, expressões e comandos

## Tópicos

* [Capítulo 2: Variáveis, expressões e comandos](https://panda.ime.usp.br/pensepy/static/pensepy/02-Conceitos/conceitos.html)
* comando de atribuição
* escrita na tela (= input())
* leitura via teclado (= print())
* classes str e int
* função de conversão int()
* [Comando de repetição: while](https://panda.ime.usp.br/aulasPython/static/aulasPython/while.html)
* expressões lógicas e operadores relacionais

## Estrutura de um programa

Utilizaremos a seguinte estrutura para um programa em Python.

**def** main():

*# comandos*

...

*#------*

*# a linha a seguir inicia a execução do programa*

main()

## Variáveis

Uma **variável** é um nome que se refere a um valor. Um comando de atribuição cria uma nova variável e lhe dá um valor.

Variáveis são usadas para guardarmos valores que serão usados mais tarde no programa.

## Atribuição

Um **comando** é uma unidade de código que o interpretador pode executar. Um comando de **atribuição** tem a seguinte forma:

variável = expressão

Significado:

1. o valor da expressão é calculado;
2. a variável é criada;
3. a variável passa a fazer referência/guardar o valor

Exemplos

num = 15

soma = 0 *# este é um comentário*

soma = soma + num

a = 3

b = 4

Cada nome de variáveis é uma letra ou o símbolo '\_' (underscore) seguida de letras, números e '\_'. Atenção, maiúscula é diferente de minúscula. Assim maior e Maior são duas variáveis diferentes.

## Exercício 2.1

Dados dois inteiros a e b, calcular a sua soma.

Tentativa 0: execute o programa abaixo e veja o que acontece.



1

2

3

4

5

6

7

8

9

def main():

a = 3

b = 4

soma = a + b

print("A soma de a + b eh igual a soma")

main()

(aula02\_ex1\_1)

Run Save Load

Observe que tudo entre aspas (”) define um texto ou string. Mas queremos imprimir o **valor** das variáveis e não o **nome** delas.

### print()

A função print() é utilizada para o programa exibir mensagens. Por exemplo:

**print**("Olá, mundo!")

**print**("A soma é", soma)

Escreve na tela do computador

Olá, mundo!

A soma é XXXX

onde XXXX é o valor ao qual a variável soma se refere.

Tentativa 1: execute o programa abaixo e veja o que acontece.

Observe que o print() recebeu dois **valores** separados por vírgula:

* um string (entre aspas) e
* o nome de uma variável

O string é impresso diretamente (pois esse é o seu **valor**) e, ao invés de imprimir o nome da variável soma, o print() exibe o valor ao qual a variável soma se refere, ou, dito mais simplesmente, o valor da variável soma.

Mas e como imprimir os valores de a e b ao invés de seus nomes?

Tentativa 2: execute o programa abaixo e veja o que acontece.



1

2

3

4

5

6

7

8

9

def main():

a = 3

b = 4

soma = a + b

print("A soma de", a, "+", b, "eh igual a", soma)

main()

(aula02\_ex1\_3)

Run Save Load

Agora sim, a mensagem ficou bem mais clara. Mas como fazer com que os valores a serem somados sejam definidos por um usuário?

Tentativa 3: execute o programa abaixo e veja o que acontece.



1

2

3

4

5

6

7

8

9

def main():

a = input("Digite o primeiro numero: ")

b = input("Digite o segundo numero: ")

soma = a + b

print("A soma de", a, "+", b, "eh igual a", soma)

main()

(aula02\_ex1\_4)

Run Save Load

### input()

Um programa pode utilizar a função input() para receber os dados ou valores que um usuário fornece através do teclado. O formato típico de um comando leitura é

variável = input("Prompt")

O programa pára e espera pela digitação de algum texto seguido do ENTER. "Prompt" é opcional e pode indicar o que programa deseja. Por exemplo,

nome = input("Qual é o seu nome? ")

**print**(nome, ", me fale sobre você.")

O valor que o usuário fornece e que será retornada pelo input() é sempre um string e não um número. Quando o operador + é realizado sobre strings, eles são concatenados (“grudados”) em vez de adicionados. Como veremos mais tarde, a operação de concatenar strings através do operador + é muito útil.

No momento desejamos adicionar dois números inteiros. Precisamos, portanto, de um maneira para converter um string em um número inteiro, para que o Python obtenha a soma desses números. Para ler um valor e converté-lo para um número inteiro utilizamos a função de conversão int().

### Strings

Uma **string** é texto entre aspas (") ou apóstrofo (')

### Valores

Programas manipulam valores. Valores podem ser de diferentes **tipos** ou **classes**. Até agora vimos que um **valor** pode ser um número inteiro (classe int) ou um texto (classe str). Por exemplo, "Oi" é um string (valor da classe str), 23 é um inteiro (valor da classe int), e "23" é um string (valor da classe str).

**>>>** type("Oi")

<class 'str'>

**>>>** type(23)

<class 'int'>

**>>>** type("23")

<class 'str'>

>>>

### Operacões sobre strings

O operador + concatena dois strings.

Tentativa 4: execute o programa abaixo e veja como ele se comporta.



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

def main():

# a\_str e b\_str guardam strings

a\_str = input("Digite o primeiro numero: ")

b\_str = input("Digite o segundo numero: ")

# a\_int e b\_int guardam inteiros

a\_int = int(a\_str) # converte string/texto para inteiro

b\_int = int(b\_str) # converte string/texto para inteiro

# calcule a soma entre valores que são números inteiros

soma = a\_int + b\_int

# imprima a soma

print("A soma de", a\_int, "+", b\_int, "eh igual a", soma)

main()

(aula02\_ex1\_5)

Run Save Load

### int()

A função int() converte um dado string para um número inteiro.

**>>>** a = int("123")

**>>>** type (a)

<class 'int'>

**>>> print**("valor inteiro =", a)

valor inteiro = 123

>>>

O string deve ser algo como "123" ou "-5", pois senão ocorrerá um erro.

**>>>** a = int("1.5")

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

ValueError: invalid literal for int() with base 10: '1.5'

**>>>** a = int("a156")

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

ValueError: invalid literal for int() with base 10: 'a156'

>>>

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

ValueError: invalid literal for int() with base 10: 'a156'

Como não precisamos guardar as respostas na forma de texto, podemos simplificar o programa compondo as funções int() e input() da seguinte forma:

Tentativa 5:



1

2

3

4

5

6

7

8

9

def main():

a = int(input("Digite o primeiro numero: "))

b = int(input("Digite o segundo numero: "))

soma = a + b

print("A soma de", a, "+", b, "eh igual a", soma)

main()

(aula02\_ex1\_6)

Run Save Load

## Exercício 2.2

Dada uma sequência de números inteiros diferentes de zero, terminada por um zero, calcular a sua soma. Por exemplo, para a sequência:

12 17 4 -6 8 0

o seu programa deve escrever o número 35.

Tente escrever a sua solução primeiro e, depois, clique [aqui](https://panda.ime.usp.br/aulasPython/static/aulasPython/exercicios/ex22.html) para ver uma.



1

def main():

num = int(input("Digite um inteiro: "))

soma = 0

while num != 0:

soma = soma + num

num = int(input("Digite um inteiro: "))

print("A soma eh", soma)

#----------------------------------------------

main()

(aula02\_ex2\_tentativa)

Run Save Load

### Comando de repetição: while

Leitura sobre [o comando de repetição while](https://panda.ime.usp.br/aulasPython/static/aulasPython/while.html).

### Expressões lógicas

Condições ou **expressões lógicas** (boolean expressions) são expressões cujo valor é verdadeiro (=``True`` em Python) ou falso (=``False`` em Python) e usam, entre outros, os operadores relacionais:

* > (maior);
* >= (maior ou igual);
* < (menor);
* <= (menor ou igual);
* == (igual); ou
* != (diferente).

## Exercício 2.3

Nota: Exercício 4 da [lista sobre inteiros](http://www.ime.usp.br/~macmulti/exercicios/inteiros/index.html).

Dados números inteiros n e k, com k >= 0, calcular n elevado a k. Por exemplo, dados os números 3 e 4 o seu programa deve escrever o número 81.

Tente escrever a sua solução abaixo primeiro e, depois, clique [aqui](https://panda.ime.usp.br/aulasPython/static/aulasPython/exercicios/ex23.html) para ver uma.



1

2

def main():

n = int(input("Type n value:"))

k = int(input("Type k value:"))

calc = n \*\* k

print(calc)

main()

(aula02\_ex3\_tentativa)

Run Save Load

### Exercício 2.4

Nota: Exercício 8 da [lista sobre inteiros](http://www.ime.usp.br/~macmulti/exercicios/inteiros/index.html).

Dado um número inteiro n >= 0, calcular n!.

Tente escrever a sua solução abaixo primeiro e, depois, clique [aqui](https://panda.ime.usp.br/aulasPython/static/aulasPython/exercicios/ex24.html) para ver uma.



1def main():

n = int(input("Digite o valor de n: "))

fat = 1

i = 2

while i <= n:

fat = fat\*i

i = i + 1

print("O valor de %d! eh =" %n, fat)

#--------------------------------------------

main()

2