



Aula 06 - Tipos primitivos

▼ Dificuldade	★
☰ Tags	<code>.isalnum</code> <code>.isalpha</code> <code>.islower</code> <code>.isnumeric()</code> <code>.isupper</code> <code>bool()</code> <code>float</code> <code>int</code> <code>str</code>

Tipos primitivos e saída de dados

<https://youtu.be/hdDHg1p3YVc>

Toda linguagem de programação trabalha com **tipos primitivos**, o Python tem vários tipos primitivos, **mas há quatro que são fundamentais**.

Se tentarmos fazer uma **conta** sem qualquer tipo primitivo.

```
num1 = input('Digite um número: ')\nnum2 = input('Digite outro número: ')\nprint('A soma desses números é: ', num1+num2)
```

O que será retornado vai ser **um número seguido de outro**, e não **a soma dos dois**. Pois o Python entende o que será colocado na variável como uma **string** e não como um número.

Mas colocando o que será digitado como um **número inteiro**, a soma será feita!

```
num1 = int(input('Digite um número: '))\nnum2 = int(input('Digite outro número: '))\nprint('A soma desses números é: ', num1+num2)
```

Os quatro tipos primitivos básicos

int → Números inteiros

▼ Exemplos

- 5
- 345
- 231
- 34594
- -345

float → Números decimais

▼ Exemplos

- 4.3
- 54.34
- -940.23
- 13.2
- 7.0

bool → Valores lógicos

▼ Exemplos

- **True**
- **False**

str → Texto

▼ Exemplos

- 'Olá'
- '7.45'
- 'Seja bem-vindo'
- 'Qualquer coisa entre aspas'

Ao invés de concatenar uma **variável** na string assim:

```
print('A soma desses números é: ', num1+num2)
```

É possível fazer desse jeito:

```
print('A soma desses números é: {}'.format(num1+num2))
```

Com o print ainda melhor, usando o **.format**:

```
print('A soma entre {} e {} é {}'.format(num1, num2, num1+num2))
```

Para colocar **mais de uma** variável no print, basta usar o **{ }** e no **.format** colocar na ordem qual variável será substituída.

Usando valores lógicos.

```
n = bool(input('Digite algo: '))  
print(n)
```

Se usarmos o **valor booleano** e digitarmos algo o valor retorna **True**.

Se não digitarmos nada em n, o valor retorna **False**.

Verificando se o valor digitado é **numérico**:

```
n = input('Digite algo: ')  
print(n.isnumeric())
```

Se digitarmos um **número**, o Python perceberá que o valor é **numérico**, mesmo sem colocar o valor como número. O método **isnumeric()** identifica a possibilidade do valor ser convertido para número ou não.

Há também a **verificação** se o valor digitado é formado somente por **letras**.

```
n = input('Digite algo: ')  
print(n.isalpha())
```

Novamente, o **método** identifica a possibilidade do valor digitado ser convertido em **str**.

Se o valor for formado por **letras** e **números** é possível usar outro método.

```
n = input('Digite algo: ')
print(n.isalnum())
```

O método `.isalnum()` verifica se o valor digitado é formado por **letras** e **números**.

Há diversos outros métodos verificadores, o `.isupper()`, que verifica se o que foi digitado é somente formado por **letras maiúsculas**, o `.islower()`, que verifica se o valor é formado somente por **letras minúsculas**, etc.

Desafio 4

Faça um programa que leia algo pelo teclado e mostre na tela o seu tipo primitivo e todas as informações possíveis sobre ele.

```
n = input('Digite algo: ')
print('Esse valor é do tipo: {}'.format(type(n)))
print('Esse valor é formado por letras? {}'.format(n.isalpha()))
print('Esse valor é formado por números? {}'.format(n.isnum()))
print('Esse valor é formado por números e letras? {}'.format(n.isalnum()))
print('Esse valor é capitalizado? {}'.format(n.istitle()))
```

https://www.youtube.com/watch?v=tHYxjJxtJko&list=PLHz_AreHm4dm6wYOIW20Nyg12TAjmMGT-&index=4