

IO (Input / Output)

Até o momento nossos scripts em python não parecem muito divertidos, uma vez que seu funcionamento é basicamente estático, realizamos operações matemáticas, podemos salvar textos e números dentro de variáveis, mas não podemos fazer muito mais do que isso, mas esse paradigma está prestes a mudar, vamos aprender como acrescentar uma camada de interatividade aos nossos scripts.

Logo na primeira aula ensinamos uma das principais formas de apresentar conteúdo de maneira textual para o usuário do terminal por meio do comando `print()`. Este comando na verdade é uma função, não se desespere, vamos aprender o que são funções em python logo logo, o que você precisa saber neste momento é que funções recebem valores como **argumentos** e utilizam estes valores para executar instruções específicas.

Como o próprio nome diz, `print()` exibe, de maneira textual, valores para o nosso terminal. Podemos imprimir strings, ints, floats, além de uma gama de outros tipos de dados.

```
print(2)
print(3.14)
print("Hello World")
```

Print também funciona para variáveis:

```
pi = 3.14
two = 2
greet = "Hello"

print(pi)
print(two)
print(greet)
```

Legal, o comando `print()` ajuda a dar um pouco mais de interatividade aos nossos scripts mas não permite que o usuário interaja de maneira ativa com o programa, mas não por muito tempo, existe uma outra função, a `input()`, que permite que o programa capte valores digitados diretamente no terminal e utilize na execução do programa, veja um exemplo:

```
x = input()
```

No exemplo acima, o programa captura o que o usuário digitar no console e armazena em uma variável para posterior uso, dessa forma, podemos fazer programas mais dinâmicos que respondem de maneira diferente dependendo da interação do usuário. Como exemplo, vamos escrever um programa que efetua o cálculo do quadrado de um número arbitrário digitado pelo usuário.

```
print("Digite um número: ")
num = input()
print(num * num)
```

Se você tentar executar esse programa, vai observar que temos alguns problemas. Primeiro, é exibido digite um número, e o cursor do terminal automaticamente vai para a linha seguinte, podemos resolver isso removendo o primeiro `print()` e passando o texto diretamente na função `input`, veja:

```
num = input("Digite um número: ")
print(num * num)
```

O segundo problema é na verdade um erro de execução, em aplicações de console, as operações de entrada e saída são sempre tratadas como textos, por este motivo, o nosso `num` não é um tipo numérico como `int` ou `float`, mas na verdade é do tipo `str`. Como você deve imaginar, não é possível

realizar operações matemáticas com textos, mesmo que estes se pareçam com números, antes de realizar operações matemáticas, temos que converter o texto, em uma representação numérica, como um int.

```
num_str = input("Digite um número: ")
num = int(num_str)
print(num * num)
```

Parece complexo em um primeiro momento, mas não se preocupe vamos entender o que está sendo feito linha por linha:

Linha 1 - É obtido o valor digitado pelo usuário, em formato textual, ou seja uma string (str)

Linha 2 - Convertemos o valor de uma representação textual, para uma representação numérica, usando a função construtora `int()`.

Linha 3 - O cálculo é realizado sem erros quando num passa a ser um int ao invés de um `str`.

Repare que em python, precisamos nos atentar ao sistema de tipos (str, int, float etc), caso contrário vamos encontrar problemas na execução dos nossos scripts.

Exercícios

1. Crie um script que recebe dois valores por meio da interação do usuário e realiza a divisão do primeiro número pelo segundo.