

## Tipos de resultados impressos em prints no python

O comando print no python é extremamente versátil e usado de várias formas para exibição de informações no console. Abaixo alguns exemplos:

- Print básico:

```
Print("olá Mundo")
```

exibe uma mensagem simples.

- Print com múltiplos argumentos:

```
print("olá", "mundo!", sep="-")
```

aqui, os argumentos são separados por um delimitador definido pelo parâmetro "sep".

- Print formatado com f-strings :

```
Nome = "João"
```

```
Print(f"Meu nome é {nome}")
```

Permite inserir variáveis diretamente na string

- Print com end personalizado:

```
Print("Linha 1", end=" ")
```

```
Print("Linha 2")
```

O parâmetro "end" define o que será adicionado ao final da linha (por padrão, é uma nova linha "\n")

- Print com formatação antiga (%):

```
Print("Meu nome é &s e eu tenho %d anos." % ("João", anos))
```

Método antigo, mas ainda funcional.

- Print com .format():  
`Print("Meu nome é {} e tenho {} anos." .format("João", 25))`  
Substitui os {} pelos valores fornecidos

## Módulos no python

Módulos são arquivos python que contêm definições e implementações de funções, classes e variáveis. Eles ajudam a organizar e reutilizar código. Abaixo alguns tipos de módulos.

- **Módulos embutidos (built-in):** São módulos que vêm com o python e não precisam ser instalados.  
EX: `math` (funções matemáticas)  
`Import math`  
`Print(math.sqrt(16))` #saida – 4.0
- **Random** (gerador de números aleatórios)  
EX: `Import random`  
`Print(rando.randint(1,10))` #número aleatório de 1 a 10
- **Módulos externos:** São módulos criados por terceiros e instalados via `pip`.  
EX: `numpy` (para computação numérica)

## Pandas (para análise de dados)

Para instalar: no terminal digite `pip install numpy pandas`

- **Módulos personalizados:** Você pode criar seus próprios módulos.

```
EX: arquivo meu_modulo.py  
def saudacao(nome):  
    return f"Olá, {nome}!"
```

#para usar o módulo:

```
Import meu_modulo  
Print(meu_modulo.saudacao("maria"))  
#saida: Olá, Maria
```

- **Pacotes:** São coleções de módulos organizados em diretórios. Um pacote contém um arquivo `__init__.py` para ser reconhecido como tal.

## Diferença Entre Funções e Módulos em Python

Funções:

- **O que são?** Blocos de código reutilizáveis que realizam uma tarefa específica.
- **Objetivo:** Tornar o código mais organizado, modular e fácil de manter.

- **Como criar?** Usando a palavra-chave def.
- **Exemplo:**

```
Def saudacao(nome):
```

```
    Return print(f"olá, {nome}")
```

# Aqui, a função saudacao pode ser chamada várias vezes com diferentes valores para nome.

Módulos:

- **O que são?** Arquivos Python que contêm funções, classes e variáveis relacionadas. Servem para organizar e reutilizar código em projetos maiores.
- **Objetivo:** Agrupar funcionalidades em arquivos separados para facilitar a manutenção e a reutilização.
- **Como criar?** Basta salvar um arquivo com extensão .py e importar onde necessário.
- **Exemplo:**
  1. Crie um arquivo chamado meu\_modulo.py:

```
def saudacao(nome):
```

```
    Return print(f"olá, {nome}")
```

2. use o módulo em outro arquivo importando ele

OBS! está é a maior diferença entre módulo e função

- **Funções:** São blocos de código dentro de um programa ou módulo.
- **Módulos:** São arquivos que agrupam funções, classes e variáveis para serem reutilizados em outros programas.

