



print(), input() e comentários

Comentários

▼ Comentário simples

- Utiliza o símbolo `#`
- O interpretador ignora o que está escrito após o `#` (na mesma linha)
- `CTRL /` transforma em comentário

▼ Comentários de múltiplas linhas

- Utiliza o símbolo `"""` `"""` ou `'''` `'''` para definir linhas que não serão exibidas

! Na realidade, esse caso NÃO é um comentário, apesar de ser utilizado como tal. Isso ocorre pois o interpretador lê as linhas, mesmo sem executá-las

Print()

▼ Exemplo básico

```
print("Olá mundo")
```

▼ Quebra de linha (`\n`)

- Utiliza o símbolo `\n`

▼ Exibir com uma tabulação (`\t`)

- `\t` representa uma tabulação horizontal, que é usada para adicionar espaço entre textos em uma linha. Ele insere um "tab", que geralmente equivale a 4 ou 8 espaços.
- É útil para a exibição de diversas colunas, tendo o devido espaçamento entre elas.

▼ Definir separação (`sep`)

- `Print(1, 2, 3, sep = " ")`
 - Exibe: 1 2 3
- `Print(1, 2, 3, sep = "-")`
 - Exibe: 1-2-3

▼ Definir final (`end`)

- `Print(1, 2, 3, end = ".")`
 - Exibe: 1 2 3.
- `end = "\n"`
 - Útil para terminar o print pulando uma linha

▼ Escape (`\`)

- É utilizado para evitar problemas quando há aspas dentro de aspas que definem uma string
 - Exemplo: `print ("Teste de " caractere " de escape")` - Não exibe caractere
 - Solução: `print ("Teste de \" caractere \" de escape")` - Exibe "caractere"

! Alternativa: `print ("Teste de 'caractere' de escape")` - Inverter as aspas

▼ Exibir caracteres ocultos (`r`)

- Exemplo: `print (r "Caractere oculto \n ")`
 - Exibe: Caractere oculto \n

input()

▼ Sintaxe

```
variavel = input("Pode incluir uma frase")
```

▼ Exemplo básico

```
nome = input("Digite seu nome:")  
idade = input("Digite sua idade: ")  
altura = input("Digite sua altura")  
print(f"Nome: {nome}\nIdade: {idade}\nAltura:{altura}")
```

! input() sempre recebe um valor do tipo **string**