

---

## Arakhnê Project

### Arakhnê Document

---

Reference: XXX  
Version: 0.1  
Updated: July 22, 2023  
Status: Confidential

Authors:

Project	Arakhnê Project
Document	Arakhnê Document
Reference	XXX
Version	0.1
Last Update	July 22, 2023

INVENTIA

Version History		
<i>Version</i>	<i>Date</i>	<i>Updates</i>
height		

---

## Arakhnê Project

### Arakhnê Document

---

Reference: XXX  
Version: 0.1  
Updated: July 22, 2023  
Status: Confidential

Authors:

Python = Linguagem programação  
tipo de tipagem = Dinamica/Forte  
str -> string -> texto  
Strings são textos que estão dentro das aspas

## 0.2/ COERÇÃO

conversão de tipos , coerção type conversion, typecasting, coercion é o ato de converter um tipo em outro  
tipos imutaveis e primitivos str, int , float, bool

## 0.3/ SEPARADOR

nesse trecho de codigo sep indica que os numeros inteiros seram separados por - e ao final e o end barra  
n indica que havera uma quebra de linha

```
print(12, 34, 1011, sep= "-", end= '\n##')  
print(9, 10, sep= "-", end= '\n')  
print(56, 78, sep= '-', end= '\n')
```

## 0.4/ F-STRINGS

consigo inserir dentro de uma string as variaveis dentro do codigo

```
nome = 'Luiz Otavio'  
altura = 1.80  
peso = 95  
imc = peso / altura ** 2
```

```
"f-strings"  
linha_1 = f'{nome} tem {altura:.2f} de altura,'  
linha_2 = f'pesa {peso} quilos e seu imc e '  
linha_3 = f'{imc:.2f}'
```

```
print(linha_1)  
print(linha_2)  
print(linha_3)
```

```
# Luiz Otavio tem 1.80 de altura,  
# pesa 95 quilos e seu IMC e  
# 29.320987654320987
```

## 0.5/ FORMATAÇÃO STRINGS USANDO FORMAT

setando a quantidade de casas apos a virgula

```

b = 'BBBBBB'
c = 1.1
string = 'b={nome2} a={nome1} a={nome1} c={nome3:.2f}'
formato = string.format(
    nome1=a, nome2=b, nome3=c
)

print(formato)

```

## 0.6/ OPERADORES LOGICOS

Operadores logicos and (e) or (ou) not (não)

and - Todas as condições precisam ser verdadeiras. Se qualquer valor for considerado falso, a expressão inteira será avaliada naquele valor São considerados falsy (que vc já viu) 0 0.0 '' False Também existe o tipo None que é usado para representar um não valor

### 0.6.1/ EXEMPLO AND

```

entrada = input('[E]ntrar [S]air: ')
senha_digitada = input('Senha: ')

senha_permitida = '123456'

if entrada == 'E' and senha_digitada == senha_permitida:
    print('Entrar')
else:
    print('Sair')
#Avaliacao de curto circuito
print(True and False and True)
print(True and 0 and True)

```

### 0.6.2/ EXEMPLO OR

esse exemplo serve para verificar senha ou se não há senha

```

# entrada = input('[E]ntrar [S]air: ')
# senha_digitada = input('Senha: ')

# senha_permitida = '123456'

# if (entrada == 'E' or entrada == 'e') and senha_digitada ==
#     senha_permitida:
#     print('Entrar')
# else:
#     print('Sair')

# Avaliacao de curto circuito
senha = input('Senha: ') or 'Sem senha'
print(senha)

```

usado para inverter expressões convém as vezes usar dentro de um print

```
@@ -0,0 +1,7 @@  
# Operador logico "not"  
# Usado para inverter expressões  
# not True = False  
# not False = True  
# senha = input('Senha: ')  
print(not True) # False  
print(not False) # True
```

#### 0.6.4/ OPERADOR NOT IN

```
# Operadores in e not in  
# Strings são iteráveis  
# 0 1 2 3 4 5  
# o t a v i o  
# -6-5-4-3-2-1  
# nome = 'Otavio'  
# print(nome[2])  
# print(nome[-4])  
# print('vio' in nome)  
# print('zero' in nome)  
# print('tio' not in nome)  
# print('zero' not in nome)  
  
nome = input('Digite seu nome: ')  
encontrar = input('Digite o que deseja encontrar: ')  
  
if encontrar in nome:  
    print(f'{encontrar} esta em {nome}')  
else:  
    print(f'{encontrar} não esta em {nome}')
```

#### 0.7/ INTERPOLAÇÃO DE STRING COM PORCENTAGEM EM PYTHON

```
"""  
Interpolacao basica de strings  
s - string  
d e i - int  
f - float  
x e X - Hexadecimal (ABCDEF0123456789)  
"""  
  
nome = 'Luiz'  
preco = 1000.95897643  
variavel = '%s, o preco e R$%.2f' % (nome, preco)  
print(variavel)  
#conversao de inteiro decimal para hexadecimal  
print('0 hexadecimal de %d e %08X' % (1500, 1500))
```

```
"""
Formatacao basica de strings
s - string
d - int
f - float
.<numero de digitos>f
x ou X - Hexadecimal
(Caractere)(><^)(quantidade)
> - Esquerda
< - Direita
^ - Centro
= - Forca o numero a aparecer antes dos zeros
Sinal - + ou -
Ex.: 0>-100,.1f
Conversion flags - !r !s !a
"""
```

```
variavel = 'ABC'
print(f'{variavel}')
print(f'{variavel: >10}')
print(f'{variavel: <10}.')
print(f'{variavel: ^10}.')
print(f'{1000.4873648123746:0=+10,.1f}')
print(f'0 hexadecimal de 1500 e {1500:08X}')
print(f'{variavel!r}')
```