



Revisão Python

APOIO CTISM Informática
Monitora: Andriza Campanhol



Primeiro de tudo...

```
print('Olá Mundo!')
```



Tipos e Conversões

n = 20 #Inteiro

n = 20.5 #Float

x = True #Booleano

s = 'CTISM' #String

int(var)

#converte para inteiro

float(var)

#converte para float

bool(var)

#converte para booleano

Operadores Relacionais e Lógicos

Nome do Operador	Operador	Descrição
menor que	<	operador menor que
maior que	>	operador maior que
menor ou igual	<=	operador menor que ou igual a
maior ou igual	>=	operador maior que ou igual a
igual	==	operador igualdade
diferente	!=	operador desigualdade
e lógico	and	resulta True quando ambos operadores são True
ou lógico	ou	resulta True quando ao menos um operador é True
não lógico	not	nega o valor (ex: True se torna False)

Exemplo

```
print('Olá Mundo!')  
  
print(5==10)  
print(10 > 5)  
print((5 >= 1) and (5 <= 10))
```

Saída:

```
Olá Mundo!  
False  
True  
True
```

Operações Aritméticas

Nome da operação	Operação
Soma	+
Subtração	-
Divisão	/
Multiplicação	*
Exponenciação	**
Radiciação	** 1/2
Módulo	%
Divisão inteira	//

Exemplo

```
op = (6 + 4) / 2
op2 = (2*2 + 0.5) // 2
op3 = 40 % 2

print(op)
print(op2)
print(op3)
```

Saída:

```
5.0
2.0
0
```

Operações com Sequências

Nome da operação	Operação	Descrição
Indexação	[]	acessa um elemento da sequência
Concatenação	+	combina duas sequências
Repetição	*	concatena um certo número de vezes
Pertinência	in	responde se um item pertence a sequência
Comprimento	len	fornece o número de itens da sequência

Criação de Listas

```
lista = [1, 4, 21, True, 0,  
2021]  
print(lista)
```

```
[1, 4, 21, True, 0, 2021]
```

```
lista2 = [0] * 5  
print(lista2)
```

```
[0, 0, 0, 0, 0]
```

```
lista3 = list(range(10))  
print(lista3)
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Métodos em Listas

Nome da método	Exemplo de uso	Descrição
append	lista.append(item)	adiciona um novo item ao final da lista
insert	lista.insert(i,item)	insere um item na i-ésima posição da lista
pop	lista.pop()	remove e retorna o último item da lista
pop	lista.pop(i)	remove e retorna o i-ésimo item da lista
sort	lista.sort()	modifica uma lista para ficar ordenada
reverse	lista.reverse()	modifica uma lista, inverte a ordem dos itens
del	del lista[i]	exclui o item na posição i
index	lista.index(item)	retorna o índice da primeira ocorrência do item
count	lista.count(item)	retorna o número de ocorrências do item
remove	lista.remove(item)	remove a primeira ocorrência do item

Exemplo

```
minhaLista = [1024, 3, True, 6.5]
minhaLista.append(False)
print(minhaLista)
minhaLista.insert(2,4.5)
print(minhaLista)
print(minhaLista.pop())
print(minhaLista)
print(minhaLista.pop(1))
print(minhaLista)
minhaLista.pop(2)
print(minhaLista)
minhaLista.sort()
print(minhaLista)
minhaLista.reverse()
print(minhaLista)
print(minhaLista.count(6.5))
print(minhaLista.index(4.5))
minhaLista.remove(6.5)
print(minhaLista)
del minhaLista[0]
print(minhaLista)
```

Saída:

```
[1024, 3, True, 6.5, False]
[1024, 3, 4.5, True, 6.5, False]
False
[1024, 3, 4.5, True, 6.5]
3
[1024, 4.5, True, 6.5]
[1024, 4.5, 6.5]
[4.5, 6.5, 1024]
[1024, 6.5, 4.5]
1
2
[1024, 4.5]
[4.5]
```

if, else e elif

```
a = 2  
b = 4  
c = 10
```

```
if b > a:  
    print("b é maior que a!")  
elif c > a:  
    print("c é maior que a!")  
else:  
    print("b e c são menores que a!")
```

saída:

b é maior que a!

for

```
lista = ["Pedro", "Maria", "João", "Bianca"]
```

```
for i in range(0,5):  
    print(i)
```

```
for l in lista:  
    print(l)
```

Saída:

0

1

2

3

4

Pedro

Maria

João

Bianca

while

```
contador = 0
```

```
while contador < 10:  
    print(contador)  
    contador += 1
```

Saída:

0

1

2

3

4

5

6

7

8

9

Referências

- Material sobre Introdução ao Python PET-SI (Minicurso)
- https://panda.ime.usp.br/pythonds/static/pythonds_pt/01-Introducao/07-revisaoPython.html
- <https://panda.ime.usp.br/algoritmos/static/algoritmos/01-revisaoPython.html>