

# PYTHON

professor



Lázaro  
Santos

Palavras-Chave  
e Tipos





# Tipos Básicos de Dactos





# TIPOS BÁSICOS

## Inteiros

2, 5, 2007, -55

5007  
5  
-3





# TIPOS BÁSICOS

Ponto flutuante

```
5.990e120, 3.1415, 5.0
```

3.986

1.1e7840

4.0002





# TIPOS BÁSICOS

String

```
"Olá!", "Lázaro", "Bom dia!"
```

**"Bom  
dia!"**



# TIPOS BÁSICOS

## String

```
texto = """  
Agora  
eu consigo  
pular linhas  
!!!  
"""
```

```
texto2 = '''  
Eu  
também  
😁  
!!!  
'''
```





# TIPOS BÁSICOS

## Booleanos

```
x, y = True, False
```

True

False



# TIPOS BÁSICOS

```
x, y = 3, 2
print(x + y) # = 5
print(x - y) # = 1
print(x * y) # = 6
print(x / y) # = 1.5
print(x // y) # = 1
```

```
print(x % y) # = 1
print(-x) # = -3
print(abs(-x)) # = 3
print(int(3.9)) # = 3
print(float(3)) # = 3.0
print(x ** y) # = 9
```





# INDEXAR E FATIAR

```
s = "0 papa mais novo tinha 11 anos"
print(s[0])           # '0'
print(s[2:4])         # 'pa'
print(s[-3:-1])       # 'no'
print(s[-3:])         # 'nos'
```





# FUNÇÕES PARA STRINGS

```
s = "smartphone"
print(s.startswith("smart")) # True
print(s.endswith("phone")) # True
print("another".find("other")) # 2
print("cheat".replace("ch", "m"))
# 'meat'
print(len("Rumpelstiltskin")) # 15
print("ear" in "earth") # True
```





# FUNÇÕES PARA STRINGS

```
pula_linha = '\n'  
print(pula_linha)
```

```
y = "  This is lazy\t\n"  
print(y.strip()) # 'This is lazy'  
print("DrDre".lower()) # 'drdre'  
print("stop".upper()) # 'STOP'
```





# Tipos Complexos de Dados





# LISTAS

```
l = [1, 2, 2]  
print(len(l)) # 3
```





# FUNÇÕES DE LISTAS

```
[1, 2, 2].append(4) # [1, 2, 2, 4]  
[1, 2, 4].insert(2,2) # [1, 2, 2, 4]  
[1, 2, 2] + [4] # [1, 2, 2, 4]
```

```
[1, 2, 2, 4].remove(1) # [2, 2, 4]
```





# LISTAS COMO PILHAS

```
stack = [3]
stack.append(42) # [3, 42]
stack.pop() # 42 (stack: [3])
stack.pop() # 3 (stack: [])
```







# DICIONÁRIOS

```
calories = {'apple' : 52,  
            'banana' : 89,  
            'choco' : 546}
```







# REVERTER E ORGANIZAR

```
[1, 2, 3].reverse() # [3, 2, 1]
```

```
[2, 4, 2].sort() # [2, 2, 4]
```







*Palavras-Chave*







# PALAVRAS-CHAVE/RESERVADAS

False	True	and	or
not	break	continue	class
def	if	elif	else
for	while	in	is
None	lambda	return	





# Ordem de Exemplos







# ORDEM DOS OPERADORES

1º Parênteses ()

2º Exponenciação \*\*

3º Multiplicação e Divisão \* / %

4º Adição e Subtração + -





# ORDEN DOS OPERADORES

Qual será o resultado?

```
operação = (((1+2) ** 2 / 3) - 3) + 5
```



# REFERÊNCIA BIBLIOGRÁFICA:



MAYER, Christian. Coffee Break Python Slicing: 24 Workouts to Master Slicing in Python, Once and for All. Amazon: Amazon, 2018.

