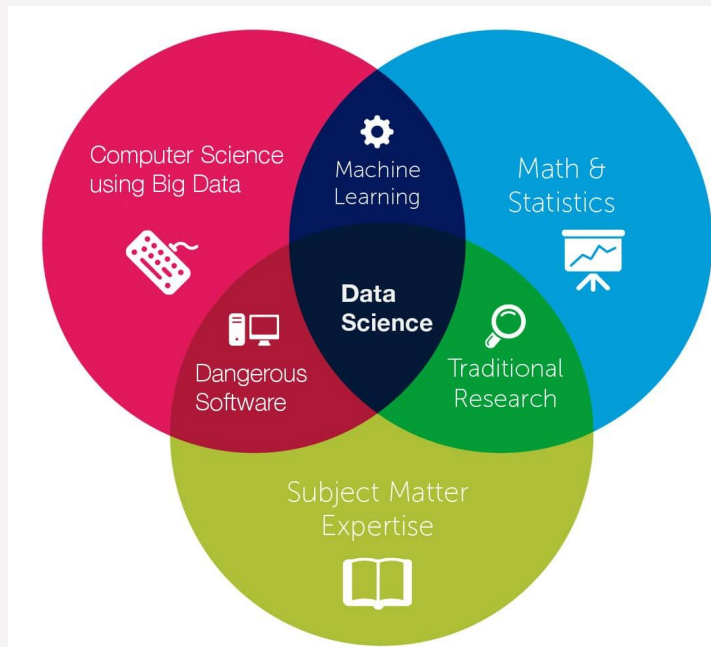
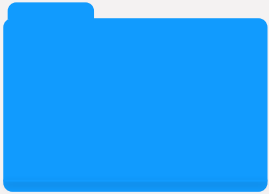


# Ciência de Dados





Visão Geral



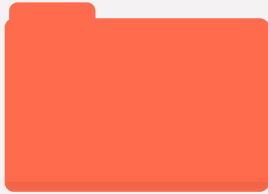
Variáveis



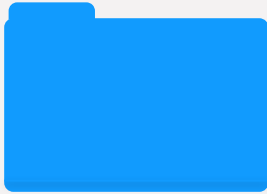
Controle de  
Fluxo



Procedimentos



Entrada &  
Saída



Pandas



Visão Geral



Variáveis



Controle de  
Fluxo



Procedimentos

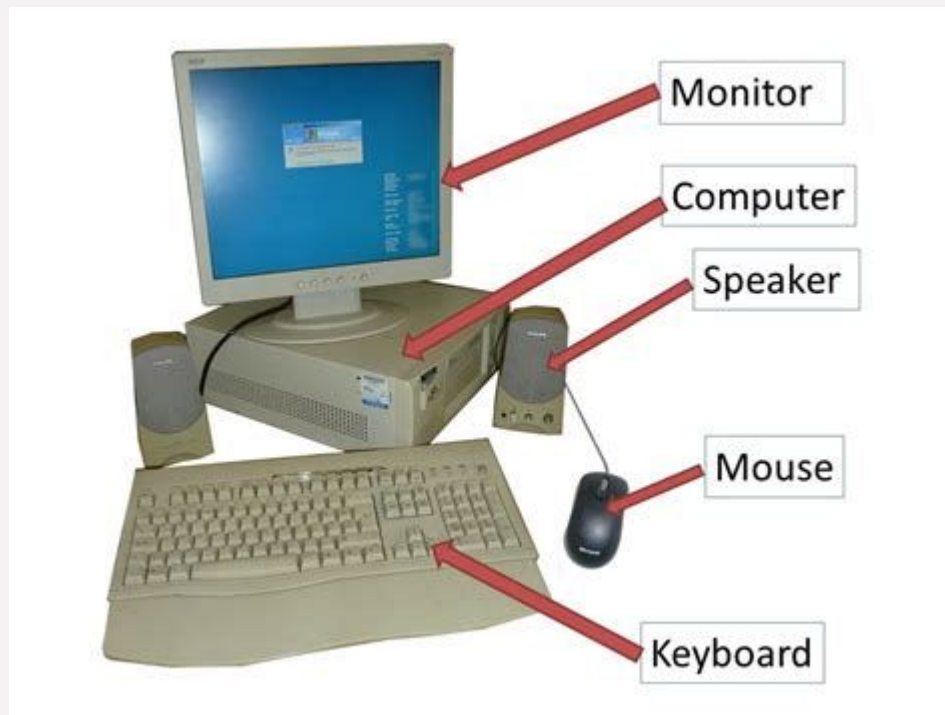


Entrada &  
Saída

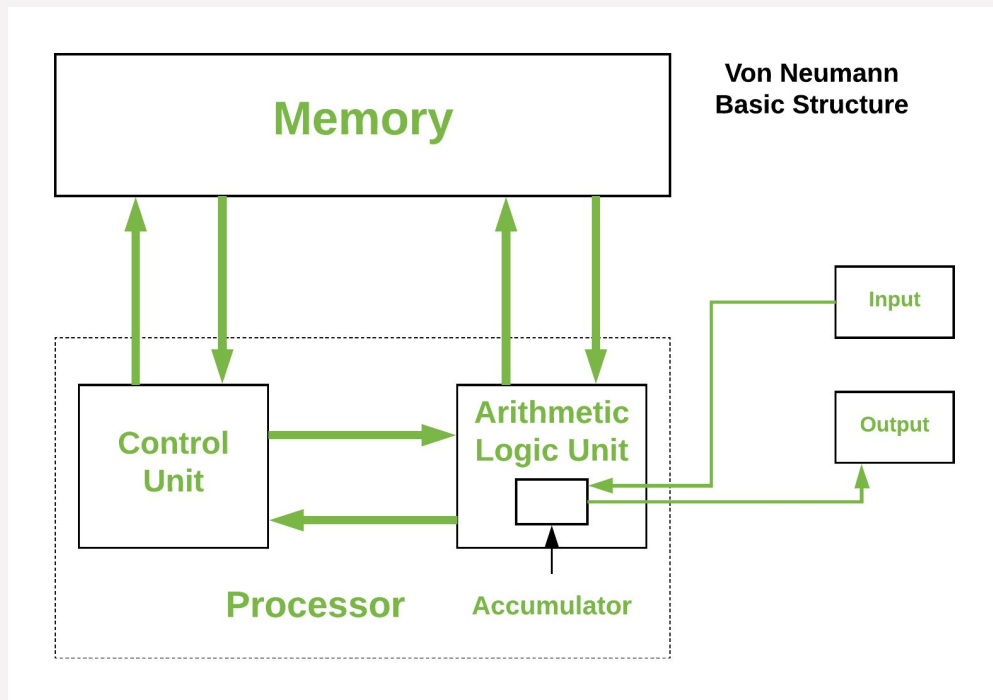


Pandas

# O que é um computador?



# ...por dentro



# Hardware vs Software

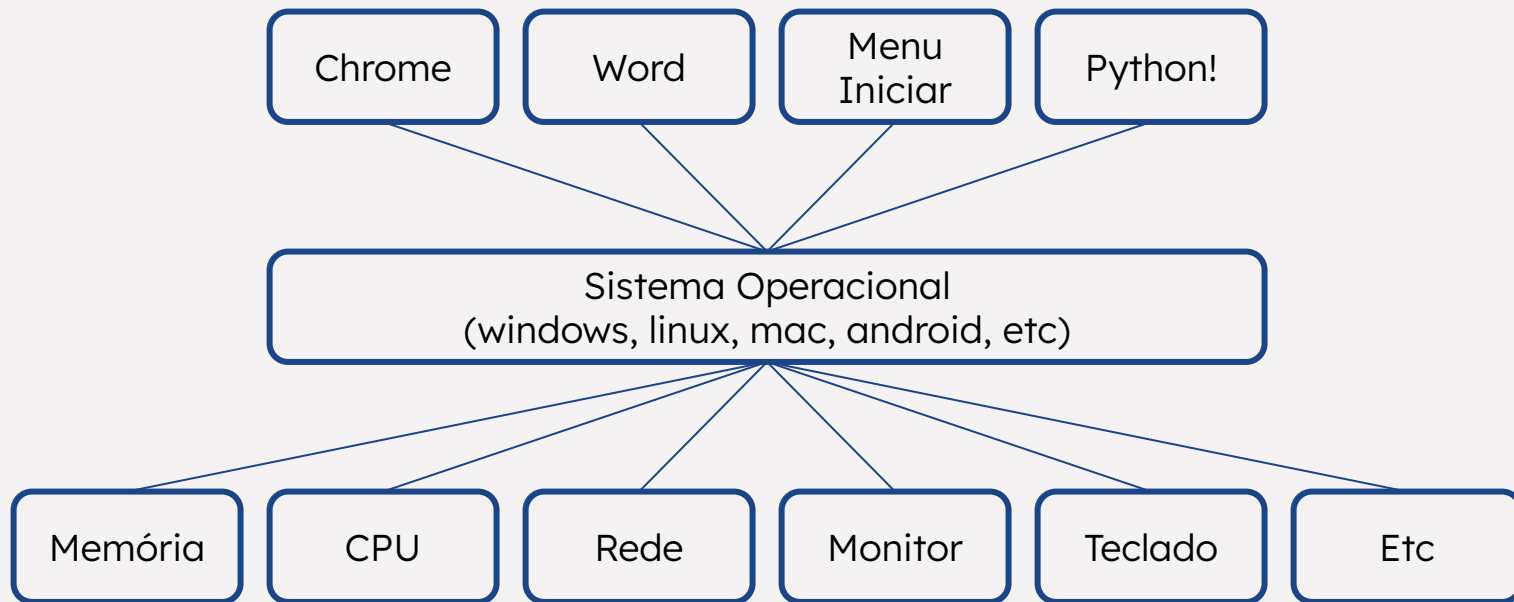


Para saber mais:

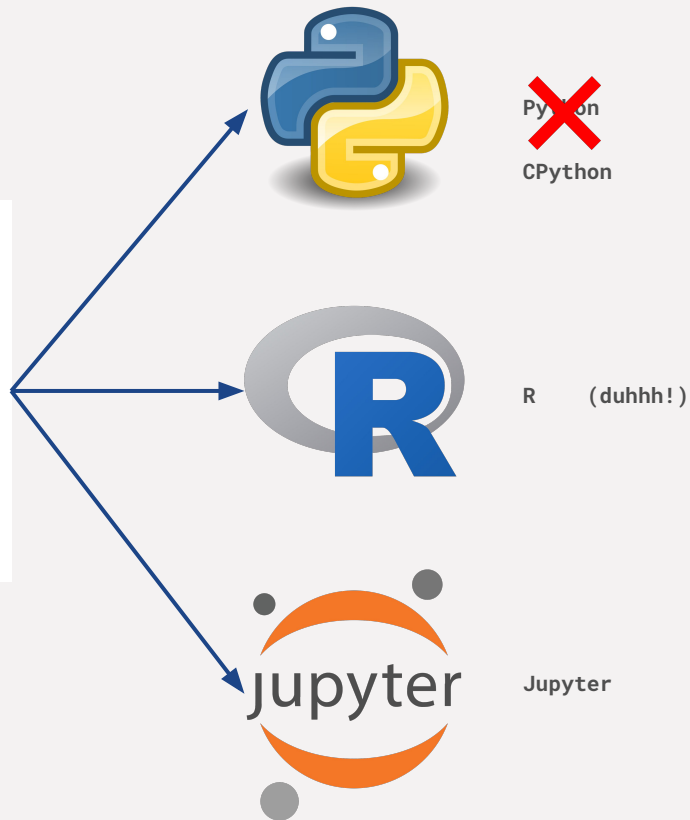
[https://pt.wikipedia.org/wiki/Arquitetura\\_de\\_von\\_Neumann](https://pt.wikipedia.org/wiki/Arquitetura_de_von_Neumann)

From nand to tetris: [https://youtu.be/LqirVc5SIW0?list=PLrDd\\_kMiAuNmSb-CKWQqq9oBFN\\_KNMTal](https://youtu.be/LqirVc5SIW0?list=PLrDd_kMiAuNmSb-CKWQqq9oBFN_KNMTal)

# Camadas dos Programas

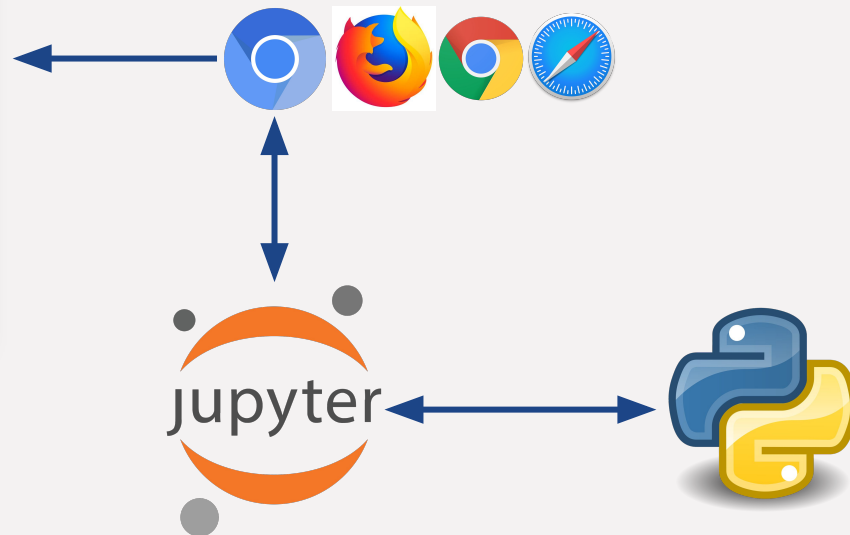
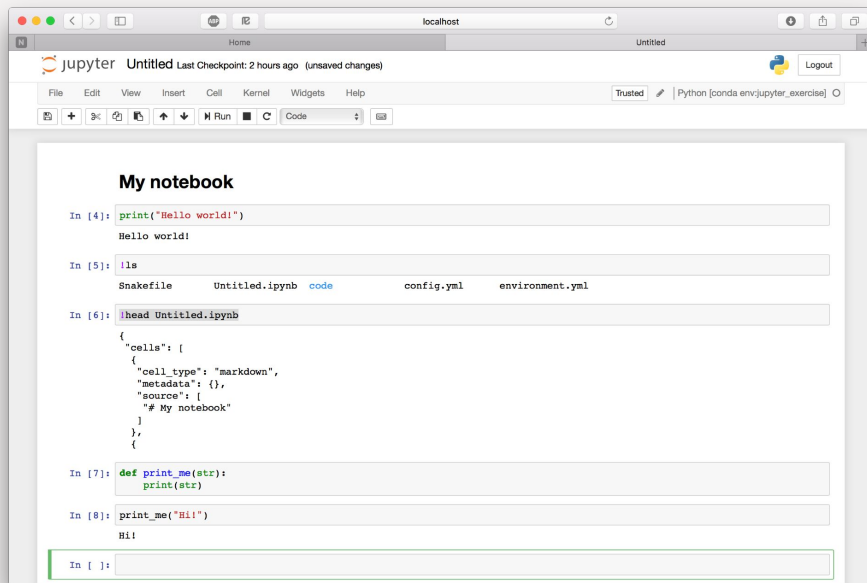


# Instalação



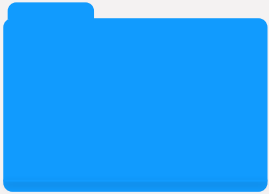


# Arquitetura





Visão Geral



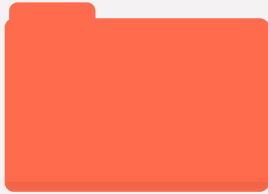
Variáveis



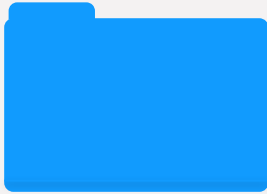
Controle de  
Fluxo



Procedimentos



Entrada &  
Saída



Pandas



Visão Geral



Variáveis



Controle de  
Fluxo



Procedimentos

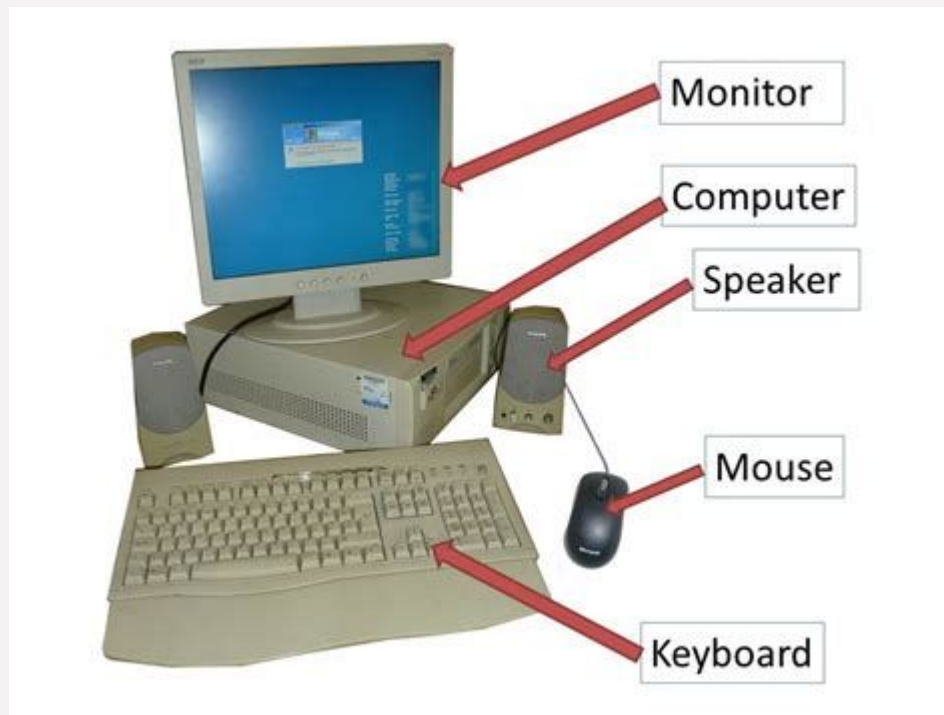


Entrada &  
Saída



Pandas

# De volta ao computador...



# ...por dentro



# Binary Digit

Não

# Binary Digit

Sim

# Binary Digit





# Binary Digit



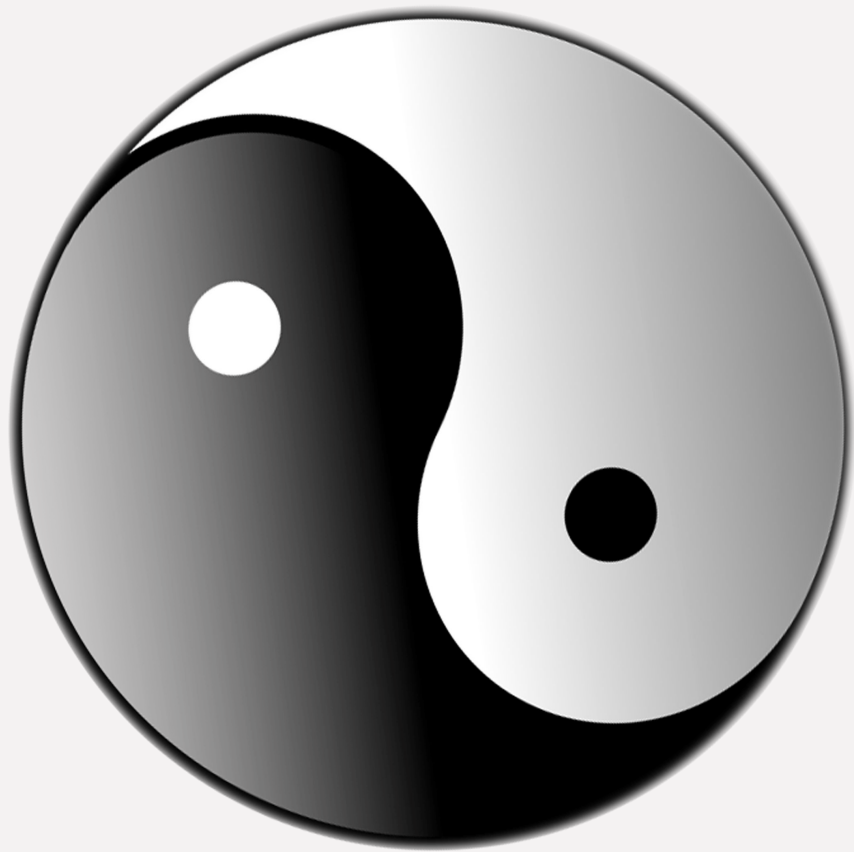
# Binary Digit

0

# Binary Digit

1

# Binary Digit



# Binary Digit

0

1

Binary Digit  
Bi †  
Bit



# Binary Digit

0

# Binary Digit

1

Binary Digit

Textos



Binary Digit

Imagens

Binary Digit

Áudio

# Binary Digit

Vídeos

# Binary Digit

Páginas da web

Binary Digit

Números

# Então só conta até 2?

00000000

0

# Então só conta até 2?

00000001  
1

# Então só conta até 2?

00000010  
2



# Então só conta até 2?

00000011

3

# Então só conta até 2?

00000100

4

# Então só conta até 2?

00000101

5

# Então só conta até 2?

00000110

6

# Então só conta até 2?

00000111

7

# Então só conta até 2?

00001000

8

# Então só conta até 2?

00001001

9

# Então só conta até 2?

00001010  
10



# Então só conta até 2?

00001011

11

# Então só conta até 2?

00001100

12

# Então só conta até 2?

00001101

13

# Então só conta até 2?

00001110

14

# Então só conta até 2?

00001111

15

# Então só conta até 2?

00010000

16

# Então só conta até 2?

...

# E como representa texto?

100 0001	101	65	41	A
100 0010	102	66	42	B
100 0011	103	67	43	C
100 0100	104	68	44	D
100 0101	105	69	45	E
100 0110	106	70	46	F
100 0111	107	71	47	G
100 1000	110	72	48	H
100 1001	111	73	49	I
100 1010	112	74	4A	J
100 1011	113	75	4B	K
100 1100	114	76	4C	L
100 1101	115	77	4D	M
100 1110	116	78	4E	N
100 1111	117	79	4F	O
101 0000	120	80	50	P
101 0001	121	81	51	Q

110 0001	141	97	61	a
110 0010	142	98	62	b
110 0011	143	99	63	c
110 0100	144	100	64	d
110 0101	145	101	65	e
110 0110	146	102	66	f
110 0111	147	103	67	g
110 1000	150	104	68	h
110 1001	151	105	69	i
110 1010	152	106	6A	j
110 1011	153	107	6B	k
110 1100	154	108	6C	l
110 1101	155	109	6D	m
110 1110	156	110	6E	n
110 1111	157	111	6F	o
111 0000	160	112	70	p
111 0001	161	113	71	q



# Curiosidade

11111111

255

# Chega de teoria...

Modo de navegação

↑

↓

<enter>

entra no modo de edição

<ctrl> + <enter>

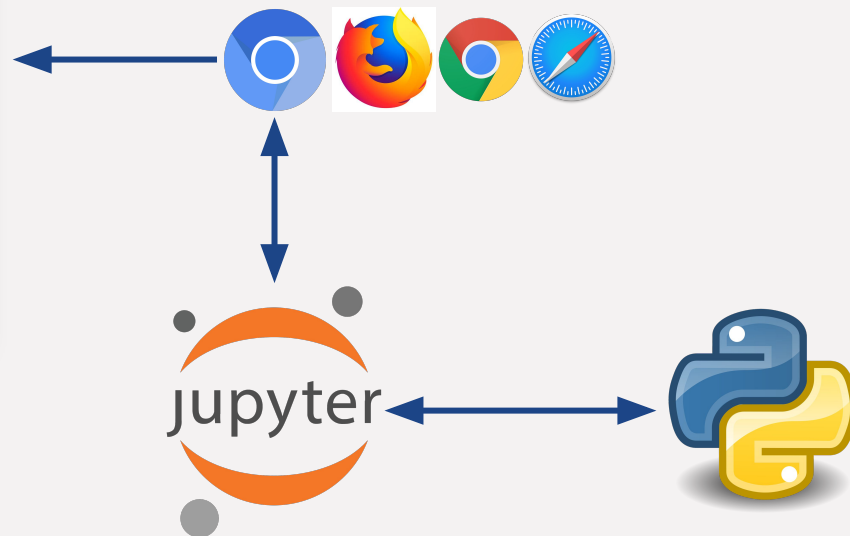
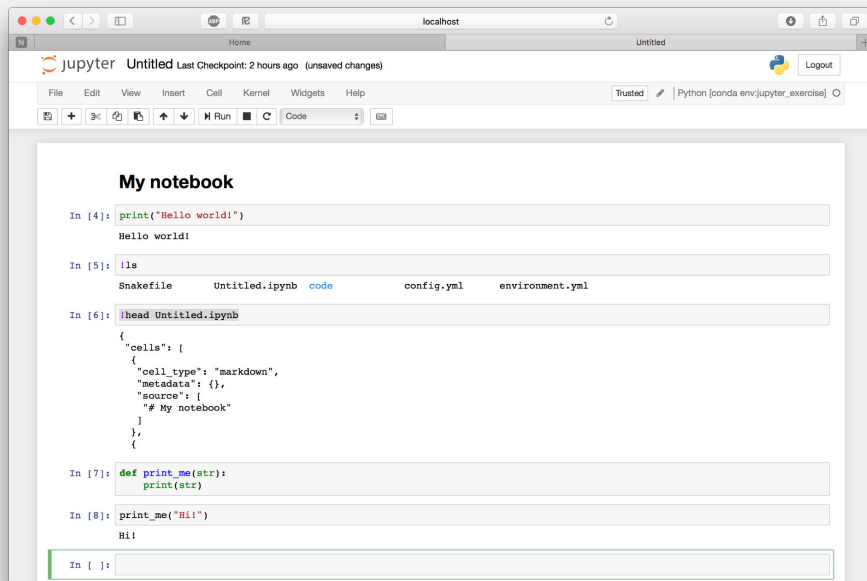
executa a célula

Modo de edição

<esc>

sai do modo de edição

# Arquitetura



# Chega de teoria...

Modo de navegação

↑

↓

<enter> entra no modo de edição

<ctrl> + <enter> executa a célula

Modo de edição

<esc> sai do modo de edição

# Calculadora de IMC

$$\text{IMC} = \text{peso} / (\text{altura} * \text{altura})$$

# Calculadora de IMC 2

edinho: peso=80 kg; altura=1.74m

Rei momo: peso=160 kg; altura=1.80m

Quão maior o imc do rei momo é maior do que o meu?



# Calculadora de IMC 2 - pró

```
imc1 = 80 / (174 * 174)
```

```
imc2 = 160 / (180 * 180)
```

```
imc2 / imc1
```

Nota

= ←

= <=

= :=

= ==



# Voltando pra teoria...

$x = 1$

$y = 2$

$x + y$        $????$

# Voltando pra teoria...

$$x = 1101$$

$$y = 2202$$

$$x * y \qquad \text{????}$$

# Voltando pra teoria...

x = 1000

x = x + 1

x                ????

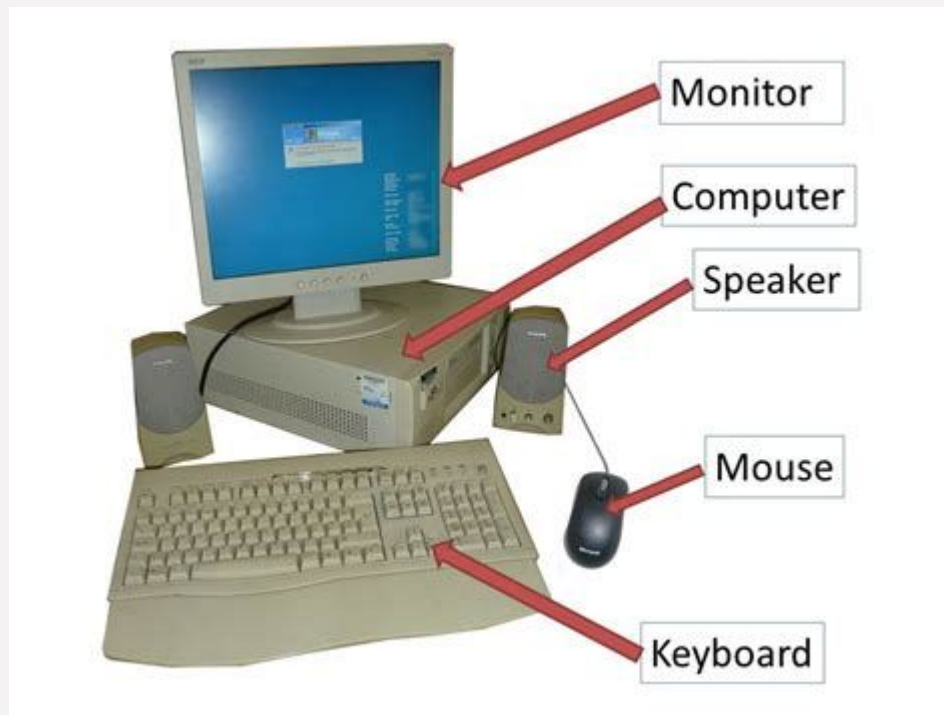
# Voltando pra teoria...

x = 1000

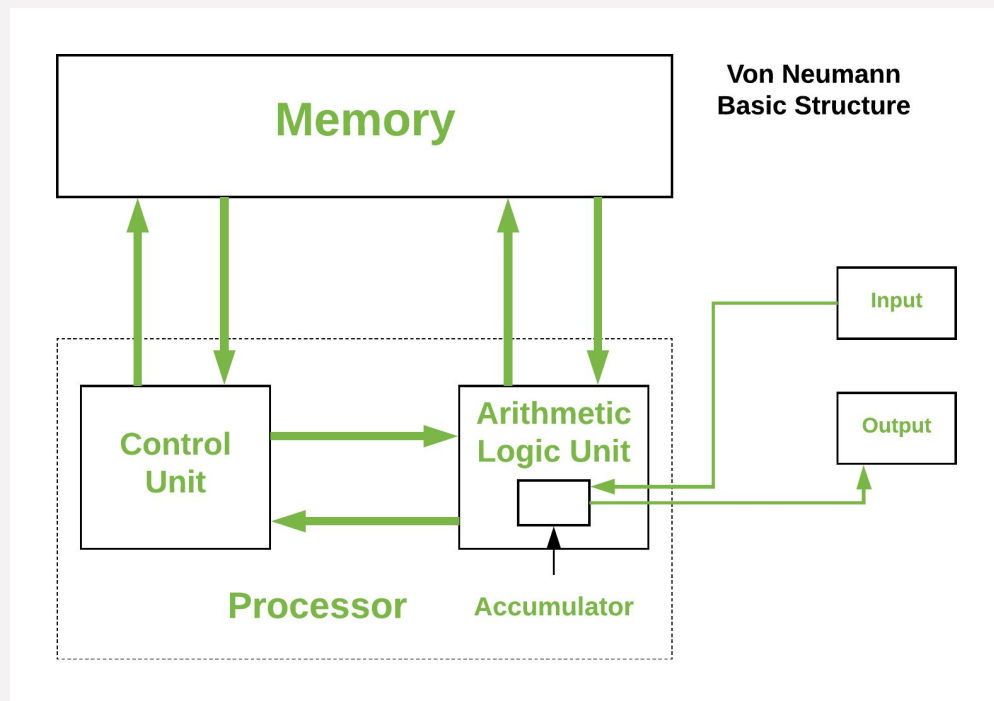
x = x + 1

x                ????

# Ola novamente...



# ...não entendo nada disso



# Voltando pra teoria...

x = 1000

x = x + 1

x

# 3 comandos

x = 1000

x = x + 1

x



# 3 comandos

x = 1000

Comando 1

x = x + 1

Comando 2

x

Comando 3

# Comando 1

```
x = 1000
```



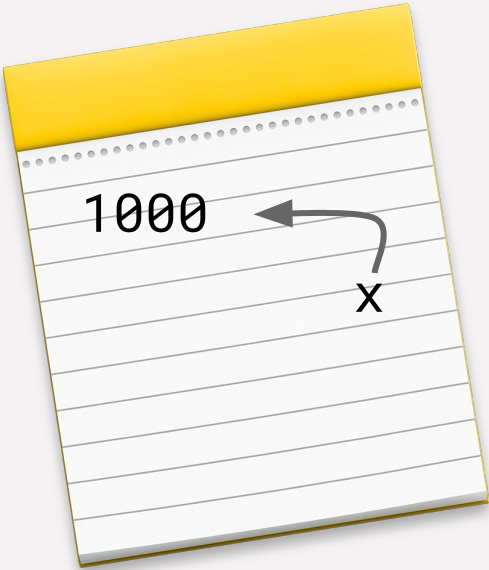
# Comando 1

```
x = 1000
```



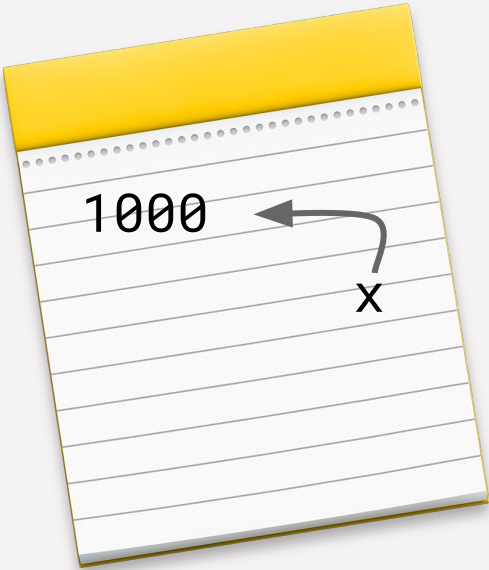
# Comando 1

x = 1000



# Comando 1

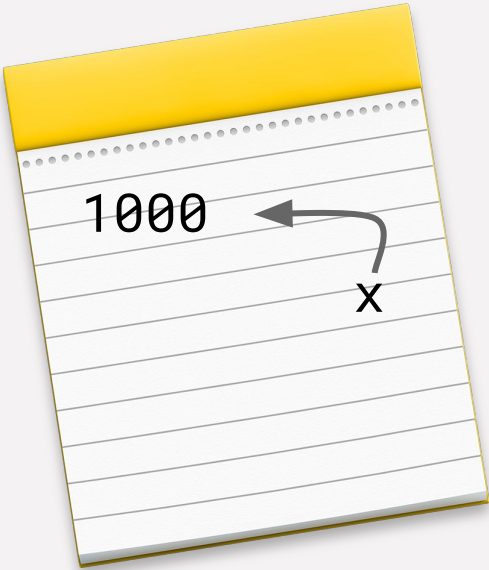
x = 1000



FIM!

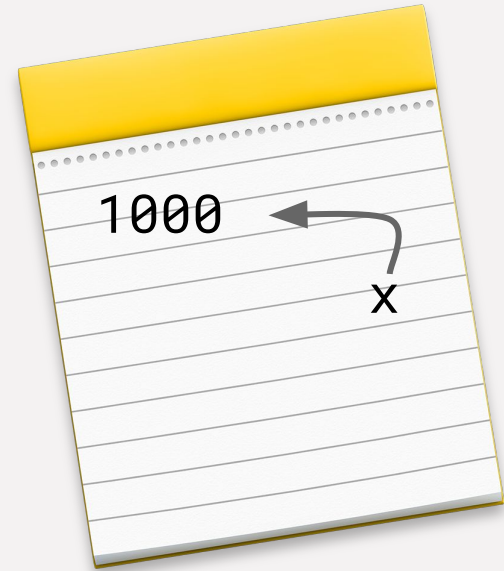
# Comando 2

```
x = x + 1
```



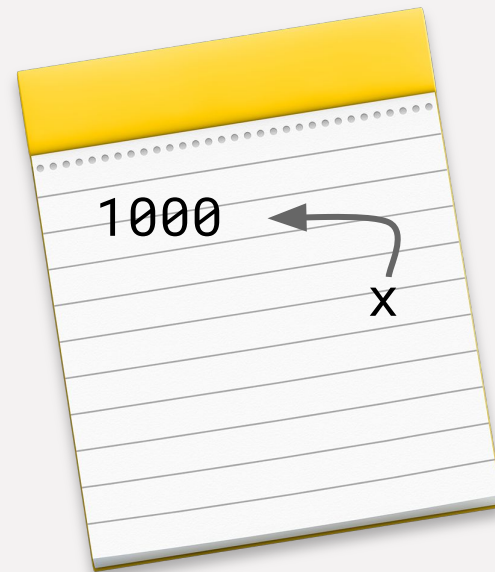
# Comando 2

`x = x + 1`



# Comando 2

$$x = x + 1$$



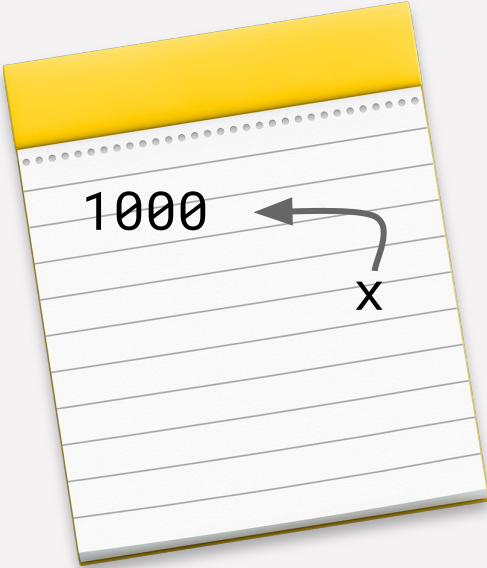


# Comando 2

```
x = x + 1
```



Parte 1

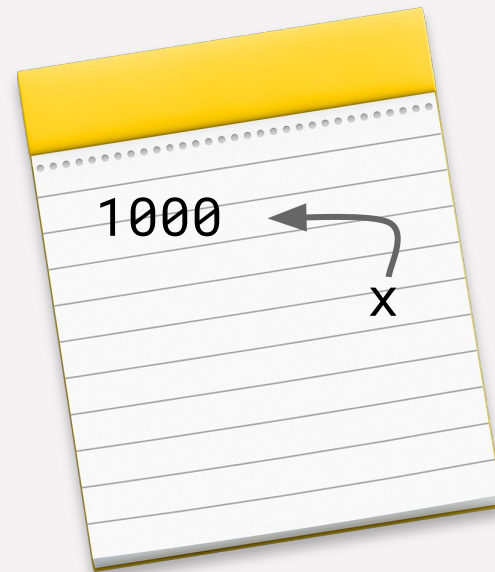


# Comando 2

$x = x + 1$

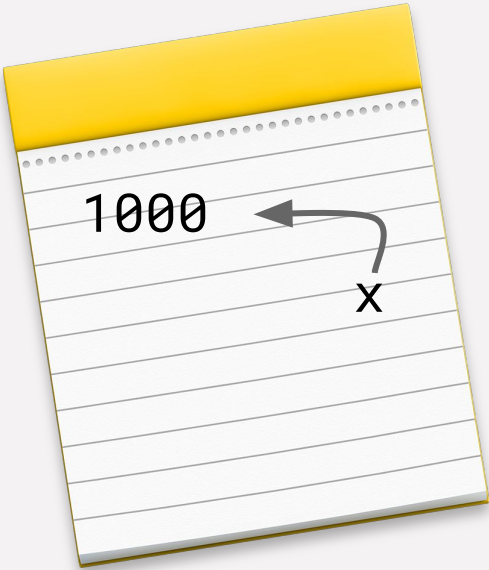
Parte 1

Parte 2



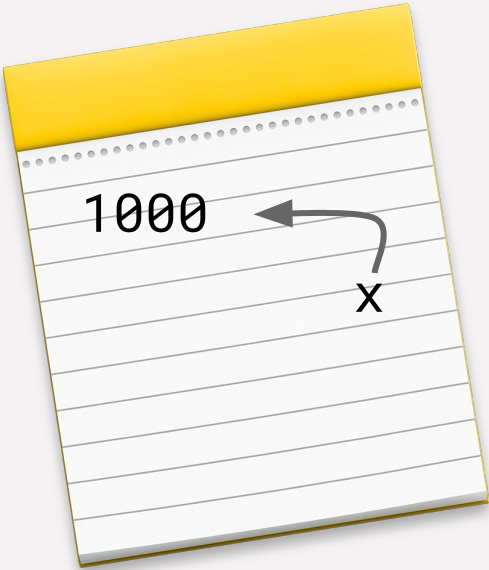
# Comando 2 - parte 1

`x + 1`



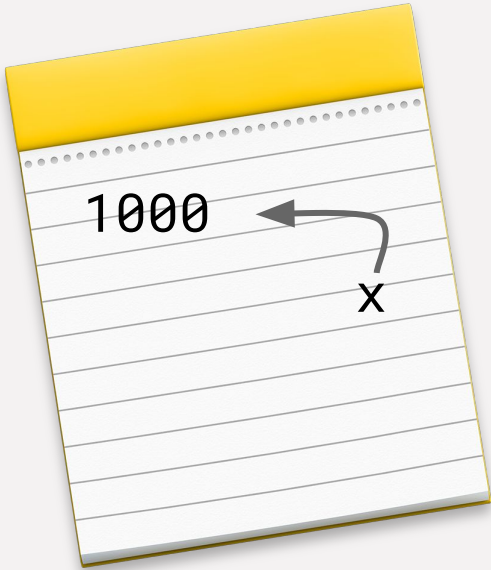
# Comando 2 - parte 1

1000 + 1



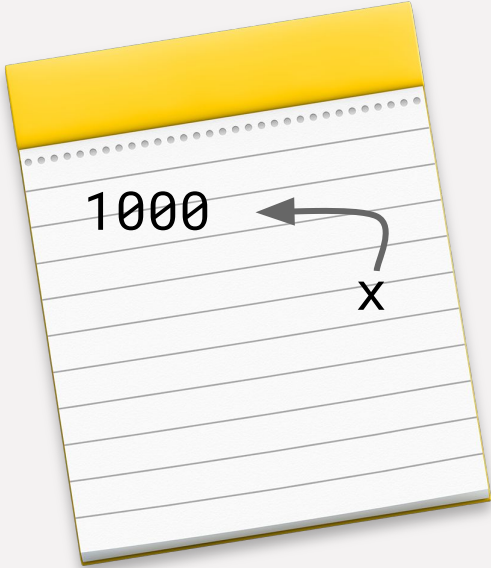
# Comando 2 - parte 1

1001



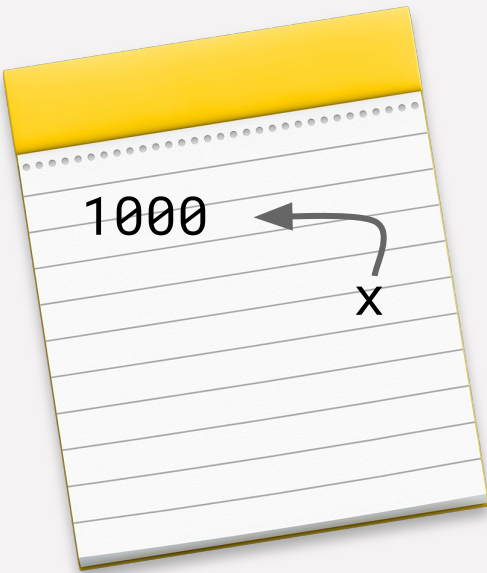
# Comando 2 - parte 1

1001



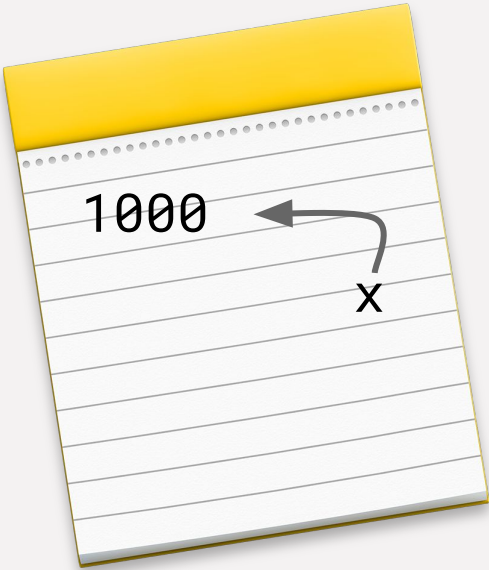
# Comando 2 - parte 2

```
x = x + 1
```



# Comando 2 - parte 2

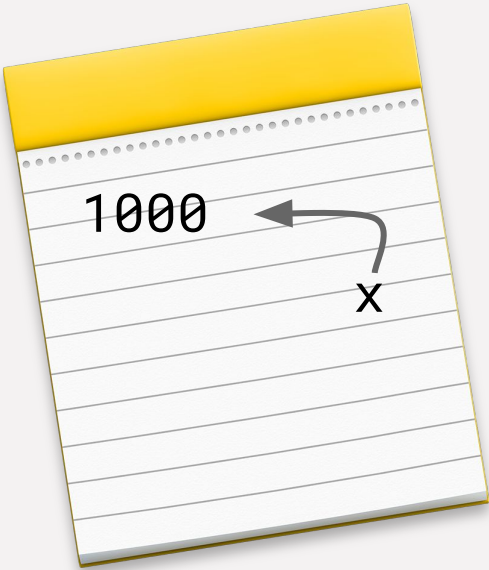
$$x = x + 1$$





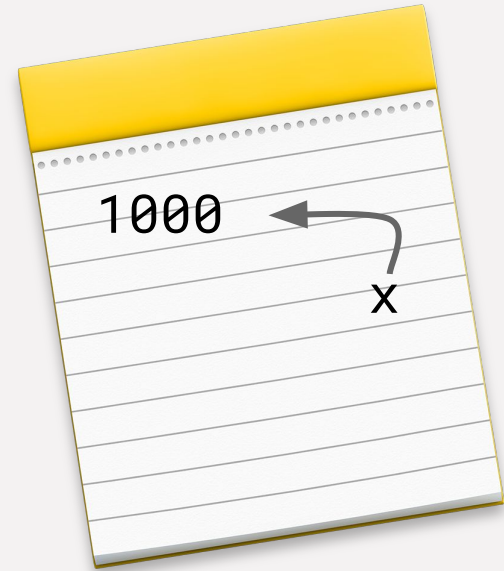
# Comando 2 - parte 2

x = 1001



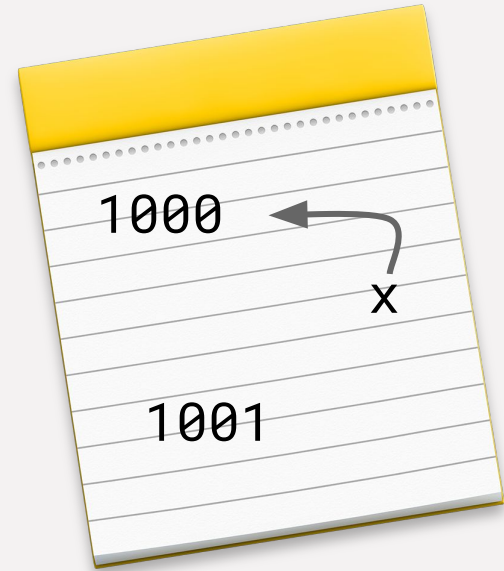
# Comando 2 - parte 2

x = 1001



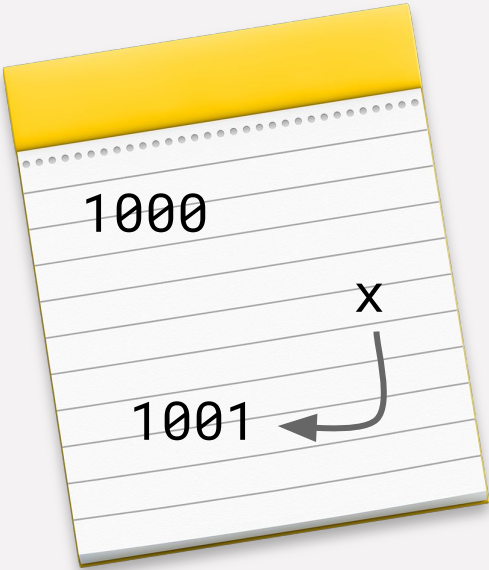
# Comando 2 - parte 2

x = 1001



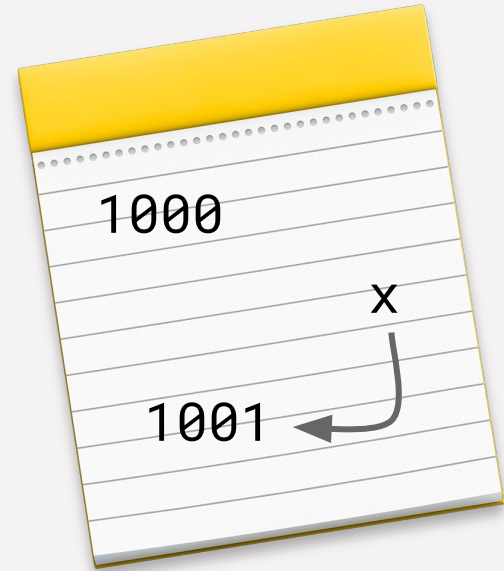
# Comando 2 - parte 2

x = 1001



# Comando 2 - parte 2

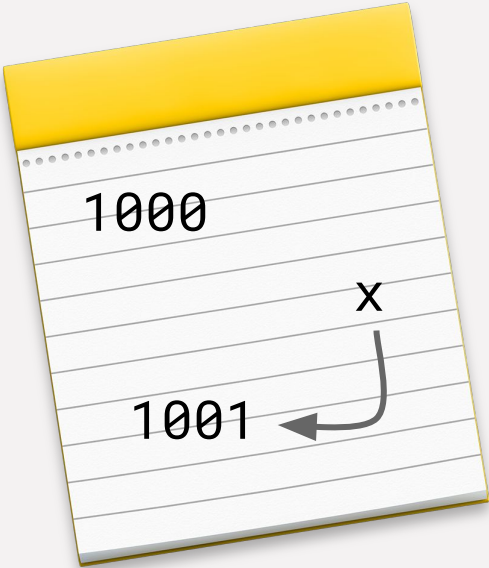
x = 1001



FIM!

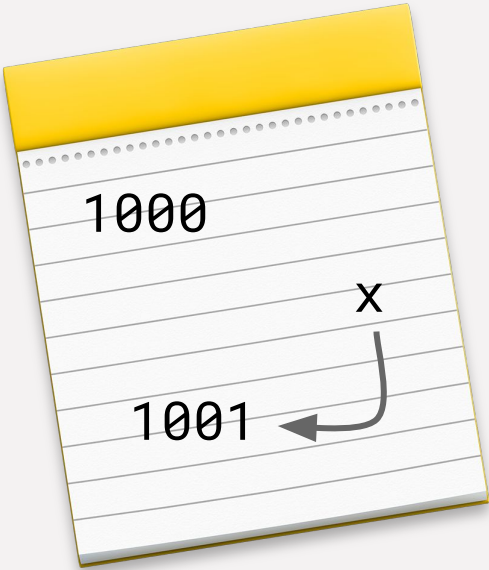
# Comando 3

x



# Comando 3

X



FIM!

# Voltando pra teoria...

x = 1000

x = x + 1

x                ????



# Voltando pra teoria...

$$x = 1$$

$$y = x$$

$$x + y \qquad \text{????}$$

# Voltando pra teoria...

$$x = 10$$

$$y = 2$$

$$z = x + y + 2$$

$$z = z + x + 20$$

Agora uma outra coisa...

# Voltando pra teoria...

x = 1

y = "x"

x + y        ????

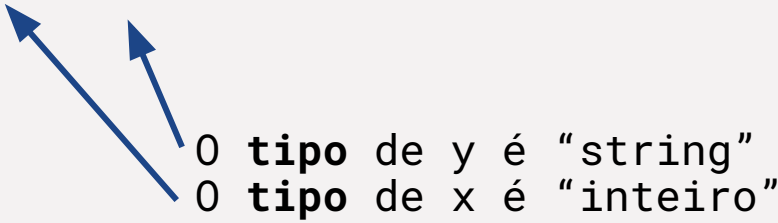
# Voltando pra teoria...

`x = 1`

`y = "a"`

Tudo é representado na memória como 0 ou 1.  
Além do valor da variável, o python também armazena o **tipo** da variável.

`x + y`      `????`



# Tipos das variáveis

```
i = 1  
type(i)
```

```
s = "a"  
type(s)
```

```
f = 1.1  
type(f)
```

# Tipos das variáveis

```
s = "a"
```

```
i = 100
```

```
s * a
```

# Tipos das variáveis

```
s = "nan "  
i = 8  
  
(s * i) + "batman!"
```



# Tipos das variáveis

```
i = 1  
type(i)
```

```
li = [1, 2, 3]  
type(li)
```

```
s = "a"  
type(s)
```

```
f = 1.1  
type(f)
```

# Listas

```
li = [1, 2, 3]
```

```
len(li)
```

```
li[1]
```

```
li[0]
```

```
li[-1]
```

# Listas

```
altura = 1.74
```

Qual o peso para um IMC de 22?

```
pesos = [60, 65, 70, 75, 80, 85, 90]
```

```
pesos[0] / altura * altura
```

```
pesos[1] / altura * altura
```

```
pesos[2] / altura * altura
```

```
pesos[3] / altura * altura
```

```
...
```

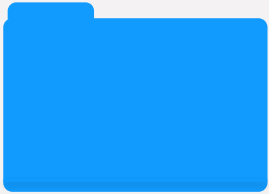
# Listas

```
pesos = [40, 41, 42, 43, 44, ..., 120]
```

```
alturas = [1.40, 1.41, 1.42, ..., 2.10]
```



Visão Geral



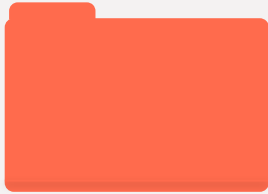
Variáveis



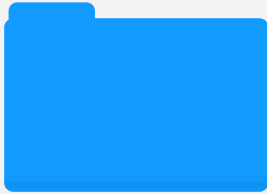
Controle de  
Fluxo



Procedimentos



Entrada &  
Saída



Pandas



Visão Geral



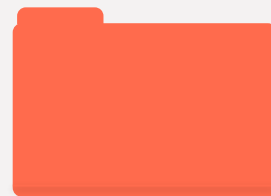
Variáveis



Controle de  
Fluxo



Procedimentos



Entrada &  
Saída

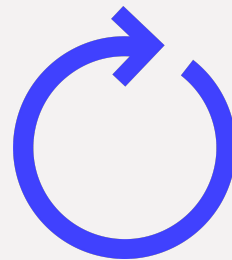


Pandas

# Loops (laços)

```
altura = 1.74  
pesos = [60, 65, 70, 75, 80, 85, 90]
```

```
for peso in pesos:  
    imc = peso / (altura * altura)  
    print(imc)
```



# Loops (laços)

```
altura = 1.74  
pesos = [60, 65, 70, 75, 80, 85, 90]
```

```
for peso in pesos:  
    print(peso)  
    print(altura)  
    imc = peso / (altura * altura)  
    print(imc)
```



# Loops (laços)

```
altura = 1.74  
pesos = [60, 65, 70, 75, 80, 85, 90]
```

```
for peso in pesos:  
    print(peso)  
    print(altura)  
    imc = peso / (altura * altura)  
    print(imc)
```

# Loops (laços)

```
altura = 1.74  
pesos = [60, 65, 70, 75, 80, 85, 90]
```

```
for peso in pesos:  
    print(peso)  
    print(altura)  
    imc = peso / (altura * altura)  
    print(imc)
```

# Loops (laços)

```
altura = 1.74  
pesos = [60, 65, 70, 75, 80, 85, 90]
```

```
for peso in pesos:  
    print(peso)  
    print(altura)  
    imc = peso / (altura * altura)  
    print(imc)
```

# Loops (laços)

```
altura = 1.74  
pesos = [60, 65, 70, 75, 80, 85, 90]
```

```
for peso in pesos:  
    print(peso)  
    print(altura)  
    imc = peso / (altura * altura)  
    print(imc)
```

# Loops (laços)

```
altura = 1.74
pesos = [60, 65, 70, 75, 80, 85, 90]
```

```
for peso in pesos:
    print(peso)
    print(altura)
    imc = peso / (altura * altura)
    print(imc)
```

# Loops (laços)

```
altura = 1.74  
pesos = [60, 65, 70, 75, 80, 85, 90]
```

```
for peso in pesos:  
    print(peso)  
    print(altura)  
    imc = peso / (altura * altura)  
    print(imc)
```

# Loops (laços)

```
altura = 1.74  
pesos = [60, 65, 70, 75, 80, 85, 90]
```

```
for peso in pesos:  
    print(peso)  
    print(altura)  
    imc = peso / (altura * altura)  
    print(imc)
```

# Loops (laços)

```
altura = 1.74  
pesos = [60, 65, 70, 75, 80, 85, 90]
```

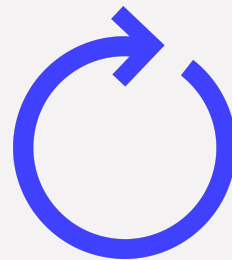
```
for peso in pesos:  
    print(peso)  
    print(altura)  
    imc = peso / (altura * altura)  
    print(imc)
```



# Loops (laços)

```
altura = 1.74  
pesos = [60, 65, 70, 75, 80, 85, 90]
```

```
for peso in pesos:  
    print(peso)  
    print(altura)  
    imc = peso / (altura * altura)  
    print(imc)
```



# Loops (laços) - exercício

Calcule a soma total dos pesos:

```
pesos = [60, 65, 70, 75, 80, 85, 90]
```

# Loops (laços) - exercício 2

Calcule a média dos pesos:

```
pesos = [60, 65, 70, 75, 80, 85, 90]
```

# Loops (laços) - exercício 3

Calcule o desvio padrão dos pesos:

```
pesos = [60, 65, 70, 75, 80, 85, 90]
```

# Loops (laços) - exercício 3

Calcule o desvio padrão dos pesos:

`pesos = [60, 65, 70, 75, 80, 85, 90]`

$$s = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n - 1}}$$

# Loops (laços) - exercício 3

Calcule o desvio padrão dos pesos:

`pesos = [60, 65, 70, 75, 80, 85, 90]`

$$s = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n - 1}}$$

```
média = ...  
numerador = (p1 - média) ** 2 + (p2 - média) ** 2 +  
denominador = n - 1  
desvio_padrao = (numerador / denominador) ** (0.5)
```