

**01 flertando com python**



**O prompt  
interativo esse  
ilustre  
desconhecido**

# Chamando

```
$ python
```

# Temos

```
$ python
Python 2.6.5 (r265:79063)
[GCC 4.4.3] on linux2
Type "help", "copyright", "credits"
or "license" for more information.
>>>
```

# Seu melhor amigo

```
>>> a = 5
```

```
>>> print a
```

```
5
```

```
>>> b = 4
```

```
>>> print a + b
```

```
9
```

# Hello world

```
>>> from datetime import datetime
>>> from time import sleep
>>> while True: #rodar para sempre
...     hora = datetime.now()
...     print hora.strftime("%H:%M:%S")
...     sleep(1) #aguardar um segundo
... 
```

# Por partes

**True** e **False** com a primeira letra em Maiuscula

```
>>> while True: #rodar para sempre
        ^ marca o inicio do bloco
...     hora = datetime.now()
...     print hora.strftime("%H:%M:%S")
...     sleep(1) #aguardar um segundo
    ^^^^ a identacao tem que ser constante
```

padrão são 4 espaços, mas voce tem liberdade de escolher desde que seja consistente

# Blocos

**if / elif  / else**

**for / else**

**while / else**

**try / except / finally**

**class / def**



# Sequências

## Principais classes

- string e unicode
- listas e tuplas

# Implica

```
len(s)      # length
min(s)      # ou max(s)
s[i]        # iesimo item de s (base 0)
s[i:j]      # slice do iesimo a jesimo item
s[i:j:k]    # mesma coisa com passo k
s + t       # concatenação
s * i       # ou i * s; i copias de s
x in s      # se s tem o elemento x
x not in s  # ou nao tem
```

# Strings

- Imutaveis 🍌
- suportam operações de sequência
- demarcadas com ' ou " 🍌 🍖
- multilinhas com ''' ou ""''

# Strings como sequências

```
a = "abcdef"
len(a)      # 6
min(a)      # 'a'
a[2]        # 'c'
a[2:5]      # 'cde'
a[2:5:2]    # 'ce'
a + "zx"    # 'abcdefzx'
a * 2       # 'abcdefabcdef'
"b" in a    # True
"b" not in a # False
```

# Strings como strings

```
a = ' abcd \n '  
a.upper() # ' ABCD \n '  
a.strip() # 'abcd'  
a.islower() # True  
# isalnum/isalpha/isdigit  
# islower/isspace/istitle/isupper  
a.startswith(" ") # True  
# endswith  
a.find("abc") # 2  
a.split("b") # [' a', 'cd \n ']
```