



Python para Aplicações em Eletrônica

Aula 01



Conceitos Básicos

Características Básicas

- Linguagem Interpretada
- Permite interação com o interpretador (teste e *debugging*)
- Permite programação orientada a objetos
- Linguagem de fácil aprendizagem e utilização:
 - *Easy-to-learn*
 - *Easy-to-read*
 - *Easy-to-maintain*
- Possui *Garbage Collection*
- *Portable e scalable (Será??)*

Execução do Código

- Execução de arquivo .py

```
ex01.py (~Dropbox/Ensino/Minicursos/Python/Aula01/Codigos)
Arquivo  Editar  Ver  Pesquisar  Ferramentas  Documentos  Ajuda

print "\n\nall work and no play makes jack a dull boy!!!"

a=1+20

# Iniciando um loop (Isso eh um comentario)
for i in range(1,10):
    print i, " "
    print "0_o "
# Fim do loop
print a
```

```
gilmar@poseidon ~/Dropbox/Ensino/Minicursos/Python/Aula01/Codigos
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda

gilmar@poseidon ~/Dropbox/Ensino/Minicursos/Python/Aula01/Codigos $ python ex01.py

all work and no play makes jack a dull boy!!!
1
0_o
2
0_o
3
0_o
4
0_o
5
0_o
6
0_o
7
0_o
8
0_o
9
0_o
21
gilmar@poseidon ~/Dropbox/Ensino/Minicursos/Python/Aula01/Codigos $
```

Execução do Código

- Execução interativa

```
gilmar@poseidon ~  
Arquivo Editar Ver Pesquisar Terminal Ajuda  
gilmar@poseidon ~ $ python  
Python 2.7.12 (default, Dec 4 2017, 14:50:18)  
[GCC 5.4.0 20160609] on linux2  
Type "help", "copyright", "credits" or "license" for more information.  
>>> █
```

```
gilmar@poseidon ~  
Arquivo Editar Ver Pesquisar Terminal Ajuda  
gilmar@poseidon ~ $ python  
Python 2.7.12 (default, Dec 4 2017, 14:50:18)  
[GCC 5.4.0 20160609] on linux2  
Type "help", "copyright", "credits" or "license" for more information.  
>>> print "\n\nall work and no play makes jack a dull boy!!!"  
  
all work and no play makes jack a dull boy!!!  
>>> a=1+20  
>>> █
```

```
gilmar@poseidon ~  
Arquivo Editar Ver Pesquisar Terminal Ajuda  
gilmar@poseidon ~ $ python  
Python 2.7.12 (default, Dec 4 2017, 14:50:18)  
[GCC 5.4.0 20160609] on linux2  
Type "help", "copyright", "credits" or "license" for more information.  
>>> print "\n\nall work and no play makes jack a dull boy!!!"  
  
all work and no play makes jack a dull boy!!!  
>>> █
```

```
gilmar@poseidon ~  
Arquivo Editar Ver Pesquisar Terminal Ajuda  
>>> a=1+20  
>>> # Iniciando um loop (Isso eh um comentario)  
... for i in range(1,10):  
...     print i, " "  
...     print " 0_o "  
... # Fim do loop  
...  
1 0_o  
2 0_o  
3 0_o  
4 0_o  
5 0_o  
6 0_o  
7 0_o  
8 0_o  
9 0_o  
>>> print a █
```

Execução do Código

- “Comentário” `#!/usr/bin/python`

```
ex01_b.py (~Dropbox/Ensino/Minicursos/Python/Aula01/Codigos) - + x
Arquivo Editar Ver Pesquisar Ferramentas Documentos Ajuda
[Icons]
#!/usr/bin/python
print "\n\nall work and no play makes jack a dull boy!!!"

a=1+20

# Iniciando um loop (Isso eh um comentario)
for i in range(1,10):
    print i, " "
    print " 0_o "
# Fim do loop

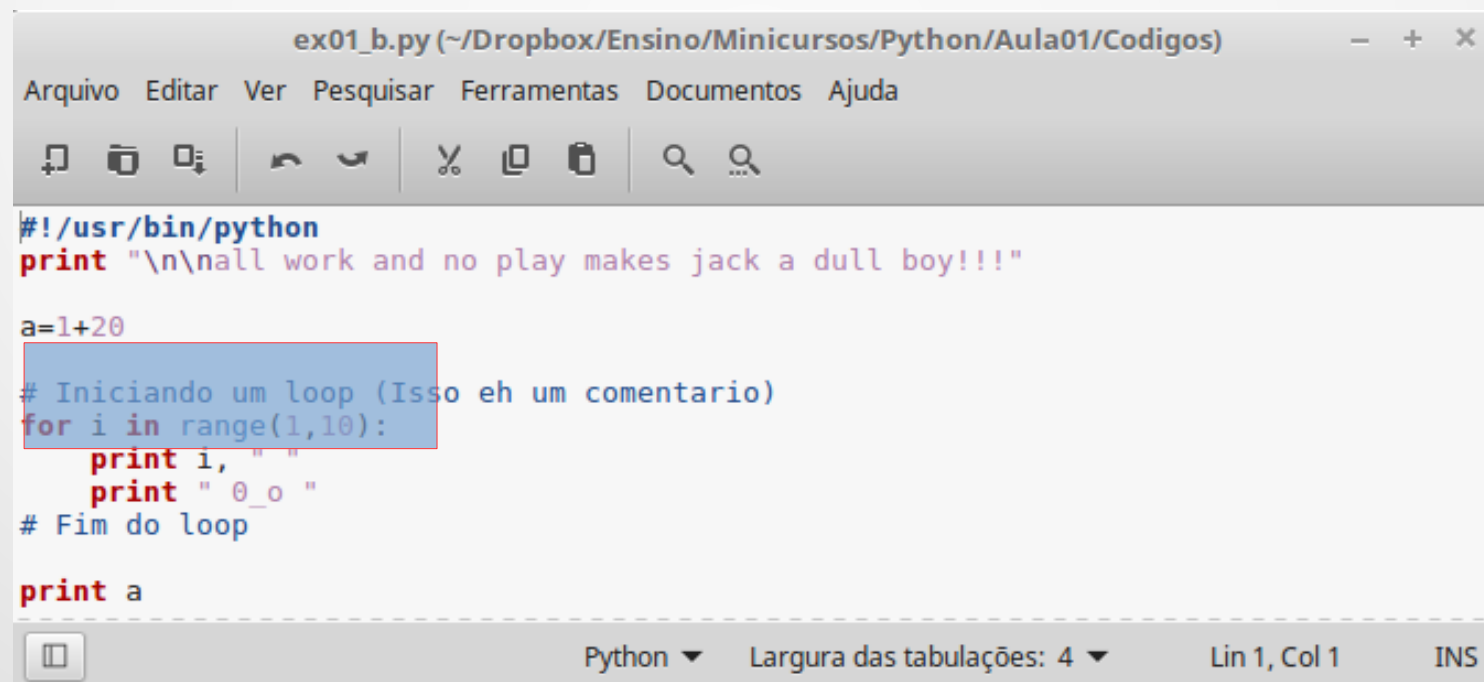
print a
```

```
gilmar@poseidon ~/Dropbox/Ensino/Minicursos/Python/Aula01/Codigos - + x
Arquivo Editar Ver Pesquisar Terminal Ajuda
gilmar@poseidon ~/Dropbox/Ensino/Minicursos/Python/Aula01/Codigos $ ./ex01.py
Unescaped left brace in regex is deprecated, passed through in regex; marked by
<-- HERE in m/%{ <-- HERE (.*)}/ at /usr/bin/print line 528.
Error: no such file "\n\nall work and no play makes jack a dull boy!!!"
./ex01.py: line 6: erro de sintaxe próximo do `token' não esperado `('
./ex01.py: line 6: `for i in range(1,10):'
gilmar@poseidon ~/Dropbox/Ensino/Minicursos/Python/Aula01/Codigos $ ./ex01_b.py

all work and no play makes jack a dull boy!!!
1
 0_o
2
 0_o
3
 0_o
4
 0_o
5
 0_o
6
 0_o
7
```

Elementos Básicos de Sintaxe

- Linhas e Identação:
 - Não é necessário usar “;” para determinar o fim de uma instrução
 - Linhas em branco são ignoradas
 - Utiliza-se um “tab” para definir pertinencia de código (uso do “:”)



```
ex01_b.py (~/.Dropbox/Ensino/Minicursos/Python/Aula01/Codigos)
Arquivo  Editar  Ver  Pesquisar  Ferramentas  Documentos  Ajuda

#!/usr/bin/python
print "\n\nall work and no play makes jack a dull boy!!!"

a=1+20
# Iniciando um loop (Isso eh um comentario)
for i in range(1,10):
    print i, " "
    print " 0_o "
# Fim do loop

print a
```

Python ▾ Largura das tabulações: 4 ▾ Lin 1, Col 1 INS

Elementos Básicos de Sintaxe

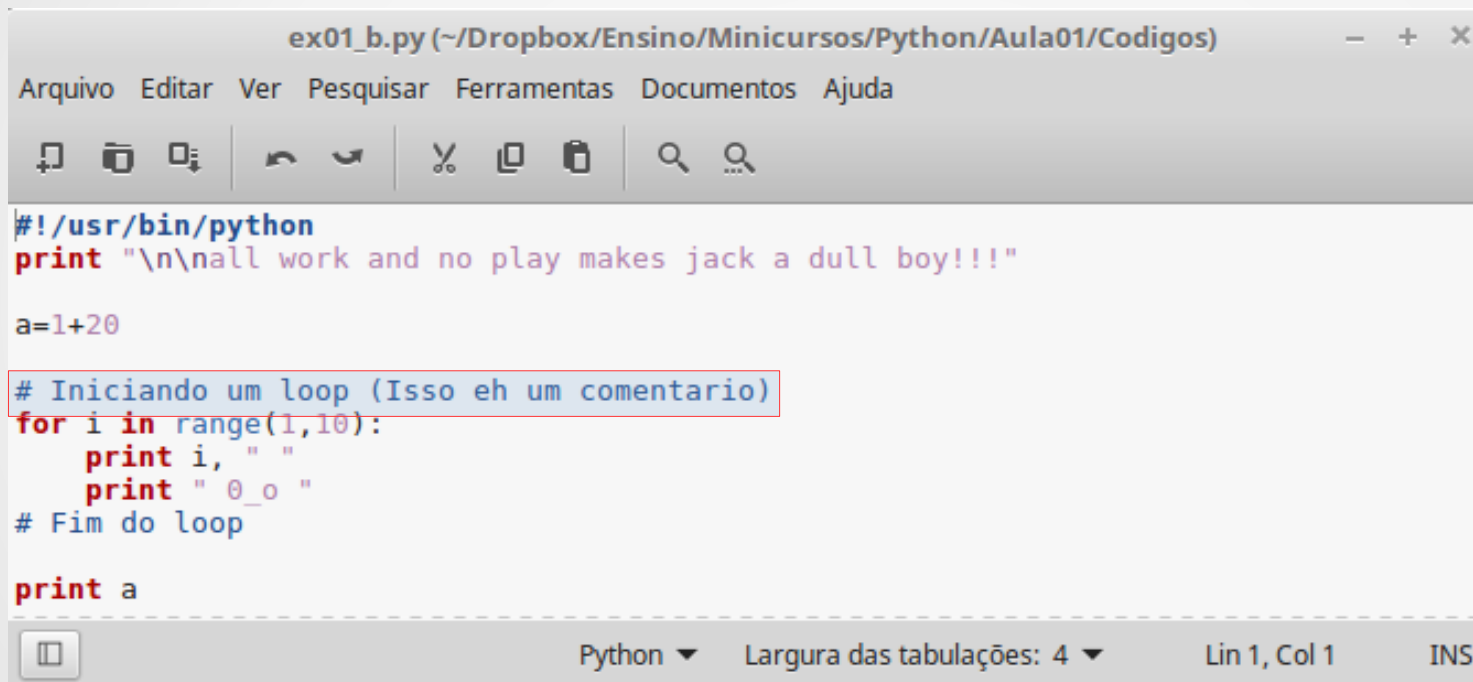
- Instrução em múltiplas linha

```
ex01_b.py (~/.Dropbox/Ensino/Minicursos/Python/Aula01/Codigos) - + x
Arquivo Editar Ver Pesquisar Ferramentas Documentos Ajuda
[Icons]
#!/usr/bin/python
a=1+20\
+10\
    +15\
+21
print "Valor de ", a
```

```
gilmar@poseidon ~/.Dropbox/Ensino/Minicursos/Python/Aula01/Codigos - + x
Arquivo Editar Ver Pesquisar Terminal Ajuda
gilmar@poseidon ~/.Dropbox/Ensino/Minicursos/Python/Aula01/Codigos $ ./ex01_b.py
Valor de 67
```


Elementos Básicos de Sintaxe

- Comentários (#)



The screenshot shows a Python IDE window titled "ex01_b.py (~/.Dropbox/Ensino/Minicursos/Python/Aula01/Codigos)". The menu bar includes "Arquivo", "Editar", "Ver", "Pesquisar", "Ferramentas", "Documentos", and "Ajuda". The toolbar contains icons for file operations and search. The code editor displays the following Python code:

```
#!/usr/bin/python
print "\n\nall work and no play makes jack a dull boy!!!"

a=1+20

# Iniciando um loop (Isso eh um comentario)
for i in range(1,10):
    print i, " "
    print " 0_o "
# Fim do loop

print a
```

The status bar at the bottom indicates "Python", "Largura das tabulações: 4", "Lin 1, Col 1", and "INS".

Elementos Básicos de Sintaxe

- Definição de *strings*
 - Podem ser utilizadas ‘ ‘, ou “ “
 - Não tem diferença



Variáveis e Operadores

Variáveis

- Tipos Básicos
 - Números
 - *Strings*
 - Listas
 - Tuplas
 - Dicionários

Variáveis

```
1 #!/usr/bin/python
2 n = 1
3 f = 74.34
4 nome = 'A casa de Jose'
5 print n
6 print f
7 print nome
8 print nome[2:4]
9
10 lista = ['a', 'banana', 2.23, 'casa', 4]
11 tupla = ('a', 'banana', 2.23, 'casa', 4)
12
13 print lista
14 print lista[1:3]
15
16 print tupla
17 print tupla[1:3]
18
19 lista[1] = 7
20 print lista
21
22 tupla[1] = 7
23 print tupla
24
```

```
gilmar@poseidon ~/Dropbox/Ensino/Minicursos/Python/Aula01/Codigos
Arquivo Editar Ver Pesquisar Terminal Ajuda
gilmar@poseidon ~/Dropbox/Ensino/Minicursos/Python/Aula01/Codigos $ python ex03.py
1
74.34
A casa de Jose
ca
['a', 'banana', 2.23, 'casa', 4]
['banana', 2.23]
('a', 'banana', 2.23, 'casa', 4)
('banana', 2.23)
['a', 7, 2.23, 'casa', 4]
Traceback (most recent call last):
  File "ex03.py", line 22, in <module>
    tupla[1] = 7
TypeError: 'tuple' object does not support item assignment
gilmar@poseidon ~/Dropbox/Ensino/Minicursos/Python/Aula01/Codigos $
```

Operadores Básicos

- Algébricos
 - + (soma), - (subtração), * (multiplicação), / (divisão)
 - % (resto), ** (exponencial), // (*floor division*)
- Atribuição
 - = , +=, -=, *=, /=, **=, //=

Operadores Básicos

```
1 #!/usr/bin/python
2 a = 7; b=3
3
4 print "a+b:", (a+b)
5 print "a-b:", (a-b)
6 print "a*b:", (a*b)
7 print "a/b:", (a/b)
8 print "a**b:", (a**b) #a^b
9 print "a//b:", (a//b)
10
11 c = a
12 c = c+4
13
14 d=a
15 d+=4
16 print "c:", c
17 print "d:", d
18
19 d**=2
20 print "d**=2:", d
```

```
a+b: 10
a-b: 4
a*b: 21
a/b: 2
a**b: 343
a//b: 2
c: 11
d: 11
d**=2: 121
```

Operadores Básicos

- Comparação
 - == (igualdade), != ou <> (não igualdade)
 - >, >=, <, <=
- Pertinência
 - **in** e **not in**

Operadores Básicos

```
1 #!/usr/bin/python
2 print "3==4:", (3==4)
3 print "3<>4:", (3<>4)
4 print "3!=4:", (3!=4)
5
6 print "casa==casa:", ('casa'=='casa')
7 a='casa'; b='casa'; c='cas'; d='dado'; e='Dado'
8 print "a==b:", (a==b)
9
10
11 print "3>4:", (3>4)
12 print "3>=4:", (3>=4)
13 print "3<4:", (3<4)
14 print "3<=4:", (3<=4)
15 print "a>=c:", (a>=c)
16 print "c>=a:", (c>=a)
17 print "c>d:", (c>d)
18 print "c>e:", (c>e)
19
20 a = [3, 'casa', 3.75, 27]
21 print "a:", a
22 print "3 in a", (3 in a)
23 print "4 in a", (4 in a)
24 print "casa in a", ('casa' in a)
25 print "cas in a", ('cas' in a)
26
27 pa = [3, 27]
28 print "pa in a", (pa in a)
```

```
3==4: False
3<>4: True
3!=4: True
casa==casa: True
a==b: True
3>4: False
3>=4: False
3<4: True
3<=4: True
a>=c: True
c>=a: False
c>d: False
c>e: True
a: [3, 'casa', 3.75, 27]
3 in a True
4 in a False
casa in a True
cas in a False
pa in a False
```



Estruturas de Controle



Estruturas de Controle

Estruturas de Decisão

- If (condição) :

```
1 #!/usr/bin/python
2 a=1; b=7; c=15
3
4 if (a>=b):
5     print "a>=b"
6 elif (b<c):
7     print "a<b<c"
8 else:
9     print "a<b e b>=c"
10
11 lista=[1,2,3,4,5]
12
13 print "lista:", lista
14 if (a in lista): print "a estah na lista"
15 else: print "a nao estah na lista"
16
17 if (b in lista): print "b estah na lista"
18 else: print "b nao estah na lista"
```

```
a<b<c          2 a=1; b=7;
lista: [1, 2, 3, 4, 5]
a estah na lista (a>=b)
b nao estah na lista
```

Estruturas de Decisão

- Múltiplas condições: **and** e **or**

```
20 if (b in lista) or (a in lista): print "a ou b estao na lista"
21 elif (b in lista) and (a in lista): print "a e b estao na lista"
22
23 if (b not in lista) and (c not in lista): print "nem b e nem c estao na lista"
```

```
a ou b estao na lista
nem b e nem c estao na lista
```

Estruturas de Repetição

- While (condição):

```
1 #!/usr/bin/python
2
3 i=0
4 while (i<10):
5     print "i:", i
6     i+=2;
7
8 k=0
9 while (True):
10     print "k:", k
11     k+=3
12     if (k>=17):
13         print "k>17\nsaindo..."
14         break
15
16 print "Saiu com k:",k
```

```
i: 0
i: 2
i: 4
i: 6
i: 8
k: 0
k: 3
k: 6
k: 9
k: 12
k: 15
k>17
saindo..
Saiu com k: 18
```

Estruturas de Repetição

- *for it_var in sequence:*

```
1 #!/usr/bin/python
2 seq=range(1,5)
3 print seq
4
5 for i in seq:
6     print "i:",i
7
8 for i in range(17,23):
9     print "i:",i
10
11 nome = 'Jose Gilmar'
12 print nome
13 for letra in nome:
14     print letra
15
16 lista = [1,3,'casa',5.431,'azul', 'bolo']
17 print lista
18 for elemento in lista:
19     print elemento
20
21 for k in range(0,len(lista)):
22     if (lista[k]=='azul'):
23         print "Elemento encontrado na posicao ", k
24     else: print lista[k]
25
26 lista2=['casa', 'bolo', 'bola', 'dudu', 'bala']
27 print lista2
28 for k in range(0,len(lista2)):
29     if ('a' in lista2[k]):
30         print lista2[k], " possui a letra a"
```

```
[1, 2, 3, 4]
i: 1
i: 2
i: 3
i: 4
i: 17
i: 18
i: 19
i: 20
i: 21
i: 22
Jose Gilmar
J
o
s
e
G
i
l
m
a
r
[1, 3, 'casa', 5.431, 'azul', 'bolo']
1
3
casa
5.431
azul
bolo
1
3
casa
5.431
Elemento encontrado na posicao 4
bolo
['casa', 'bolo', 'bola', 'dudu', 'bala']
casa possui a letra a
bola possui a letra a
bala possui a letra a
```

Estruturas de Repetição

- Não se deve usar parênteses no *for*

```
for (i in range(17,23)):  
    print "i:",i
```

```
for (i in range(17,23)):  
    ^  
SyntaxError: invalid syntax
```


Estruturas de Repetição

- Não se deve usar parênteses no *for*

```
for (i in range(17,23)):  
    print "i:",i
```

```
for (i in range(17,23)):  
    ^  
SyntaxError: invalid syntax
```

Estruturas de Repetição

- Não se deve usar parênteses no *for*

```
for (i in range(17,23)):  
    print "i:",i
```

```
for (i in range(17,23)):  
    ^  
SyntaxError: invalid syntax
```



Tópico especial: Leitura de teclado



Tópicos especiais

Leitura de Teclado

- É possível receber comandos e valores *via* teclado.
- Utiliza-se a função **raw_input()** (python 2.xx) ou **input()** (python 3.xx)

```
1 #!/usr/bin/python
2 ## Utilizando entradas do teclado
3
4 ## Python 2.xx - v = raw_input()
5 ## Python 3.xx - v = input()
6
7 nome = raw_input('Digite o seu nome: |')
8 idade= int(raw_input('Digite a sua idade: '))
9 print nome, "::", idade, " anos"
```

```
Digite o seu nome: José Gilmar
Digite a sua idade: 31
José Gilmar :: 31  anos
```

Leitura de Teclado

- Na linha 8, é definido que idade é um número inteiro.
- Caso o usuário não respeite, ocorrerá erro de execução.
- Mais a frente veremos formas de tratar isso.

```
Digite o seu nome: José
Digite a sua idade: ad
Traceback (most recent call last):
  File "ex09.py", line 8, in <module>
    idade= int(raw_input('Digite a sua idade: '))
ValueError: invalid literal for int() with base 10: 'ad'
```

```
Digite o seu nome: José
Digite a sua idade: 31.2
Traceback (most recent call last):
  File "ex09.py", line 8, in <module>
    idade= int(raw_input('Digite a sua idade: '))
ValueError: invalid literal for int() with base 10: '31.2'
```

Adicionando Elementos a uma Lista

- É possível adicionar novos elementos a uma lista utilizando a função **append()**.

```
lista = []  
print lista  
lista.append('primeiro item')  
print lista  
lista.append(2)  
print lista  
lista.append('terceiro item')  
print lista
```

```
[]  
['primeiro item']  
['primeiro item', 2]  
['primeiro item', 2, 'terceiro item']
```



Exercícios

Exercícios

1) Criar um código que armazene os dados (nome, data de nascimento, sexo e profissão) de **n** pessoas em listas. Após cadastrar as **n** pessoas, o programa deve exibir os dados de cada uma delas.

- *No início da execução do programa, o usuário deve inserir quantos cadastros serão realizados.*
- *Após a realização dos cadastros, todos os itens devem ser exibidos na forma:*
"nome::data_de_nascimento::sexo:profissao"
- *O programa termina depois disso*

Exercícios

2) Adicione ao código anterior a funcionalidade de, após gravar os dados das pessoas, o usuário poder consultar os dados de uma pessoa através do seu **id** (pode ser a ordem de armazenamento).

- *Ao invés de encerrar, o programa deve permitir ao usuário escolher um determinado **id** (neste caso ele irá apresentar as informações do respectivo usuário) indefinidamente ou terminar a aplicação (através de um comando a ser definido por você)*

Exercícios

3) Altere o código anterior permitir ao usuário selecionar uma operação (cadastrar novo usuário, exibir cadastro completo, busca por nome e delatar cadastro);

- *Ao invés de definir um número n de cadastros, o usuário poderá selecionar uma das operações (incluindo a de adicionar um novo cadastro), indefinidamente, ou terminar a aplicação;*
- *Após executar a operação, o programa deve permitir ao usuário escolher uma nova operação.*