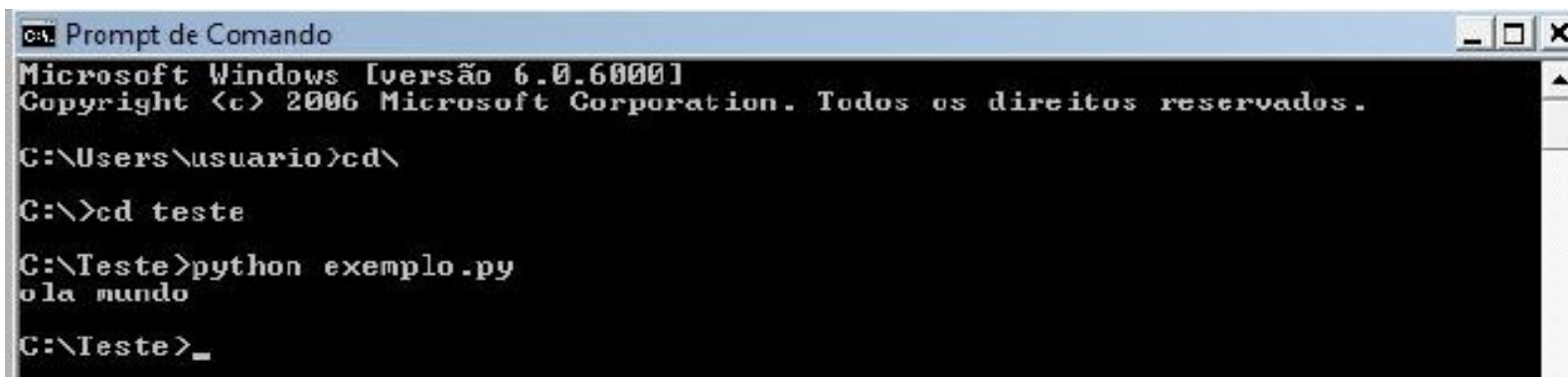


PROGRAMAÇÃO DE  
COMPUTADORES

CONCEITOS FUNDAMENTAIS COM  
**PYTHON**

# Primeiro Exemplo

- Digite no Bloco de Notas do seu computador: `print "Alo mundo"` e salve com o nome `exemplo.py`, dentro de uma pasta chamada teste que você deverá criar na unidade `c:\` de seu computador;
- Em seguida, abra o prompt de comando do windows;
- Saia da pasta em que está localizado pelo comando `cd\`;
- Na unidade `c:\` acesse a pasta teste, para isso digite `cd teste`;
- Estando dentro da pasta teste digite `python exemplo.py`;
- Perceba que seu pequeno programa será executado.



```
C:\> Prompt de Comando
Microsoft Windows [versão 6.0.6000]
Copyright (c) 2006 Microsoft Corporation. Todos os direitos reservados.

C:\Users\usuario>cd\
C:\>cd teste
C:\Teste>python exemplo.py
ola mundo
C:\Teste>_
```

# Segundo Exemplo

- Você pode criar pequenos trechos de programas diretamente no interpretador Python. Para isso, basta você digitar Python no prompt de comandos e a estrutura para digitação será fornecida.
- O segundo exemplo você deverá criar diretamente no prompt.
- Usando o comando print você deverá testar:

<i>print 1 + 1</i>	<i>print 21 % 2</i>	<i>print 4 // 3</i>
<i>print 2 - 1</i>	<i>print 20 % 2</i>	<i>print 3 * 8 + 2</i>
<i>print 15 / 3</i>	<i>print 4 ** 2</i>	<i>print (3*8) + 2</i>

- Crie outros testes com números maiores, menores, decimais e negativos.

# Sobre os números e operadores matemáticos...

- Não custa anotar:
  - Operadores que são novidades (resto da divisão, exponenciação, etc;)
  - Ordem de precedência das operações, lembre-se que os símbolos  $\{ \}$  e  $[ ]$  não são usados para agrupar as operações;
  - Números decimais são separados por ponto (.);
  - Quando todos os números de uma expressão de divisão são inteiros o resultado retornado será inteiro;

# Variáveis e tipos de dados

- Em Python variáveis não precisam ser declaradas;
- Python utiliza tipagem dinâmica o que permite que o tipo ao qual a variável está associada pode variar durante a execução do programa;
- Tipos de dados:

Tipo inteiro (int)	num = 1
Tipo real (float)	num = 1.23
Tipo texto (string)	eu = 'meu nome'

# Variáveis e tipos de dados

- Realize os testes abaixo:

<pre>eu = 'meu nome' print eu</pre>	<pre>a = 'segunda' b = '-' c = 'feira' dia = a+b+c</pre>
<pre>d= 1.87 print d</pre>	<pre>x = 3 print x * 5</pre>
<pre>a = 1 b = 4 print a + b</pre>	<pre>print 'x' * 5</pre>

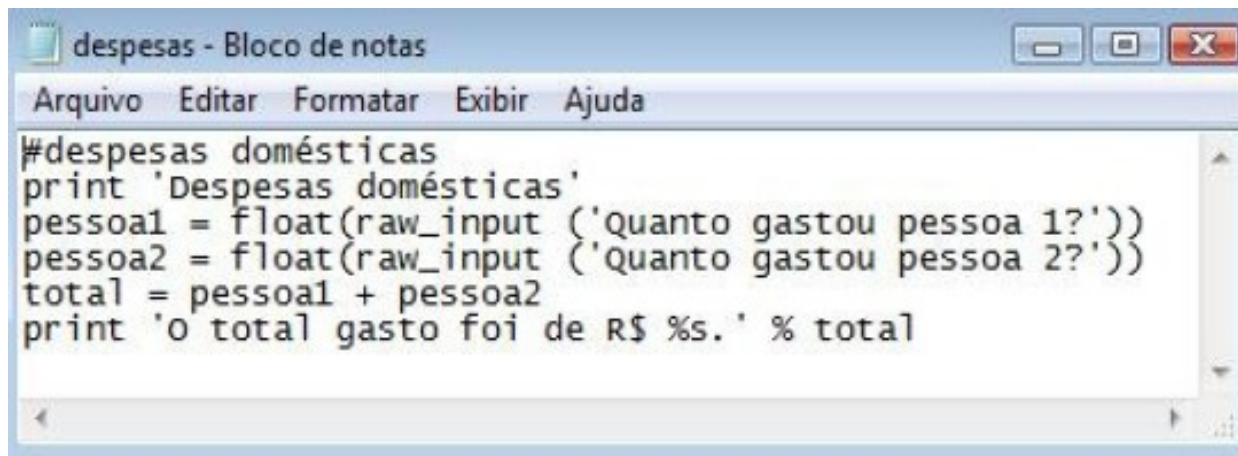
```
print 1, 2, 3; 4; 5
```

```
print 1.2151E3 + 6.28E-1
```

Mais alguns  
Testes  
importantes

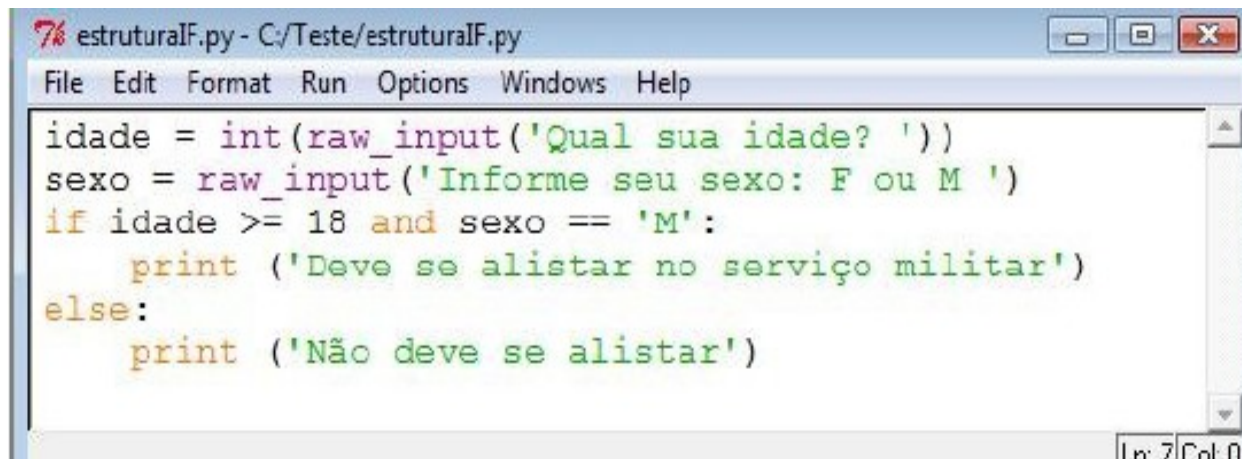
# Entrada e saída de dados

- Nesse momento você já deve ter claro em sua mente qual é o comando de saída utilizado pelo Python. Não??? O comando de saída utilizado pelo interpretador Python é o `print`, usado por você diversas vezes até esse ponto do manual.
- Já o comando de entrada ainda é novidade. Esse comando é representado pela função `input()`.
- Exemplo:
  - Crie no Bloco de Notas o programa abaixo e o execute na linha de comando.



```
#despesas domésticas
print 'Despesas domésticas'
pessoa1 = float(raw_input ('Quanto gastou pessoa 1?'))
pessoa2 = float(raw_input ('Quanto gastou pessoa 2?'))
total = pessoa1 + pessoa2
print 'O total gasto foi de R$ %s.' % total
```

# Estrutura de Seleção

A screenshot of a Python script editor window titled 'estruturalF.py - C:/Teste/estruturalF.py'. The window has a menu bar with 'File', 'Edit', 'Format', 'Run', 'Options', 'Windows', and 'Help'. The code inside is as follows:

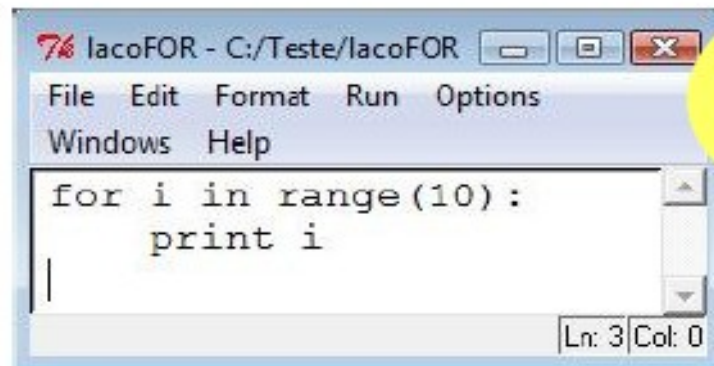
```
idade = int(raw_input('Qual sua idade? '))
sexo = raw_input('Informe seu sexo: F ou M ')
if idade >= 18 and sexo == 'M':
    print ('Deve se alistar no serviço militar')
else:
    print ('Não deve se alistar')
```

The code uses color coding: keywords like 'if', 'else', 'print', and 'int' are in orange, strings are in green, and variables/numbers are in black. The indentation for the 'if' block is clearly visible.

- O código acima merece total atenção, em especial sobre dois itens:
  - A presença de dois pontos sinalizando o início de um bloco de comando;
  - Todas as linhas dos blocos de comando estão igualmente indentadas.
- A partir desse momento passaremos a utilizar o IDLE do interpretador Python. Para isso localize o atalho para IDLE (Python GUI). Em File solicite uma nova janela e digite o código acima. Quando terminar, basta pressionar F5.



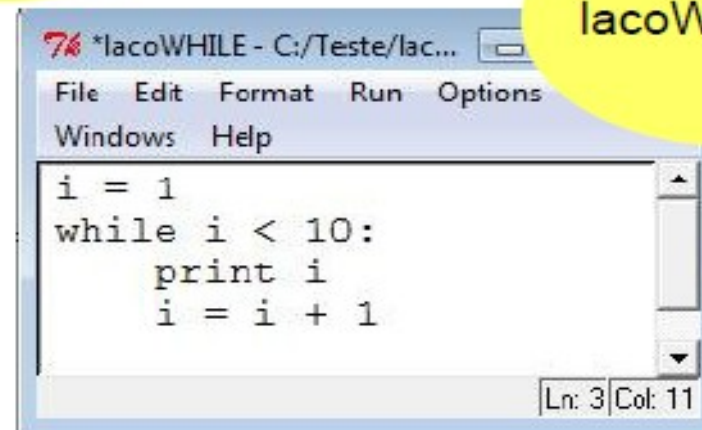
# Estruturas de Repetição



```
File Edit Format Run Options  
Windows Help  
for i in range(10):  
    print i  
|
```

Ln: 3 Col: 0

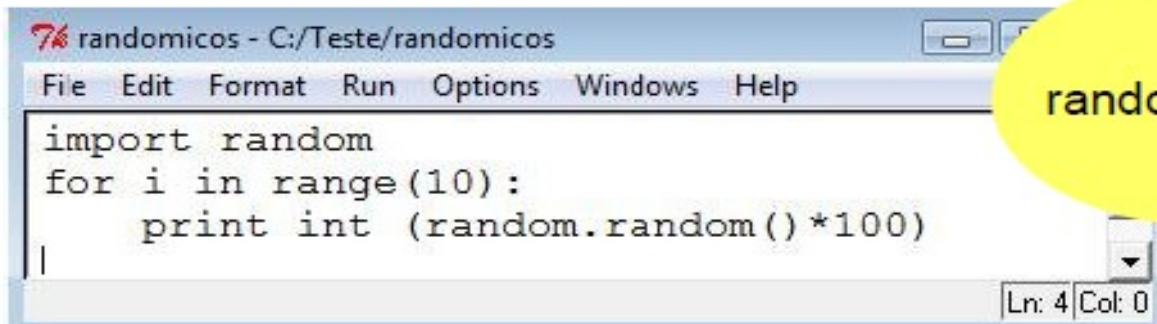
lacoFOR.py



```
*lacoWHILE - C:/Teste/lac...  
File Edit Format Run Options  
Windows Help  
i = 1  
while i < 10:  
    print i  
    i = i + 1  
|
```

Ln: 3 Col: 11

lacoWHILE.py



```
randomicos - C:/Teste/randomicos  
File Edit Format Run Options Windows Help  
import random  
for i in range(10):  
    print int (random.random()*100)  
|
```

Ln: 4 Col: 0

randomicos.py

# Exercícios

- 1 - Construa um programa que indique a classe eleitoral de uma pessoa (eleitor facultativo, eleitor obrigatório, não eleitor).
- 2 - Construa um programa que leia as quatro notas bimestrais de um aluno e informe sua média final.
- 3 - Construa um programa que leia as quatro notas bimestrais de 40 alunos e informe quantos foram aprovados e quantos não foram aprovados. Considere para aprovação média igual ou superior a 6.0.

# Desafios

- 1 - Construa um jogo onde três valores são gerados automaticamente e o jogador precisa acertar a soma deles. O jogador terá uma chance para acertar.
- 2 - Construa um jogo onde o computador sorteia um número de 1 a 100 e o jogador precisa acertar qual foi esse número. A cada tentativa do jogador, o computador deve informar se o número indicado é maior ou menor que o escolhido. O jogador terá três chances para acertar.
- 3 - Construa um jogo de perguntas e respostas sobre um determinado tema escolhido pelo grupo. O jogador deverá ter três chances para acertar cada pergunta. A pontuação deverá ser exibida ao final.

# Observações

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---