

COMPUTAÇÃO 1 - PYTHON

AULA 1 TEÓRICA: INTRODUÇÃO

SILVIA BENZA BAREIRO

SILVIA@BENZA.COM.PY

CONHECENDO A TURMA

Experiência com programação e uso do computador

Quantos já programaram antes ?

Quais linguagens ?

Quantos tem computador em casa com acesso a Internet ?

Qual Sistema Operacional ?

Quantos são calouros ?

Quem veio de outro curso ?

Nível de inglês ?

POR QUE PROGRAMAR?

É divertido

- Melhor que usar programas feitos por outros
- Fazer programas = resolver quebra-cabeças
- Programação como arte

É útil

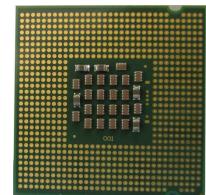
- Programação como ferramenta
- Pode ser aplicado a quase qualquer atividade
 - Arte / Ciência / Filosofia / Entretenimento

OS COMPUTADORES QUEREM SER ÚTEIS!

Os computadores foram criados com um propósito – fazer coisas para a gente!

Mas PRECISAMOS falar a linguagem deles para poder descrever o que queremos que façam

Os usuários não devem se preocupar com isso – alguém já colocou varias instruções nos nossos computadores e celulares para que o usuário só escolha o que usar.



USUÁRIOS VS. PROGRAMADORES

Os usuários vêm o computador como um conjunto de ferramentas a serem utilizadas – Word, Excel, WoW, Dota, LoL, etc.

Os programadores aprendem como funciona o computador e as linguagens de programação

Os programadores possuem ferramentas que ajudam a criar novas ferramentas – para varias pessoas ou pra eles mesmos

OBJETIVO DA DISCIPLINA

**Aprender a construir ferramentas ou programas de
computadores**

Exemplos de programas?

OBJETIVO DA DISCIPLINA

Aprender a construir ferramentas ou programas de computadores

Exemplos de programas?

- Explorer, Firefox, Google Chrome
- Facebook
- Windows
- Word,Powerpoint
- Media Player, iTunes

OBJETIVO DA DISCIPLINA

Aprender a construir ferramentas ou programas de computadores

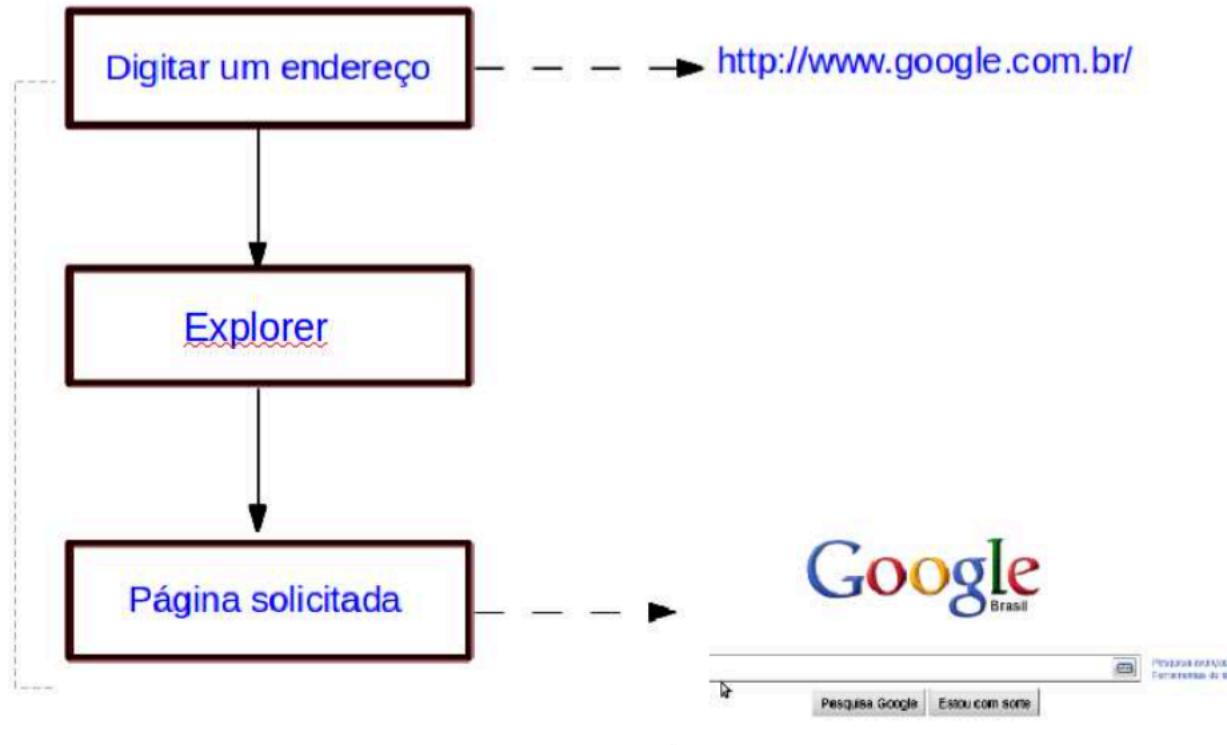
Exemplos de programas?

- Explorer, Firefox, Google Chrome

Qual a “tarefa” que o Explorer deve realizar?

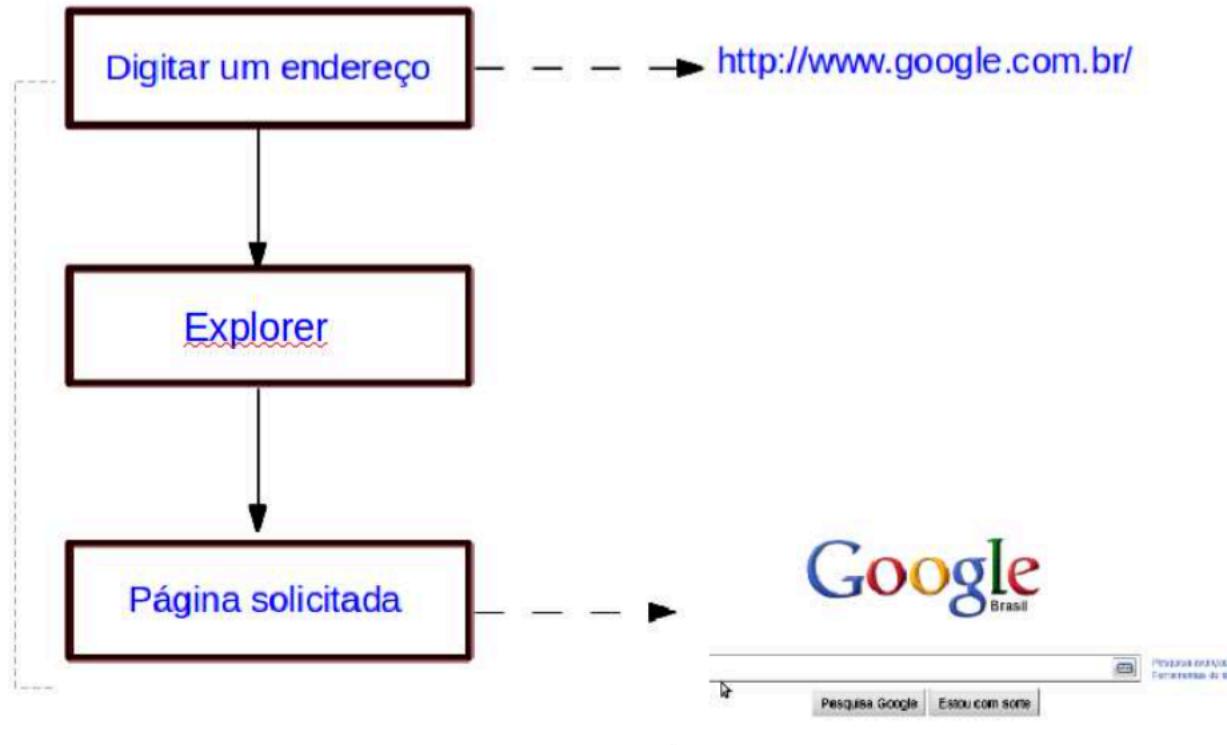
OBJETIVO DA DISCIPLINA

Qual a “tarefa” que o Explorer deve realizar?



OBJETIVO DA DISCIPLINA

Como o Explorer realiza esta “tarefa”?



EXEMPLO DE INSTRUÇÕES

while music is playing:

Left hand out and up

Right hand out and up

Flip Left hand

Flip Right hand

Left hand to right shoulder

Right hand to left shoulder

Left hand to back of head

Right hand to back of head

Left hand to right hit

Right hand to left hit

Left hand on left bottom

Right hand on right bottom

Wiggle

Wiggle

Jump



EXEMPLO DE INSTRUÇÕES

```
while music is playing:  
    Left hand out and up  
    Right hand out and up  
    Flip Left hand  
    Flip Right hand  
    Left hand to right shoulder  
    Right hand to left shoulder  
    Left hand to back of head  
    Right ham to back of head  
    Left hand to right hit  
    Right hand to left hit  
    Left hand on left bottom  
    Right hand on right bottom  
    Wiggle  
    Wiggle  
    Jump
```

EXEMPLO DE INSTRUÇÕES

```
while music is playing:  
    Left hand out and up  
    Right hand out and up  
    Flip Left hand  
    Flip Right hand  
    Left hand to right shoulder  
    Right hand to left shoulder  
    Left hand to back of head  
    Right hand to back of head  
    Left hand to right hip  
    Right hand to left hip  
    Left hand on left bottom  
    Right hand on right bottom  
    Wiggle  
    Wiggle  
    Jump
```

EXEMPLO DE INSTRUÇÕES

O computador não consegue compreender esses erros!

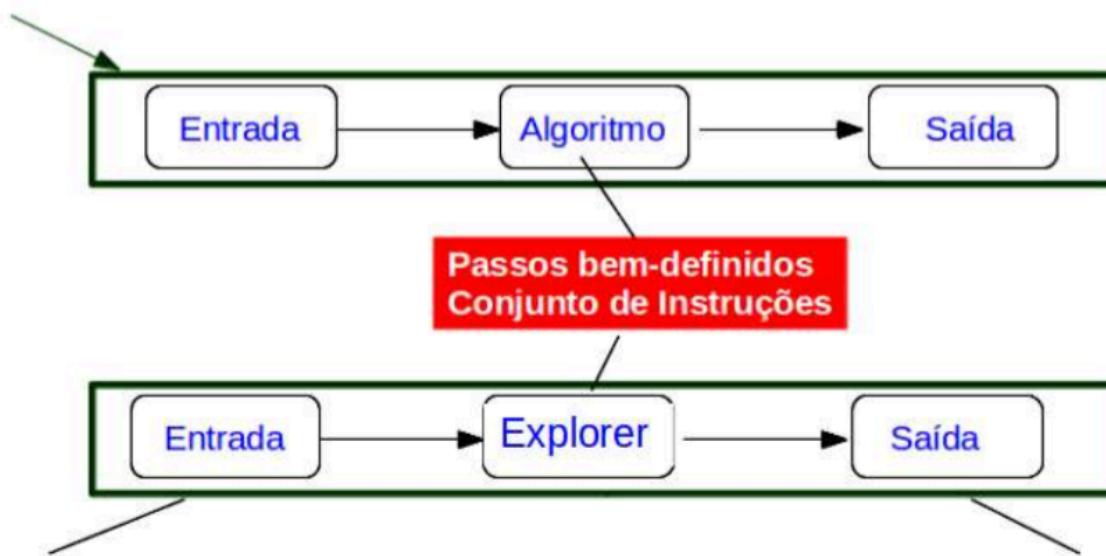
ALGORITMO

Método efetivo expresso como um conjunto de instruções que devem ser feitas para realizar uma tarefa.

Características

- **Finitude:** deve sempre terminar após um número finito de passos.
- **Bem-definido:** cada passo de um algoritmo deve ser precisamente definido (sem ambiguidades).
- **Entradas:** deve ter zero ou mais entradas (informações que são fornecidas antes do algoritmo iniciar).
- **Saídas:** deve ter uma ou mais saídas (resultado final do algoritmo).
- **Efetividade:** todas as operações devem ser suficientemente básicas de modo que possam ser executadas com precisão em um tempo finito por uma pessoa.

ALGORITMO



<http://www.google.com.br/>

Google
Brasil



ALGORITMOS E PROGRAMAS

Algoritmo = método para solucionar um problema

Estruturas de dados = método para organizar informações

Programa = algoritmos + estruturas de dados expressos de forma a ser entendidos pelo computador

Programas tipicamente processam dados de entrada e produzem dados de saída

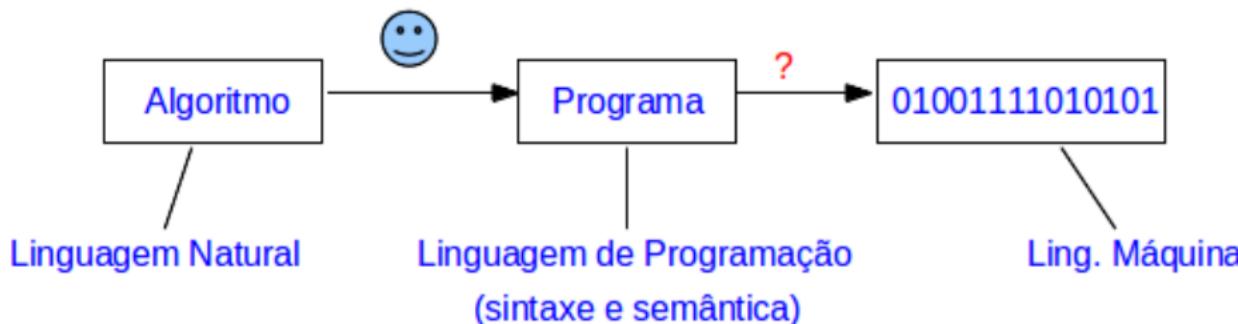
PROGRAMAS DE COMPUTADOR

Conjunto de instruções que descrevem como uma tarefa deve ser realizada por um computador. Ou seja, o computador deve ser capaz de “entender” as instruções.



O computador “entende” linguagem de máquina:
01011100110.

Como traduzir um algoritmo para código de máquina?



LINGUAGENS DE PROGRAMAÇÃO

Linguagem artificial com sintaxe e semântica bem definidos

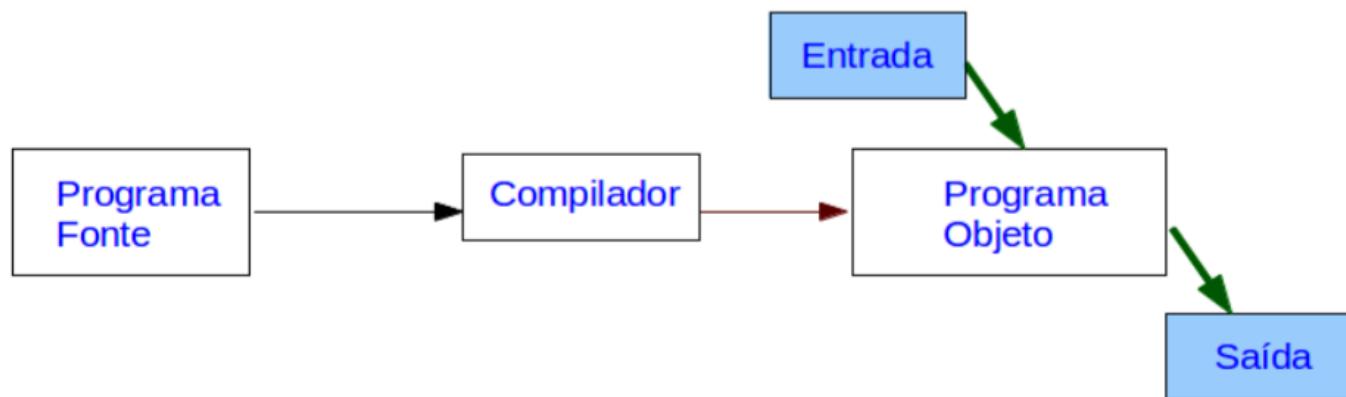
Usada para exprimir programas

Mais “técnica” do que linguagens naturais

**Menos “técnica” do que a linguagem nativa dos
computadores (linguagem de máquina)**

COMPILADOR

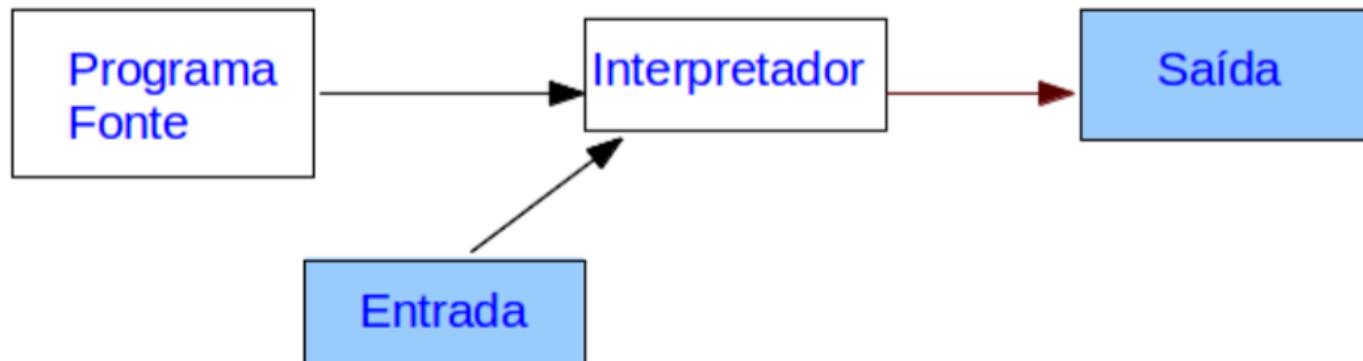
- Programas que traduzem programas escritos em linguagem de programação para programas equivalentes escritos em linguagem de máquina
- O primeiro é chamado de programa fonte, enquanto que o segundo é chamado de programa objeto



INTERPRETADOR

Simulam uma “máquina virtual” ■

Programa fonte é lido, entendido e as instruções são executadas imediatamente



POR QUE PYTHON?

Simples o suficiente para um curso introdutório

Muitos recursos

- Orientação a Objetos
- Escalável (módulos, classes, controle de exceções)
- Biblioteca embutida extensa e grande número de módulos fornecidos por terceiros

Grande variedade de aplicações

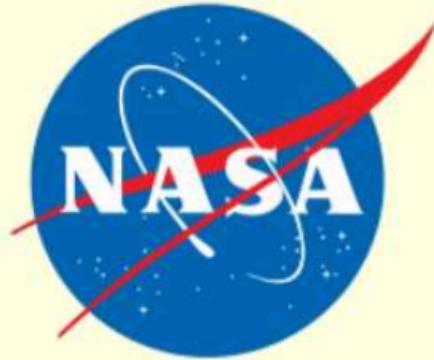
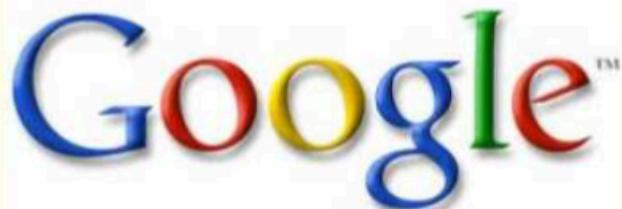
Linguagem interpretada (script)

Multi-plataforma

Grátis!

Comunidade bastante grande

QUEM USA PYTHON?



YAHOO!® GROUPS

O QUE VAMOS PRECISAR?

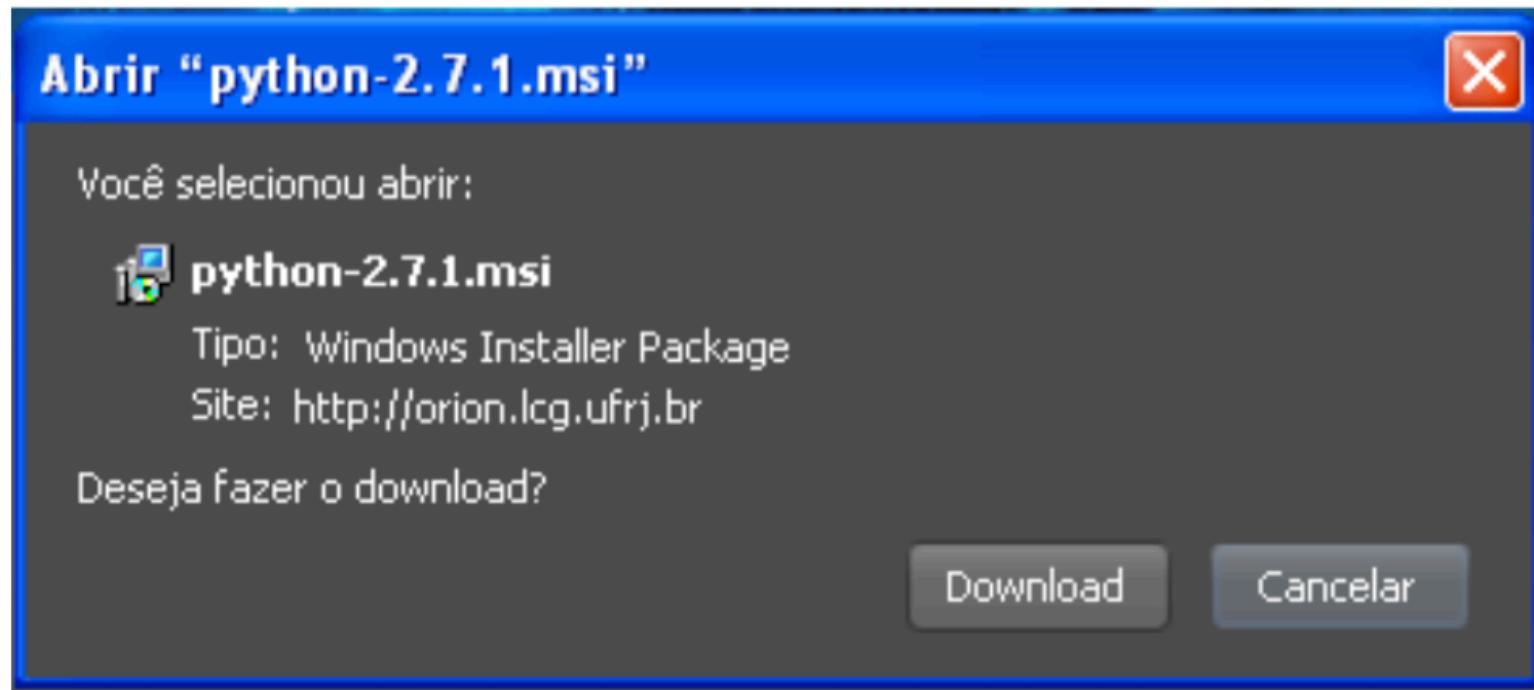
Uma implementação da linguagem

- <http://www.python.org>
- Implementação pronta para baixar (Windows)
- Linux normalmente já vem com python instalado

Um editor de textos

- Qualquer editor serve
- Ambiente IDLE inclui um editor
 - Incluído na distribuição Windows

INSTALANDO O INTERPRETADOR PYTHON 2.7



INSTALANDO O INTERPRETADOR PYTHON 2.7



INSTALANDO O INTERPRETADOR PYTHON 2.7



INSTALANDO O INTERPRETADOR PYTHON 2.7



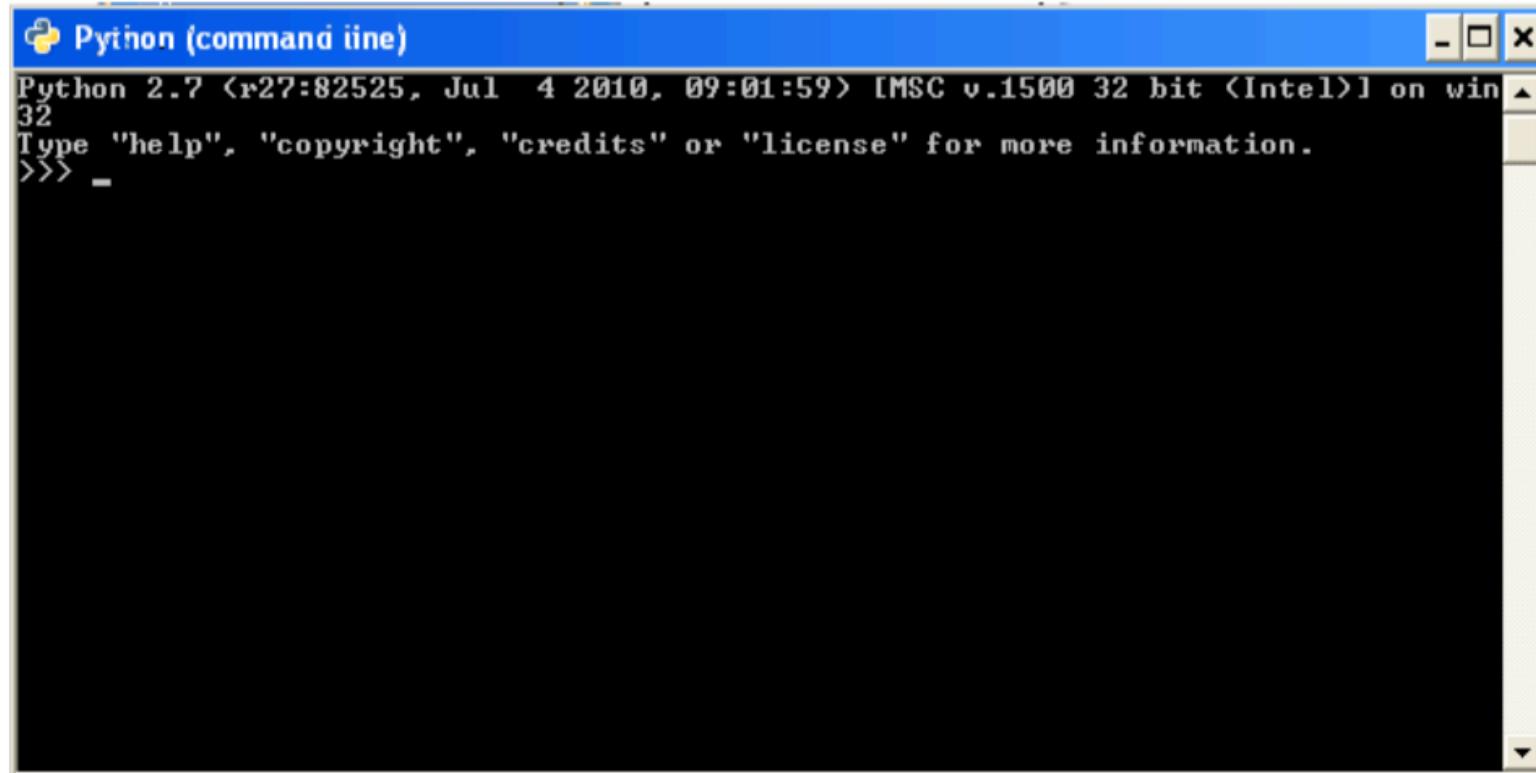
INSTALANDO O INTERPRETADOR PYTHON 2.7



INSTALANDO O INTERPRETADOR PYTHON 2.7



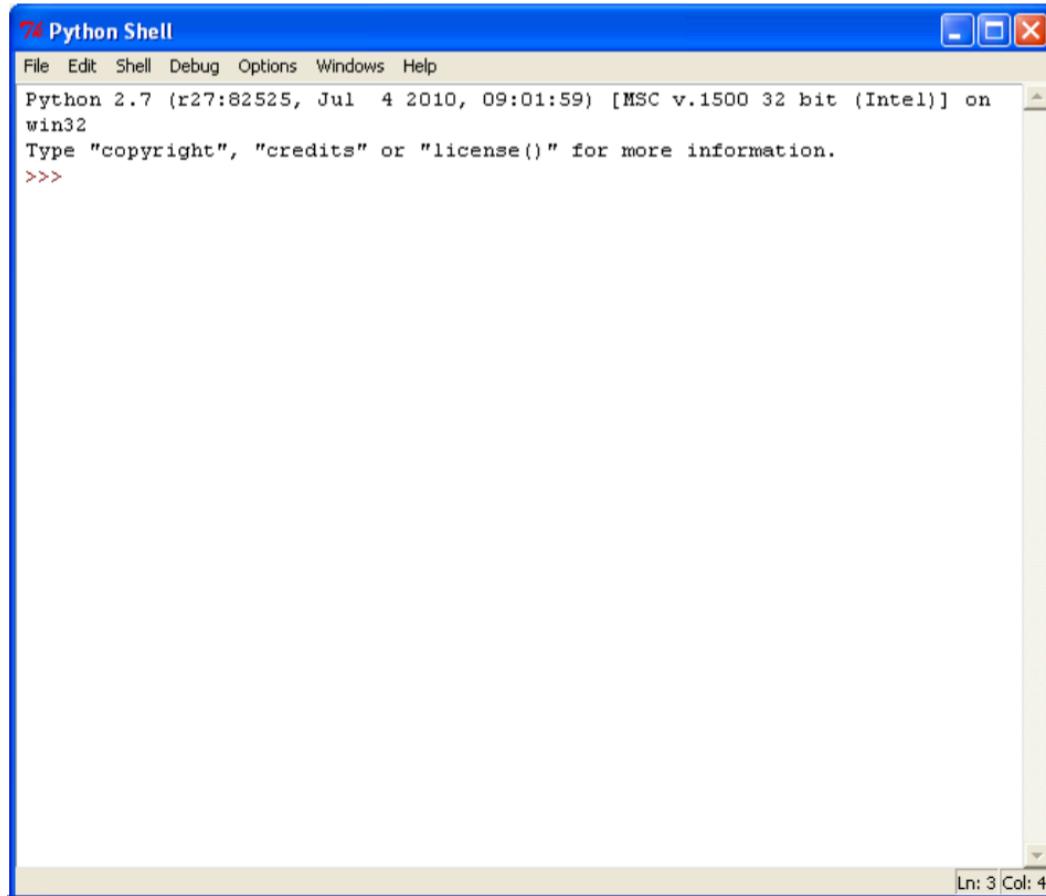
INSTALANDO O INTERPRETADOR PYTHON 2.7



INSTALANDO O INTERPRETADOR PYTHON 2.7



INSTALANDO O INTERPRETADOR PYTHON 2.7



PYTHON TUTOR

Pythontutor - <http://pythontutor.com/>

[Start using Online Python Tutor now](#)

For instance, here is a visualization showing a program that recursively finds the sum of a (cons-style) linked list. Click the “Forward” button to see what happens as the computer executes each line of code.

```
1 def listSum(numbers):
2     if not numbers:
3         return 0
4     else:
5         (f, rest) = numbers
6         return f + listSum(rest)
7
8 myList = (1, (2, (3, None)))
9 total = listSum(myList)
```

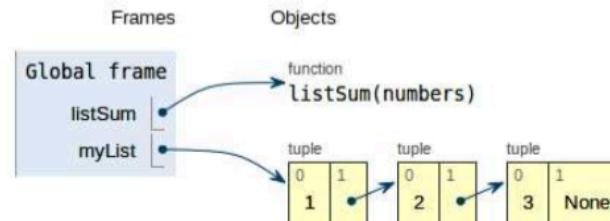
[Edit code](#)



< Back

Step 3 of 22

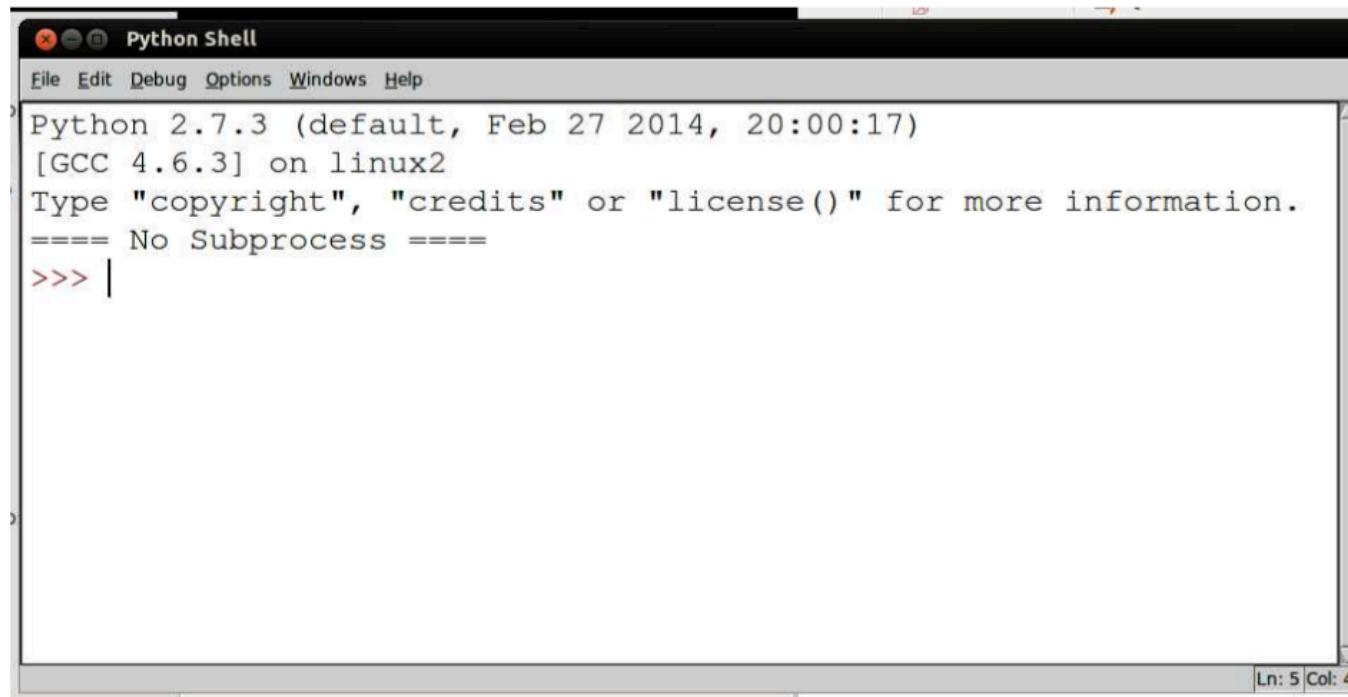
Forward >



**MAS,
VAMOS LÁ**

PYTHON – MODO INTERATIVO

Experiência com programação e uso do computador



A screenshot of a Python Shell window titled "Python Shell". The window has a menu bar with File, Edit, Debug, Options, Windows, Help. The main area displays the following text:

```
Python 2.7.3 (default, Feb 27 2014, 20:00:17)
[GCC 4.6.3] on linux2
Type "copyright", "credits" or "license()" for more information.
===== No Subprocess =====
>>> |
```

In the bottom right corner of the window, there is a status bar with "Ln: 5 Col: 4".

PYTHON COMO CALCULADORA

O Interpretador python pode ser usado como calculadora

Por exemplo, as quatro operações aritméticas são denotadas pelos símbolos

- + adição
- - subtração
- * multiplicação
- / divisão

PYTHON COMO CALCULADORA

```
>>> 10
10
>>> # Um comentário é precedido do caractér "#"
... # Comentários são ignorados pelo interpretador
... 10+5
15
>>> 10-15 # Comentários podem aparecer também após código
-5
>>> 10*3
30
>>> 10/3
3
>>> 10/-3 # Divisão inteira retorna o piso
-4
>>> 10%3 # Resto de divisão inteira simbolizado por %
1
```

PRIMEIROS PASSOS - OPERADORES

adição	+
subtração	-
multiplicação	*
divisão	/ ou //
exponenciação	**
módulo	%

Regras de precedência

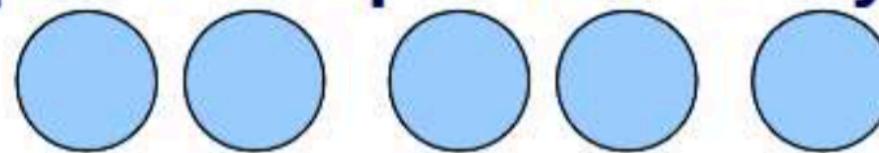
- 1 Expressões entre parênteses
- 2 Exponenciação
- 3 Multiplicação, Divisão e Módulo (*)
- 4 Adição e Subtração (*)

(*) Esquerda para direita

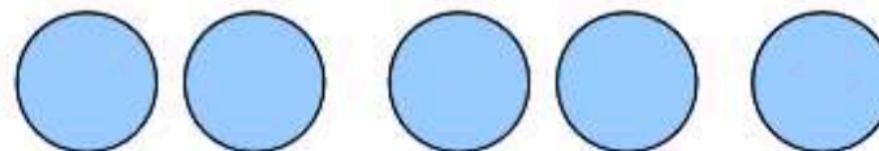
PRIMEIROS PASSOS - OPERADORES

Qual a sequência de operações ?

$$Z = p * r \% q + w / x - y$$



$$Z = a * x ^\star 2 + b * x + c$$



PRIMEIROS PASSOS - OPERADORES

Qual a sequência de operações ?

$$Z = p * r \% q + w / x - y$$

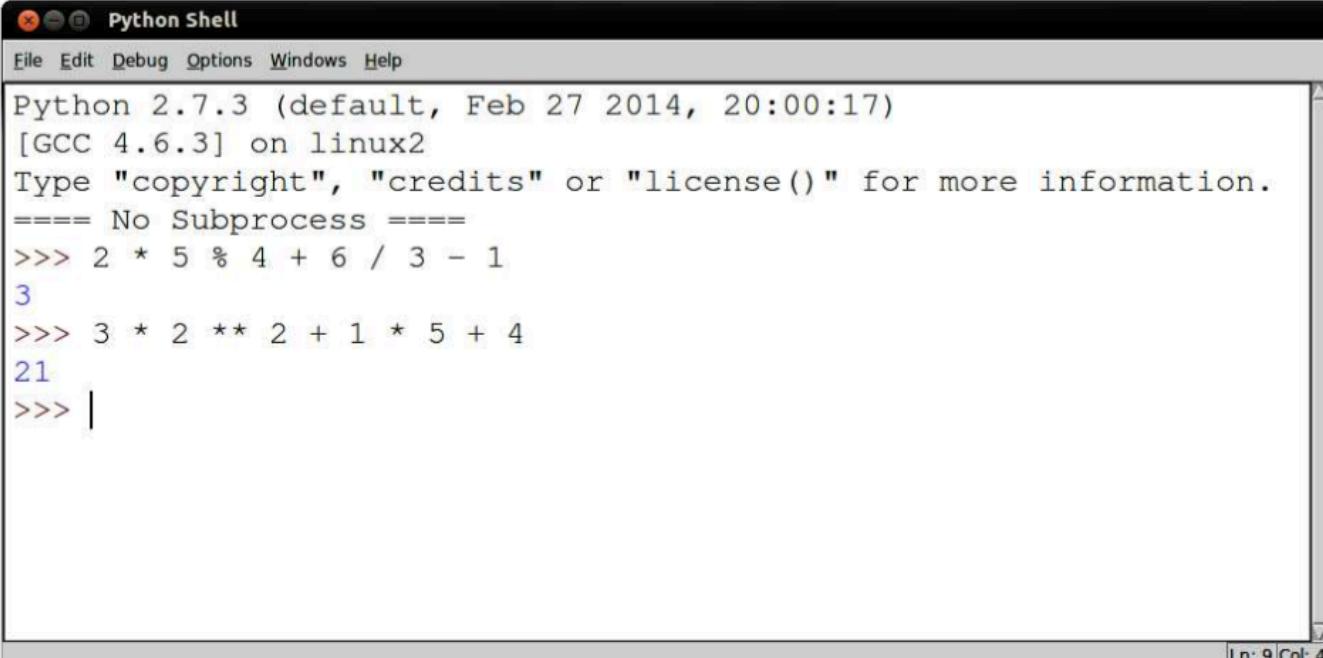
1 2 4 3 5

$$Z = a * x ** 2 + b * x + c$$

2 1 4 3 5

PRIMEIROS PASSOS - OPERADORES

Qual a sequência de operações ?



The screenshot shows a Python Shell window with the title "Python Shell". The menu bar includes "File", "Edit", "Debug", "Options", "Windows", and "Help". The window displays the following Python session:

```
Python 2.7.3 (default, Feb 27 2014, 20:00:17)
[GCC 4.6.3] on linux2
Type "copyright", "credits" or "license()" for more information.
===== No Subprocess =====
>>> 2 * 5 % 4 + 6 / 3 - 1
3
>>> 3 * 2 ** 2 + 1 * 5 + 4
21
>>> |
```

The session demonstrates the evaluation of arithmetic expressions based on operator precedence. In the first expression, multiplication and division are performed before modulus and addition. In the second expression, exponentiation is evaluated before multiplication and addition.

PRIMEIROS PASSOS

Exercício: Escreva funções que dados dois números x e y , retorna:

soma → soma($2,3$) = $2 + 3 = 5$

subtração → subt($7,4$) = $7 - 4 = 3$

divisão → divis($8,2$) = $8 / 2 = 4$

produto → prod($3,5$) = $3 * 5 = 15$

PRIMEIROS PASSOS

Exercício: Escreva funções que dados dois números x e y , retorna:

soma → soma(2,3) = 2 + 3 = 5

subtração → subt(7,4) = 7 - 4 = 3

divisão → divis(8,2) = 8 / 2 = 4

produto → prod(3,5) = 3 * 5 = 15

COMO DEFINIR UMA FUNÇÃO EM PYTHON

```
def nome-função( lista-parâmetros ):  
    return valor de retorno
```

COMO DEFINIR UMA FUNÇÃO EM PYTHON

```
def nome-função( lista-parâmetros ):  
    return valor de retorno
```

Soma

- Nome da Função: soma
- Parâmetros: x,y
- Valor de Retorno: : x+y

COMO DEFINIR UMA FUNÇÃO EM PYTHON

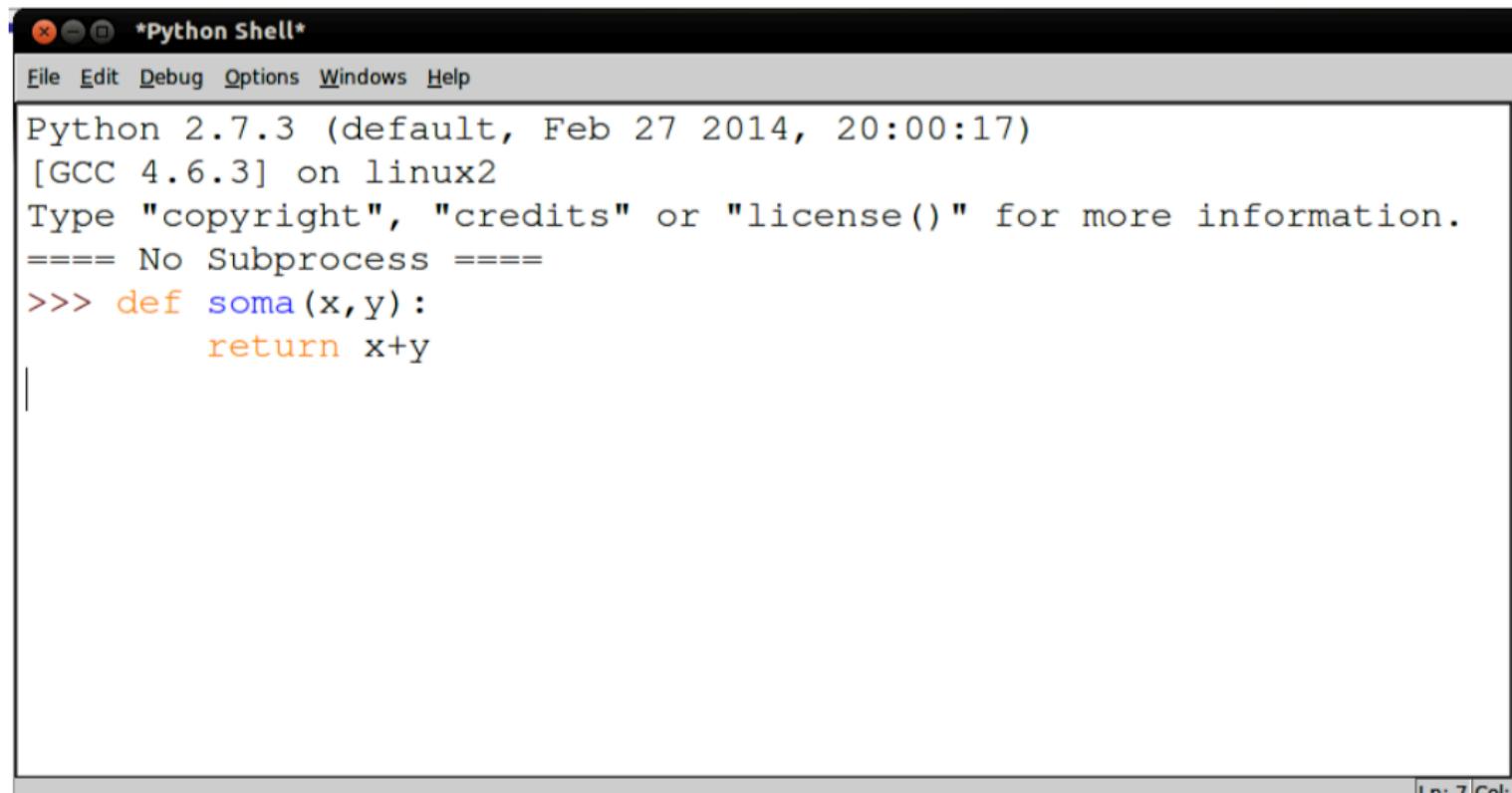
```
def nome-função( lista-parâmetros ):  
    return valor de retorno
```

Soma

- Nome da Função: soma
- Parâmetros: x,y
- Valor de Retorno: : x+y

```
def soma (x,y):  
    return x+y
```

EXECUTANDO FUNÇÕES INTERATIVAMENTE

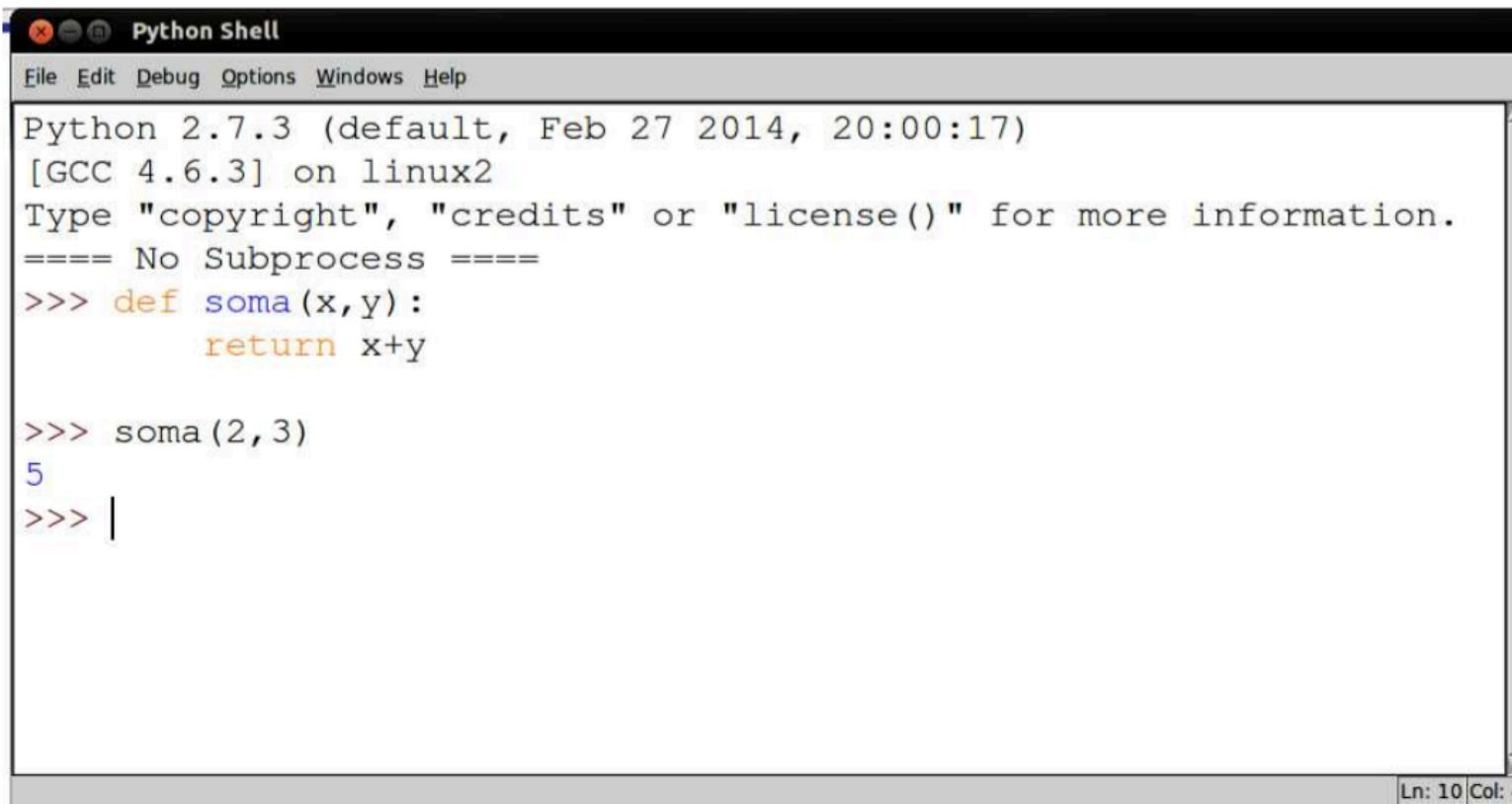


A screenshot of a Python Shell window titled '*Python Shell*'. The window has a menu bar with File, Edit, Debug, Options, Windows, and Help. The main area displays the Python startup message and a user-defined function:

```
Python 2.7.3 (default, Feb 27 2014, 20:00:17)
[GCC 4.6.3] on linux2
Type "copyright", "credits" or "license()" for more information.
===== No Subprocess =====
>>> def soma(x,y):
    return x+y
```

The status bar at the bottom right shows 'In: 7 Col: 0'.

EXECUTANDO FUNÇÕES INTERATIVAMENTE



A screenshot of a Python Shell window. The title bar says "Python Shell". The menu bar includes "File", "Edit", "Debug", "Options", "Windows", and "Help". The main window displays the following text:

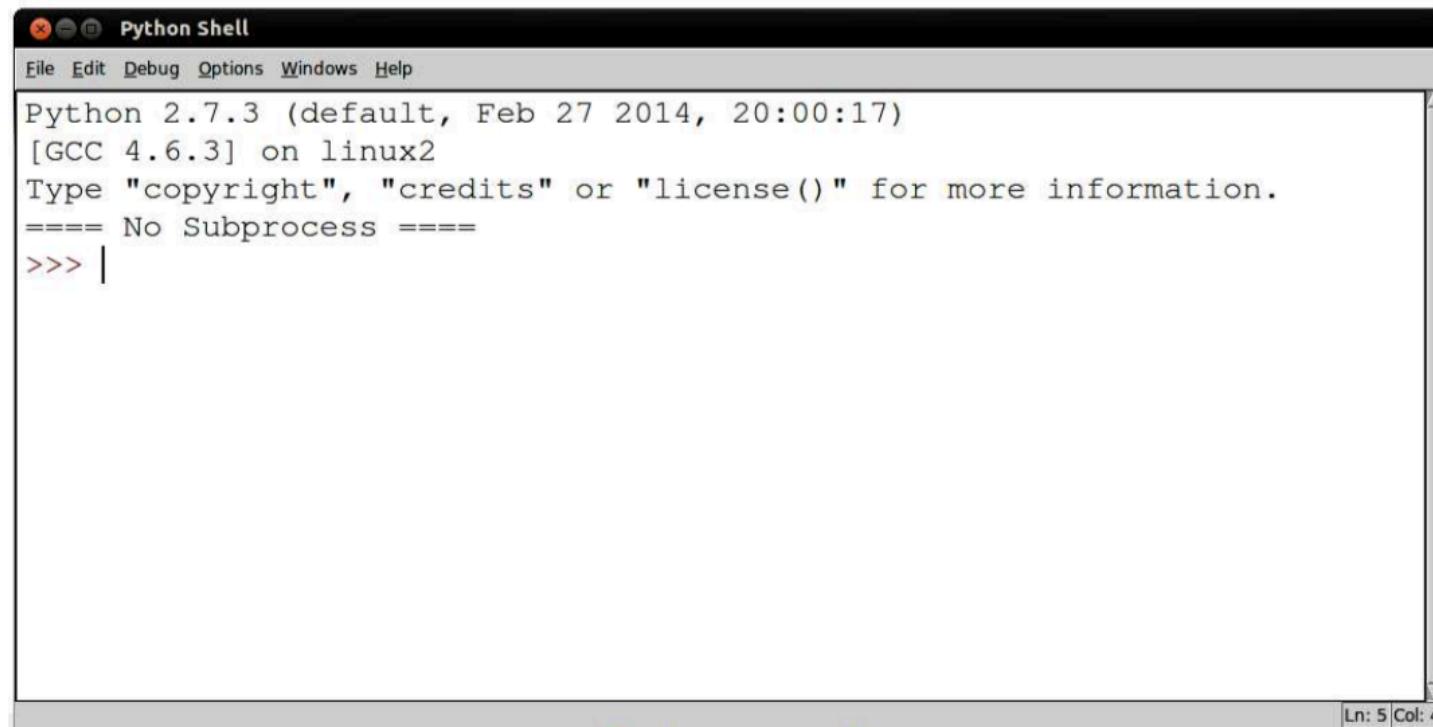
```
Python 2.7.3 (default, Feb 27 2014, 20:00:17)
[GCC 4.6.3] on linux2
Type "copyright", "credits" or "license()" for more information.
===== No Subprocess =====
>>> def soma(x,y):
    return x+y

>>> soma(2,3)
5
>>> |
```

In the bottom right corner of the window, there is a status bar with "Ln: 10 Col: 4".

EDITOR IDLE

Abrir o editor IDLE: 'File > New Window' ou **Ctrl + N**

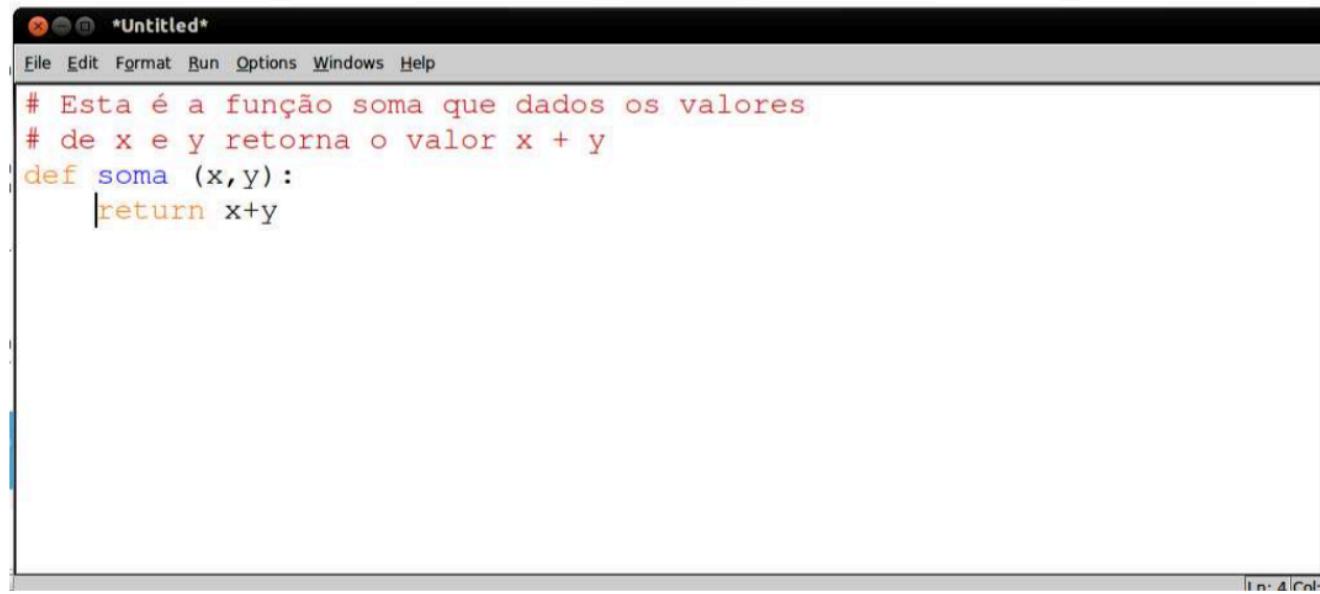


EDITOR IDLE



Escreva o código da função no editor IDLE

EDITOR IDLE



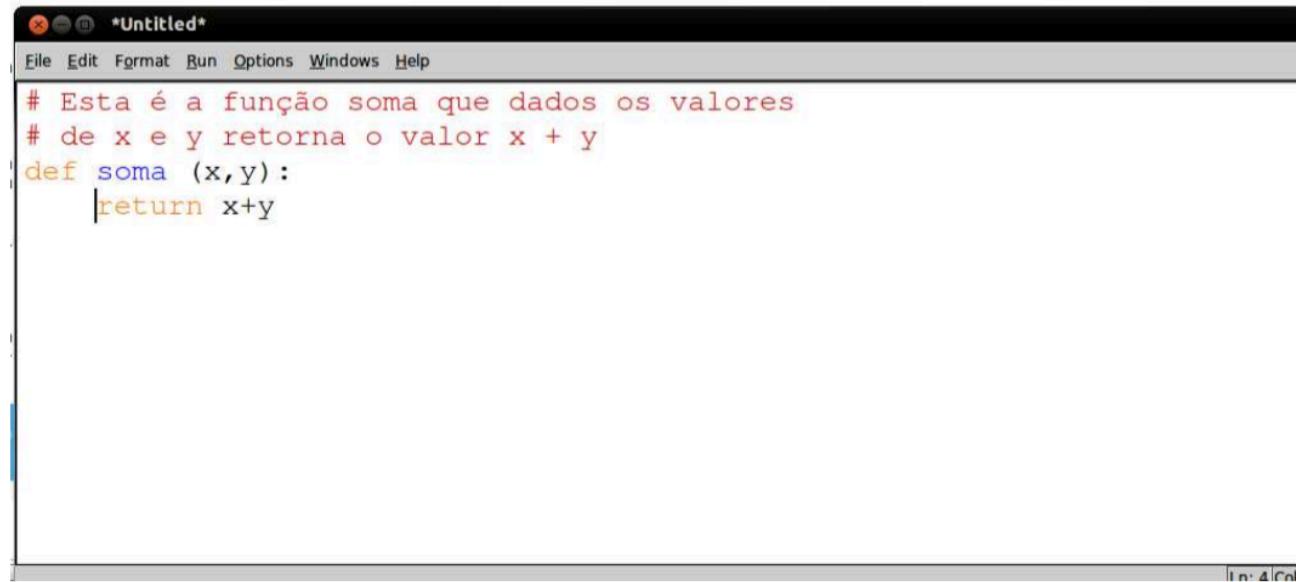
The image shows a screenshot of the Python IDLE editor window titled '*Untitled*'. The menu bar includes File, Edit, Format, Run, Options, Windows, and Help. The code area contains the following Python code:

```
# Esta é a função soma que dados os valores
# de x e y retorna o valor x + y
def soma (x,y):
    return x+y
```

The status bar at the bottom right indicates Line: 4 Col: 4.

**Escreva o código da função no editor IDLE
Utilize # quando quiser fazer algum comentário na sua
função.**

EDITOR IDLE



A screenshot of the Python Idle Editor window titled '*Untitled*'. The menu bar includes File, Edit, Format, Run, Options, Windows, and Help. The code in the editor is:

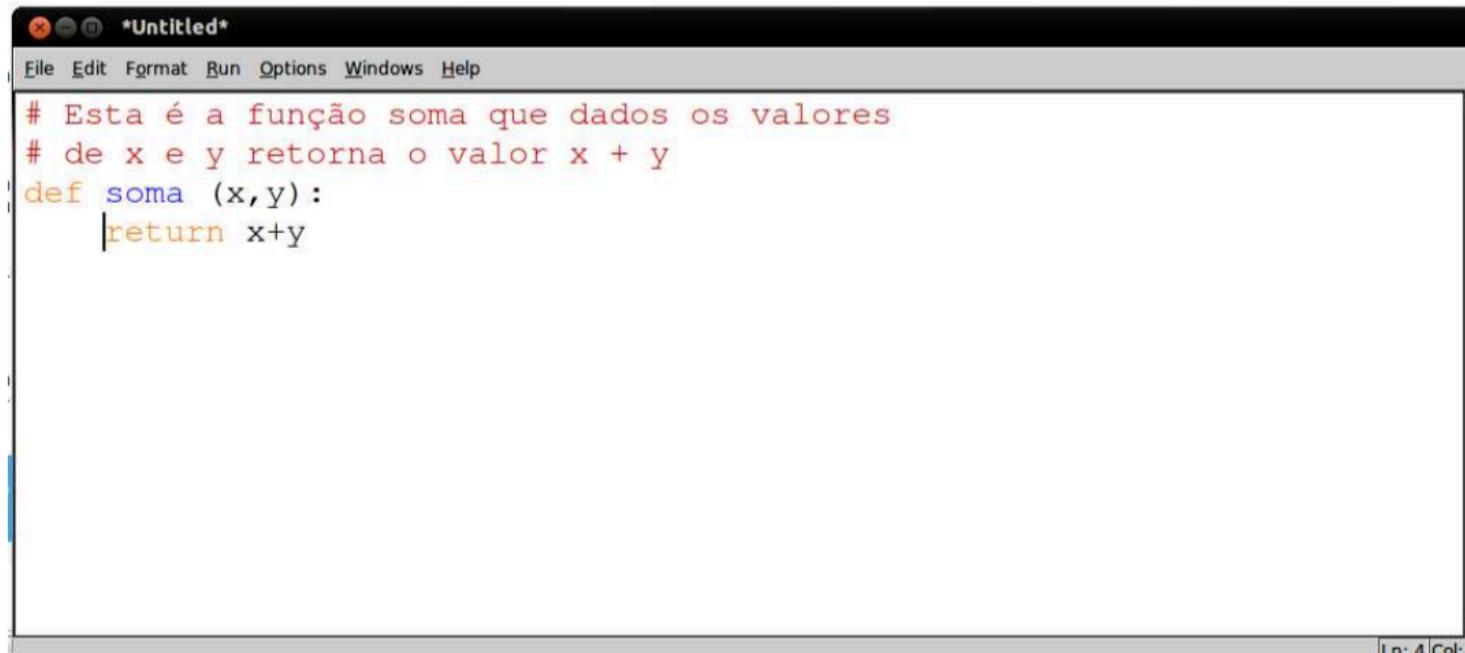
```
# Esta é a função soma que dados os valores
# de x e y retorna o valor x + y
def soma (x,y):
    return x+y
```

Guarde seu código em um arquivo:

selecione 'File > Save' ou Ctrl + S

Podemos usar o seguinte nome para este arquivo: **funcoes.py**

EDITOR IDLE



```
*Untitled*
File Edit Format Run Options Windows Help
# Esta é a função soma que dados os valores
# de x e y retorna o valor x + y
def soma (x,y):
    return x+y
```

Execute sua função a partir do Editor IDLE:

selecione Run Module (F5)

COMO DEFINIR UMA FUNÇÃO EM PYTHON

Como ficam as outras funções ?

COMO DEFINIR UMA FUNÇÃO EM PYTHON

Como ficam as outras funções ?

```
def nome-função( lista-parâmetros ):  
    return valor de retorno
```

```
def soma (x,y):  
    return x+y
```

```
def subt(x,y):  
    return x-y
```

```
def prod(x,y):  
    return x*y
```

```
def divis(x,y):  
    return x/y
```

COMO DEFINIR UMA FUNÇÃO EM PYTHON

Como ficam as outras funções ?

```
def nome-função( lista-parâmetros ):  
    return valor de retorno
```

```
def soma (x,y):  
    return x+y
```

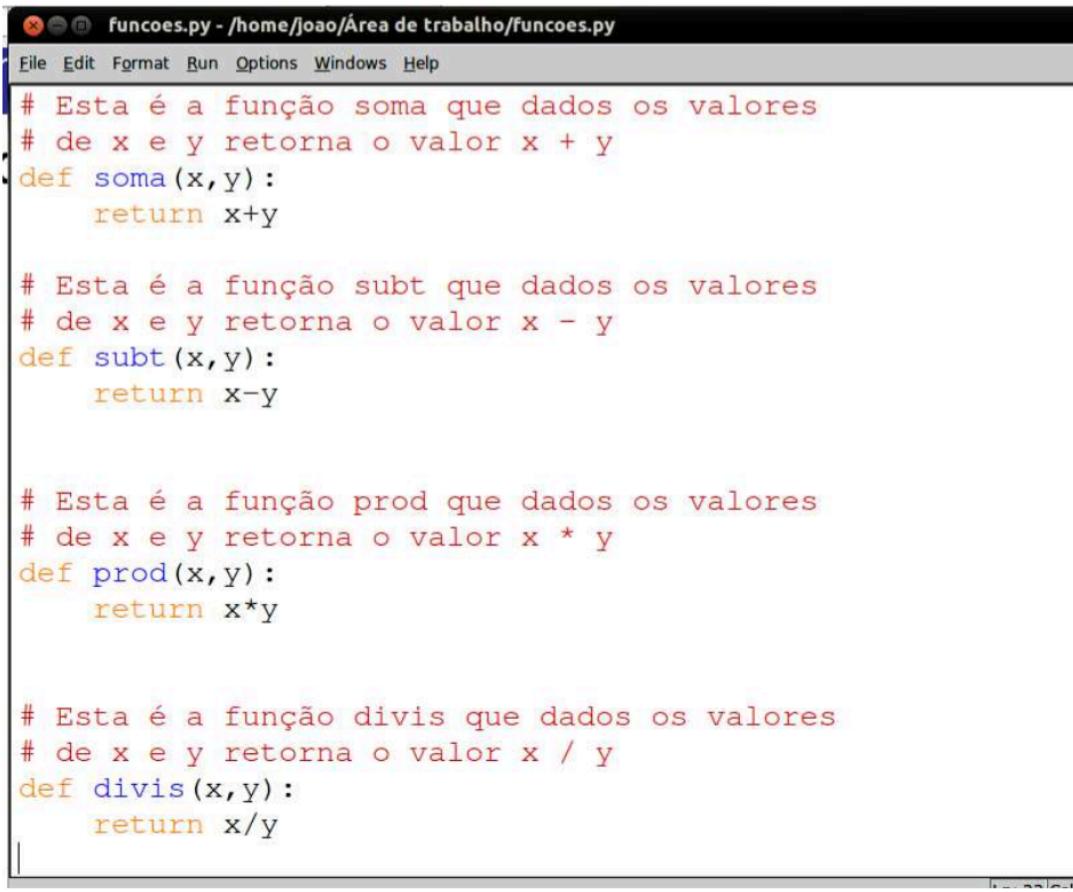
```
def subt(x,y):  
    return x-y
```

```
def prod(x,y):  
    return x*y
```

```
def divis(x,y):  
    return x/y
```

Acrescente estas funções no seu arquivo `funcoes.py`

COMO DEFINIR UMA FUNÇÃO EM PYTHON



```
funcoes.py - /home/joao/Área de trabalho/funcoes.py
File Edit Format Run Options Windows Help
# Esta é a função soma que dados os valores
# de x e y retorna o valor x + y
def soma(x,y):
    return x+y

# Esta é a função subt que dados os valores
# de x e y retorna o valor x - y
def subt(x,y):
    return x-y

# Esta é a função prod que dados os valores
# de x e y retorna o valor x * y
def prod(x,y):
    return x*y

# Esta é a função divis que dados os valores
# de x e y retorna o valor x / y
def divis(x,y):
    return x/y
```

EXERCÍCIO

Escreva uma função que dado um número, retorna o seu quadrado.

EXERCÍCIO

Escreva uma função que dado um número, retorna o seu quadrado.

```
# Esta é uma maneira de calcular o quadrado  
# de um número usando multiplicação
```

```
def quadrado(x):  
    return x*x
```

```
# Esta é uma maneira de calcular o quadrado  
# de um número usando exponenciação
```

```
def quadrado(x):  
    return x**2
```

EXERCÍCIOS

Faça uma função que:

1. Calcule a área de um retângulo dados seus dois lados. Teste pelo menos para os seguintes pares de entrada:
 - 5 e 7; resposta esperada é 35
 - 15 e 2; resposta esperada é 30
 - 500 e 700; resposta esperada é 350000
 - 5 e 0; resposta esperada é 0
2. Calcule a área da coroa circular (anel) formada por dois círculos de raios $r1$ e $r2$ ($r1 > r2$ e $Pi = 3.14$). Teste pelo menos para os seguintes pares de entrada:
 - 2 e 1; resposta esperada é 9.42
 - 15 e 5; resposta esperada é 628
 - 100 e 0; resposta esperada é 31400

EXERCÍCIOS

3. Calcule o resultado e o resto da divisão de dois números inteiros (a função deve retornar os dois valores no mesmo comando *return*).
4. Calcule a ordenada de uma função de segundo grau dados os parâmetros *a*, *b*, *c* e a abscissa.
5. Dado o valor da conta de um restaurante, calcule a gorjeta do garçom, considerando 10% do valor da conta.
6. Calcule a média de dois números. Teste pelo menos para os seguintes pares de entrada:
 - - 5 e 7;
 - 2 e -2;
 - 5 e 5;
 - 3 e 4;
 - 3.0 e 4.0;
7. Calcule a média ponderada de dois números com os respectivos pesos. ☰

EXERCÍCIOS

8. Calcule a distância que a correnteza arrasta um barco que atravessa um rio. São conhecidas: a velocidade da correnteza, a largura do rio e a velocidade do barco perpendicular à correnteza.

9. Calcule o saldo final de uma conta, dado o saldo inicial, o número de meses e a taxa de juros mensal (juros simples).

$$\text{Saldo Final} = \text{Saldo Inicial} (1 + \text{juros.meses})$$

10. Calcule o erro entre o valor da soma de uma PG infinita a partir de 1.0 e a soma dos n primeiros termos dessa PG. **A soma dos termos de uma PG é $1/(1 - q)$, onde q é a razão e $0 \leq q < 1$.**

11. Calcule o tempo total de prova de um corredor de maratona em horas, minutos e segundos, dados: o tempo de partida (hh,mm,ss), e o tempo de chegada (hh,mm,ss).

12. Calcule o valor da gorjeta (10%) e o quanto cada pessoa de um grupo deve pagar (divisão equalitária). São dados o valor total da conta do restaurante e o número de pessoas na mesa.

13. Calcule a área da superfície de um cubo que tem c por aresta.

BIBLIOGRAFÍA

**Aprendendo Python - David Ascher & Mark Lutz,
Editora Bookman, 2^a edição, 2007, 566 páginas.**

**Beginning Python from Novice to Professional - Magnus Lie
Hetland, Apress, 2005, 640 pages.**

**Python Programming, an Introduction to Computer Science -
John Zelle, Franklin, Beedle, & Associates, 517 pages.**

Programming Python - Mark Lutz, O'Reilly, 2001, 1255 pages.

**Python Cookbook - Alex Martelli & David Ascher, O'Reilly, 2002,
575 pages.**

***Rapid GUI Programming with Python and Qt* - Mark
Summerfield. Prentice Hall, 2007, 584 pages.**

COMPUTAÇÃO 1 - PYTHON

AULA 1 TEÓRICA: INTRODUÇÃO

SLIDES BASEADOS NOS TRABALHOS:

AULA1 TEÓRICA DO DCC UFRJ

**AULA DE INTRODUÇÃO DO CLAUDIO ESPERANÇA DO
PESC**

**AULA DE INTRODUÇÃO DO CURSO PYTHON FOR
EVERYBODY DO CHARLES SEVERANCE, UNIVERSITY
OF MICHIGAN SCHOOL OF INFORMATION**