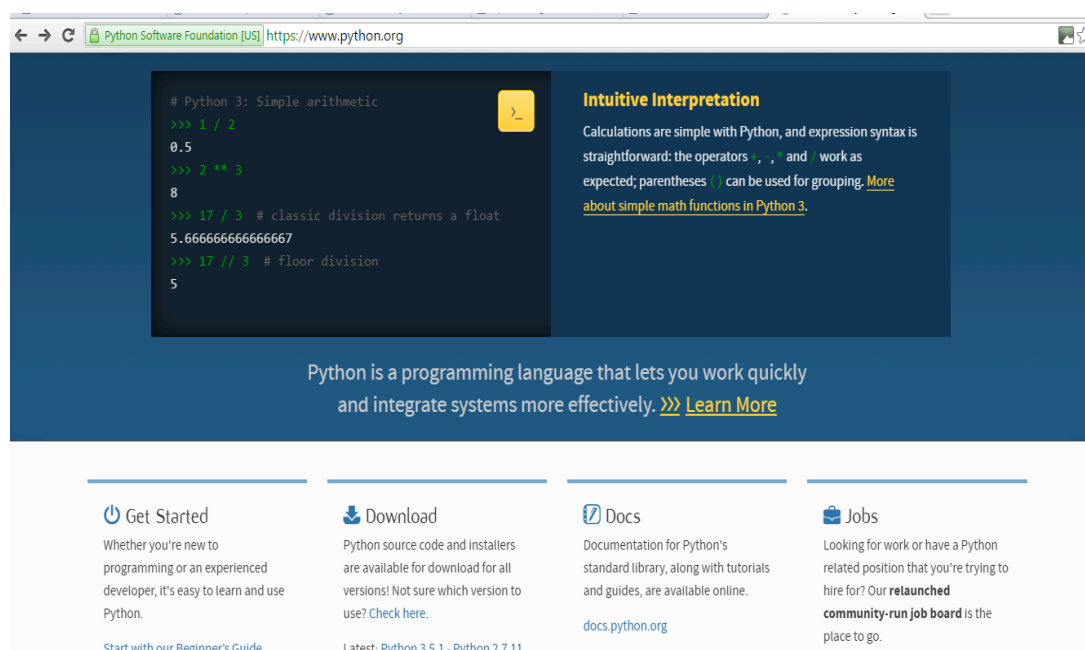


## Passo-a-passo para desenvolver um Programa usando a Linguagem Python

### 1) Para instalar o ambiente Python no seu computador:

É necessário instalar o **Interpretador da Linguagem Python**. O interpretador Python não vem instalado com o Microsoft Windows, ou seja, deve-se fazer download da Internet. No Mac OS X ou no Linux, provavelmente já esteja instalado juntamente com o Sistema Operacional.

Python é Software Livre e pode ser baixado tranquilamente e gratuitamente no site <http://www.python.org>. Na Figura 1, pode ser observada a opção DOWNLOAD na segunda coluna abaixo, na página web oficial da linguagem.



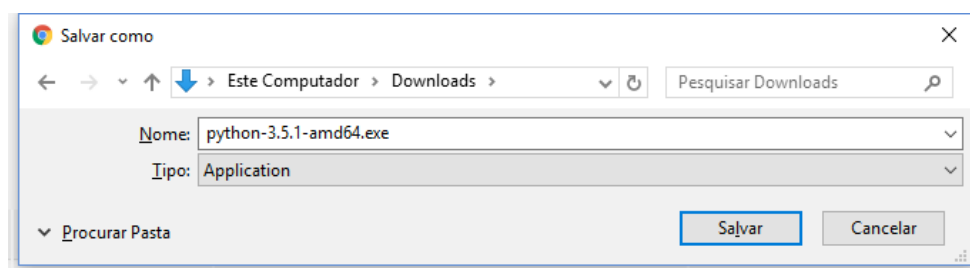
**Figura 1:** Site da Python Foundation – Site Oficial da Linguagem Python

A seguir, na Figura 2, aparecem as opções para download, de acordo com o Sistema Operacional utilizado. Não esqueça também, de verificar se seu sistema é 32 ou 64 bits, antes de fazer a instalação (no Windows, essa informação está em Painel de Controle – Sistema).

Files					
Version	Operating System	Description	MD5 Sum	File Size	GPG
<a href="#">Gzipped source tarball</a>	Source release		be78e48cdfc1a7ad90efff146dce6cfe	20143759	<a href="#">SIG</a>
<a href="#">XZ compressed source tarball</a>	Source release		e9ea6f2623ffcd871b7b19113fde80	14830408	<a href="#">SIG</a>
<a href="#">Mac OS X 32-bit i386/PPC installer</a>	Mac OS X	for Mac OS X 10.5 and later	c66bddc2a4a560496e68fb16600143a7	25709672	<a href="#">SIG</a>
<a href="#">Mac OS X 64-bit/32-bit installer</a>	Mac OS X	for Mac OS X 10.6 and later	1c41a4bd7e6644b8680fc2508cebf1ed	24038487	<a href="#">SIG</a>
<a href="#">Windows help file</a>	Windows		cc3e73cbe2d71920483923b731710391	7719456	<a href="#">SIG</a>
<a href="#">Windows x86-64 embeddable zip file</a>	Windows	for AMD64/EM64T/x64, not Itanium processors	b07d15f515882452684e0551decad242	6832590	<a href="#">SIG</a>
<a href="#">Windows x86-64 executable installer</a>	Windows	for AMD64/EM64T/x64, not Itanium processors	863782d22a521d8ea9f3cf41db1e484d	29627072	<a href="#">SIG</a>
<a href="#">Windows x86-64 web-based installer</a>	Windows	for AMD64/EM64T/x64, not Itanium processors	6a14ac8dfb70017c07b8f6cb622daa1a	963360	<a href="#">SIG</a>
<a href="#">Windows x86 embeddable zip file</a>	Windows		6e783d8fd44570315d488b9a9881ff10	6023182	<a href="#">SIG</a>
<a href="#">Windows x86 executable installer</a>	Windows		4d6fdb5c3630cf60d457c9825f69b4d7	28743504	<a href="#">SIG</a>
<a href="#">Windows x86 web-based installer</a>	Windows		6dfcc4012c96d84f0a83d00cfd8f8bb8	937680	<a href="#">SIG</a>

**Figura 2:** Página de Download do Ambiente de Programação em Python

Você deve salvar o arquivo de instalação **.exe** (ou **.msi**) em alguma de suas pastas (como por exemplo a pasta Downloads) ou na Área de Trabalho mesmo (Figura 3).



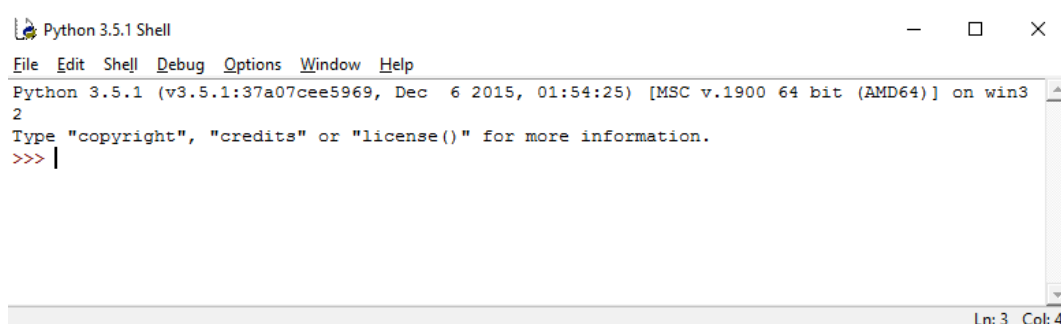
**Figura 3:** Arquivo de Instalação do Python (versão 3.5.1 para amd64 bits)

## 2) Usando o Interpretador Python

IDLE é uma interface gráfica para o interpretador Python que permite também a edição e execução dos programas.

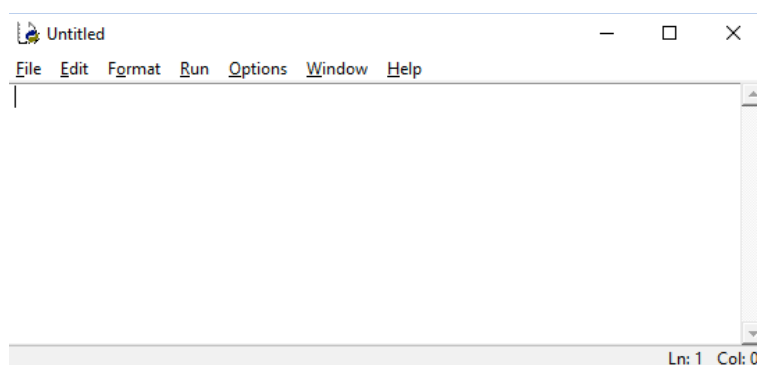
- No Windows, deve ter uma pasta no menu **Iniciar > Programas > Python 3.5 > IDLE**
- No Linux, abra o terminal e digite: **idle-python3.5 &**
- No Mac OS X, abra o terminal e digite: **IDLE3.5 &**

A janela inicial do IDLE versão 3.5.1, no Windows 10 – 64 bits, é mostrada na Figura 4. No Mac OS X, no Linux ou em outra versão do Windows, talvez não seja igual a janela da figura, mas será parecida.



**Figura 4:** Janela Inicial IDLE Python

É possível usar qualquer Editor de Textos para fazer um programa na Linguagem Python e depois apenas usar o Interpretador para Executar o programa, ou então, pode-se usar o próprio Editor do Interpretador, da seguinte forma: na janela do Interpretador, conforme Figura 4 (acima), clique no menu **File >> New File** (ou Ctrl + N). Aparecerá a janela a seguir:



**Figura 5:** Janela do Editor de Textos do IDLE

### 3) Digitando o código-fonte dos programas e salvando

Essa janela do Editor de Textos é parecida com a do Interpretador, mas tem alguns menus diferentes.

É neste ambiente que você deve digitar os códigos-fontes dos programas e, depois de **Salvar** o arquivo através do menu **File >> Save** (ou Ctrl + S).

Ao salvar um arquivo estando neste editor do IDE, deve-se escolher apenas o nome do arquivo, pois a extensão aparecerá automaticamente. Os programas em Python tem a extensão **.py**

### 4) Executando os Programas

Para executar seus programas, você deve ir no menu, na opção **Run >> Run Module** (ou F5).

### 5) Comandos básicos

Escriver na Tela: **print**

Ler do Teclado: **input**

6) **Tipos de Dados:** quando armazenamos uma informação em uma variável, devemos informar o tipo de informação.

Números inteiros: **int**

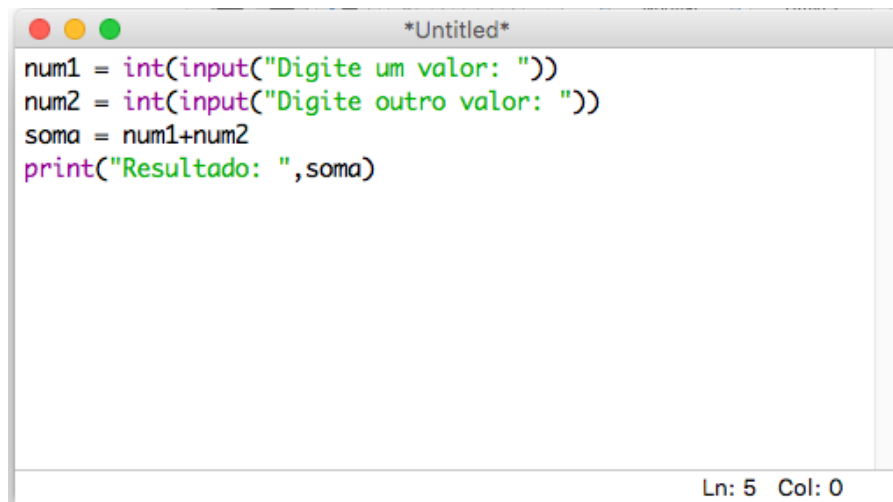
Números “quebrados”: **float**

Caracteres (string): **str**

## 7) Exemplos de Programas em Python

### 1ºPasso) Digitar o código-fonte do programa: File – New File

**exemplo1.py:** soma dois números e escreve o resultado na tela.



```
num1 = int(input("Digite um valor: "))
num2 = int(input("Digite outro valor: "))
soma = num1+num2
print("Resultado: ",soma)
```

Ln: 5 Col: 0

### 2ºPasso) Salvar o Arquivo: File – Save (escolher o local e o nome do arquivo: **exemplo1.py**)

### 3ºPasso) Executar o Programa: Run – Run Module (F5)



```
Python 3.6.0 (v3.6.0:41df79263a11, Dec 22 2016, 17:23:13)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: /Users/flaviacarvalho/Desktop/exemplo1.py =====
Digite um valor: 4
Digite outro valor: 5
Resultado: 9
>>> |
```

Ln: 9 Col: 4

- 8) **exemplo2.py**: soma dois números, escreve o resultado na tela e depois testa se o resultado é maior ou igual a 10 (TESTE **if** - **else**).

```
exemplo2.py - /Users/flaviacarvalho/Desktop/exemplo2.py (3.6.0)
num1 = int(input("Digite um valor: "))
num2 = int(input("Digite outro valor: "))
soma = num1+num2
print("Resultado: ",soma)
if soma >= 10:
    print("Resultado maior ou igual a 10!")
else:
    print("Resultado menor que 10!")
|
```

Ln: 9 Col: 0

```
Python 3.6.0 Shell
Python 3.6.0 (v3.6.0:41df79263a11, Dec 22 2016, 17:23:13)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: /Users/flaviacarvalho/Desktop/exemplo2.py =====
Digite um valor: 6
Digite outro valor: 7
Resultado: 13
Resultado maior ou igual a 10!
>>>
```

Ln: 10 Col: 4

```
Python 3.6.0 Shell
Python 3.6.0 (v3.6.0:41df79263a11, Dec 22 2016, 17:23:13)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: /Users/flaviacarvalho/Desktop/exemplo2.py =====
Digite um valor: 2
Digite outro valor: 3
Resultado: 5
Resultado menor que 10!
>>> |
```

Ln: 10 Col: 4

- 9) **exemplo3.py**: soma dois números, escreve o resultado na tela, testa se o resultado é maior ou igual a 10 e pergunta se o usuário quer fazer outra soma, ou seja, se quer repetir todo programa novamente (LAÇO **while**).

```
exemplo3.py - /Users/flaviacarvalho/Desktop/exemplo3.py (3.6.0)
resposta = str(input("Gostaria de somar dois numeros S/N? "))
while resposta == "s":
    num1 = int(input("Digite um valor: "))
    num2 = int(input("Digite outro valor: "))
    soma = num1+num2
    print("Resultado: ",soma)
    if soma >= 10:
        print("Resultado maior ou igual a 10!")
    else:
        print("Resultado menor que 10!")
    resposta = str(input("Deseja fazer nova soma S/N? "))
print("FIM")
```

Ln: 14 Col: 0

```
Python 3.6.0 Shell
Python 3.6.0 (v3.6.0:41df79263a11, Dec 22 2016, 17:23:13)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: /Users/flaviacarvalho/Desktop/exemplo3.py =====
Gostaria de somar dois numeros S/N? s
Digite um valor: 4
Digite outro valor: 5
Resultado: 9
Resultado menor que 10!
Deseja fazer nova soma S/N? s
Digite um valor: 7
Digite outro valor: 9
Resultado: 16
Resultado maior ou igual a 10!
Deseja fazer nova soma S/N? n
FIM
>>> |
```

Ln: 18 Col: 4

## 10) Exercícios Práticos

**Exercício1)** Faça um programa usando a Linguagem Python para ler um valor (do teclado) e escrever (na tela) o seu antecessor.

**Exercício2)** Faça um programa usando a Linguagem Python para ler as dimensões de um retângulo (base e altura), calcular e escrever a área do retângulo.

**Exercício3)** Faça um programa que converta metros para centímetros. Lembrando que 1m = 100cm.

**Exercício4)** Faça um programa que leia o valor do lado de um quadrado, calcule a área do quadrado (lado\*lado) e em seguida mostre o dobro desta área para o usuário.

**Exercício5)** Faça um programa que pergunte quanto a pessoa ganha por hora (salário por hora) e o número de horas trabalhadas no mês. Calcule e mostre o total do seu salário no referido mês.

**Exercício6)** Faça um programa para ler o salário mensal atual de um funcionário e o percentual de reajuste. Calcular e escrever o valor do novo salário.

**Exercício7)** As maçãs custam R\$ 1.30 cada se forem compradas menos de uma dúzia, e R\$ 1.00 se forem compradas pelo menos 12. Escreva um programa que leia o número de maçãs compradas, calcule e escreva o custo total da compra.

**Exercício8)** Faça um programa para pedir que a pessoa digite “f” para feminino ou “m” para masculino. O programa deve aceitar somente as letras “f” ou “m”, se o usuário digitar outra letra, o programa não deve aceitar e pedir para o usuário digitar novamente, informando ao usuário a mensagem “Resposta errada! Digite ‘F’ ou ‘M’”.

**Exercício9)** Faça um programa para ler as notas da 1a. e 2a. avaliações de um aluno, calcular e imprimir a média (simples) desse aluno. Ao final, perguntar se o usuário deseja fazer “NOVO CÁLCULO (S/N)?”. Se for respondido “S” deve retornar e executar um novo cálculo da média, caso contrário deverá encerrar o programa.