

Aula 00 – Apresentação do Python

A LINGUAGEM DE PROGRAMAÇÃO PYTHON

Python é uma linguagem de programação. Isso significa basicamente 2 coisas:

- 1) existem regras que determinam como as palavras são dispostas, já que é uma linguagem;
- 2) o texto descreve instruções para o computador realizar tarefas.

Ou seja, podemos escrever um documento - que chamamos de código fonte - em Python para o computador ler e realizar nossos desejos e tarefas.

Python tem algumas características interessantes:

a) Linguagem interpretada	<ul style="list-style-type: none"> ✓ ou seja, o interpretador do Python executa o código fonte diretamente, traduzindo cada trecho para instruções de máquina; ✓ Isto é, o código Python é interpretado por um programa escrito na linguagem C e C++. ✓ Este programa que entende e executa os códigos Python, é chamado de interpretador Python, ou então, a Máquina Virtual Python. ✓ Todas as vezes em que executarmos um programa escrito em Python, este só será executado se a Máquina Virtual do Python estiver instalado no computador em que está sendo executado.
b) Alto nível	<ul style="list-style-type: none"> ✓ ou seja, o interpretador se vira com detalhes técnicos do computador. ✓ Assim, desenvolver um código é mais simples do que em linguagens de baixo nível, nas quais o programador deve se preocupar com detalhes da máquina;
c) Propósito geral	<ul style="list-style-type: none"> ✓ ou seja, podemos usar Python para desenvolver programas em diversas áreas. ✓ Ao contrário de linguagens de domínio específico, que são especializadas e atendem somente a uma aplicação específica;
d) em tipos dinâmicos	<ul style="list-style-type: none"> ✓ ou seja, o interpretador faz a magia de descobrir o que é cada variável.

Apesar de ser simples de aprender, Python é uma linguagem bastante poderosa e flexível. Essa combinação resulta em um rico ecossistema que melhora a produtividade dos desenvolvedores.

O [Python](#) é uma linguagem ímpar, projetada para ser simples e desenhada para ser eficaz. Seu uso vem crescendo exponencialmente e atualmente, diversas universidades ao redor do mundo estão ensinando Python como a primeira linguagem de programação. Isso porque, a linguagem possui uma fácil leitura e por conseguinte, acaba tendo uma fácil compreensão.



Criada em 1991 pelo holandês [Guido van Rossum](#), Python é uma linguagem de alto nível e de propósito geral. A linguagem é conhecida por ter uma sintaxe muito simples e por ser amigável para iniciantes na programação.

Ele decidiu usar esse nome porque estava lendo um roteiro de **Monty Python**, um grupo de comédia inglês da década de 1970. A documentação oficial do Python contém muitas referências aos filmes e personagens desse grupo.

Um resumo das versões desta linguagem:

- Versão 1.0 foi publicada em Janeiro de 1994.
- Versão 2.0 foi publicada em Outubro de 2000.
- Versão 3.0 foi publicada em Dezembro de 2008. 3.7 é a versão mais recente.

Devido a facilidade em codificar e tendo como resultado um código limpo e legível, o Python acaba se tornando uma linguagem fácil de dar manutenção

Python pode ser usada em muitos tipos de aplicações,

- a) desde pequenos softwares de linha de comando,
- b) jogos com a biblioteca **PyGame**,
- c) aplicações web com o framework **Django**,
- d) sistemas de inteligência artificial e **data science** e
- e) até mesmo aplicativos para celulares com o framework **Kivy**.
- f) mais usada no meio da ciência de dados e da inteligência artificial.

Vale a pena ressaltar que essas duas áreas estão em crescente expansão, o que faz a demanda por programadores Python, no mercado de trabalho, aumentar bastante.

Pelo seu poderio e sua facilidade de escrita, a linguagem é extensivamente usada por grandes empresas, como:

- ✓ Wikipedia,
- ✓ Google(Youtube),
- ✓ Amazon,
- ✓ Facebook,
- ✓ Dropbox,
- ✓ IBM,
- ✓ Yahoo,
- ✓ Mozilla,
- ✓ Instagram,
- ✓ CERN,
- ✓ NASA,
- ✓ Spotify
- ✓ Etc.

CARACTERÍSTICAS DO PYTHON

O Python é uma linguagem que proporciona alta produtividade, isto é, se consegue escrever grandes aplicações em pouco tempo e, com poucas linhas de código. Essa característica decorre da maneira como a linguagem está projetada e, dos princípios empregados pela comunidade na construção das bibliotecas. Um programa em Python, muitas vezes, parece a um texto escrito em Inglês. E também, trechos de código Python, podem ser lido por leigos e estes, até conseguirão entender, a maior parte do que o programador está mandando o computador fazer.

Em Python conseguimos trabalhar com os conceitos de **Programação Orientada a Objetos (POO)** de maneira simples. Conseguimos facilmente representar um tipo de dado abstrato, inclusive, podemos criar um novo tipo de dados através da junção de outras estruturas, utilizando, por exemplo, a herança ou herança múltipla.

Diferentemente de outras linguagens, em Python, não há restrições quanto ao acesso a membros de classes. A linguagem parte do princípio que o programador sabe o que está fazendo, ao contrário das linguagens C++ e Java, que partem do princípio que as estruturas devem ser protegidas de uma possível má utilização.

1.2 Exemplos

Vamos ver alguns exemplos sobre o uso de Python no mundo real.

1.2.1 BitTorrent

O protocolo Torrent é muito utilizado para transferir quantidades grandes de dados para diversos computadores. O primeiro programa a implementar esse protocolo foi desenvolvido inteiramente em Python, pela BitTorrent, Inc.!

1.2.2 Django

Django é um conjunto de pacotes para desenvolvimento web. E é baseado em Python :)
Um objetivo de Django é desenvolver facilmente websites complexos e que lidam com grandes bancos de dados.

Alguns sites desenvolvidos em Django:

- ✓ Instagram,
- ✓ The Washington Times,
- ✓ Disqus,
- ✓ Mozilla,
- ✓ National Geographic.

1.2.3 Dropbox

O popular serviço de armazenamento de dados em Nuvem Dropbox tem diversas partes da infraestrutura feita em Python.

O aplicativo para computadores é feito em Python e grande parte da infra estrutura dos servidores deles também é!

1.2.4 Estudo sobre erupções solares

Não somente a indústria utiliza Python, muitos pesquisadores utilizam em diversas áreas científicas. É possível de modo bem simples estudar as erupções solares desde 1992 até hoje. O Observatório Real da Bélgica tem um banco de dados sobre o número de manchas solares, e disponibilizam online para estudos. Veja como é o código para visualizar a atividade solar desde 01/01/1992 em cada parte (norte e sul) do Sol:

```
import pandas as pd # isso aqui gerencia os dados
import matplotlib.pyplot as plt # isso aqui permite fazer gráficos

# pega os dados solares de WDC-SILSO, Royal Observatory of Belgium, Brussels
sun = pd.read_table('http://sidc.oma.be/silso/INFO/sndhemcsv.php', sep=';', encoding
    ,→= "ISO-8859-1", header=-1)

# faz o gráfico de cada hemisfério do sol
fig = plt.figure()
plt.scatter(sun[3], sun[5], label='Norte', alpha=0.5)
plt.scatter(sun[3], sun[6], label='Sul', alpha=0.5)

plt.title("Atividade solar diária\n")

plt.ylabel('Número de manchas solares')
```

```
plt.xlabel('Ano')  
plt.legend(loc='upper right')  
plt.show()
```

Listagem 1: Estudo sobre erupções solares

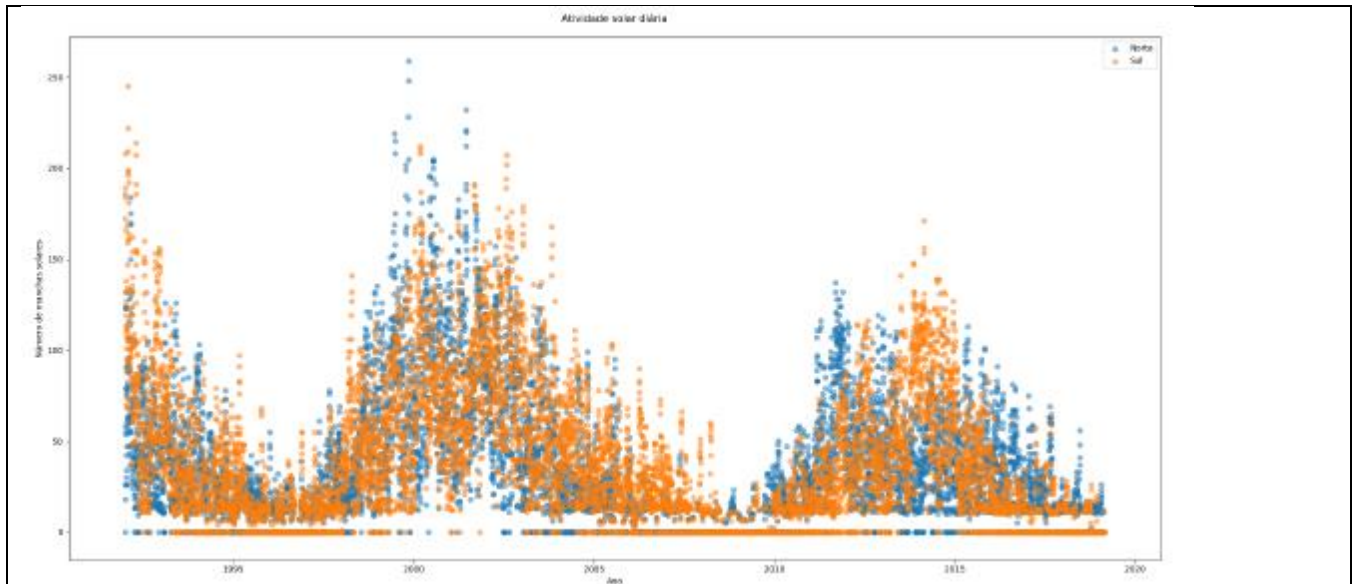


Figura 1: Erupções solares desde 1992, separadas por hemisférios (norte e sul) do Sol.

1.2.5 Física de Partículas

O premio Nobel de 2013 em Física foi para os cientistas que estudaram como as partículas elementares adquirem massa, conhecido como Mecanismo de Higgs. Uma nova partícula foi descoberta em 2012: o Bóson de Higgs. Você pode ler todos os detalhes super técnicos neste artigo aqui que os cientistas do CERN publicaram.

Caso você esteja interessado apenas na parte computacional, você pode refazer toda a análise dos dados experimentais utilizando Python!

Existe uma apostila online que ensina passo a passo como obter os dados experimentais e simulações teóricas para reproduzir o gráfico a seguir.

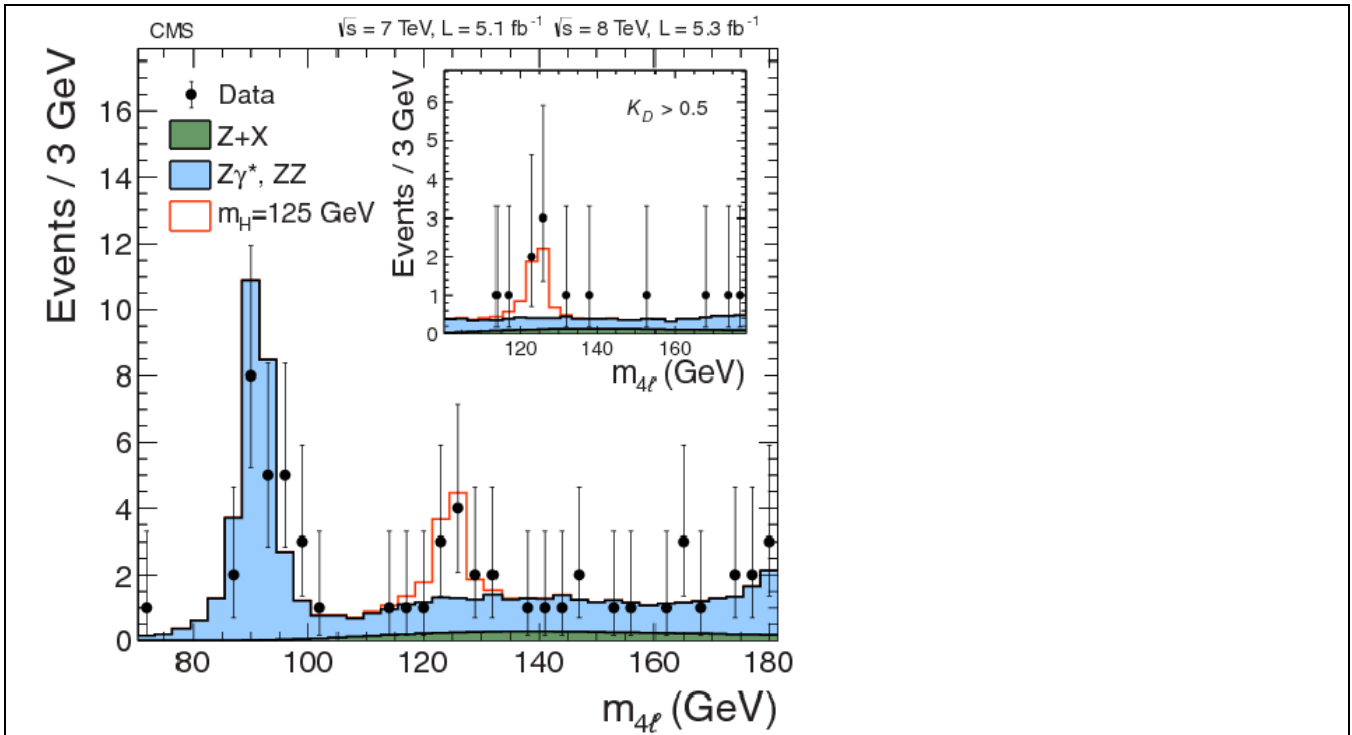


Figura 2: Reprodução dos resultados científicos publicados em 2012 sobre a descoberta do Bóson de Higgs. Gráfico obtido com o código publicado na apostila online¹⁰, que utiliza dados públicos do experimento CMS, no CERN.

1.2.6 The Sims

O jogo The Sims tem partes feitas em Python3. Isso permite o desenvolvimento de mods para o jogo em Python \o/



Figura 3: Capa do jogo The Sims 4, da Electronic Arts.

Por que Python 3?

O ano de 2014 foi um ano chave na adoção de Python3, por grande parte da comunidade. Muitos projetos relevantes foram portados ou lançaram versões compatíveis nesse ano. A família 3.4 tem maior aceitação que as anteriores da versão 3 e, inclusive, foi adotada como versão padrão em alguns sistemas operacionais. Python 3 nunca chamou tanta atenção como agora, então nada mais justo que um livro o tenha como assunto base.

É uma linguagem de uso gratuito e de código-fonte aberto, compatível com os principais sistemas operacionais, como: Linux, OSX, Windows, BSDs etc. Ela conta com uma vasta biblioteca padrão e documentação que possibilitam que muitas coisas sejam feitas sem dependências adicionais.

2) SO e MÁQUINA VIRTUAL PYTHON

O Python é composto por um conjunto de bibliotecas e um arquivo executável, no caso, o interpretador da linguagem Python. Quando instalamos o Python em nosso computador, o que fazemos de fato, é instalar um programa que executa código Python e adicionar, no Path do sistema, o local em que o interpretador da linguagem Python está contido.

O programa executável do Python, isto é, o interpretador, é também chamado de **Máquina Virtual do Python**. A expressão [Máquina Virtual](#) é decorrente do fato que o programa Python propriamente dito, executa instruções que juntas, formam programas, e essas instruções são executadas pelo interpretador de maneira análoga as instruções de arquivos executáveis que são executadas pelo Computador.

BAIXANDO E INSTALANDO O PYTHON

O Python é distribuído gratuitamente e toda sua documentação pode ser encontrada no site da organização Python: <https://www.python.org/>.

Para fazermos o download do Interpretador Python, temos que acessar o site, em seguida clicar no menu Downloads. Feito isso, a página de download será aberta, como pode ser visto na imagem a seguir:



Caso queiras ir [direto à página de download do Python](#), utilize este link.

Baixe o instalador conforme o Sistema Operacional que vc está utilizando! Também é necessário escolher entre a versão de 32bits ou 64bits, conforme a arquitetura do seu processador!

USUÁRIOS LINUX / OUTRAS DISTROS

Nesta aula não foi contemplado a instalação do Python em outros sistemas operacionais. Usuários do MacOS e do Ubuntu já possuem o Python instalado de fábrica, e também, as maiores e principais distros Linux, já vem com o Python instalado por padrão.

QUAL O PROBLEMA ENTÃO?

A versão do Python que está instalada. Nesse curso, trabalhamos com a versão 3x do Python, mas a maioria dos sistemas operacionais vem com o Python 2.7 de fábrica.

QUAL A RAZÃO DE HAVER 2 VERSÕES 2.7.xxxx E 3.xxx.xxx?

A versão 3 do Python sofreu mudanças estruturais significantes, a mais simples, podemos dizer, num primeiro momento, é o fato de você ser obrigado a utilizar parêntesis na função `print()`. Essa é uma única mudança, há diversas outras, principalmente na parte de Orientação a Objetos e a estrutura interna das classes.

Com o objetivo de evitar o **caos** e o trauma de toda e qualquer mudança abrupta, optou-se, num primeiro momento, por utilizar 2 versões do Python em paralelo. Por isso é que temos a versão 2.7.x e a versão 3.x. O Python 2.7.x suporta a maior parte das mudanças do Python 3, mas o contrário não é verdade. Ou seja, o Python 3 não suporta o Python 2.7.

Mas não se preocupe com isso nesse momento! Porém, não esqueça: **vamos trabalhar com a versão 3.x do Python neste curso!**

ONDE SABER MAIS

Existem 2 versões OFICIAIS: a versão 2.7 e a versão 3x do Python. A versão 2.7 é a que será descontinuada em 2022 e a mesma, vem, gradativamente, sofrendo diversas modificações. Assim, o que acontece, é que não haverá uma mudança abrupta, e sim, ocorrerá que a versão 2.7 se "encontrará" com a versão 3.x.x.

A cada nova versão do Python 2.7, há mudanças que devem ser feitas em código legado a fim de manter a compatibilidade. Logo, o que acontece é, que ambas versões caminham juntas e paralelas. As mudanças a nível de código são implementadas nas duas versões, porém, como já dito, a versão 2.7 não funciona na versão 3.x.

2.1 Linux

Provavelmente você já tem o Python instalado e configurado. Para ter certeza que ele está instalado e descobrir qual versão, abra um terminal e execute o comando:

```
$ python --version
```

Se o resultado do comando for Python 3.6.5 (ou alguma versão igual ou superior a 3.5) o Python já está instalado corretamente.

Caso o resultado do comando anterior tenha sido Python 2.7.13 (ou qualquer versão do Python 2) tente rodar o seguinte comando, pois seu computador pode ter ambas versões 2 e 3 instaladas:

```
$ python3 --version
```

Caso tenha aparecido a mensagem `bash: python: command not found`, você pode instalá-lo da seguinte maneira:

2.1.1 No Ubuntu e Debian

```
$ sudo apt install python3
```

2.1.2 No ArchLinux

```
$ sudo pacman -Sy python
```

2.2 Mac OS X

Obtenha o instalador na sessão de downloads para Mac OS X do Python19. Clique duas vezes no `Python.mpkg` para abrir o instalador.

Para ter certeza que ele está instalado e descobrir qual versão, abra um terminal e execute o comando:

```
$ python --version
Python 3.6.5
```

2.3 Windows

Obtenha o arquivo de instalação última versão compatível com a arquitetura do seu computador no site oficial do Python. O arquivo x86 provavelmente funcionará para todos computadores.

A seguir, execute o instalador e uma imagem similar a essa aparecerá:



Figura 1: Deve ser selecionada a opção Add Python 3.6 to PATH e depois continuar a instalação até o fim.

<https://www.python.org/downloads/>

3) Ambientes de Desenvolvimento

RECURSOS DA LINGUAGEM

São várias as características do Python que o tornam uma linguagem tão promissora e, que atrai a cada dia, mais desenvolvedores. A seguir, foram elucidados alguma dessas características e junto, uma breve descrição.

Características	Descrição
fácil de aprender	<ul style="list-style-type: none"> ✓ O Python é uma linguagem fácil de ser aprendida e poderosa para trabalharmos. ✓ Ela possui uma sintaxe limpa e clara, como também, contém um conjunto de bibliotecas estáveis e bem estruturadas.
fácil leitura e compreensão	<ul style="list-style-type: none"> ✓ A sintaxe da linguagem é minimalista, isso é, mantém somente o necessário, o que torna o código escrito, muitas vezes, a um texto em inglês.

fácil manutenção	✓ Em decorrência da simplicidade sintática e da excelente estruturação das bibliotecas, a manutenção de códigos, seja aquele que desenvolvemos ou mesmo de terceiros, muito mais fácil e compreensível.
multiplataforma	✓ O interpretador do Python é escrito com a Linguagem C e C++, assim, o mesmo pode ser portado a todas as plataformas que possuam compiladores para a linguagem. ✓ Tendo em vista que o C++ é a linguagem mais difundida e a base de praticamente toda a informática, temos compiladores nativos ou portados para quase todas as plataformas existentes.
modo interativo	✓ Sendo o Python uma linguagem interpretada, foi possível o desenvolvimento de ferramentas interativas, isto é, ferramentas majoritariamente em linha de comando, onde podemos executar instruções e analisar a saída.

Para desenvolvermos em Python, podemos utilizar qualquer editor de textos ou então, o simples Bloco de Nota do Windows. Por padrão, o interpretador do Python executa textos [códigos] compostos por caracteres ASCII e/ou Unicode e por isso, é indiferente o programa utilizado para escrever os códigos. Isso porque, um arquivo de código é basicamente um arquivo de texto com a extensão *.py.

Atualmente, é comum utilizarmos programas para construir programas, isto é, utilizamos softwares que são desenvolvidos para agilizar e automatizar nossas tarefas de programação. Estes softwares são conhecidos como IDE (**Integrated Development Environment**), que numa tradução livre significa, Ambiente de Desenvolvimento Integrado.

Existem centenas de IDE no mercado, muitas gratuitas, outras pagas. Os recursos que estas oferecem vão desde a compilação e execução automática do código, até o gerenciamento e geração de código automatizado.

IDE (Integrated Development Environment - Ambiente de Desenvolvimento Integrado, em português) é um editor de texto que possui ferramentas e recursos que facilitam a vida do programador.

Entre as ferramentas e recursos, podemos citar:

- Identificar quais variáveis foram declaradas.
- Identificar erros no código.
- Personalizar o ambiente de trabalho.
- Ocultar parte do código para melhor visualização.

Para programarmos em Python, temos diversas opções de IDE disponíveis, a maioria é gratuita ou então, possui uma versão livre. Inclusive, existe IDE para desenvolvimento em Python escritas com a própria linguagem Python, como o é o caso do NINJA IDE, uma excelente ferramenta, multiplataforma, gratuita e open source.

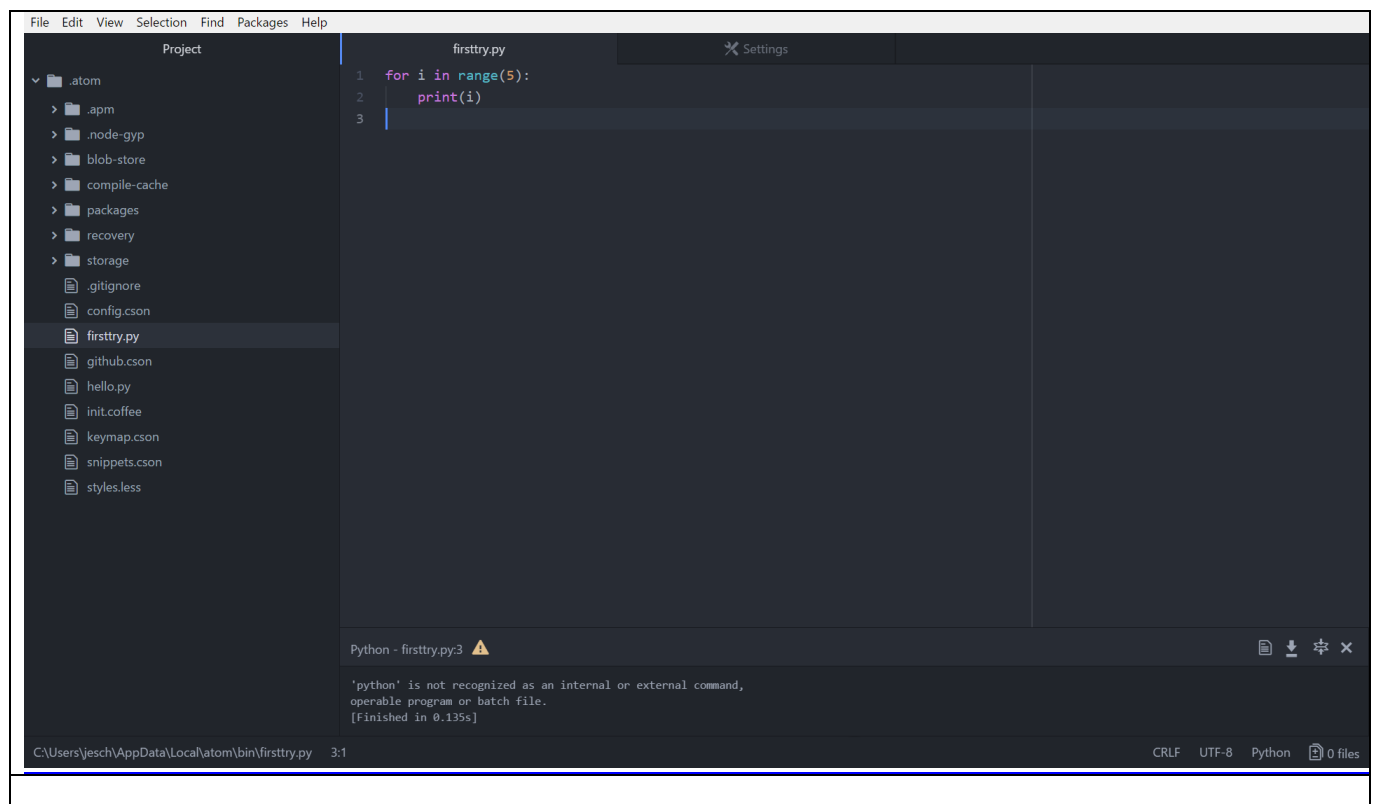
3.1 Ambientes gráficos

3.1.1 ATOM

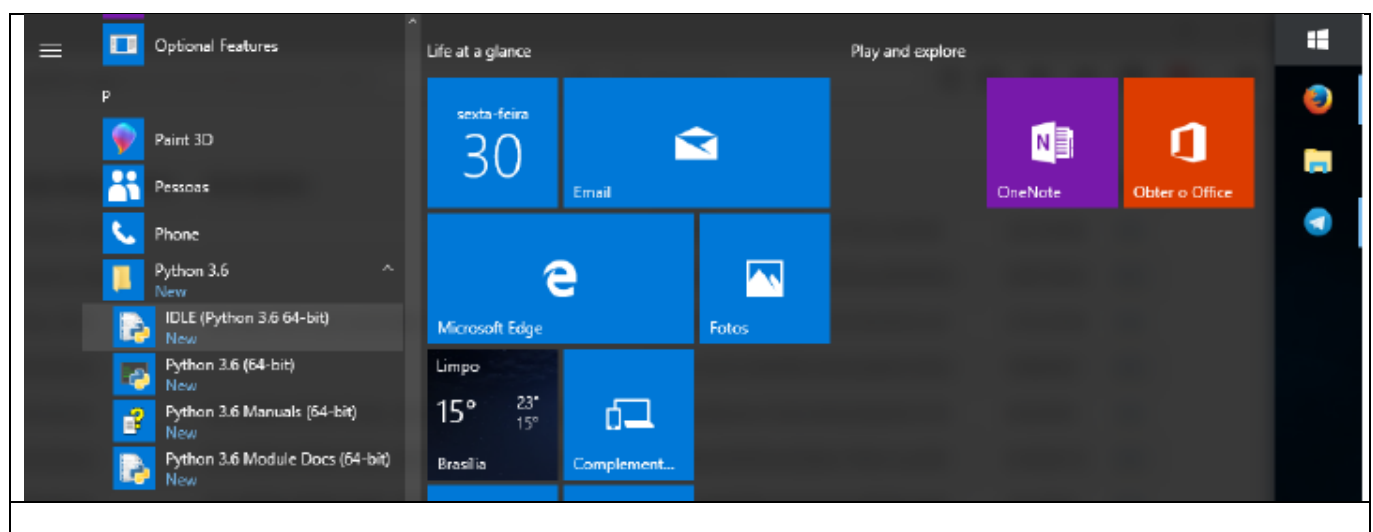
O programa ATOM é um IDE open-source que apresenta diversos pacotes para personalizar.

No site oficial do [ATOM21](#), você encontrará um link para a Documentação do programa. Na documentação, é possível acessar o [manual22](#) que mostrará passo a passo como instalar o programa (tanto para Windows como para Linux).

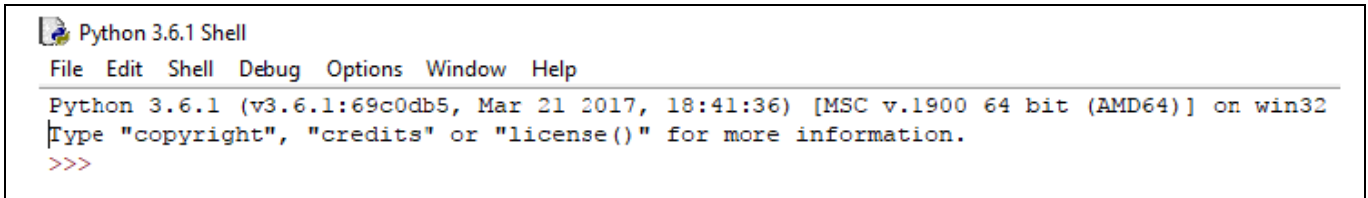
Caso você tenha alguma dúvida, é aconselhável entrar na seção de [discussão23](#). Nessa página, você encontrará respostas para diversas dúvidas, e possivelmente, para a sua.

Links:<https://atom.io><https://flight-manual.atom.io/getting-started/sections/installing-atom/#platform-windows><https://discuss.atom.io/>**3.1.2 IDLE**

Para usuários Windows é recomendado utilizar o IDLE. Ele é composto pelo interpretador do Python e um editor de texto para criar programas, e já vem junto com o Python. Após seguir o [Guia de instalação do Python](#) (Página 21), o menu inicial deve estar da seguinte forma:



Ao abrir o IDLE (Python 3.X), aparecerá uma janela como na imagem abaixo:

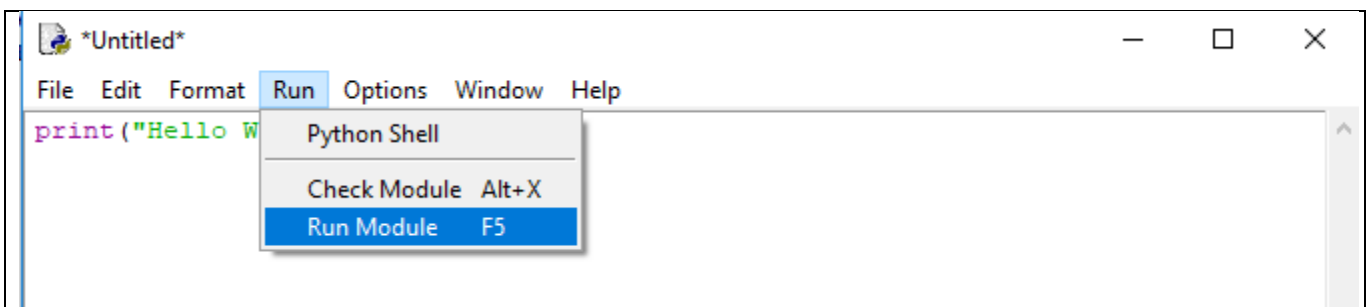
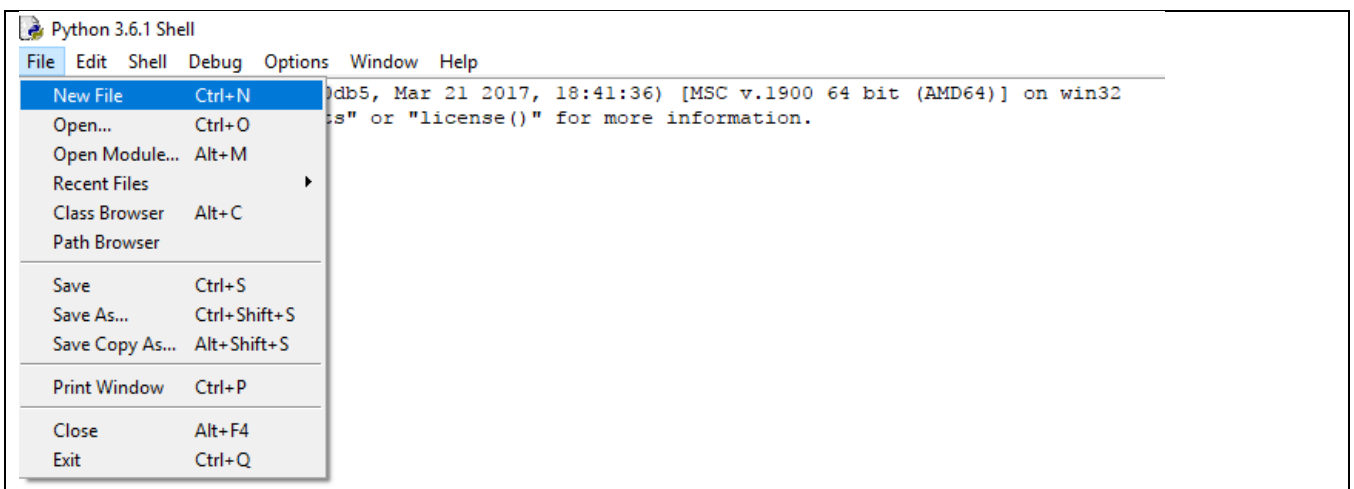


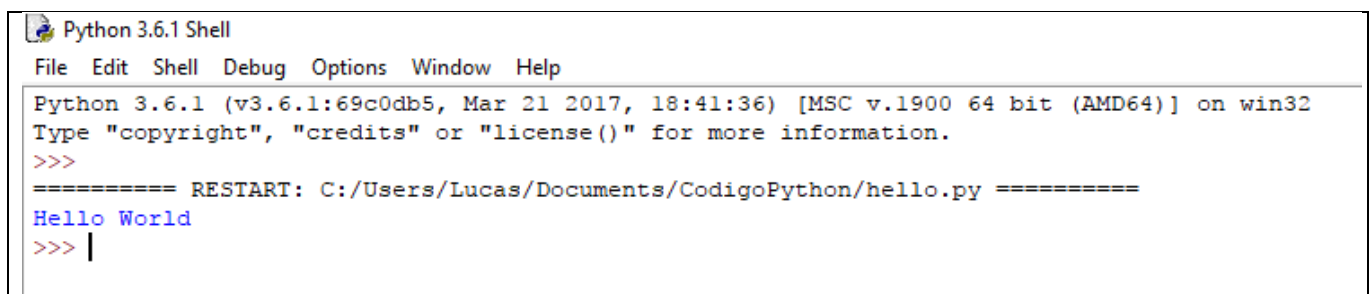
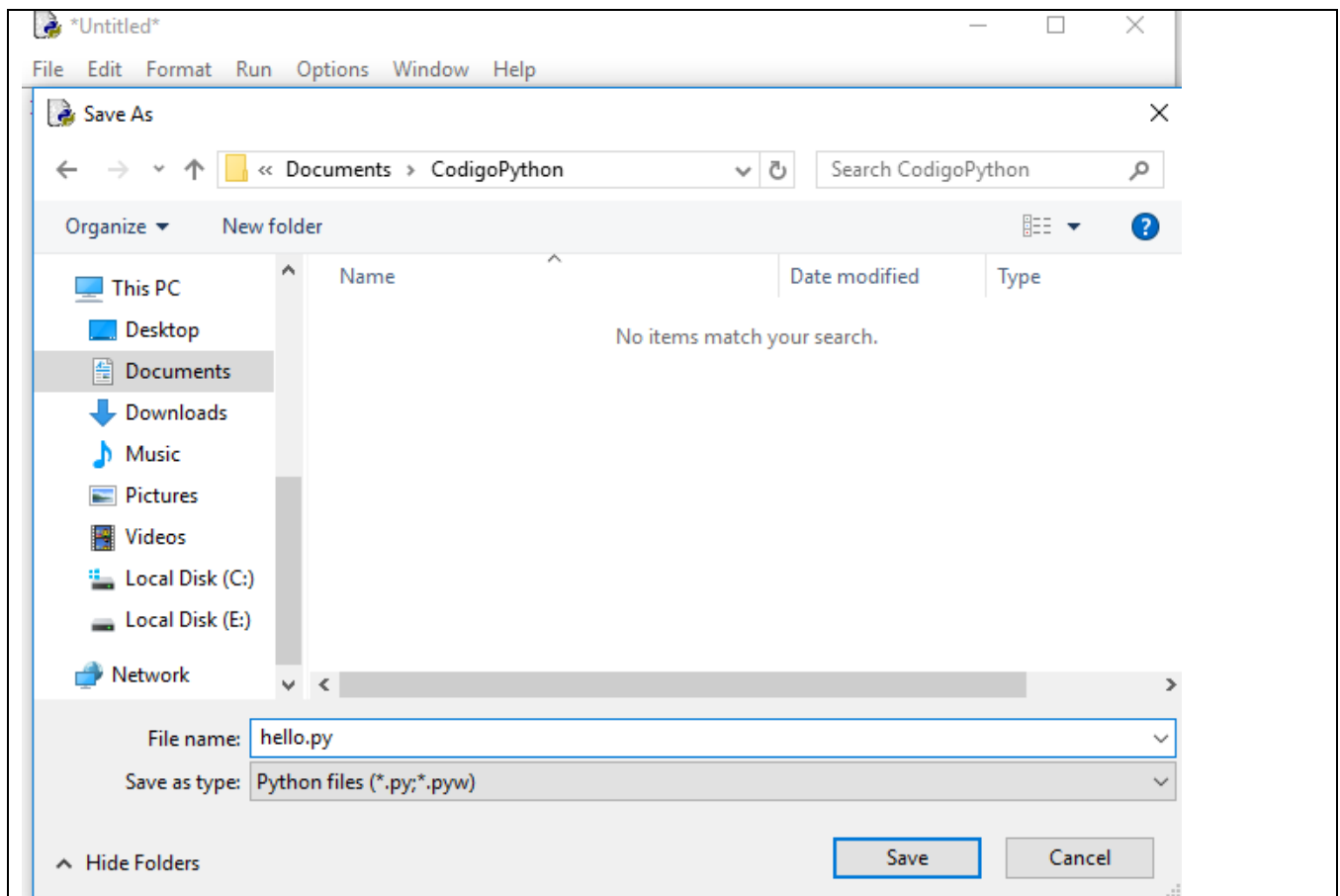
No IDLE é possível digitar comandos para o interpretador do Python e, também, é possível criar e digitar em um arquivo. Para fazer isso, no menu clique em File -> New File (Ou pressione as teclas Ctrl + N juntas)

Para rodar um programa, clique em Run -> Run Module (Ou aperte a tecla F5)

Caso o arquivo ainda não tenha sido salvo, é necessário salvá-lo antes de executá-lo. Não esqueça de prefixar o nome do arquivo com .py (extensão do Python):

Após isso, o resultado da execução do código deve aparecer na janela anterior do IDLE:





A palavra IDLE é o acrônimo de **Integrated Development and Learning Environment**. O mesmo é um programa escrito em Python para ser utilizado como um ambiente de desenvolvimento integrado para execução de Script's. A aplicação vem junto com a maioria das instalações do Python e está presente na distribuição oficial desde a versão 2.3.

O editor está construído 100% em Python, logo, toda a parte gráfica foi desenvolvida utilizando a biblioteca Tkinter, um bind (versão) da biblioteca Tk disponibilizada nativamente para Python.

De forma geral, podemos dizer que o objetivo do IDLE é proporcionar uma maneira rápida e fácil para o uso das funções e bibliotecas do Python, como também, proporcionar uma plataforma de estudo simples de ser utilizada, disponibilizando num único lugar, todos os recursos e bibliotecas do Python.

REPL

A palavra R.E.P.L é um acrônimo de [Read-Eval-Print Loop](#), também chamado de Interactive Toplevel or Language Shell.

Esse tipo de ferramenta apareceu com a linguagem LISP e fornece uma forma rápida e fácil para testar funções e estudar a linguagem.

Para maiores informações, veja o artigo na [Wikipedia](#).

UBUNTU

Se você estiver utilizando o Ubuntu será necessário instalar o IDLE manualmente. Para isso, abra o terminal e digite:

```
sudo apt-get install idle
```

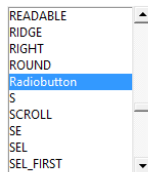
CODE COMPLETE

O IDLE vem evoluindo bastante nas últimas versões. Atualmente, o mesmo possui um **code complete** bastante inteligente, que exibe os parâmetros das função, ou então, todas as funções que podem ser invocadas por determinada classe ou objeto.

Type "copyright", "credits" or "l

```
>>> import tkinter
```

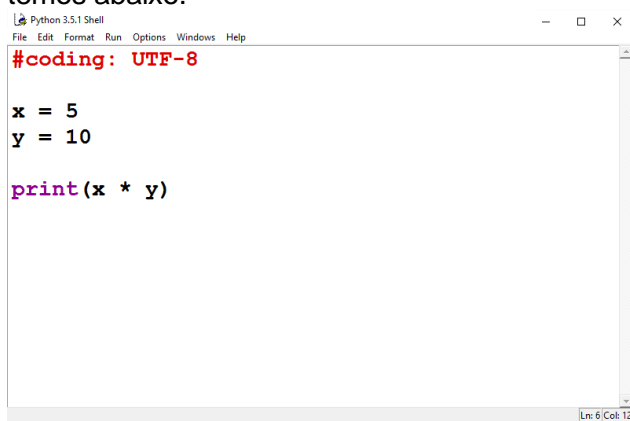
```
>>> tkinter.Radiobutton|
```



As teclas de atalho para chamar o CodeComplete são: Ctrl + Espaço. A forma como chamamos o CodeComplete do IDLE é igual a 99% das IDE's.

EDITOR DE SCRIPTS

O IDLE também possui um editor de Script de várias linhas. Para utiliza-lo acesse o Menu *File -> New File*, ou então, pressione as teclas de atalho Ctrl + N. Em seguida, será aberta uma janela semelhante ao que temos abaixo.

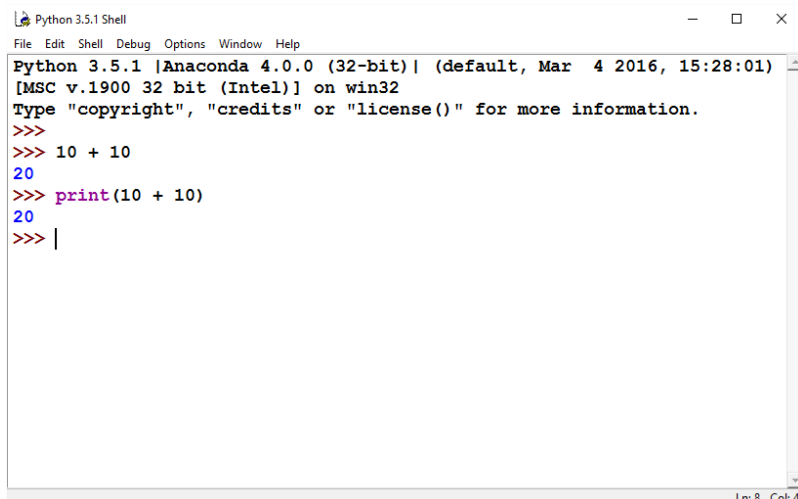


Como podemos ver na imagem, nesta janela é possível criar scripts que contenham várias linhas e em seguida, executar todo o arquivo. Para executar o código, inicialmente, salve o Script e em seguida, vá no Menu *Run -> Run Module*, ou então, pressione a tecla de atalho F5. Feito isso, o Script será executado, caso não haja erros.

FUNCIONAMENTO BÁSICO DO IDLE

Para executar uma instrução no IDLE, basta digitarmos a mesma e pressionarmos a tecla Enter. Se a instrução enviada for válida, algo será impresso na tela ou então, nada acontecerá, salvo que o Cursor se posicionará numa nova linha.

Se por exemplo, executarmos a soma entre 2 números inteiros, o resultado será impresso na linha seguinte a operação, como podemos ver na imagem a seguir:



```
Python 3.5.1 Shell
File Edit Shell Debug Options Window Help
Python 3.5.1 [Anaconda 4.0.0 (32-bit)] (default, Mar  4 2016, 15:28:01)
[MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 10 + 10
20
>>> print(10 + 10)
20
>>> |
```

Como podemos ver na imagem, se mandarmos somar 2 números o resultado é impresso na linha seguinte. O mesmo resultado pode ser obtido, passando a soma como parâmetro pela função *print()*.

```
#coding: utf-8

>>> 10 + 10
20
>>> print(10 + 10)
20
```

Após o resultado ser apresentado, o cursor é liberado para que seja possível entrar outra instrução.

RETORNANDO INSTRUÇÕES ANTERIORES

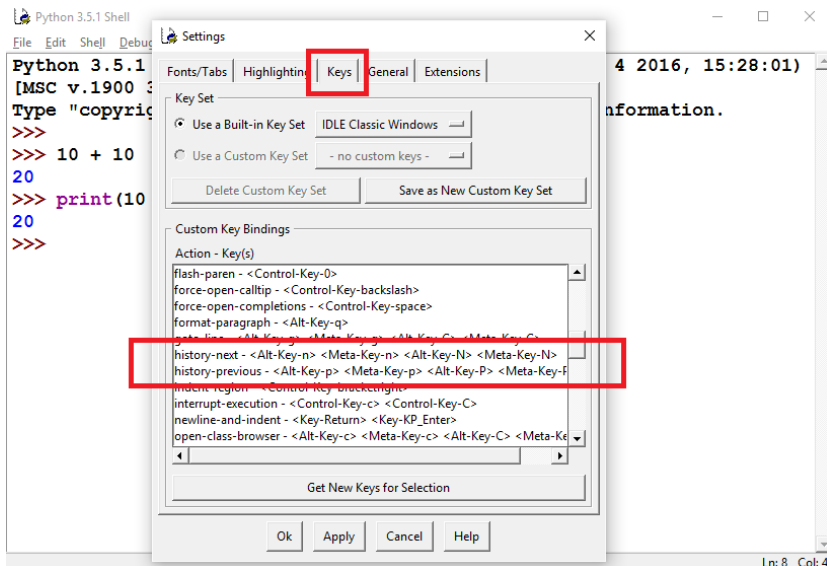
A maior crítica sobre a utilização do IDLE, sem dúvidas, está no fato das teclas de atalho para voltar a última instrução enviada. Normalmente, para retornarmos a última instrução no Prompt de Comando ou então, no Shell do Linux, utilizamos as setas para cima e para baixo. Porém, no IDLE, se utilizarmos as teclas para cima e para baixo, ocorrerá que iremos posicionar o cursor na direção pressionada.

Existe uma razão pela qual as teclas para cima e para baixo não foram utilizadas e também, rapidamente nos acostumamos com as teclas utilizadas por padrão pelo IDLE.

Alt + P - para voltar a última instrução digitada. A letra 'P' vem de "previous". Alt + N - para avançar. O 'N' vem de "Next".

Assim, temos a capacidade de avançar ou então retroceder nas instruções que digitamos.

Caso as teclas de atalho para avançar e retroceder as instruções lhe for incomodo, é possível alterá-las acessando o Menu do IDLE, para isso, vá em *Option -> Configure IDLE -> Keys* e então, defina a tecla de atalho a ação que desejares.

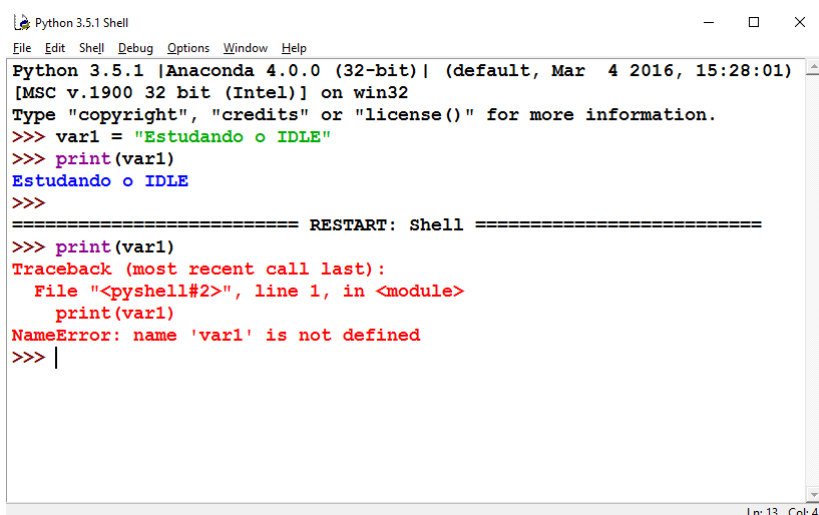


UTILIZANDO VÁRIAS SEÇÕES

Quando inicializamos o IDLE é aberta uma janela que contém um editor e é inicializado uma seção do Python. Tudo que for feito numa janela, estará disponível só e somente só na seção do Python inicializada especialmente para essa janela. Se o IDLE for reiniciado, ou então, se você estiver trabalhando com várias janelas abertas, cada uma terá uma seção independente do Python, onde poderás importar bibliotecas, definir variáveis ou então, poderás fazer o que bem entender. Assim, temos que uma variável declarada na "Janela A", não estará disponível na "Janela B", da mesma forma que, uma variável declaradas na execução do "Programa X", não estará disponível na execução do "Programa Y".

Muitas vezes, importamos diversas bibliotecas numa seção, ou então, alteramos algum funcionamento interno das bibliotecas do Python para estudarmos ou então, realizarmos testes.

Sempre que desejares, é possível reiniciar a seção em que estamos trabalhando e assim, remover quaisquer alterações feitas ou então, remover qualquer objeto criado em memória. Para isso, vá no Menu *Shell -> Restart Shell* ou então, pressione as teclas de atalho *CTRL + F6*. Em seguida, uma linha horizontal será traçada e será escrito no centro do vídeo *RESTART*, como pode ser visto na imagem abaixo:



Como a imagem acima está mostrando, a variável que foi declarada antes da reinicialização do IDLE não está mais disponível, até porque, a seção que estava sendo utilizada foi destruída e em seguida, uma nova seção do Python foi inicializada.

UTILIZANDO O ÚLTIMO RESULTADO

É comum que precisarmos utilizar o último valor que foi retornado pelo IDLE, isto é, vamos supor que nós estejamos fazendo um cálculo, logo, é comum precisarmos do último resultado. Pra isso, podemos utilizar o caractere de underline (`_`) para assim acessarmos a instância do último valor, por exemplo:

```
#realizamos uma simples soma entre 2 valores
>>> 10 + 10
20
#utilizamos o último resultado e somamos mais 10 unidades ao mesmo
>>> _ + 10
30
```

Como pode ser visto no código acima, inicialmente realizamos a soma de 2 operandos. Na sequência, utilizamos o caractere de underline para assim somarmos mais 10 unidades ao resultado da primeira expressão.

TECLAS DE ATALHO DO IDLE

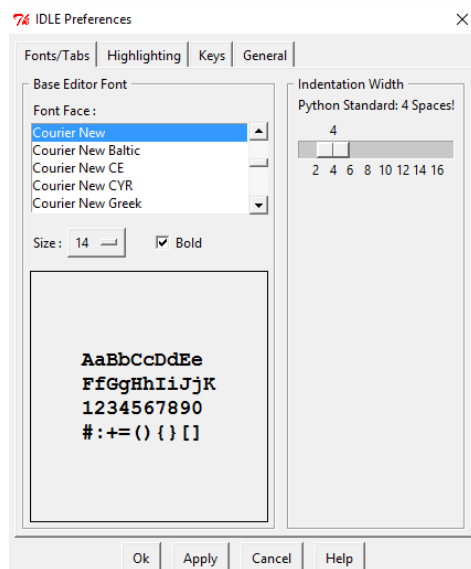
A seguir, temos as teclas de atalho que estudamos nessa aula e também, os atalhos que podem ser considerados os principais da utilização básica do IDLE.

1. Alt + P -> retorna última instrução
2. Alt + N -> avança para a próxima instrução
3. Ctrl + F6 -> Reinicia o IDLE
4. Ctrl + N -> Abre o editor de Script's de várias linhas
5. Ctrl + Enter ->
6. Ctrl + C -> interrompe a execução (Script em looping)
7. Ctrl + Espaço -> abre o Code Complete
8. Tab -> abre o Code Complete
9. Ctrl + J -> adiciona linha e move o cursor para a mesma

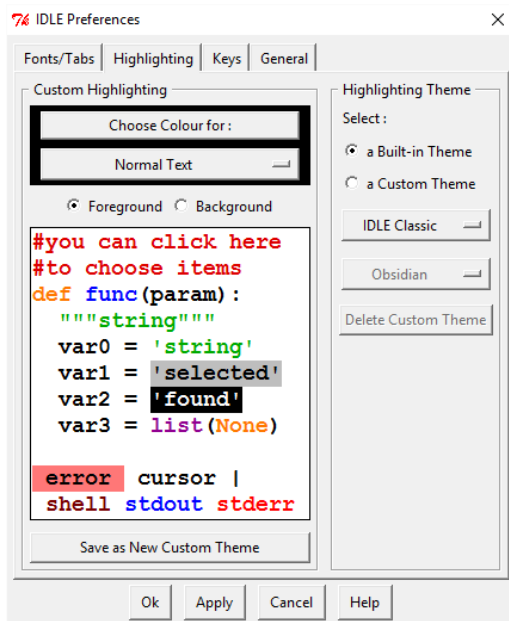
PERSONALIZANDO O IDLE

Como vimos, o IDLE é construído só e somente só com a linguagem Python, e toda a parte gráfica está escrita utilizando a biblioteca nativa Tkinter. Porém, é possível personalizarmos o IDLE das mais diversas maneiras e inclusive, somos capazes de criar nossos próprios temas e até, distribuí-los.

Para personalizar o IDLE, vá no Menu *Option -> Configure IDLE...* Em seguida, será aberta uma janela semelhante a que temos abaixo:



Como podemos ver na imagem acima, a primeira aba do menu de preferências do IDLE é *Fonts/Tabs*. Nesta, podemos escolher qual a fonte a ser utilizada pelo editor na lista contida dentro do GroupBox *Base Editor Font*. Abaixo da listagem de fontes disponíveis, há a opção *Size*, onde é possível definir o tamanho da fonte do editor e ao lado de *Size*, há um CheckBox de nome *Bold* que define se o texto deve ou não estar em Negrito.

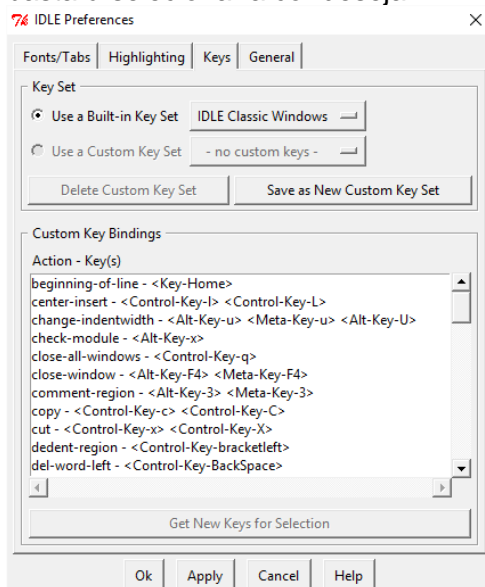


A aba *Highlighting* do IDLE permite-nos definir o sistema de cor a ser utilizado. Inicialmente, observe a caixa ao lado direito da janela de nome *Highlighting Theme*. Nesta, podemos selecionar inicialmente se a alteração das cores será feita no tema que está sendo exibido no caso, a opção a *Built-in Theme*, ou então, podemos definir a opção a *Custom Theme* que habilitará os controles abaixo.

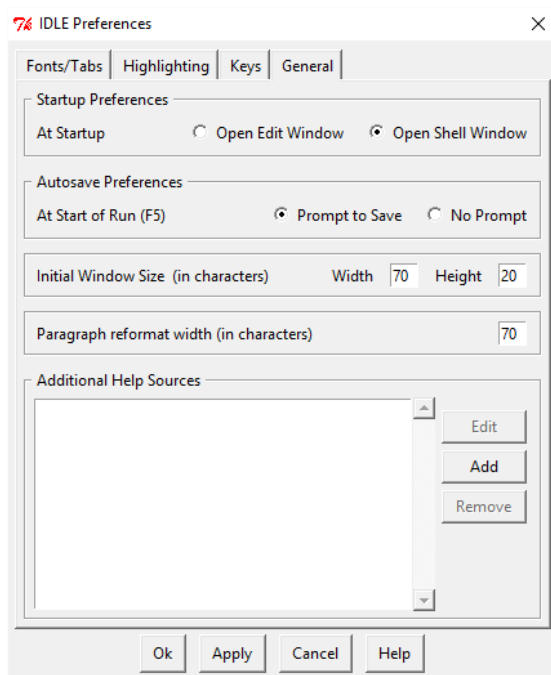
No primeiro Controle habilitado, há algumas opções de temas que estão instalados. Quando você seleciona outro tema, no mesmo momento, o editor ao lado esquerdo é alterado conforme as cores definidas no tema selecionado.

Agora, no GroupBox a esquerda de nome *Custom Highlighting*, temos a disposição todas as definições da linguagem Python. Logo, podemos alterar a cor de Background ou então, do Foreground de cada opção.

Se você for com o mouse em cima do Editor, poderás clicar em qual definição desejás alterar e assim, bastará selecionar a cor desejada.



A aba *Keys* já foi vista anteriormente, nesta, temos a definição de todas as teclas de atalho do IDLE e assim, podemos selecionar e em seguida, alterar a definição de tecla de atalho que desejamos.



A última aba é de nome *General*. Nesta, há opções configurações diversas. Inicialmente, temos o *GroupBox Startup Preferences*. Neste, podemos definir se ao abrirmos o IDLE o mesmo será aberto no modo *Shell Window*, isto é, o modo padrão, onde digitamos a instrução, pressionamos a tecla Enter e a mesma é executada.

Se definirmos a opção de abertura como sendo *Open Edit Window*, o IDLE será aberto com a tela de Edição de Scripts que estudamos anteriormente.

A opção a seguir é *Autosave Preferences*. Nesta, podemos definir se ao executarmos o Script na janela de Edição de Scripts será perguntado se desejamos salvar ou então, se o código será salvo automaticamente. É importante observar que inicialmente será perguntado onde desejamos salvar o arquivo de Script e somente em seguida, é que o salvamento será realizado automaticamente.

Em seguida temos a opção *Initial Window Size (in character)*. Está opção permite-nos definir a largura e a altura do editor. É importante observar que as dimensões são calculadas pela quantidade de caracteres e não em pixels. Quando estudarmos a biblioteca Tkinter iremos entender o porque dessa especificação!

3.1.3 Visual Studio Code

O Visual Studio Code por si só é um editor de texto criado pela Microsoft que apresenta diversos pacotes para personalizá-lo da forma que você precisa.

No site oficial do Visual Studio Code, você encontrará um link para a Documentação do programa. Na documentação, é possível acessar o manual de Python26 que mostra todos as funcionalidades que o programa possui relacionadas a Python.

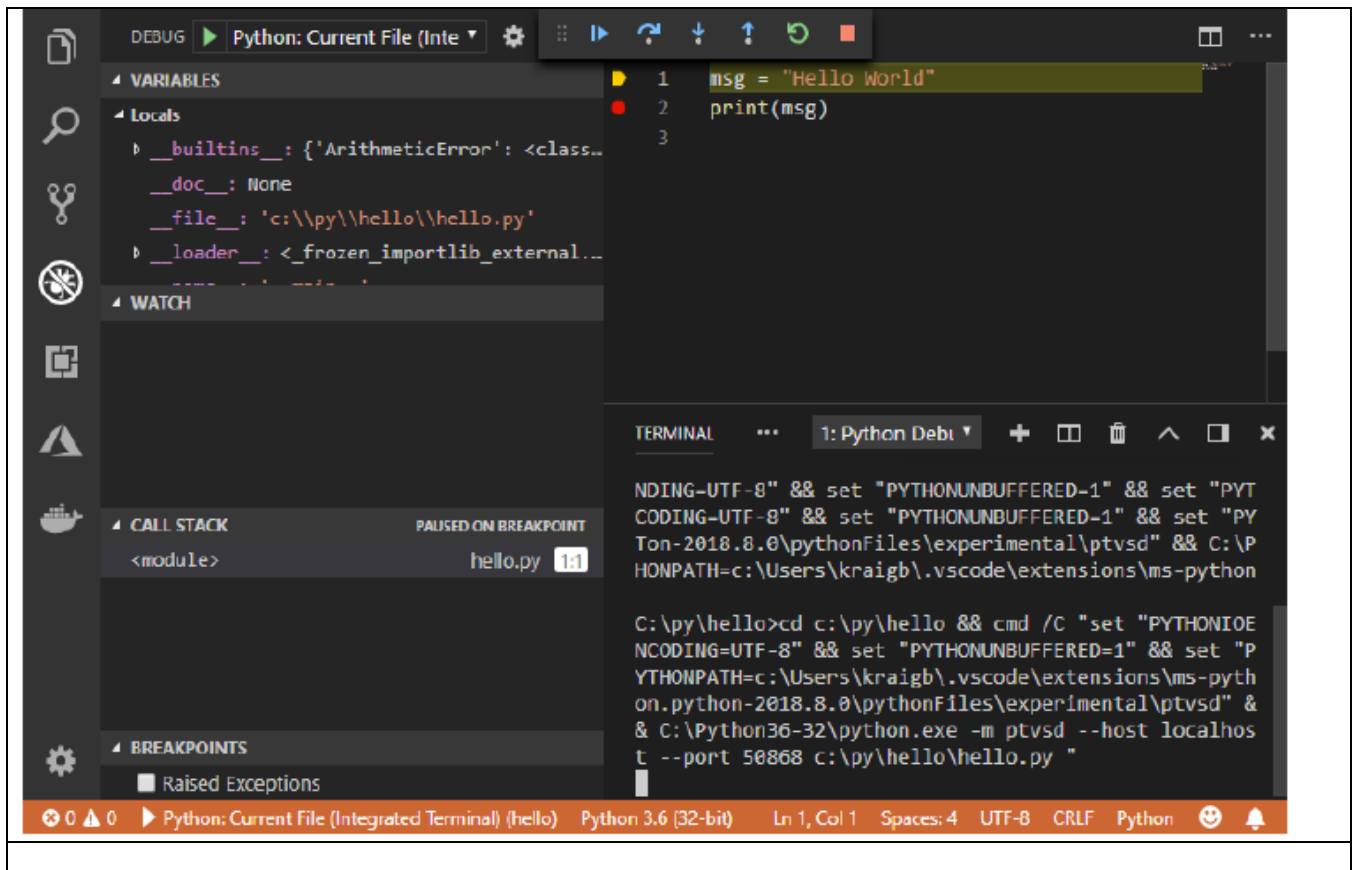
Para um passo-a-passo da instalação, você pode encontrar os detalhes de cada sistema operacional nesta página do manual.

Links

<https://code.visualstudio.com/>

<https://code.visualstudio.com/docs/languages/python>

<https://code.visualstudio.com/docs/setup/setup-overview>



3.1.4 PyCharm

O PyCharm é uma IDE desenvolvida pela empresa JetBrains, empresa esta especializada na produção de ferramentas para o desenvolvimento de software, como por exemplo, o AndroidStudio, que atualmente é, o Ambiente de Desenvolvimento oficial para a plataforma Google Android.

IDE é um programa para fazer programas. Nada tem a ver com a compilação, linkedição, interpretação ou geração de código executável.

Os produtos JetBrains, ou melhor, as IDE s da JetBrains, estão dentre as melhores ferramentas para desenvolvimento, seja em Python, C/C++, Ruby, PHP, Web e etc. Para programação em Python, temos a IDE de nome PyCharm, e este será o programa que utilizaremos para desenvolvimento de aplicações Python em todas as aulas deste curso.

Atualmente, o PyCharm é considerado a melhor IDE para desenvolvimento em Python e, recomendamos que voce utilize o mesmo em seu estudo, até porque, é gratuito e fácil de utilizar.

Você pode utilizar quaisquer IDE para desenvolvimento em Python.

O PyCharm possui uma versão gratuita e uma versão paga. Inicialmente, a versão gratuita possui muito mais recursos do que iremos precisar, porém, quando estivermos desenvolvendo para Web e utilizando framework's como Django ou Flask, é producente adquirir uma licença pela agilidade e produtividade que a versão comercial do PyCharm proporciona.

A IDE PyCharm oferece um módulo de acesso a Banco de Dados e também, trás uma maneira de testarmos códigos SQL, administrar o nosso banco, alterar, editar e excluir registros de maneira simples e rápida. Conseguimos administrar praticamente todos os bancos de dados do mercado sem a necessidade de utilizarmos outro programa, até porque, o PyCharm possui todo o módulo de acesso a base de dados incorporada a IDE.

Neste curso utilizaremos a versão comunitária do PyCharm e recomendamos que você também a utilize. Porém, sintase livre para trabalhar com qualquer editor, o resultado sempre deverá ser o mesmo, haja vista que a IDE,

isto é, o Ambiente de Desenvolvimento Integrado, gerencia a produção de código sem alterar ou influenciar como o código será executado.

INSTALAÇÃO

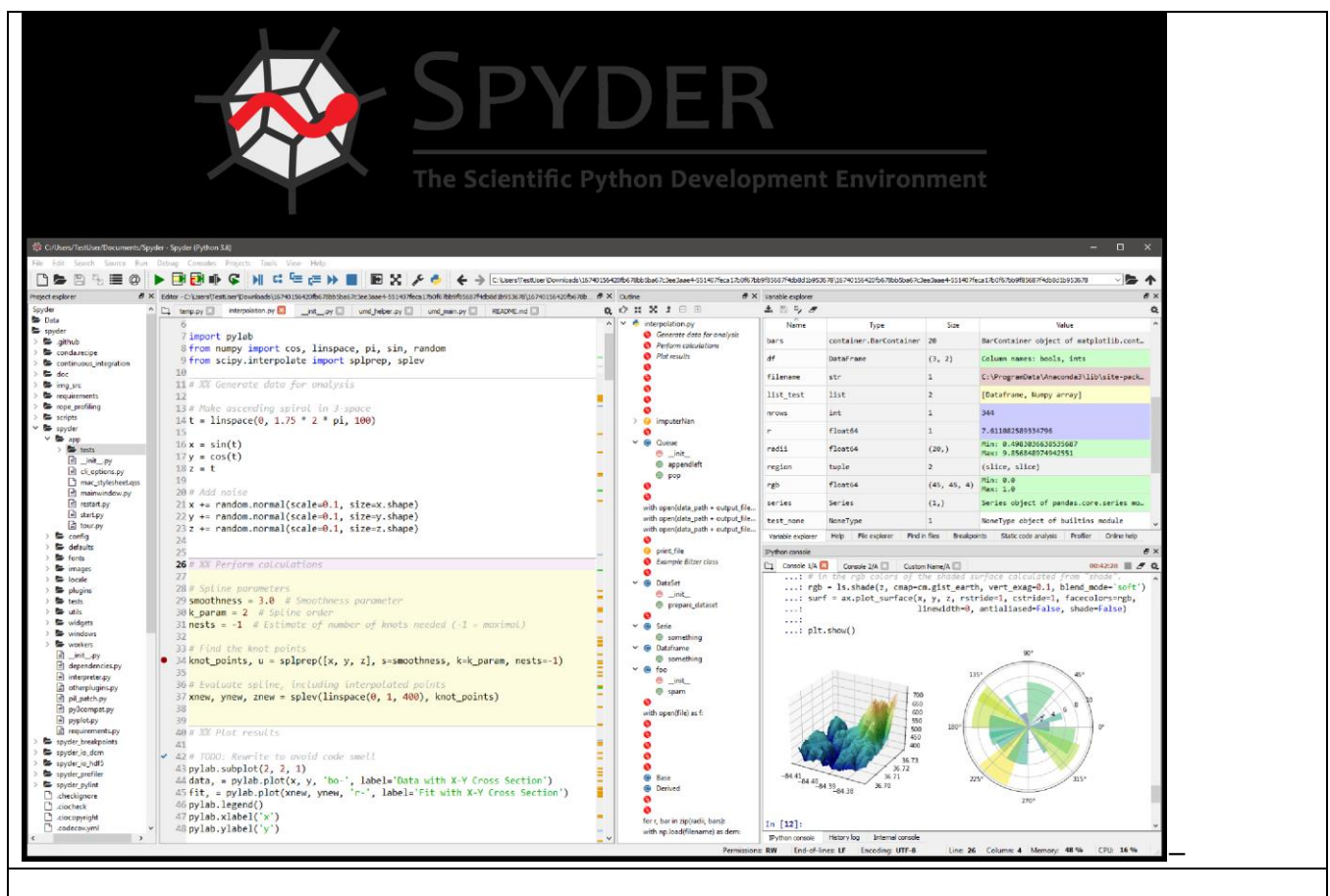
Para instalar o PyCharm em seu computador, é necessário que você instale antes o Java, caso não tenha o mesmo instalado. Na dúvida, ou seja, se não souberes se o Java está instalado, baixe-o e faça a instalação, somente para garantir!

Também é importante que você já tenha instalado o Python em seu computador. Isso porque, o PyCharm configurará o ambiente para você, do contrário, será preciso configurar qual a instalação do Python você deseja utilizar.

Após o download e instalação do Java e do Python, faça o download da última versão do PyCharm.

3.1.5 Spyder

O Spyder é uma IDE perfeita para quem utiliza o Python para Ciência de Dados, especialmente porque vem incluso no Anaconda, um pacote de várias bibliotecas e ferramentas voltadas para essa área.



Ele possui uma interface simples, com uma tabela de variáveis para fácil inspeção dos valores de cada variável no programa durante sua depuração.

A melhor forma de instalá-lo é pela instalação do Anaconda. Novamente, recomenda-se utilizar essa forma de instalação apenas para quem deseja utilizar todos os pacotes inclusos no Anaconda, e não apenas o Spyder.

Links

<https://www.anaconda.com/distribution/>

PYTHON É SOMENTE PARA WEB?

É verdade que o Python é uma excelente linguagem para a construção de WebSites, porém, a linguagem foi criada antes da popularização da Web. Por essa razão, é fácil concluir que a linguagem Python não é destinada só e somente só a construção Web, mas sim, para a construção de aplicações em geral.

4) Linha de comando

Existe também a possibilidade de trabalhar sem um ambiente gráfico, utilizando apenas o Prompt de Comando, Shell e mandar executar o seu código

4.2.1 Python Shell

Se você instalou corretamente o Python, você tem à sua disposição um interpretador interativo, popularmente conhecido como o Shell do Python.

Se você utiliza Linux ou Mac, basta abrir um terminal e digitar python. Caso utilize Windows, deverá existir uma pasta no menu iniciar chamada Python 3.7 (ou Python 3.6) que permite abrir o interpretador. Caso tenha sucesso, você deverá encontrar algo similar a esta imagem:

```
$ python
Python 3.7.2 (default, Jan 10 2019, 23:51:51)
[GCC 8.2.1 20181127] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> █
```

Para sair do Python, digite quit(). Isso encerra a seção interativa e te retorna ao terminal (ou fecha a tela preta, no caso do Windows).

Note que, ao entrar no interpretador, a primeira linha começa com Python 3.7.2. Essa é a versão utilizada do Python. Neste curso utilizamos a versão 3.6 ou mais recente. Caso a versão que aparecer em seu interpretador seja 2.7.6, encerre a seção e tente o comando python3 para utilizar a versão 3 do Python.

4.2.2 IPython

IPython é um Shell alternativo para o Python, bem poderoso. E colorido também, ao contrário do shell nativo do Python.

```
$ ipython
Python 3.7.2 (default, Jan 10 2019, 23:51:51)
Type 'copyright', 'credits' or 'license' for more information
IPython 7.3.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: █
```

Link

<https://ipython.org>

QUAL A MINHA VERSÃO DO PYTHON?

Vá no prompt/shell/bash e digite:

```
python
```

Se nada acontecer, tente digitar somente a palavra:

```
py
```

O texto que impresso deverá ser semelhante ao ao texto a seguir:

```
python
```

```
Python 3.5 (default, Mar - YYYY, hh:mm:ss) [MSC v.1600 64 bit (AMD64)] on win32  
Type "copyright", "credits" or "license()" for more information.
```

OU ENTÃO:

```
python  
Python 3.6 |Anaconda 2.2.0 (64-bit)| (default, Mar YYYY, hh:mm:ss) [MSC v.1600 64 bit (AMD64)] on  
win32  
Type "copyright", "credits" or "license()" for more information.
```

SE EU TIVER O PYTHON INSTALADO E NADA ACONTECER?

Isso ocorre nas situações quando o local de instalação do Python não está nas variáveis do sistema. As variáveis do sistema apontam para diretórios onde o SO que estamos utilizando irá buscar os programas que devem ser reconhecidos na linha de comando ou por outros softwares de maneira automática.

Nos sistemas Linux, raramente a instalação não é encontrada, mas nas versões do Windows, isso ocorre muito mais facilmente, até porque, é possível, nas instalações do Python, pedir que o mesmo não adicione o PATH [caminho ou diretório] da instalação atual, nas variáveis do sistema.

Logo, você terá que ir nas opções do sistema, no caso, em Programas Instalados e procurar por alguma instalação do Python. Se nada for encontrado, e você não tiver feito a instalação via compilação através do fontes, então o Python não está instalado.

Bibliografia

Curso Introdutório de Python. Grupy-Sanca (Maio/2019)