

**INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**
BAHIA
Campus Salvador

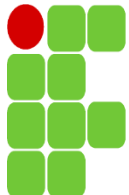
Disciplina : Programação Python

Prof. Domingos Mainart

Email: prof.mainart@gmail.com

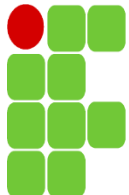
1. Introdução e História:

- Surgiu em 1989;
- Criada por Guido van Rossum;
- Monty Python and the Flying Circus;
- Licença compatível com Software Livre;
- Linguagem de altíssimo nível (VHLL) (very high-level programming language);
- Tipagem Dinâmica;
- Multiparadigma (OO, funcional e procedural);
- Compilada + Interpretada;
- Aumentar a produtividade do programador;



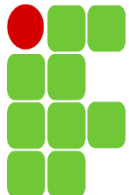
Versões e Instalação

- A implementação oficial do Python é mantida pela PSF (Python Software Foundation) e escrita em C, e por isso, é também conhecida como CPython. A versão estável mais recente está disponível para download no endereço: <http://www.python.org/download/>
- Para a plataforma Windows, basta executar o instalador. Para outras plataformas, como em sistemas Linux, geralmente o Python já faz parte do sistema, porém em alguns casos pode ser necessário compilar e instalar o interpretador a partir dos arquivos fonte.
- 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.



O que é Python?

- Python é uma linguagem de programação. Isso significa basicamente duas 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.
- Python tem algumas características interessantes
 - é interpretada, ou seja, o interpretador do Python executa o código fonte diretamente, traduzindo cada trecho para instruções de máquina;
 - é de alto nível, 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;
 - é de propósito geral, 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;
 - tem tipos dinâmicos, ou seja, o interpretador faz a magia de descobrir o que é cada variável.



O que é Python?

- A linguagem Python é uma linguagem de alto nível, interpretada, orientada a objetos com uma semântica dinâmica. Suas estruturas de alto nível, combinadas com sua tipagem de amarração dinâmica a faz muito atrativa para desenvolvimento de largos aplicativos assim como para uso como linguagem de script ou de colagem.
- A sintaxe simples do Python encoraja a reutilização de código simplificando a manutenção e a normalização de dados em módulos e pacotes distintos.

O que é Python?

- Por essas e várias outras características, Python se torna uma linguagem simples, bela, legível e amigável. É uma linguagem utilizada por diversas empresas, como Wikipedia, Google, Yahoo!, CERN, NASA, Facebook, Amazon, Instagram, Spotify;
- O desenvolvimento de Python começou no final da década de 1980, por Guido van Rossum. 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.

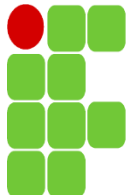
Por que utilizar Python?

- Linguagem Simples, Legível, Clara e Elegante!

Um Exemplo: Escrever um simples “nome, seja bem vindo(a)! ”

- Em C:

```
#include <stdio.h>
int main(){
    char nome[200];
    printf("Digite seu nome: ");
    scanf("%s", nome);
    printf("\n %s, Seja bem vindo(a)\n", nome);
    return 0;
}
```



Por que utilizar Python?

- Em JAVA:

```
public class Hello{  
    public static void main(String args[]) {  
        java.util.Scanner s = new java.util.Scanner(System.in);  
        System.out.print("Digite seu nome:");  
        String nome = s.nextLine();  
        System.out.println("\n" + nome + ", Seja bem vindo(a :) \n");  
    }  
}
```


Por que utilizar Python?

- Em Python:

```
nome = input("Digite seu nome: ")  
print ("\n, Seja bem vindo(a) :) ", nome);
```

Python oferece maior facilidade para entender.

A linguagem vai direto ao ponto. Simples de usar. Foco no problema, sem perder tempo na sintaxe.

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

- BitTorrent

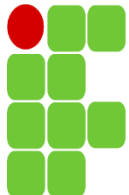
- O protocolo Torrent é muito utilizado para transferir quantidades grandes de dados para diversos computadores. O primeiro programa a implementar esse protocolo foi desenvolvido inteiramente em Python, pela BitTorrent, Inc.

- 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 bancos de dados grandões.
- Alguns sites desenvolvidos em Django: Instagram, The Washington Times, Disqus, Mozilla, National Geographic

- Dropbox

- O popular serviço de armazenamento de dados em Nuvem Dropbox tem diversas partes da infraestrutura feita em Python.5 O aplicativo para computadores é feito em Python e grande parte da infra estrutura dos servidores deles também é!



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

- 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.4
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

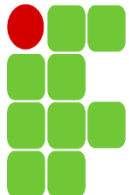
# pega os dados solares de WDC-SILSO, Royal Observatory of Belgium, Brussels
sun = pd.read_table('http://sidc.oma.be/silso/INFO/sndhemcsv.php', sep=';', encoding_
    ↵="ISO-8859-1", header=-1)

# 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()
```



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

- E o resultado desse código é a seguinte imagem:

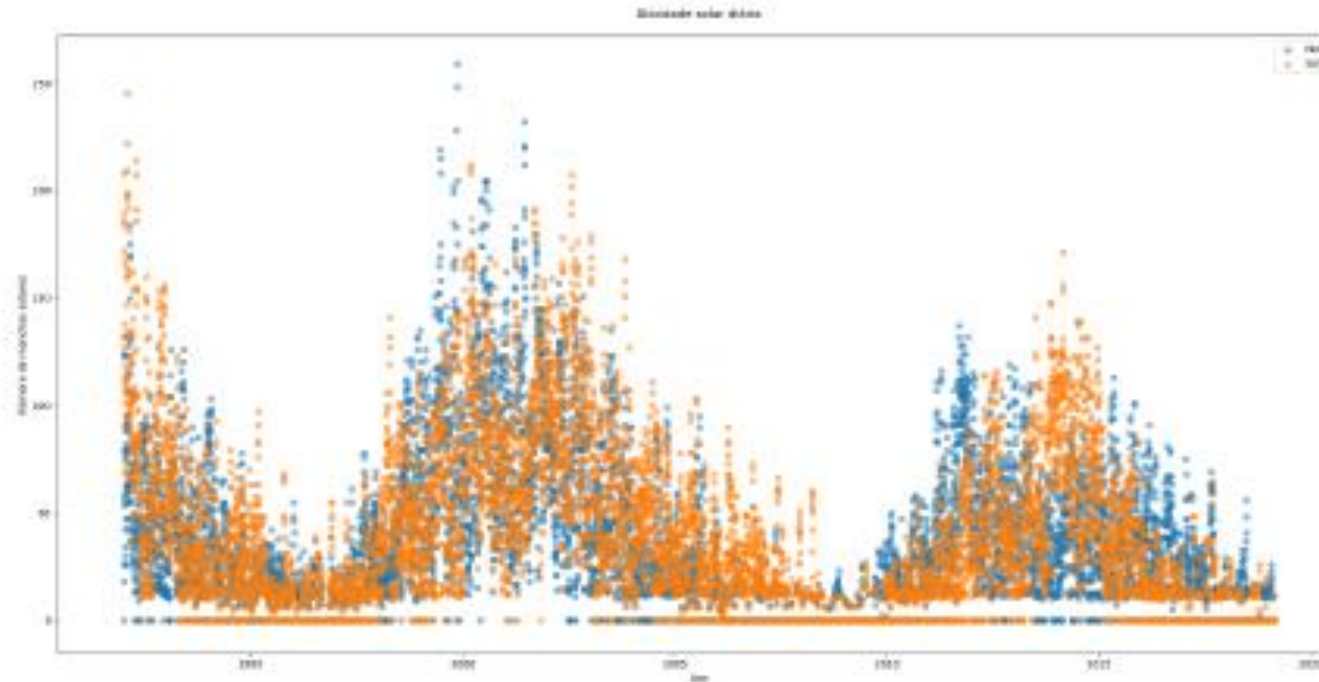
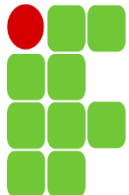


Figura 1: Erupções solares desde 1992, separadas por hemisférios (norte e sul) do Sol.



Guia de Instalação do Python

- **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:

Guia de Instalação do Python

- No Ubuntu e Debian

`sudo apt install python3`

- No ArchLinux

`$ sudo pacman -Sy Python`

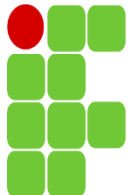
- Mac OS X

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



Guia de Instalação do Python

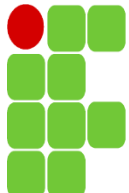
- Windows

Obtenha o arquivo de instalação última versão compatível com a arquitetura do seu computador no site oficial do Python²⁰. O arquivo x86 provavelmente funcionará para todos computadores.

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



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



Ambientes de Desenvolvimento

- Há diversos programas para desenvolvermos códigos, alguns são mais bonitinhos, outros são mais poderosos, alguns são mais simples, outros são mais amigáveis. Somente você pode responder à pergunta «Qual o melhor ambiente de desenvolvimento para mim?»
- 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.

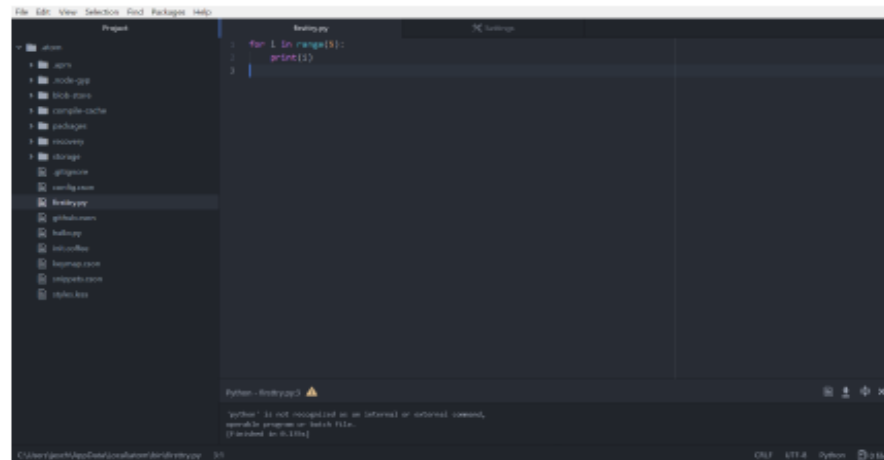
Ambientes de Desenvolvimento

- **Ambientes gráficos**

- ATOM

- O programa ATOM é um IDE open-source que apresenta diversos pacotes para personalizar.
- No site oficial do ATOM, você encontrará um link para a Documentação do programa. Na documentação, é possível acessar o manual 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 discussão. Nessa página, você encontrará respostas para diversas dúvidas, e possivelmente, para a sua.

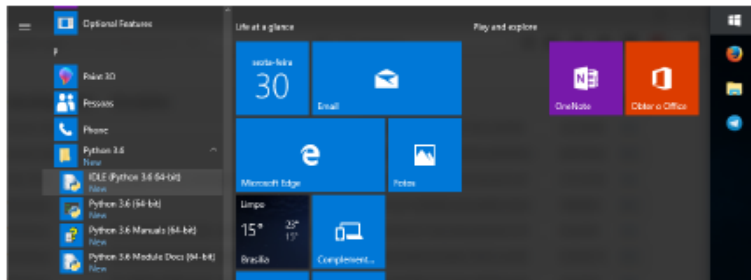


Ambientes de Desenvolvimento

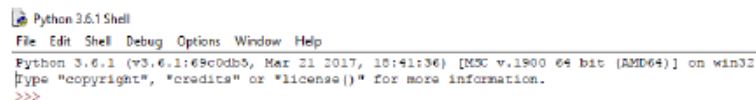
- **Ambientes gráficos**

- **IDLE**

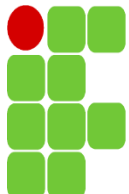
- Para usuários Windows é recomendado utilizar o IDLE. Ele é composto pelo interpretador do Python e um editor de texto para criar programas, e já vem junto com o Python. Após seguir o Guia de Instalação do Python (Página 21), o menu inicial deve estar da seguinte forma:



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

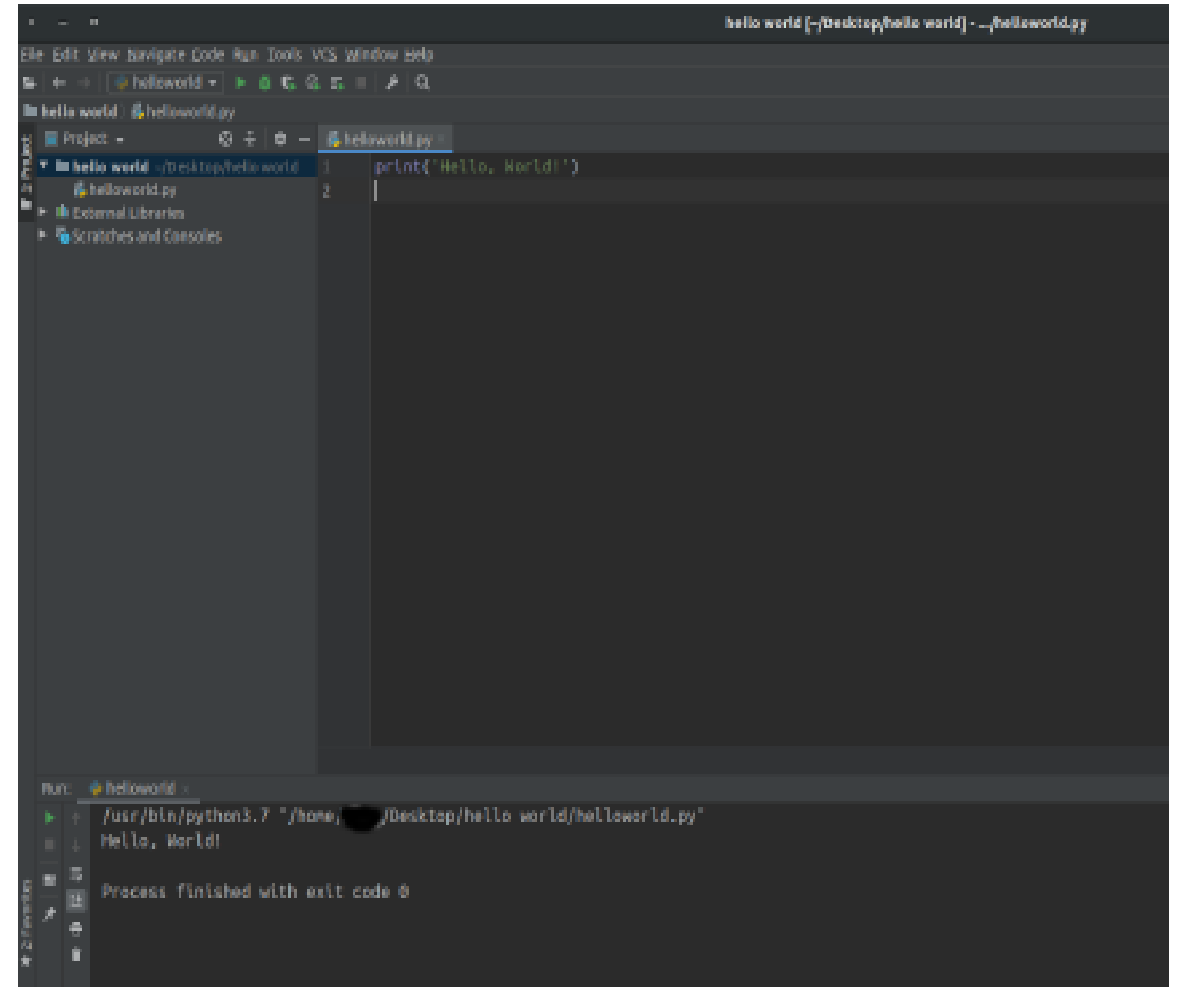


No IDLE é possível digitar comandos para o interpretador do Python e, também, é possível criar e digitar em um arquivo. Para fazer isso, no menu clique em File -> New File (Ou pressione as teclas Ctrl + N juntas) Para rodar um programa, clique em Run -> Run Module (Ou aperte a tecla F5) Caso o arquivo 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:



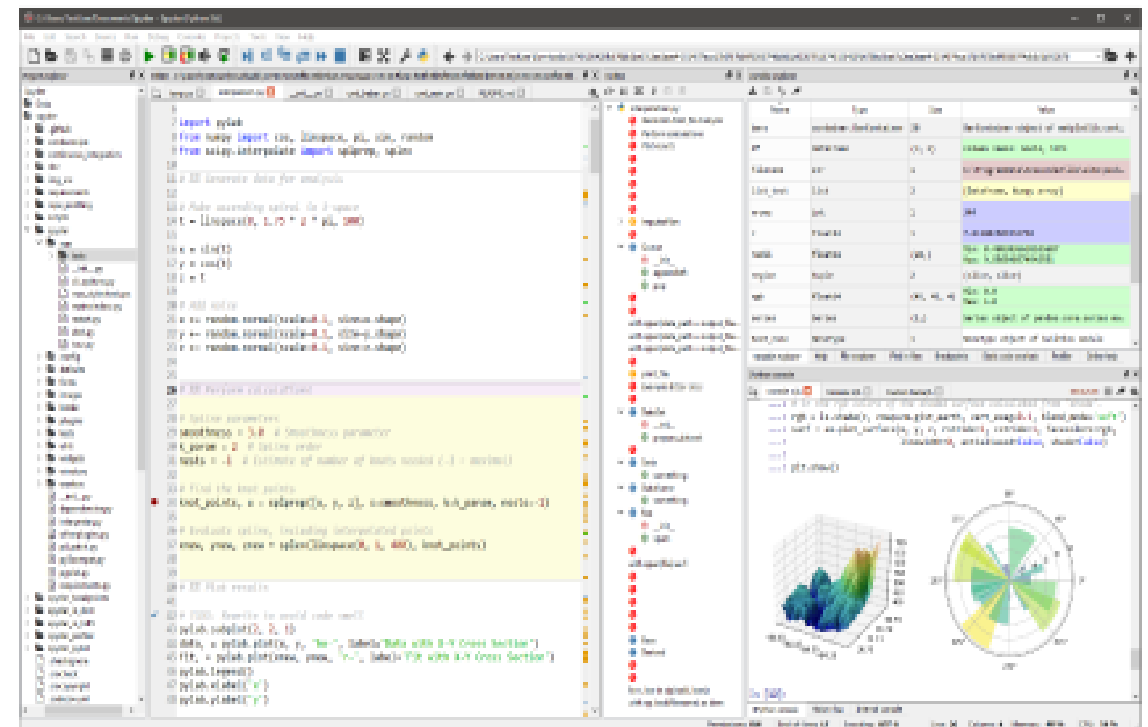
Ambientes de Desenvolvimento

- **Ambientes gráficos**
- **PyCharm**
- Esta IDE é voltada especificamente para a linguagem Python. No site oficial é possível encontrar orientações para realizar o download e instalação (Linux, Mac, Windows).
- É desenvolvido pela empresa tcheca JetBrains. Fornece análise de código, um depurador gráfico, teste de unidade integrado, integração com sistemas de controle de versão, ambiente virtual e suporta o desenvolvimento da Web com o Django, bem como Data Science com o Anaconda



Ambientes de Desenvolvimento

- **Ambientes gráficos**
- **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.



Como Compilar e Interpretar

- O código fonte é traduzido pelo Python para bytecode, que é um formato binário com instruções para o interpretador. O bytecode é multiplataforma e pode ser distribuído e executado sem fonte original.
- Por padrão, o interpretador compila o código e armazena o bytecode em disco, para que a próxima vez que o executar, não precise compilar novamente o programa, reduzindo o tempo de carga na execução.
- Quando um programa ou um módulo é evocado, o interpretador realiza a análise do código, converte para símbolos, compila (se não houver bytecode atualizado em disco) e executa na máquina virtual Python.

Como Compilar e Interpretar

- O bytecode é armazenado em arquivos com extensão “.pyc” (bytecode normal) ou “.pyo” (bytecode otimizado). O bytecode também pode ser empacotado junto com o interpretador em um executável, para facilitar a distribuição da aplicação, eliminando a necessidade de instalar Python em cada computador.

