

Fábrica de Software

Aula 01

Python

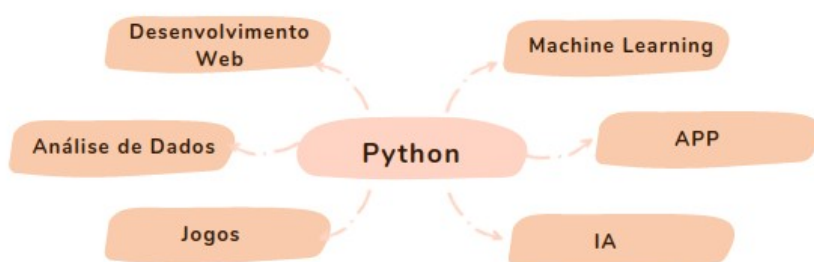


O que é a Linguagem Python?

O desenvolvimento da linguagem Python começou no final da década de 1980 por Guido Van Rossum, enquanto ele trabalhava no Centrum Wiskunde & Informatica — CWI — em Amsterdã. Seu objetivo era criar uma linguagem de script mais eficiente, capaz de acessar recursos do sistema operacional Amoeba.

O Python é uma linguagem de programação de alto nível utilizada em diferentes tipos de aplicações. Ela foi elaborada para que as pessoas desenvolvedoras escrevam instruções com menos linhas de código. Python é uma linguagem interpretada e fracamente tipada (não precisamos declarar o tipo de uma variável, por exemplo). Além disso, é uma linguagem de propósito geral.

Usos do Python



No caso, por ser de fácil compreensão, foi uma linguagem que ganhou popularidade entre as empresas devido ao seu código enxuto. Além disso, ela possui uma grande quantidade de bibliotecas disponíveis para auxiliar no nosso dia a dia, e tem uma grande comunidade de pessoas desenvolvedoras.

Se Liga!!!

O nome Python não é nada relacionado com a espécie de cobra (triste). A escolha foi feita tendo como inspiração um programa de televisão chamado Monty Python.

Algumas especificações dessa linguagem:

Multiparadigma: o que significa que podemos programar com base em diferentes estilos de programação, como a programação funcional, a procedural e a orientada a objetos

Multiplataforma: tem versões para diversos sistemas operacionais, entre eles: Windows, Linux e macOS

Linguagem Interpretada: contém um interpretador, que é responsável por traduzir o código fonte em linguagem de máquina e, assim, executar o programa.

Orientada a Objetos: No Python, o paradigma orientado à objetos funciona de forma similar às outras linguagens, porém com algumas mudanças em sua implementação

Por que Python é tão valorizado?

1. É fácil de aprender

A linguagem Python tem a sintaxe simples e o mais objetiva possível. Um exemplo é que não precisamos utilizar o sinal de ponto e vírgula ao final de cada declaração e as chaves para delimitar os comandos de estruturas de repetição e condicionais, como acontece em outras linguagens de programação.

2. É portátil e multiplataforma

A linguagem Python pode ser instalada em diferentes sistemas operacionais, como Windows, Linux e macOS. Além disso, existem distribuições portáteis, que podem ser utilizadas em pen drives, o que é ideal para facilitar a vida das pessoas programadoras na fase de desenvolvimento da aplicação.

3. É open source e gratuito

A linguagem Python é de código aberto, o que significa que é possível baixar seu código fonte escrito em linguagem C e realizar alterações. Também não é preciso pagar por sua licença de uso. Além disso, ele contém inúmeras bibliotecas e frameworks gratuitos que podem ser utilizados para obter recursos adicionais.

4. Oferece múltiplas possibilidades de desenvolvimento

É possível desenvolver diferentes tipos de aplicações com a linguagem Python, tanto sistemas comerciais para desktop, quanto softwares mais complexos de Machine Learning e Inteligência Artificial. As bibliotecas e frameworks disponíveis para a linguagem ampliam ainda mais a possibilidade de desenvolvimento.

Um exemplo é o framework Django, que é utilizado para o desenvolvimento de aplicações web. Por toda essa abrangência, ela foi escolhida em primeiro lugar como a linguagem de programação mais desejada pelas pessoas programadoras na pesquisa Stack Overflow de 2020.

5. É uma linguagem “curinga”

A linguagem Python é extremamente poderosa e oferece uma ampla possibilidade de desenvolvimento de sistemas. Como consequência, amplia as oportunidades de trabalho às pessoas programadoras, que podem escolher entre as diferentes possibilidades de desenvolvimento como a de aplicações para web. Além disso, sua sintaxe é simples e intuitiva, o que facilita o aprendizado.

6. Comunidade Grande

A comunidade é grande e cresce cada dia mais. Você consegue encontrar com facilidade as respostas às suas dúvidas e aumentar rapidamente suas amizades.

Comunidade

<https://python.org.br/>
<https://www.python.org/>
<https://pt.stackoverflow.com/>

Por que Python em Análise e Ciências de Dados?

Muitas bibliotecas de Análise de Dados – Com Python, você pode encontrar uma grande variedade bibliotecas de Ciência de dados (NumPy, SciPy, StatsModels, Scikit-learn, Pandas, Seaborn, etc.), que estão em desenvolvimento e crescimento gigante. São ferramentas de alto padrão e de fácil uso.

Facilidade de Aprendizado – A linguagem foi projetada para ser simples. O aprendizado é rápido e fácil!

Escalabilidade – em relação a outras linguagens / pacotes para a Ciência de dados (como MatLab, Stata, R) Python é muito mais rápido.

Visualização / Gráficos – Python não é tão bom como R (nesse sentido!), mas vamos ver mais e mais APIs (por exemplo, Plotly) e bibliotecas de visualização de dados que fazem a vantagem parcial de R insignificante em comparação com Python.

Jupyter-Notebook – esta é simplesmente uma grande ferramenta. O **Jupyter Notebook** é um ambiente que oferece essa abordagem de Notebooks, com um visual simples e muito fácil de utilizar.

REFERÊNCIAS

- Pense em Python: Pense Como um Cientista da Computação. Autor: Allen Downey. Editora: Novatec
- Introdução à Programação com Python: Algoritmos e Lógica de Programação Para Iniciantes. Autor: Nilo Ney Coutinho Menezes. Editora: Novatec
- Python Fluente: Programação Clara, Concisa e Eficaz. Autor: Luciano Ramalho. Editora: Novatec.
- Python Para Análise de Dados: Tratamento de Dados com Pandas, NumPy e IPython. Autor: Wes McKinney. Editora: Novatec.

Jupyter Notebook

O Jupyter Notebook é uma ferramenta essencial para quem trabalha ou estuda qualquer tipo de linguagem de programação. O revolucionário nesta aplicação é conseguir unir código e texto. Essa união permite que cada funcionalidade seja explicada de maneira minuciosa. Além dessa poderosa integração, a ferramenta também gera gráficos em tempo real, permitindo uma análise precisa das informações e de tudo o que ocorre no código.

Além disso, o Jupyter gera documentos com todas as informações, que são justamente chamados de Notebook. A documentação completa de um programa pode ser gerado desta forma. Desde a descrição da proposta, passando por fases críticas, a descrição do que foi desenvolvido, além dos resultados que possibilitam uma discussão. Ele é um aplicativo web open-source, ou seja, seu código é aberto e é possível que ele seja utilizado e alterado para diversos fins.

Por ser um software open-source, o Jupyter também é gratuito, já que não é necessário adquirir uma licença para utilizá-lo. Isso permite que a aplicação seja amplamente divulgada e melhorada com o esforço da própria comunidade.

Trabalhando com o Jupyter Notebook

O Jupyter Notebook é aberto no navegador, mas funciona localmente. A página inicial dá acesso às pastas do nosso sistema, onde conseguimos abrir Notebooks criados anteriormente e criar novos Notebooks. Vamos então trabalhar...

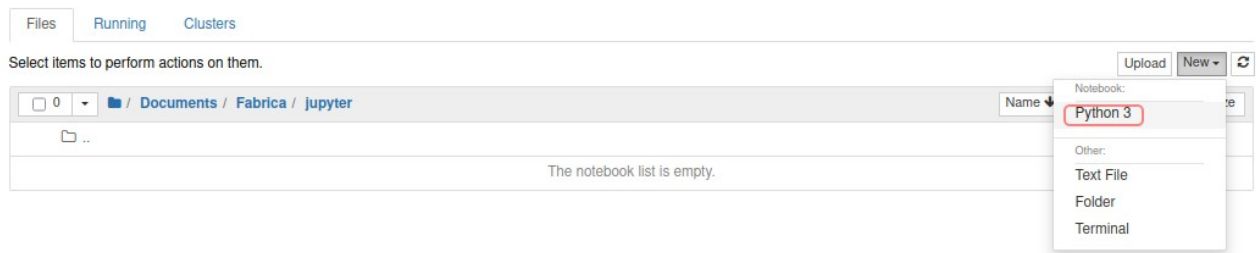
- 1 - Escreva jupyter notebook no campo de pesquisa (windows) e abra-o
- 2 - Vá até a sua pasta em Documentos
- 3- Vá em New e Folder no canto superior direito. Crie uma pasta chamada jupyter



Para renomeá-la basta selecioná-la e clicar em rename, no canto superior esquerdo.



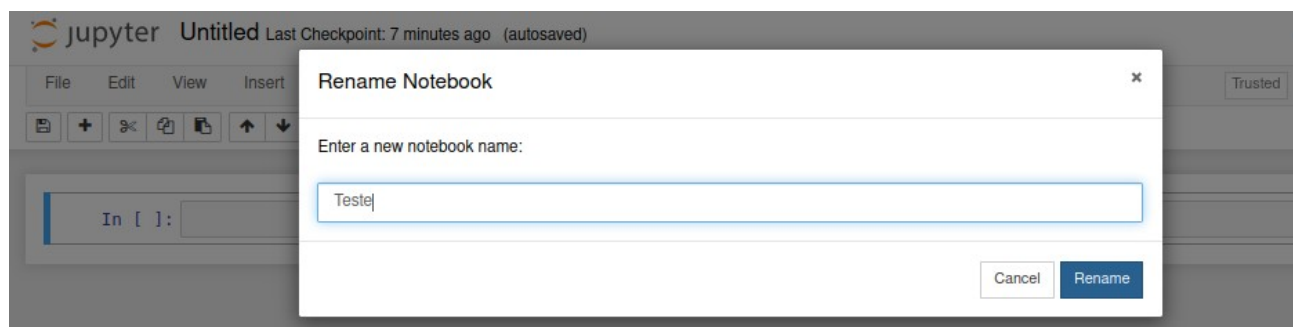
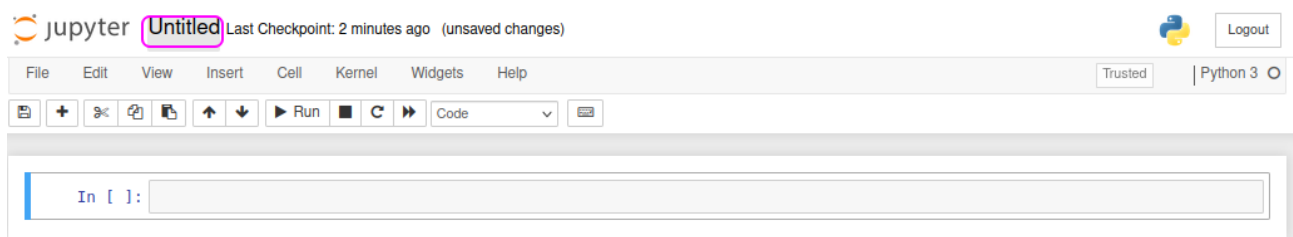
Na pasta criada, vá até o canto superior direito e crie um notebook. Basta selecionar new e Python 3.



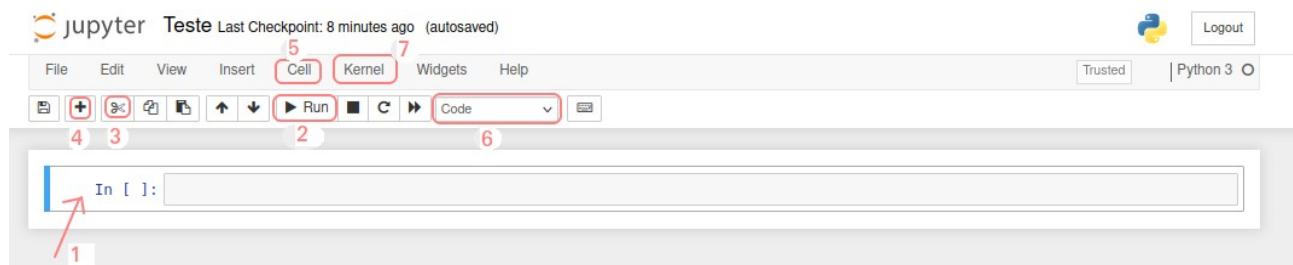
Pronto! Essa é sua área de Trabalho. Vamos ver algumas especificações e mão na massa!

Antes de tudo, vamos mudar o nome do seu Notebook. Coloque nomes simples, sem espaços e acentos. Coloque um nome auto-explicativo para ajudá-lo. O nosso notebook chamará... (som de suspense)... Teste (uau!!).

Ao lado do escrito jupyter e do símbolo, seu notebook está com o nome Untitled, mude-o para Teste. Para isso, dê duplo clique na palavra Untitled.

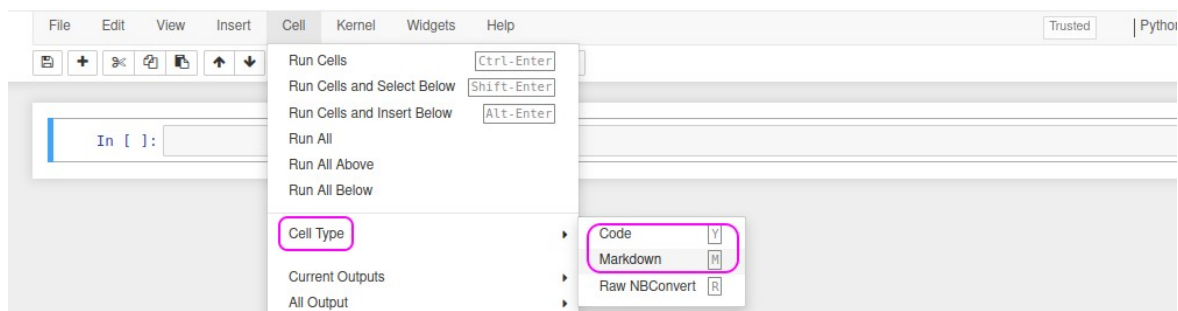


Agora vamos aprender algumas coisinhas...



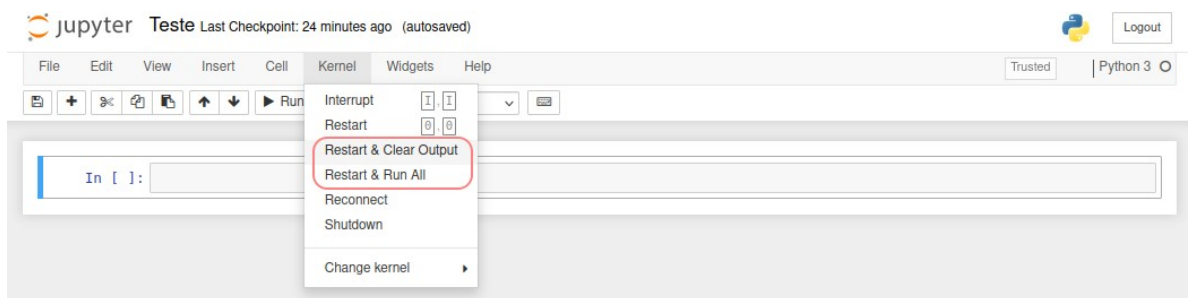
- 1- Esse local é chamado de célula. É nela que você colocará seu código.
- 2- Esse é o comando RUN. Ele irá rodar seu código. Toda vez que é aceita o código sem erros, um número aparecerá na frente da célula.
- 3- Caso queira excluir uma célula você usará a tesourinha.
- 4- Use o símbolo de + para adicionar células.

5- Na barra de ferramentas, selecione Cell, Cell Type e você verá duas opções importantes: Code e Markdown. Quando você quiser que sua célula seja apenas um comentário ou texto, selecione markdown. Para voltar ao estado de código, selecione Code.



6- A mesma mudança entre código e texto pode ser feita aqui.

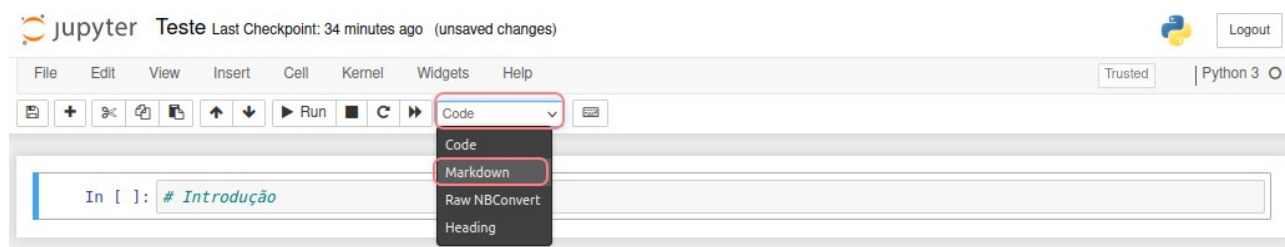
7- Kernell: Aqui, duas seleções são importantes: Restart & Clear Output (vai reiniciar e “zerar” todos os códigos) e Restart & Run All (que vai reiniciar e rodar todos os códigos de uma vez).

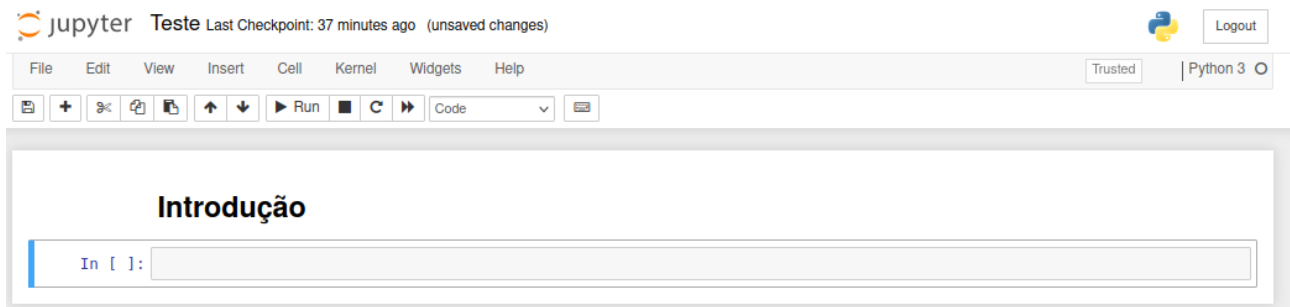


Vamos treinar alguns comandinhos...

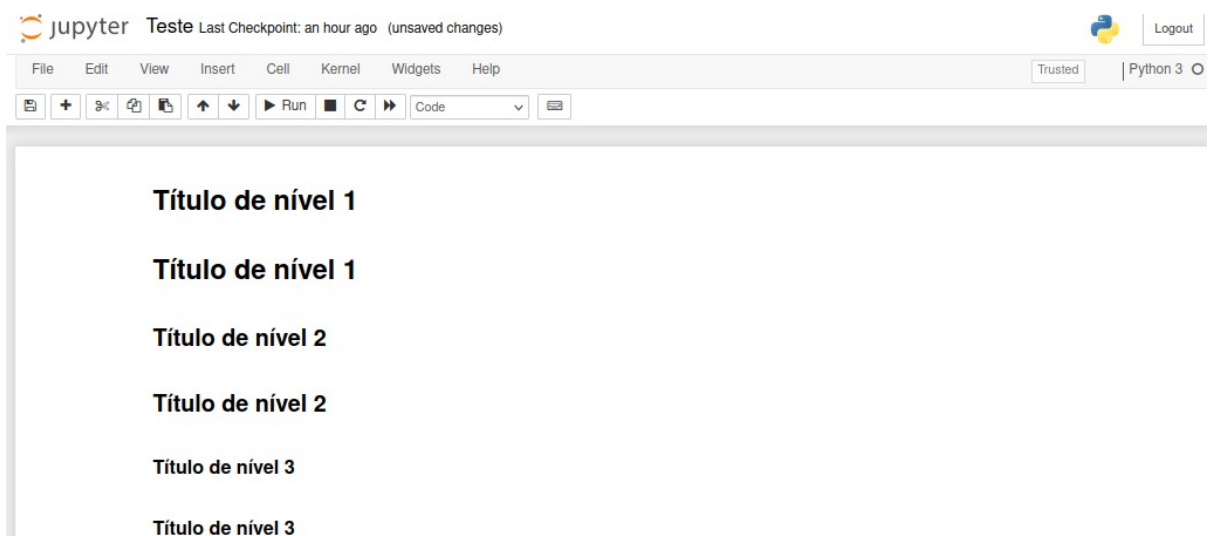
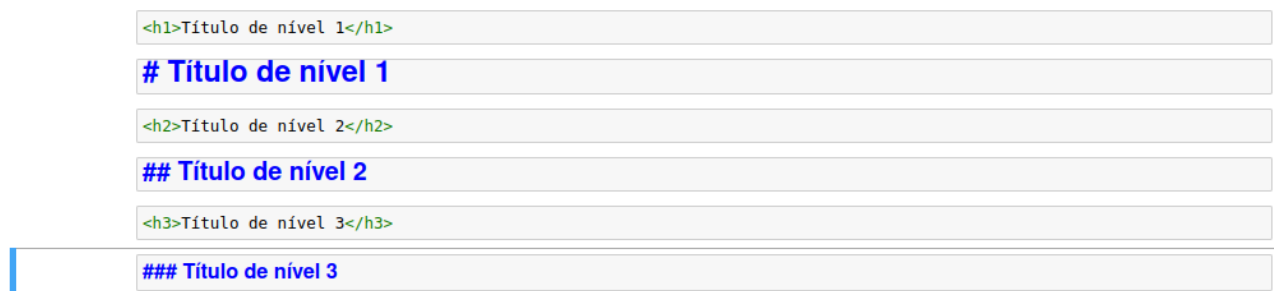
Primeiro vamos colocar um texto em forma de título:

Escreva o texto com o # na frente. Se você apenas rodar (run), ele ficará como comentário. Agora, se você mudar a célula para Markdown, ele será um título. (Não se esqueça de colocar um espaço entre a # e a palavra).





Caso queira mudar a cor, fonte, tamanho, etc...Mude com tag html ou usando #



DESAFIO

Escreva um título em azul,negrito e itálico. Separe-o com uma linha dupla. Coloque também um texto com itens de 1 a 4.

O desafio deve ficar assim:

Título

1. Primeiro item
2. Segundo item
3. Terceiro item
4. Quarto item

Atalhos do teclado

Os seguintes atalhos estão implementados por default para célula no modo comando:

- **Ctrl-Enter**: executa a célula;
- **Shift-Enter**: executa a célula e cria uma nova célula abaixo no modo de comando;
- **Alt-Enter**: executa a célula e cria uma nova célula abaixo no modo de edição;
- **Y**: muda a célula para o tipo código;
- **M**: muda a célula para o tipo *markdown*;
- **A**: insere célula acima;
- **B**: insere célula abaixo;
- **X**: recorta a célula;
- **C**: copia a célula;
- **V**: cola a célula do *clipboard* abaixo;
- **shift-V**: cola a célula do *clipboard* acima;
- **D, D**: deleta uma célula;
- **Z**: desfaz o apagar célula; e
- **L**: ativa a enumeração das linhas.

Os seguintes atalhos estão implementados por default para célula no modo de edição:

- **Tab**: autocompleta durante a digitação; e
- **Shift-Tab**: abre uma minijanela com parte da documentação sobre a parte do código durante a documentação.