

Curso Profissional: Programador/a de Informática
PSD – 11.º ano: UFCD 10793 – Fundamentos de Python

Anaconda e ambientes de desenvolvimento

Ano letivo 22/23

IPython

O [IPython](#) (agora integrado no projeto [Jupyter](#)) é um ambiente interativo para a utilização livre da linguagem Python como ferramenta de cálculo e visualização, bem como para o desenvolvimento de (pequenos) programas na linguagem Python e sua prototipagem¹ rápida. A versatilidade do ambiente IPython facilita a integração com outros ambientes e linguagens de programação.

O ipython tem os seguintes componentes:

- ▷ Uma poderosa e interativa Shell (interface de linha de comandos) para python;
- ▷ Um kernel Jupyter, para trabalhar com o código Python no Jupyter Notebook e outros frontends interativos.

O IPython está integrado no pacote da distribuição Anaconda.

O ambiente IPython surge-nos dividido em **células (cells)**, onde podemos **escrever e avaliar expressões**. Há também células de texto, onde se podem fazer anotações relevantes. O IPython providencia ainda um conjunto de extensões à linguagem Python, para interação com o sistema operativo e com outros ambientes e linguagens, bem como mecanismos de introspeção (examinação do tipo ou propriedades de um objeto em tempo de execução) e de controlo da computação e que se denominam de **magics**.

No modo **Jupyter Notebook**, a interação dá-se através de uma *interface* muito simples suportado por um *browser*, que por sua vez comunica com um *kernel* que disponibiliza um interpretador de Python.

O ambiente de desenvolvimento Jupyter Notebooks

O **Jupyter Notebooks** é um sistema cada vez mais popular que combina código dinâmico, texto descritivo, saída, imagens e interfaces interativas num único ficheiro de notebook/bloco de notas que pode ser editado, visualizado e usado num navegador/browser da web.

O Jupyter Notebook permite a utilização de mais de 40 linguagens de programação. Pode-se criar um projeto interativo que pode combinar elementos de rich text², equações, código e seus resultados (figuras, tabelas, gráficos interativos).

Os notebooks também podem ser exportados para arquivos de código bruto, documentos HTML ou PDF, ou usados para criar slides interativos ou páginas Web.

¹ Geração de modelos interativos de software para testagem das suas funcionalidades, identificando falhas no mesmo.

² Texto com formatações padrão que podem ser lidos por vários editores ou processadores de texto.

Markdown

O **Markdown**, usado no Jupyter Notebooks, é uma ferramenta de conversão de texto para HTML. Faz uso de uma sintaxe de formatação de texto simples (rich text) e de uma ferramenta de software, que converte a formatação de texto simples para tags HTML.

Diferença entre texto escrito em HTML e Markdown:

Texto e links em HTML (com parte em negrito e abertura de página inset e nova janela):

```
O melhor do mundo da tecnologia está no <b>Canal Tech!</b>
<br> aceda em: <a href="https://canaltech.com.br/"> CanalTech </a> e
<a href = "https://www.facebook.com/canaltech" target="_blank"> Fanpage do
CanalTech </a>
```

Texto em Markdown:

```
O melhor do mundo da tecnologia está no **CanalTech**
aceda em: [CanalTech](https://canaltech.com.br/) e [Fanpage do CanalTech](https://www.facebook.com/canaltech)
```

Nota: Para forçar a mudança de linha deve-se escrever 2 espaços em branco após o texto ...CanalTech*, antes de clicar na tecla enter.

Resultado:

O melhor do mundo da tecnologia está no **Canal Tech!**
aceda em: [CanalTech](https://canaltech.com.br/) e [Fanpage do CanalTech](https://www.facebook.com/canaltech)

Iniciar o Jupyter Notebook

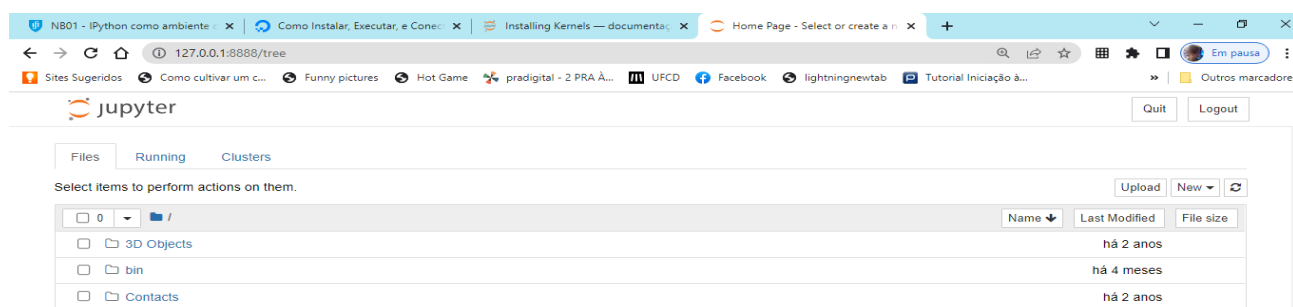
Depois de executado o ANACONDA.NAVIGATOR inicia-se o Jupyter Notebook clicando no ícone Launch

Pode-se também iniciar o Jupyter no navegador padrão: iniciar (ou abrir uma nova guia) no seguinte URL:

<http://localhost:8888/tree> ou acedendo ao menu Iniciar.

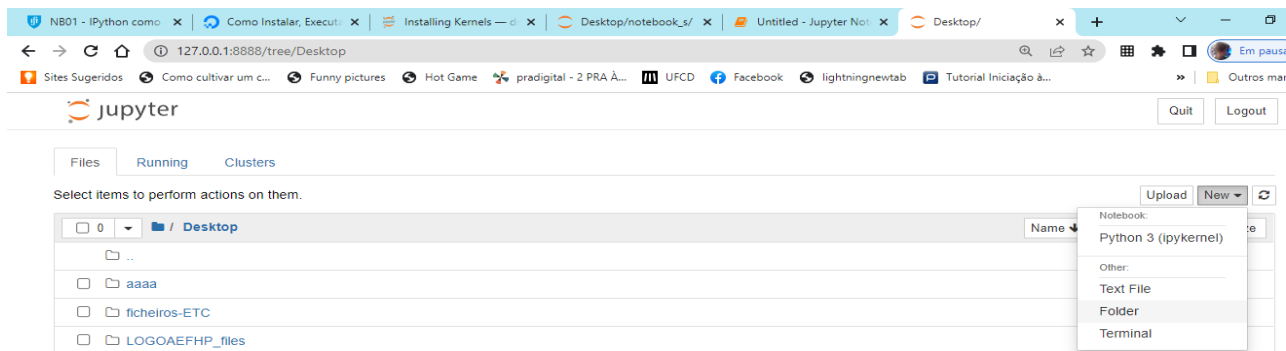


Janela do Jupyter Notebook no browser:

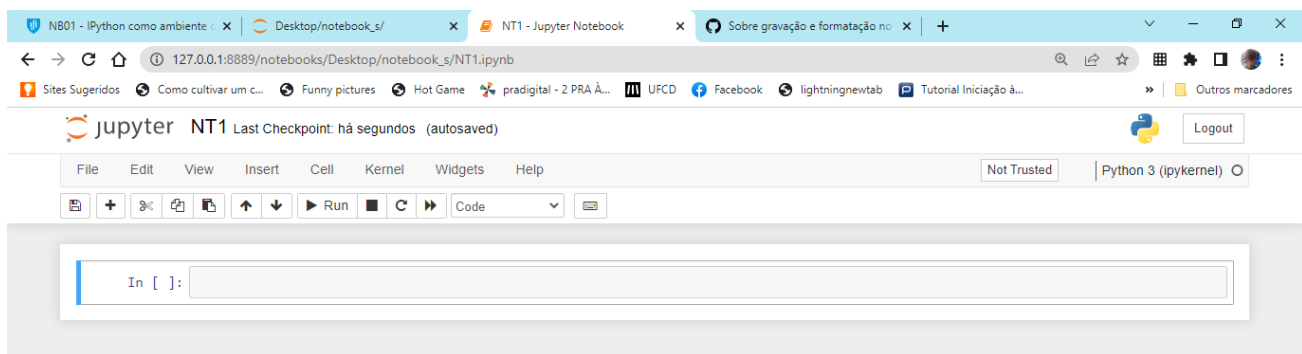
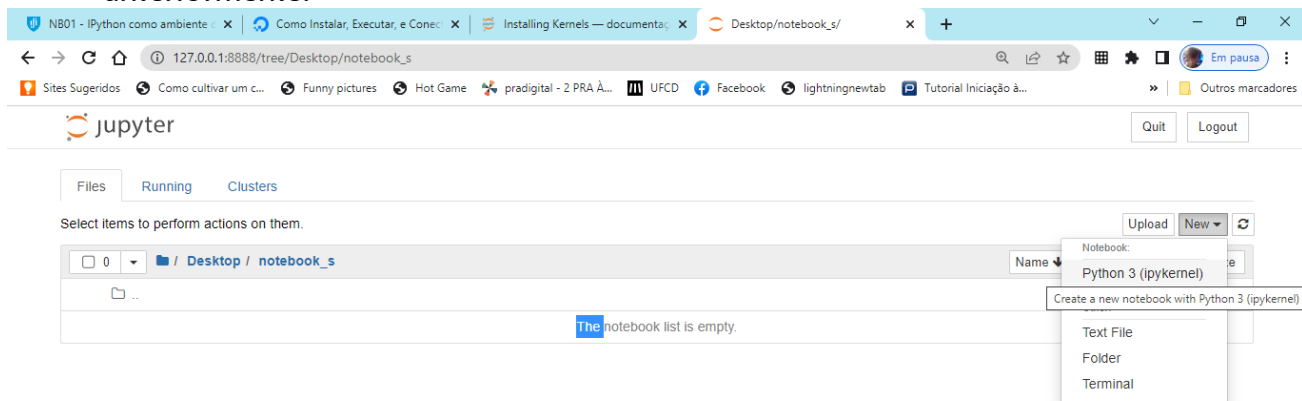


TAREFAS:

1. Começa por criar uma pasta, por exemplo no ambiente de trabalho:



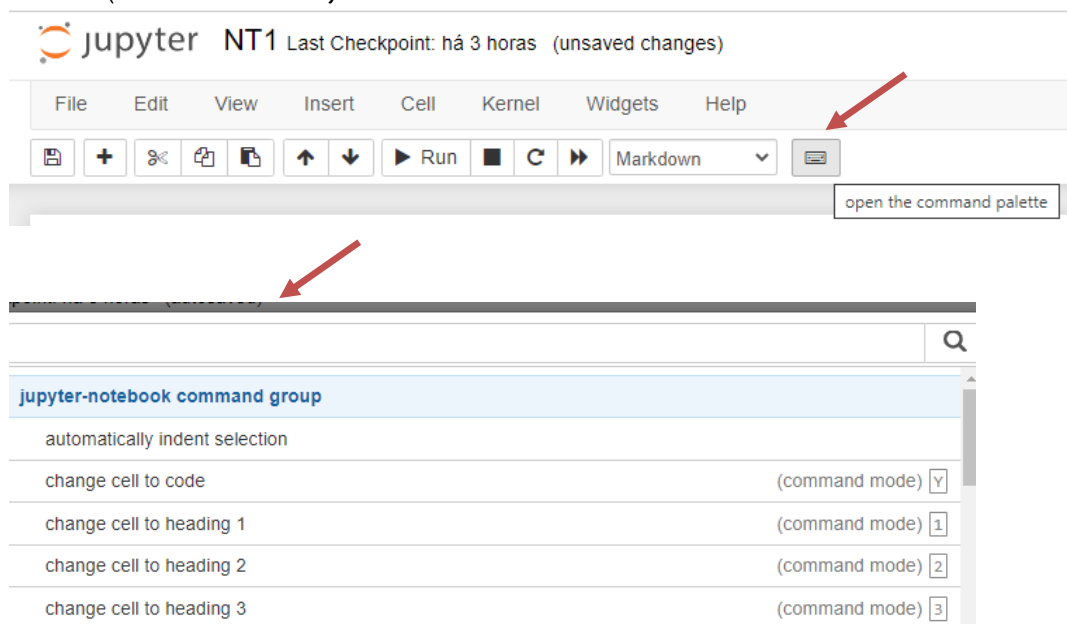
2. Cria um novo Notebook, com o nome NT1, a armazenar na pasta criada anteriormente:



Cada ficheiro criado, de extensão **ipynb**, é um ficheiro de texto que descreve o conteúdo do notebook em formato JSON.

A interface do Notebook

- Explora os menus, para teres uma ideia das suas funcionalidades. Acede, especialmente, ao comando **Palette** onde está listado o grupo de comandos do Jupyter notebook e Widgets, com as respetivas teclas de atalho-> pequeno botão com o ícone do teclado (ou Ctrl + Shift + P).



Conceito de Kernell e Cell

► Um **kernel** é um “mecanismo computacional” que executa o código contido num documento de notebook e que fornece suporte à linguagem de programação usada, na aplicação Jupyter Notebook. Para cada documento de notebook que um utilizador abre, a aplicação Web, neste caso o Jupyter Notebook, inicia um kernel que executa o código para esse notebook. Cada kernel executa o código de uma única linguagem de programação, existindo kernels disponíveis para várias linguagens, como, por exemplo, o python, Julia, Ruby e Go.

O kernel interage com as aplicações Jupyter e as suas interfaces de utilizador.

O **ipyKernel** é o kernel Jupyter de referência construído sobre o **IPython** e que fornece um ambiente poderoso para uma computação interativa em **Python**. É o ipykernel que permite ao Jupyter Notebook criar uma exibição rica de HTML, imagens, LaTeX³, som e vídeo e ainda apresentar **widgets** interativos (através do uso do pacote ipywidgets).

Quando é executada uma célula de código, esse código é executado dentro do kernel. Qualquer saída é devolvida para a célula a ser exibida. O estado do kernel persiste ao longo do tempo e entre as células — pertence ao documento como um todo e não a células individuais.

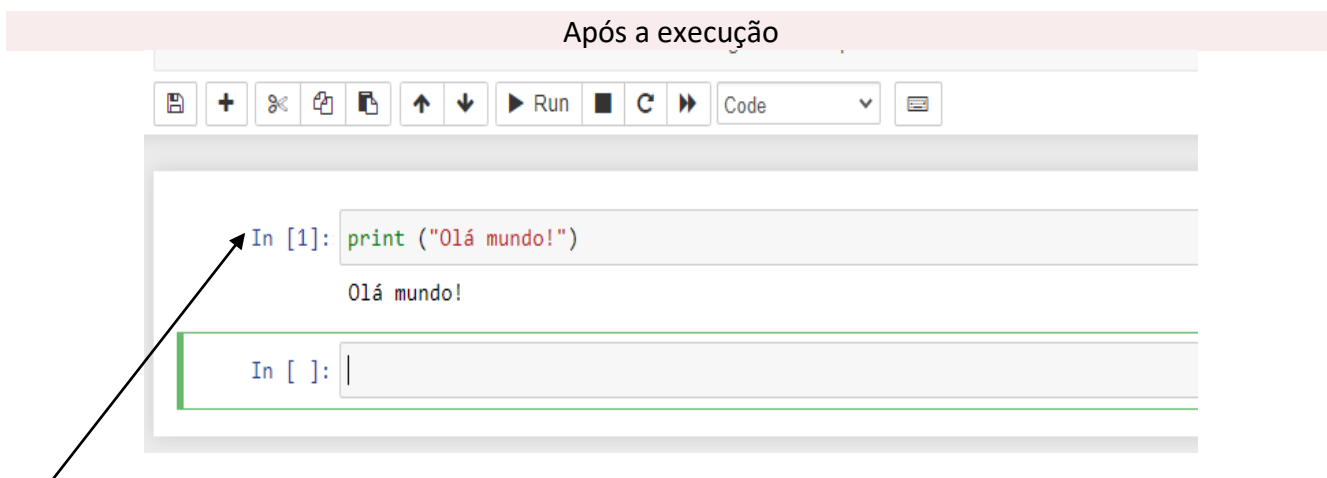
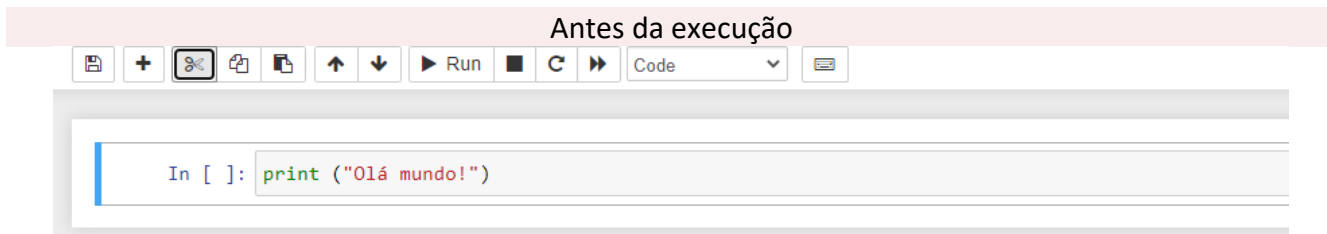
Por exemplo, ao importar-se bibliotecas ou declarar-se variáveis numa célula, estas estarão disponíveis em qualquer célula.

³ O **LaTeX** é um sistema de composição tipográfica de qualidade, que inclui recursos destinados à produção de documentos técnicos e científicos, fazendo uso da biblioteca **MathJax**, escrita em JavaScript, que é usada para exibir fórmulas matemáticas e notações nos navegadores de forma bastante fácil e eficiente.

► Uma **cell** (célula) é um contentor para o texto a ser exibido no notebook ou código a ser executado pelo kernel do notebook.

As células formam o corpo do notebook. Existem 2 tipos principais de células:

Células de código -> a ser executado pelo Kernel e que, após essa ação, exibe a saída desse código abaixo do código que o gerou.



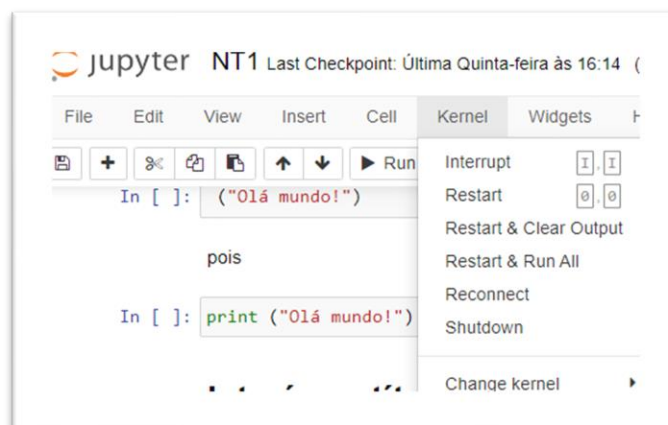
o **rótulo da célula** terá mudado de **In []** para **In [1]**. E irá ser feita uma numeração, que corresponde à ordem de execução no **Kernel**

Na maioria das vezes, quando se cria um Notebook o fluxo é de cima para baixo. Mas é comum voltar para fazer mudanças. Quando precisamos de fazer alterações numa célula anterior, a ordem de execução que podemos ver à esquerda de cada célula, como **In [1]**, pode ajudar a diagnosticar problemas pela observação da ordem em que as células foram executadas.

O rótulo da célula muda para **In[*]** quando está em execução.

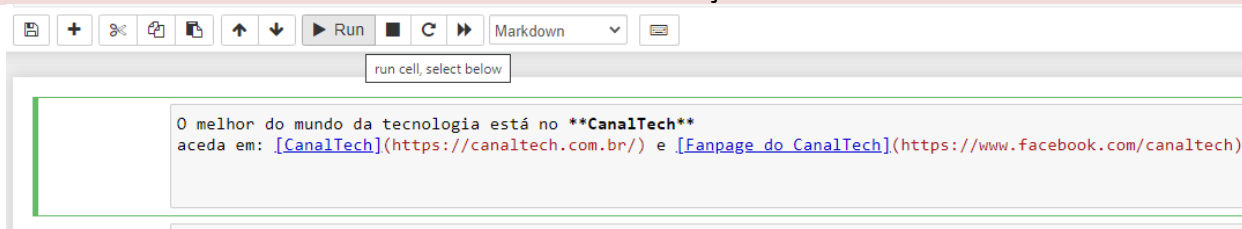
No menu Kernel, do Jupyter Notebook, existem várias opções para redefinição do Notebook criado:

- **Restart**: reinicia o kernel, limpando assim todas as variáveis etc. que foram definidas.
- **Restart Clear Output**: o mesmo que acima, mas também limpará a saída exibida abaixo das células de código.
- **Restart Run All**: o mesmo que acima, mas também executará todas as células na ordem do primeiro ao último.
- Se o kernel estiver travado/bloqueado num cálculo para pará-lo, pode-se escolher a opção **Interrupt**.

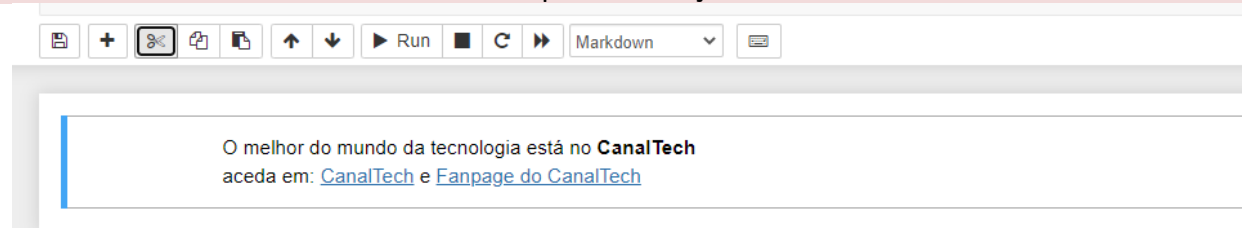


Células de Markdown -> contém texto formatado usando **Markdown** (linguagem de marcação) e exibe, após execução, a sua saída na mesma célula.

Antes da execução



Após a execução



Cor do contorno das células

Contorno **verde** — a célula está no “modo de edição”

Contorno **azul** — a célula está no “modo de comando”

Lista de algumas dos atalhos de teclado do Jupyter

- Alternar entre o modo de edição e de comando com **Esc** e **Enter**, respetivamente.
- Uma vez em modo de comando:
 - Rolar para cima e para baixo nas células com as teclas **Up** e **Down**
 - Pressionar **A** ou **B** para inserir uma nova célula acima ou abaixo da célula ativa.
 - **M** transformará a célula ativa numa célula Markdown.
 - **Y** irá definir a célula ativa para uma célula de código.
 - **D + D** (**D** duas vezes) excluirá a célula ativa.
 - **Z** irá desfazer a exclusão da célula.
 - Segurar **Shift** e pressionar **Up** ou **Down** para selecionar várias células de uma só vez. Com várias células selecionadas, **Shift + M** mesclará a seleção (juntará).
 - **Ctrl + Shift + -**, no modo de edição, dividirá a célula ativa no cursor.
 - Também podemos clicar e **Shift + Clicar** como rato na margem à esquerda das células para seleccioná-las todas.

Tarefa: Experimenta os comandos acima do ficheiro MT1.ipynb

Exemplo de utilização de Markdown

Isto é um título de nível 1

Isto é um título de nível 2

Este é um texto simples que forma um parágrafo. Adicione ênfase por meio de **negrito** e **bold**, ou *itálico* e *itálico*.

Os parágrafos devem ser separados por uma linha em branco.

* Às vezes queremos incluir listas.

* Que pode ser marcado com asteriscos.

1. As listas também podem ser numeradas.

2. Se queremos uma lista ordenada.

[É possível incluir hiperlinks](https://www.aefhp.pt)

O código embutido usa acentos graves únicos: `foo()`, e os blocos de código usam acentos graves triplos:

```
'''  
bar()  
'''
```

Ou pode ser recuado/indentado por 4 espaços:

```
foo()
```

E, finalmente, é fácil adicionar imagens: `![Alt text](http://aefhp.pt/templates/hot_coffeine/images/LOGOAEFHP.png)`

Vê como este Markdown ficaria, assim que fosse executada a célula:

Markdown escrito no jupyter Notebook:

Isto é um título de nível 1

Isto é um título de nível 2

Este é um texto simples que forma um parágrafo. Adicione ênfase por meio de **negrito** e **bold**, ou *itálico* e *itálico*.

Os parágrafos devem ser separados por uma linha em branco.

* Às vezes queremos incluir listas.

* Que pode ser marcado com asteriscos.

1. As listas também podem ser numeradas.

2. Se queremos uma lista ordenada.

[É possível incluir hiperlinks](https://www.aefhp.pt)

O código embutido usa acentos graves únicos: `foo()`, e os blocos de código usam acentos graves triplos:

```
'''  
bar()  
'''
```

Ou pode ser recuado/indentado por 4 espaços:

```
foo()
```

E, finalmente, é fácil adicionar imagens: `![Alt text](http://aefhp.pt/templates/hot_coffeine/images/LOGOAEFHP.png)`

Vê como este Markdown ficaria, assim que fosse executada a célula:

Markdown após execução da célula:

Isto é um título de nível 1

Isto é um título de nível 2

Este é um texto simples que forma um parágrafo. Adicione ênfase por meio de **negrito e bold**, ou *itálico e itálico*.

Os parágrafos devem ser separados por uma linha em branco.

- Às vezes queremos incluir listas.
 - Que pode ser marcado com asteriscos.
1. As listas também podem ser numeradas.
 2. Se queremos uma lista ordenada.

É possível incluir hiperlinks

O código embutido usa acentos graves únicos: `foo()`, e os blocos de código usam acentos graves triplos:

```
bar()
```

Ou pode ser recuado/endentado por 4 espaços:

```
foo()
```

E, finalmente, é fácil adicionar imagens:



Vê como este Markdown ficaria, assim que fosse executada a célula:

Tarefa: Reproduz o código no teu ficheiro NT1.pyynb

Como anexar imagens:

- ▷ Usar um URL para uma imagem na web (caso da tarefa anterior);
- ▷ Usar um URL local para uma imagem, a manter junto do notebook e no mesmo repositório git¹;
- ▷ Adicionar um anexo usando: “Edit > Insert Image”; o que converterá a imagem numa string e a armazenará dentro do ficheiro.ipynb do notebook . Notar que o ficheiro.ipynb ficará muito maior.

Tarefa: Abre o ficheiro `markdown-cells.ipynb`, que se encontra na Classroom, na disciplina PSD -11º ano. Importa-o para a tua pasta de ficheiros do Jupyter Notebook e abre-o.

Experimenta usar o Markdown que consta neste Notebook .

Consulta ainda o site, com formatações básicas de texto markdown:

<https://documentacao.senior.com.br/decision-center/7.0.0/markdown.htm>

1

O GitHub é um serviço baseado em nuvem que hospeda um sistema de controle de versão (VCS) chamado Git. Ele permite que os programadores colaborem e façam mudanças nos seus projetos compartilhados enquanto mantêm um registo detalhado do seu progresso.

Git é um sistema de controle de versão desenvolvido por Linus Torvalds (o criador do Linux). Qualquer programador numa equipe pode gerir o código-fonte e o seu histórico de mudanças usando ferramentas de linha de comandos de Git

O Git difere dos sistemas de controle de versão centralizados, porque oferece ramificações de recursos (ou *feature branches*). Isso significa que cada engenheiro de software na equipa pode separar uma ramificação de recursos que oferece um repositório local isolado para promover mudanças nos códigos.

Feature branches não afetam a ramificação principal, que é onde o código original do projeto está localizado. Uma vez que as mudanças tenham sido feitas e o código atualizado está pronto, a ramificação pode ser misturada (num processo de *merge*) com o *master branch*. É assim que as mudanças no projeto se tornam efetivas.

Jupyter notebook no VS Code

Pré-requisitos de software -extensões:

- VS Code Python
- Visual Studio IntelliCode

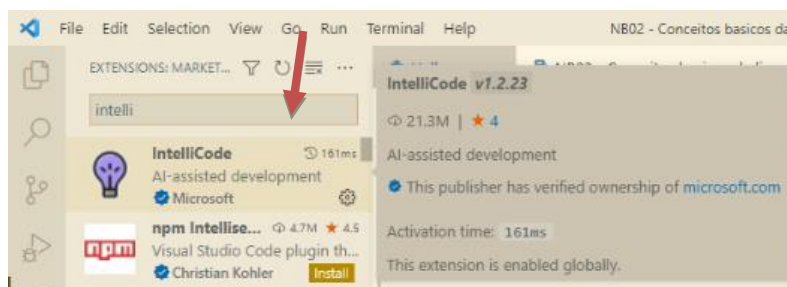
Instalação da Extensão IntelliCode

menu File > Preferences > Extension (ctrl+Shift+X)

OU clicar em



a. Escrever o nome que se pretende aqui



b. Instalar a extensão pretendida

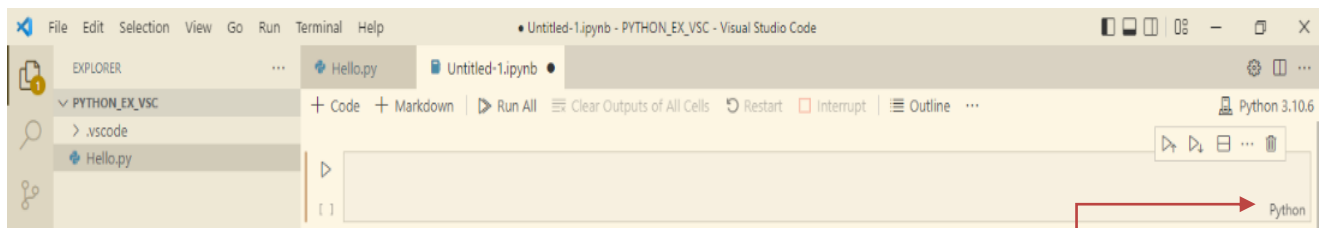
Criar um Jupyter Notebook

- Pressionar o atalho Ctrl+Shift+P, para abrir o **Command Palette** do VS Code ou na engrenagem, presente na barra lateral esquerda ou ainda no menu View.
- No Command Palette digitar Jupyter
- Clicar na opção **"Python: Specify local or remote Jupyter server for connections"**. Em seguida, clicar na opção, *Default*.
O VS Code irá iniciar um servidor do Jupyter Notebook automaticamente, sempre que for carregado ou criado um Notebook.

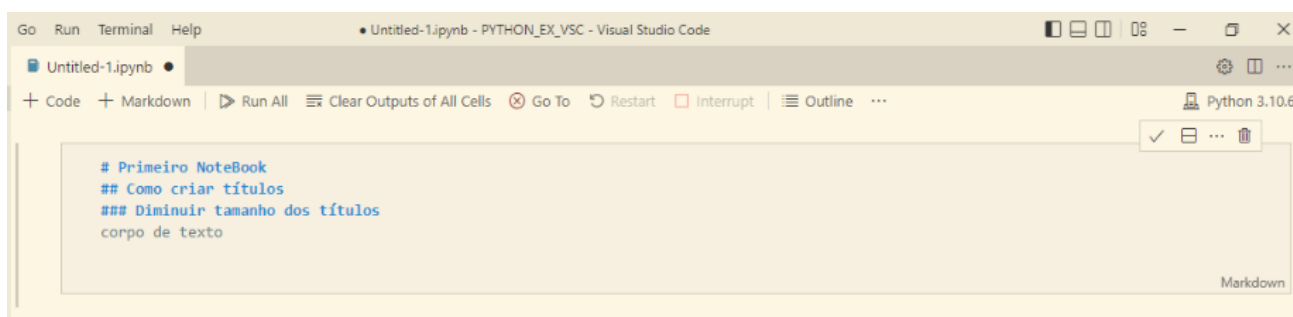
Nota: Este passo pode ser ignorado se a versão da extensão python para VS Code for recente.

- d. No Command Palette, pesquisa por Jupyter e seleciona a opção:
Create: New Jupyter Notebook

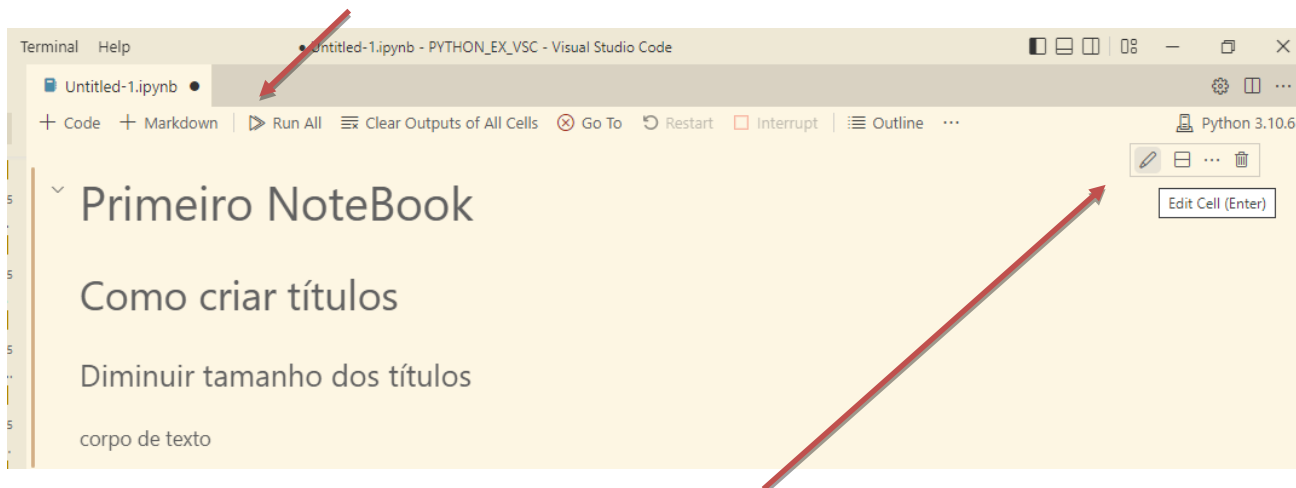
Aparecerá uma janela com o teu 1.º Notebook criado no VS Code



- e. Podes alternar entre código e Markdown clicando em _____
- f. Escreve o texto como se segue (cada cardinal irá definir o tamanho dos títulos. Sem cardinal será para o tamanho do corpo do texto)

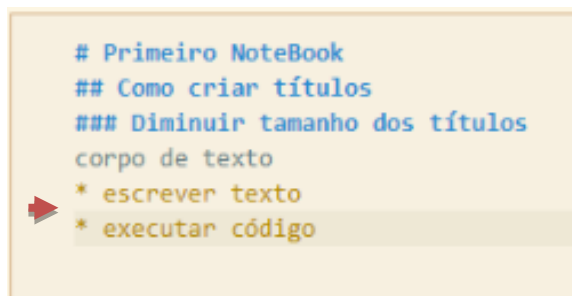


- g. Executa em: **Run All**



- h. Para reeditares o texto clica em : **Edit Cell**

- i. Cria agora Marcas, usando o símbolo asterisco * e um espaço que devem preceder o texto.
Coloca o texto com marcas a negrito, precedendo-o de 2 asteriscos **, mas sem adicionares neste último espaço entre o símbolo e o texto a escrever.



Criar uma tabela

Escolher os títulos das colunas e usar `|` para delimitar as colunas.

Depois, utilizar hífen - na segunda linha para indicar que acima estão os títulos das colunas, usando novamente `|` para delimitar colunas.

Para se especificar o tipo de alinhamento desejado nas tabelas, utilizar `:` ao lado do campo horizontal de hífen ---, na segunda linha da sua tabela:

Alinhado à esquerda > usar `:` no lado esquerdo (alinhamento padrão);

Alinhado à direita > usar `:` no lado direito;

Centrado > usar `:` dos dois lados.

Exemplo:

```
Alinhado a esquerda | Centralizado | Alinhado a direita
:----- | :-----: | -----:
Valor | Valor | Valor
```

Alinhado a esquerda	Centralizado	Alinhado a direita
Valor	Valor	Valor

j. Constrói a tabela seguinte no teu notebook

Exemplo	Valor do exemplo
Exemplo 1	R\$ 10
Exemplo 2	R\$ 8

k. Adiciona uma nova célula, mas agora para código. Escreve nela o código:

`print("Parte 1 do programa")` e executa a célula.

Face ao exposto neste documento, constatámos a importância do **Jupyter Notebook** e porque é uma aplicação muito utilizada pelos desenvolvedores de código, de material para a web e para divulgação científica, pois permite:

- ▶ Executar o código a partir de um browser, com os resultados dos cálculos anexados ao código que os gerou.
- ▶ Ver os resultados de cálculos com representações rich media, como HTML, LaTeX, PNG, SVG, PDF, etc.
- ▶ Criar e usar widgets JavaScript interativos, que vinculam visualizações e controlos interativos da interface do utilizador a cálculos reativos do lado do kernel.
- ▶ Criar texto narrativo usando a linguagem de marcação Markdown.
- ▶ Incluir equações matemáticas usando a sintaxe LaTeX no Markdown, que são processadas/renderizadas no browser pelo MathJax.

Bibliografia

<https://www.walissonsilva.com/posts/jupyter-notebook-no-visual-studio-code>
<https://jupyter-notebook.readthedocs.io/en/latest/examples/Notebook/What%20is%20the%20Jupyter%20Notebook.html>
<https://realpython.com/jupyter-notebook-introduction/#creating-a-notebook>
https://docs.jupyter.org/pt_BR/latest/running.html
<https://coderzcolumn.com/tutorials/python/how-to-display-contents-of-different-types-in-jupyter-notebook-lab#2>
<https://canaltech.com.br/software/Voce-sabe-o-que-e-Markdown/>
<https://docs.pipz.com/central-de-ajuda/learning-center/guia-basico-de-markdown#open>
https://support.zendesk.com/hc/pt-br/articles/4408846544922-Formata%C3%A7%C3%A3o-de-texto-com-Markdown#topic_xqx_mvc_43_line_break
http://web.fcm.unl.pt/moodle/help.php?file=advanced_markdown.html#linebreaks
<https://documentacao.senior.com.br/decision-center/7.0.0/markdown.htm>
<https://daringfireball.net/projects/markdown/>
https://www.hostinger.com.br/tutoriais/o-que-github#O_Que_e_Git
<https://docs.jupyter.org/en/latest/projects/kernels.html>
<https://ipython.readthedocs.io/en/stable/>
<https://acervolima.com/introducao-ao-mathjax/>
<https://docs.github.com/pt/get-started/writing-on-github/getting-started-with-writing-and-formatting-on-github/basic-writing-and-formatting-syntax>
<https://github.com/spatialaudio/nbsphinx/blob/master/doc/markdown-cells.ipynb>
<https://www.homehost.com.br/blog/tutoriais/tabela-de-cores-html/>