



Aula 1 e 2

Curso de Programação Python – 2024

iscte

INSTITUTO
UNIVERSITÁRIO
DE LISBOA



Equipa Docente



Thiago Pontes
thiago_bessa_pontes@iscte-iul.pt



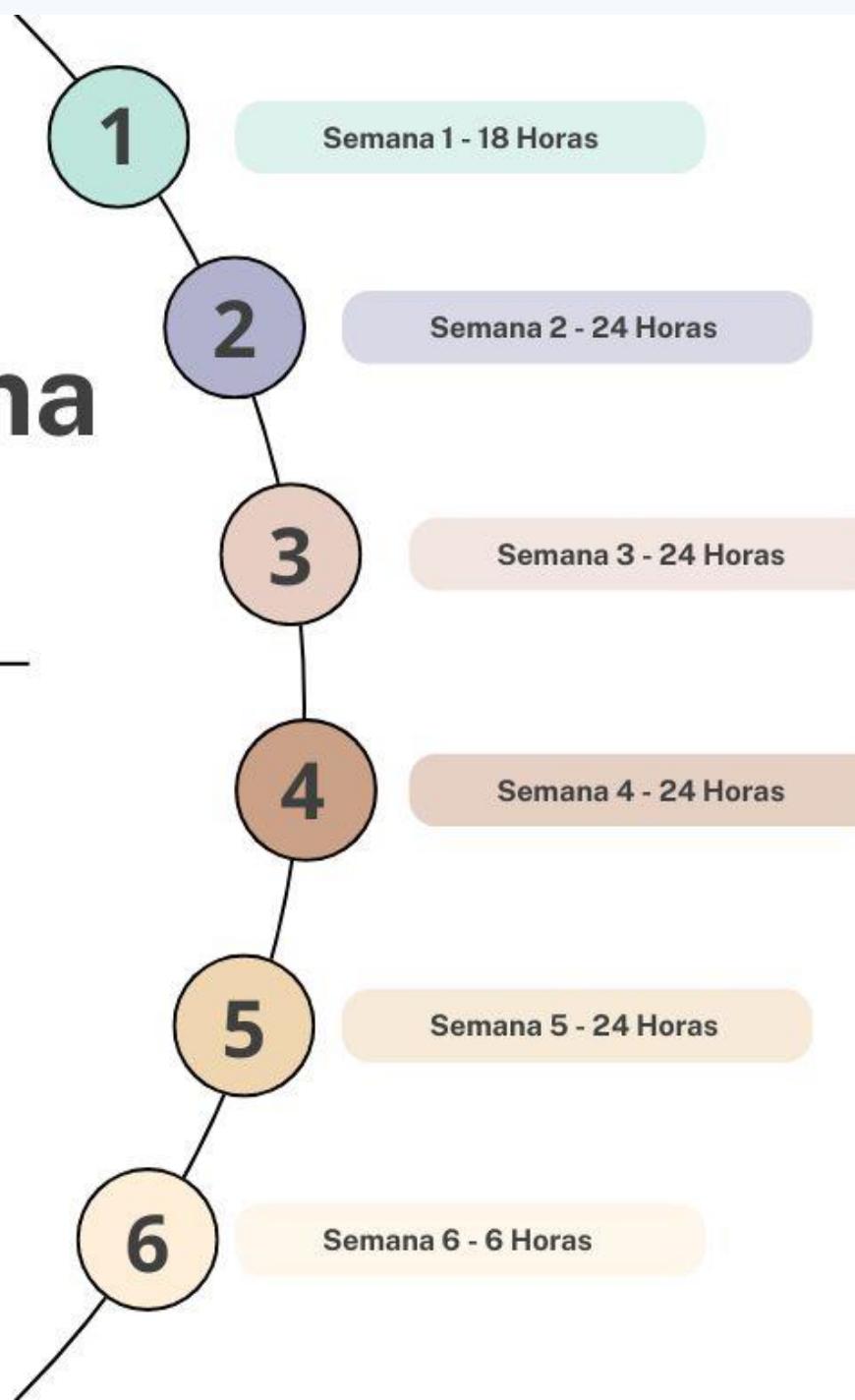
João Monge
Joao_Monge@iscte-iul.pt

Cronograma

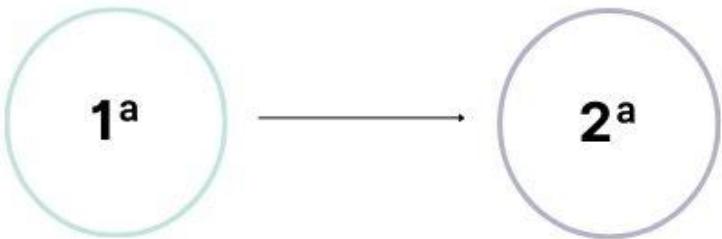
Trabalho Autónomo

Trabalho autónomo:

- estudo geral;
- revisão das aulas;
- resolução dos exercícios;
- prática dos conceitos de forma livre.



Cronograma



Prof. Thiago

Terça Feira
21/05 Aula 1 14h - 15:30h
+3 TA

Quarta Feira
22/05 TA 6h

Quinta Feira
23/05 TA 6h

Sexta Feira
24/05 Aula 2 10h - 11:30h
+3 TA

Prof. Thiago

Segunda Feira
27/05 TA 6h

Sexta Feira
31/05 Aula 4
14h - 15:30h +3 TA

Terça Feira
28/05 Aula 3 14h - 15:30h +3 TA

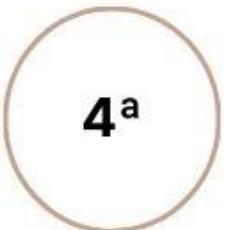
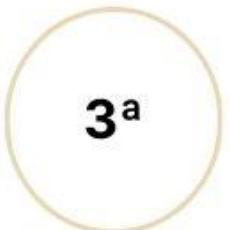
Quarta Feira
29/05 TA 6h

Quinta Feira
30/05 TA 6h



TA = Trabalho Autónomo

Cronograma



Prof. Thiago

Segunda Feira	Sexta Feira
03/06 TA 6h	07/06 Aula 6 14h - 15:30h +3 TA
Terça Feira	
04/06 Aula 5 14h - 15:30h +3 TA	
Quarta Feira	
05/06 TA 6h	
Quinta Feira	
06/06 Aula 6 6h	

Prof. Monge

Segunda Feira	Sexta Feira
10/06 TA 6h	14/06 TA 6h
Terça Feira	
11/06 Aula 7 9h - 10:30h +3 TA	
Quarta Feira	
12/06 TA 6h	
Quinta Feira	
13/06 Aula 8 11:30h -13h +3 TA	

Cronograma

5^a



6^a

Prof. Monge	
Segunda Feira	Sexta Feira
17/06 TA 6h	21/06 TA 6h
Terça Feira	
18/06 Aula 9 14h - 15:30h +3 TA	
Quarta Feira	
19/06 TA 6h	
Quinta Feira	
20/06 Aula 10 10h - 11:30h +3 TA	

Prof. Monge	
Segunda Feira	
24/06 TA 6h	
Terça Feira	
25/06 Aula 11 10h - 11:30h	
Encerramento do curso	

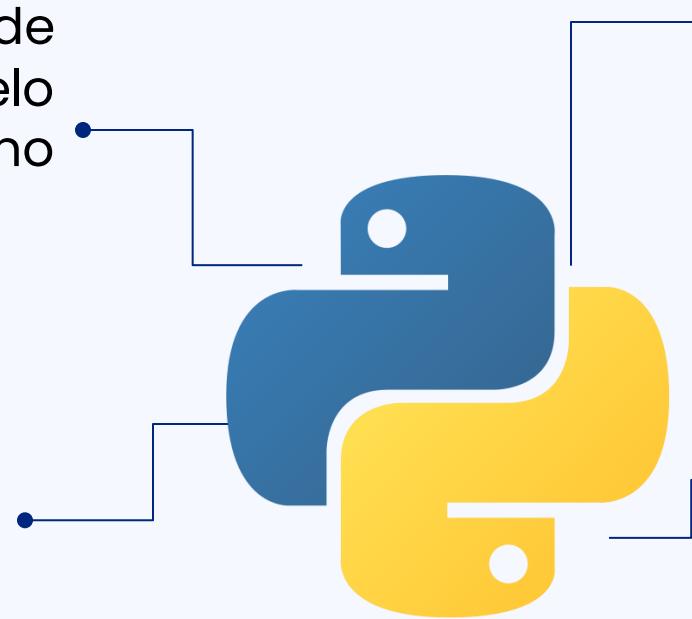


SYNCLUSIVE

Sobre o Python...

É uma linguagem de programação inventada pelo cientista Guido Van Rossum no final da década de 1980.

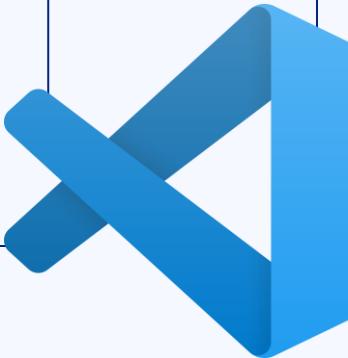
Caracteriza-se pela sua legibilidade e simplicidade.



- É a linguagem de programação mais usada em todo o mundo!
- É grátis e a versão mais recente está disponível aqui:
><https://www.python.org/downloads/>

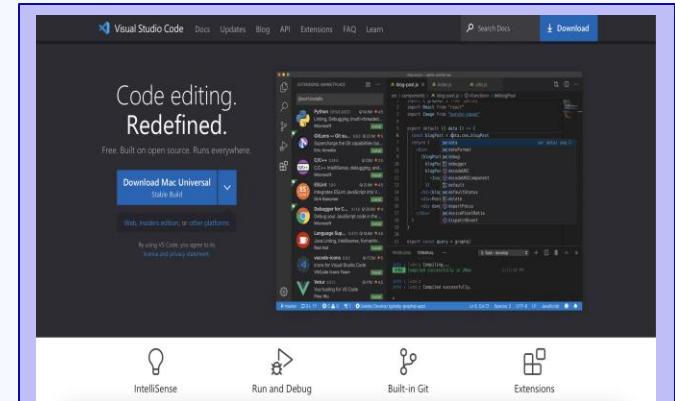
Visual Studio Code - VsCode

O Visual Studio Code é um ambiente de desenvolvimento integrado (IDE).



Permite programar em várias linguagens diferentes.

- É gratuito e o seu download pode ser feito aqui:
→ <https://code.visualstudio.com/>



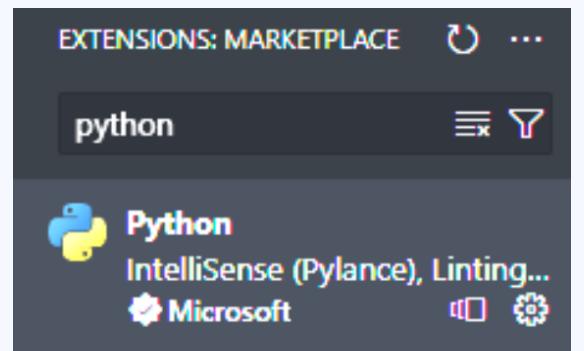
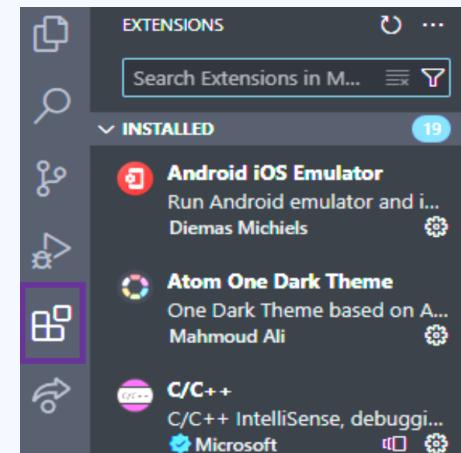
VsCode - Extensões

O que são Extensões?

Extensões são módulos adicionais que podem ser instalados para expandir e personalizar a funcionalidade do ambiente de desenvolvimento.

Como instalar a extensão do Python?

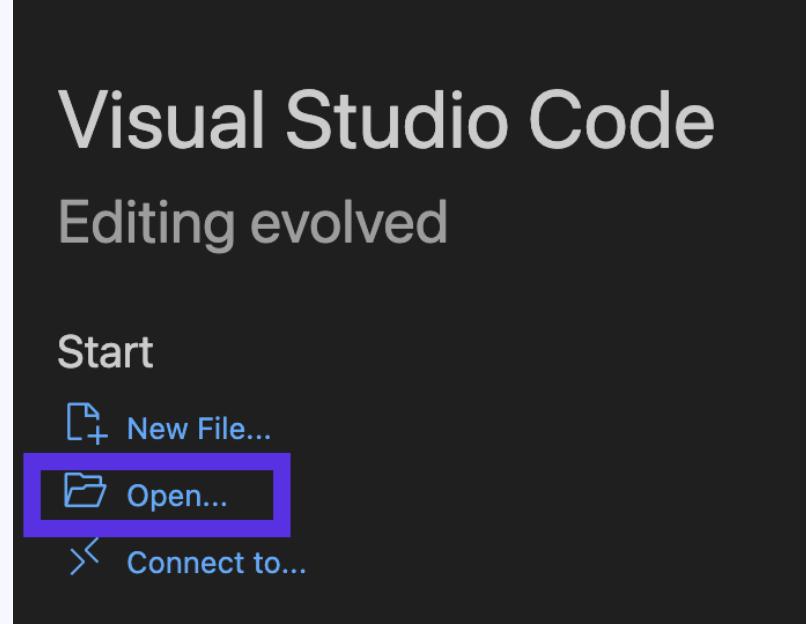
Na "área" das extensões pesquisar "python" e fazer download da versão mais recente:



Abrir Pastas e Criar Ficheiros

1. Criar e Abrir uma Pasta para o Projeto:

1. Criar uma pasta no computador (por exemplo, no ambiente de trabalho).
2. No VSCode, clicar em "Open Folder" e selecionar a pasta destinada ao projeto.



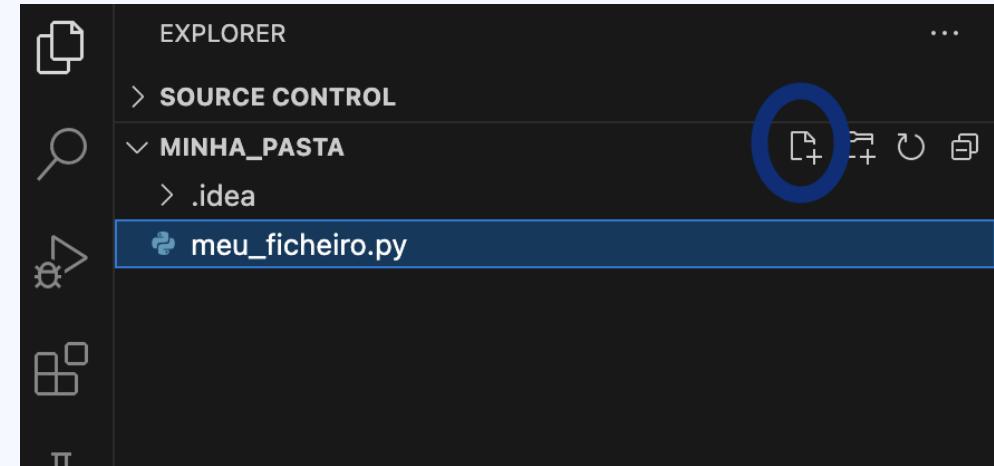
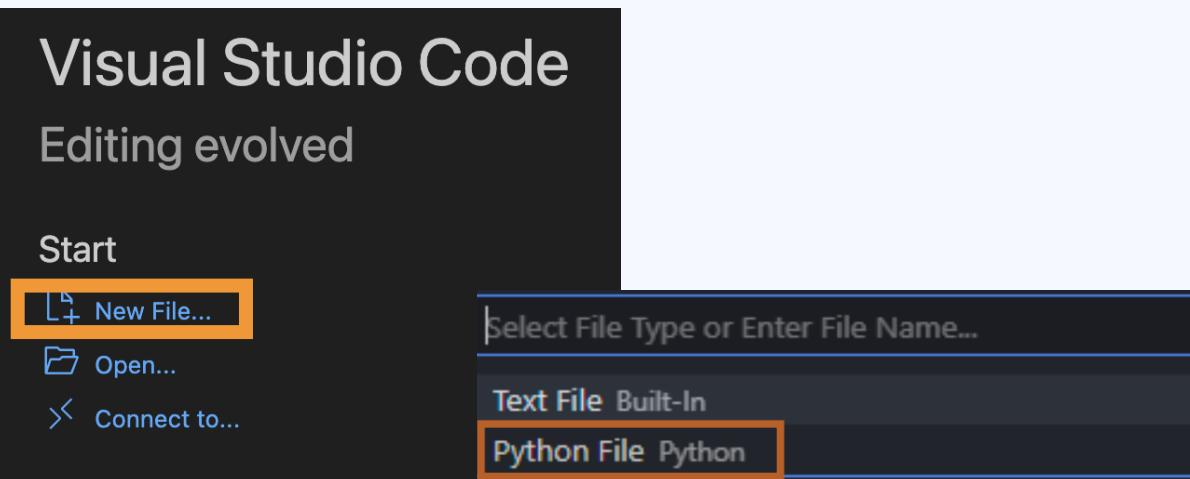
Abrir Pastas e Criar Ficheiros

2. Criar um Ficheiro Python

1. Após abrir a pasta no VSCode, clicar na opção "New File".
2. Escolher a linguagem "Python".
3. Escrever um nome para o ficheiro.

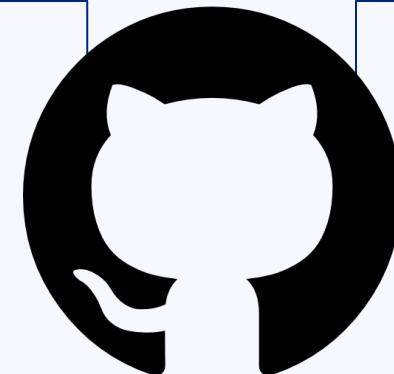
OU

1. Já dentro da pasta, clicar no icon "New File" – representado abaixo.
2. Ao dar o nome ao ficheiro ter em atenção que este tem que acabar com ".py"

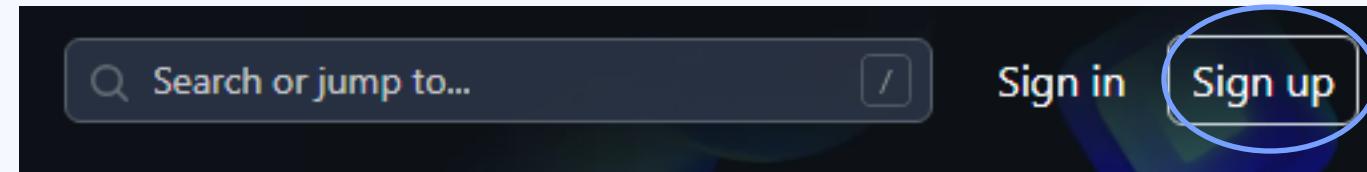


GitHub

Plataforma web que, a partir do sistema de controlo Git, fornece um ambiente colaborativo para desenvolvedores trabalharem em projetos.



- Link de acesso:
-> <https://github.com/>





Repositório

Espaço onde o projeto é armazenado, permitindo o controlo de versões e o trabalho colaborativo.



Branch

Ramificações do código principal (geralmente chamado de "main").

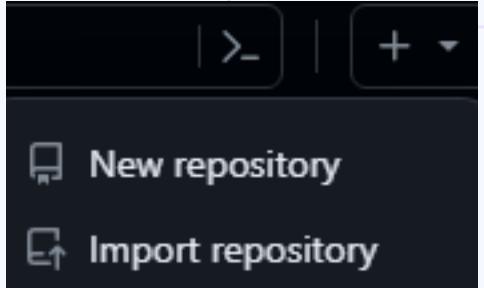


Commit

Representa uma alteração específica no seu código.

Criar um repositório no Github

1. Após fazer **Login**, no canto superior direito, clique no "+" e selecione "Novo repositório".
2. Preencha o nome do repositório, uma descrição opcional, escolha a visibilidade (público ou privado), e pode ainda marcar a opção de inicializar com um arquivo README.
3. Clique em "**Criar repositório**".



The screenshot shows the 'Create a new repository' page. At the top, it says 'Create a new repository' and 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.' Below that, it says 'Required fields are marked with an asterisk (*).'. The 'Owner' field is set to 'David-M-M-Isaac /'. The 'Repository name *' field is empty. A note says 'Great repository names are short and memorable. Need inspiration? How about [refactored-waffle](#)?' Below the name fields, there's a 'Description (optional)' field which is empty. Under 'Visibility', the 'Public' option is selected, with a note: 'Anyone on the internet can see this repository. You choose who can commit.' The 'Private' option is also available. In the 'Initialize this repository with:' section, the 'Add a README file' checkbox is checked, with a note: 'This is where you can write a long description for your project. [Learn more about READMEs](#)'. Below that, there's a 'Add .gitignore' section with a dropdown menu set to 'None'. A note says 'Choose which files not to track from a list of templates. [Learn more about ignoring files](#)'. Under 'Choose a license', a dropdown menu is set to 'None'. A note says 'A license tells others what they can and can't do with your code. [Learn more about licenses](#)'. At the bottom, a note says 'You are creating a public repository in your personal account.' and a green 'Create repository' button is at the very bottom right.

Git

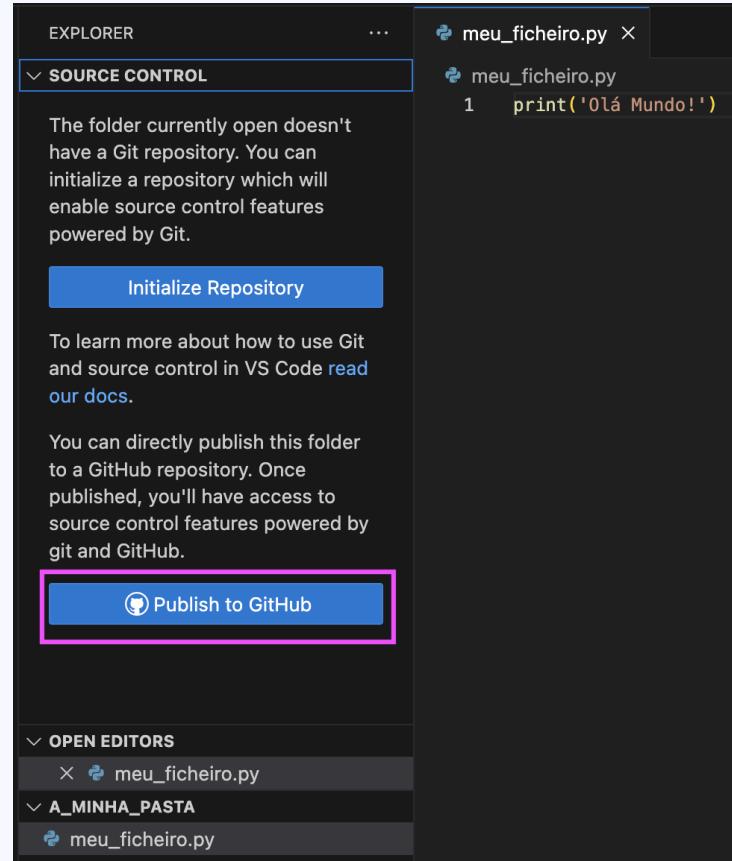
Para fazer a ligação entre o VsCode e o Github, vamos necessitar de um sistema de controlo - o **Git**.

O Git está disponível para download aqui :

- <https://git-scm.com/downloads>

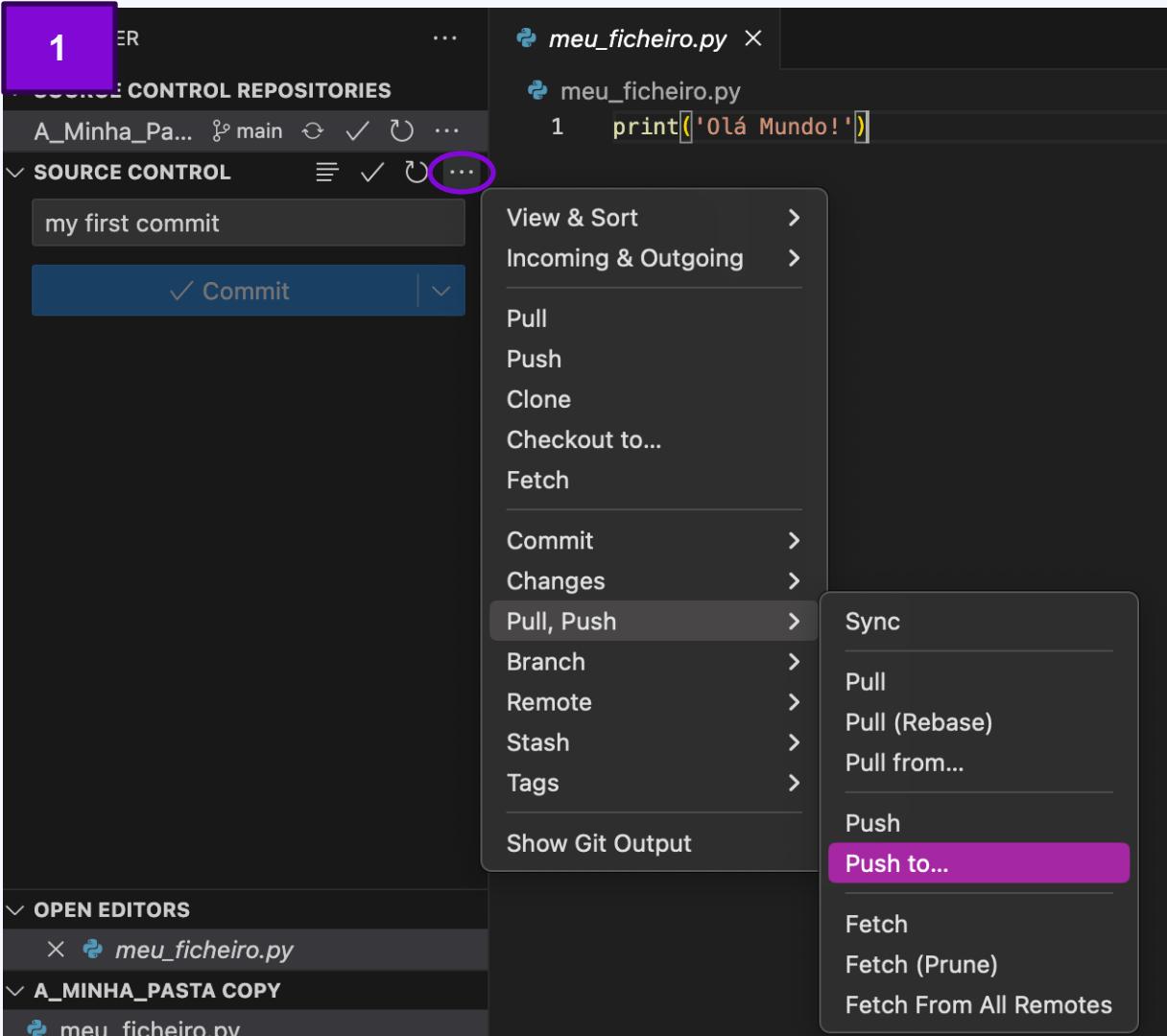


Ligar ao Vs



1. Dentro do VsCode com a pasta criada inicialmente aberta: entrar na área **Source Control** e carregar em **Publish to GitHub**.
2. Caso a sessão não se encontre iniciada, o Vscode irá solicitá-lo.

1



De seguida, fornecemos o nome do repositório, no qual queremos inserir o nosso código! - o repositório criado anteriormente

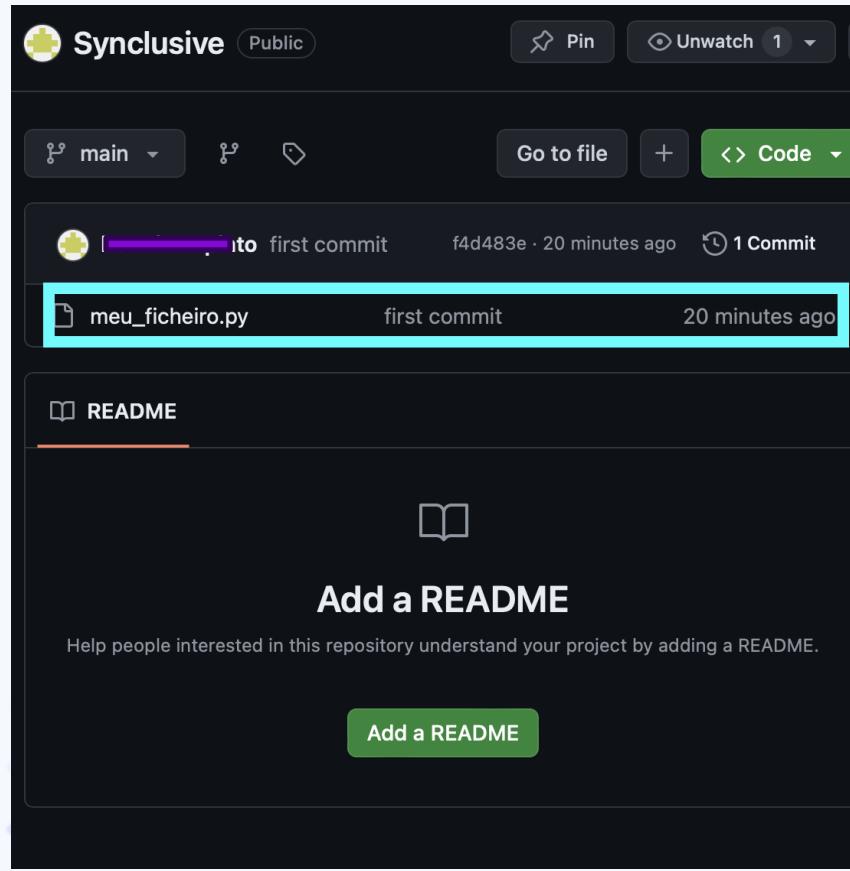
2

Synclusive

Please provide a remote name (Press 'Enter' to confirm or 'Escape' to cancel)

0 1º Commit

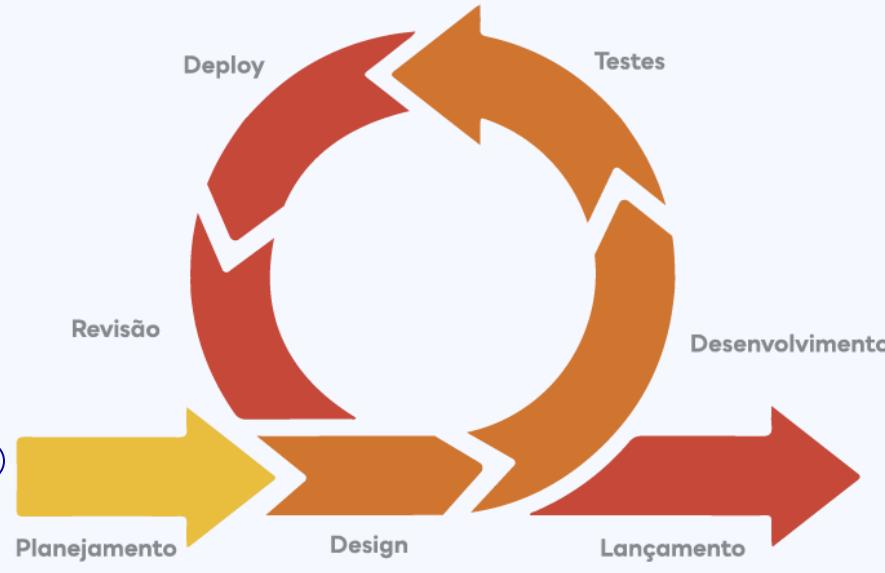
Agora, quando regressamos ao nosso repositório no GitHub, o ficheiro já se encontra lá!



... Dúvidas?

Desenvolvimento Ágil de Software

- O **desenvolvimento ágil** baseia-se num conjunto de métodos e práticas de software que enfatizam a entrega iterativa, colaboração em equipa, e a capacidade de responder rapidamente a mudanças e obstáculos de forma simples.
- Vamos procurar trabalhar recorrendo a este tipo de desenvolvimento durante as aulas do curso.

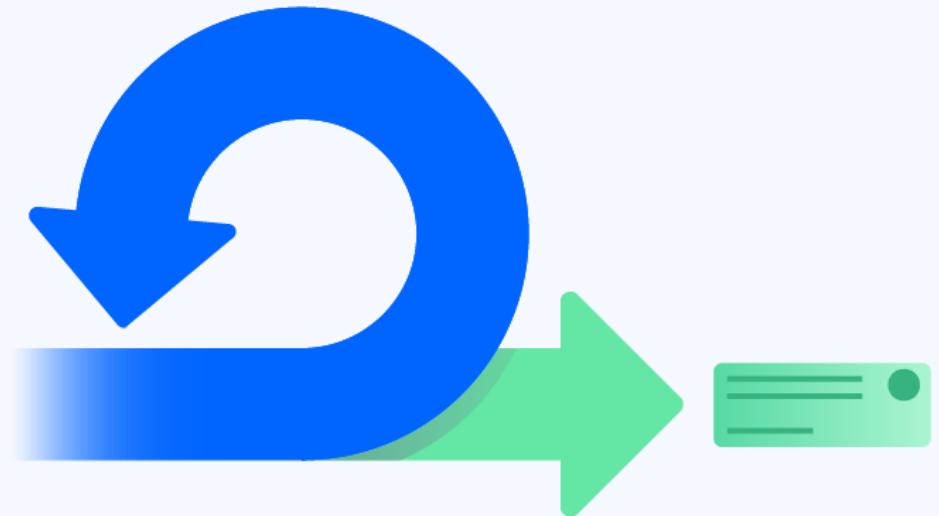


SCRUM

Este método propõe que um projeto seja dividido em diversos ciclos de atividades que promovam a eficiência do mesmo como:

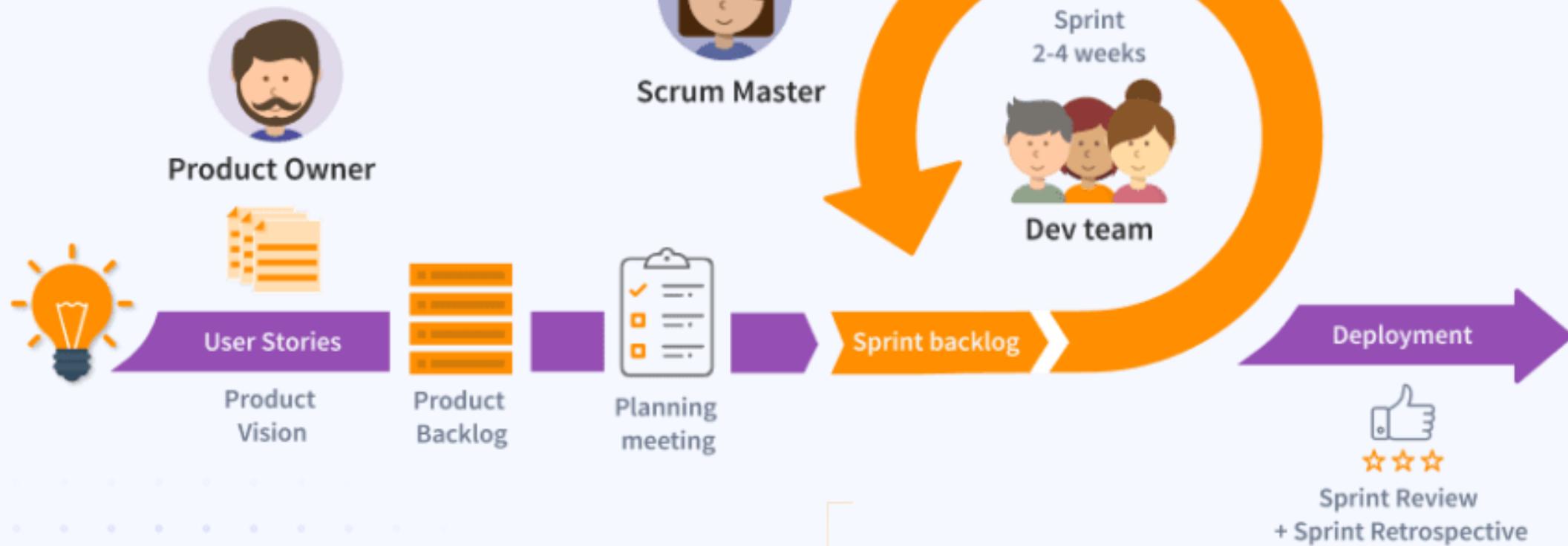
- reuniões frequentes para que a equipa possa concordar sobre o que tem que ser realizado;
- pensar em diferentes formas de melhorar o processo com alguma agilidade.

Assim, garante-se que o projeto é acompanhado e está constantemente a ser melhorado.



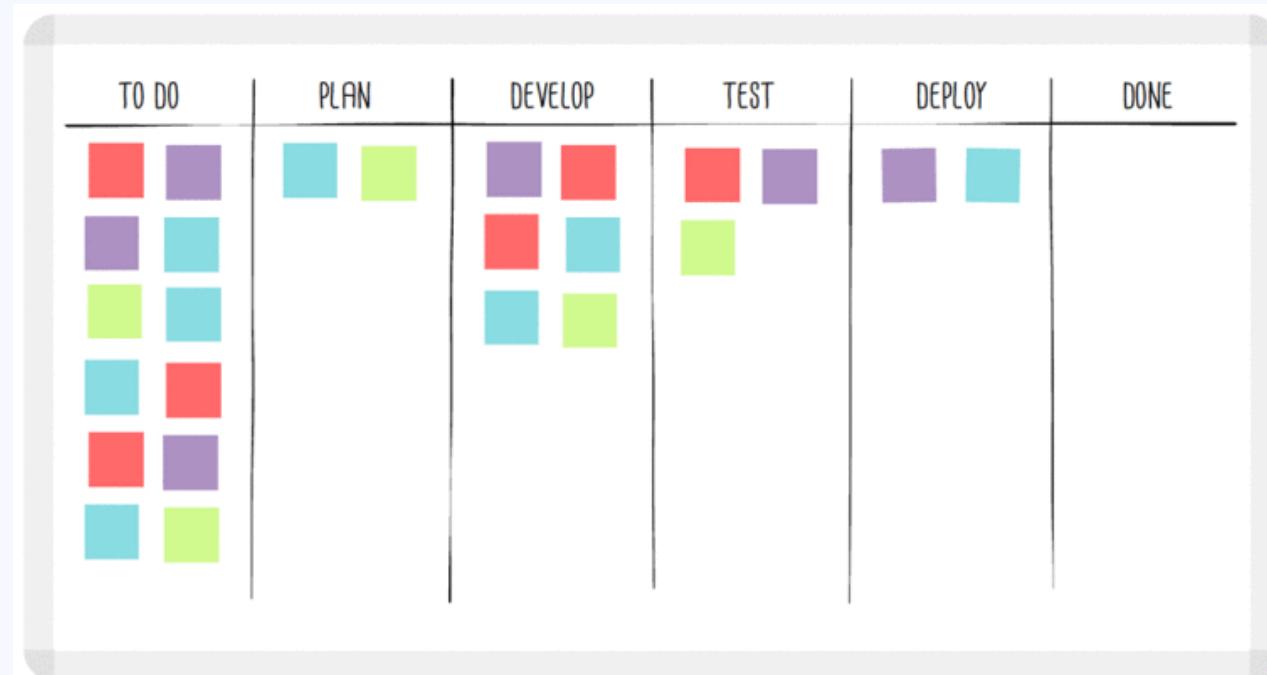
A reter...

SCRUM PROCESS



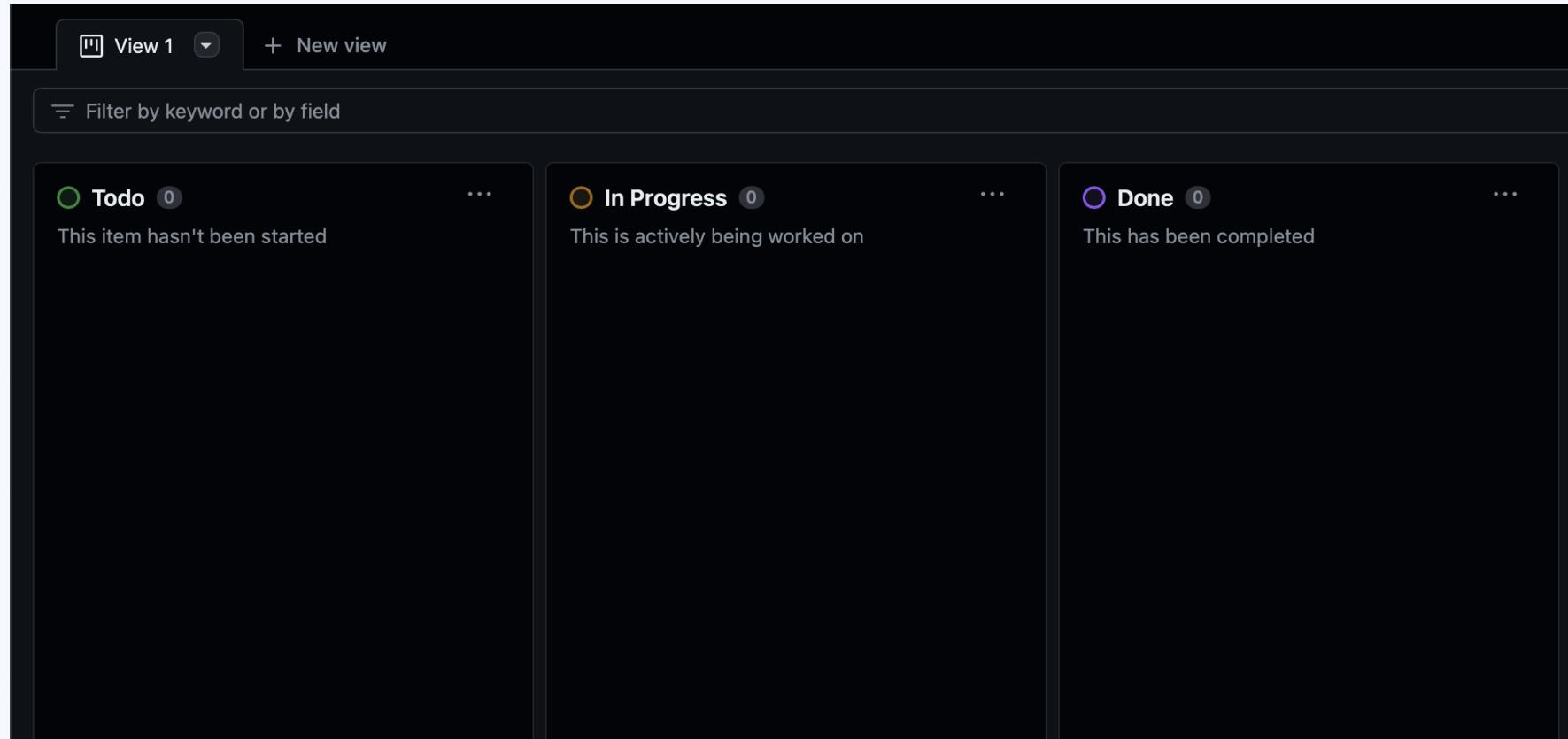
Quadro KAN BAN

- Um quadro Kanban é uma ferramenta dinâmica de gestão de projetos que facilita a visualização do trabalho necessário para a conclusão de um projeto.



NO GITHUB

-> Ao criarmos um novo projeto num repositório do **GitHub**, podemos criar um novo projeto com base num quadro (board) que irá agir como **Kan Ban**:



Tarefa:

Crie o seu primeiro projeto no repositório que foi criado nesta aula. Utilize o quadro Kan Ban para planificar o seu trabalho autónomo.

→ O Kan Ban vai ser uma ferramenta útil para organizar o trabalho a ser feito ao longo do curso ;)



Dúvidas?



Agora... vamos programar!

01

Constantes e
Variáveis

02

Sequência lógica e
instruções

03

Tipos de Dados

04

Entrada e Saída de
Dados

05

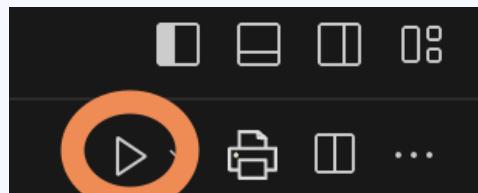
Operações lógicas,
aritméticas e relacionais



Num novo ficheiro Python, escreve o seguinte código:

```
print('Hello World!')
```

Agora, é possível correr o código escrito - carregando na seta no canto superior direito do ecrã.



O que é “impresso” na consola? 

E se mudarmos o código para:

```
print(Hello World!)
```

?

Variáveis e Constantes

Variáveis

- Em Python, uma variável é utilizada para armazenar valores na memória do computador – funciona como uma “caixa” onde podemos guardar informações e dar um nome, que permite aceder a esses valores posteriormente.
- O nome de uma variável pode conter letras, números e underscores mas nunca pode começar com uma número.
- Criação de variáveis:

```
nome = "João"  
idade = 21  
adulto = True
```



Variáveis e Constantes

Constantes

- Uma constante é um valor que não se altera durante a execução de um programa.
- Não existe um mecanismo específico para definir constantes, mas os programadores usam convenções de nome para indicar que uma variável deve ser tratada como uma constante - os seus nomes são tipicamente compostos por apenas letras maiúsculas

PI = 3.14

GRAVIDADE = 9.8

Nota: O Python não impede a modificação dos valores de uma constante – no entanto o seu nome informa o programador que o valor não deve ser alterado.

Tarefa:

Num novo ficheiro python, crie variáveis que armazenem o seu primeiro nome, apelido e idade.



Tarefa:

Num novo ficheiro python, crie variáveis que armazenem o seu primeiro nome, apelido e idade.

Solução

```
nome = "José"  
apelido = "Malhoa"  
idade = 76
```

Sequência Lógica

Código:

```
print("Olá")
print("Mundo")
print("!")
```

Output:

```
/meu_ma_pasta/vari.py
Olá
Mundo
!
```



A **reter:**

Cada instrução só
é executada após
a execução da
anterior!

Questão:

```
nome = "Carlota"  
idade = 10  
idade = 20  
  
print(idade)
```

- Com base no código apresentado, que valor se encontra armazenado na variável **idade** e irá ser impresso na consola? **10 ou 20?**

Tipos de Dados - String



Utilizamos este tipo de dados sempre que queremos recorrer a conjuntos de caracteres – palavras, frases, etc.



Exemplos:

```
variavelString = " Olá Mundo! "
sobreMim = "O meu nome é Laura e tenho 20 anos"
abc = "A B C"
```

Nota: Sempre que queremos recorrer a uma **string**, o conjunto de caracteres tem que estar entre aspas !

Tipos de Dados - Números



Integer

Utilizado em número inteiros.

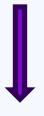


```
idade = 50  
  
numJogador = 7  
  
a = 10
```



Float

Utilizado em números decimais.



```
altura = 1.90  
  
PI = 3.14  
  
b = 10.08
```

Em números decimais deve sempre ser usado “.”

Tipos de Dados - Booleano

- Vamos supor que estamos a programar uma passadeira para peões que só assume dois estados: **Passar** ou **Não Passar**
- Podemos guardar esta informação numa variável String, mas ocuparia muito espaço na memória do computador
- Quando precisamos de guardar apenas dois valores, utilizamos um Boolean que só tem dois valores possíveis: Verdadeiro (True) e Falso (False)



1. **Nota:** True e False têm obrigatoriamente que começar com letra maiúscula.

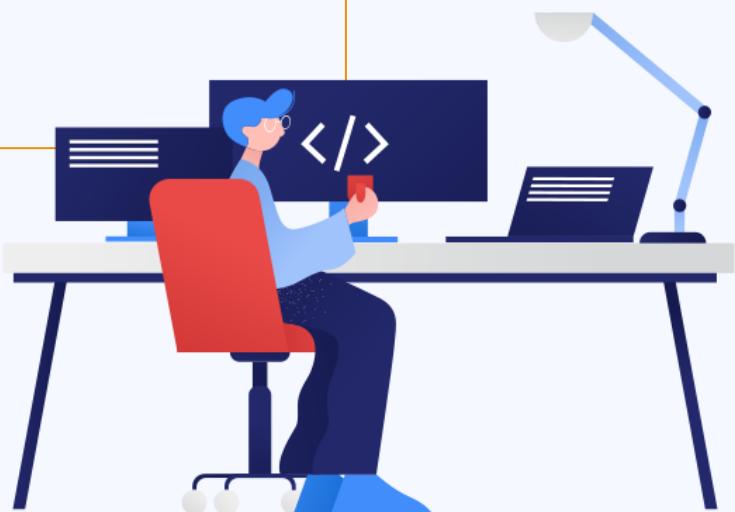
```
passarSemaforo = True
```



só aceita ter o valor **True**, caso o semáforo esteja verde e o valor **False** caso o semáforo esteja vermelho para os peões.

Tipos de Dados : Conclusão

<u>String</u>	Conjuntos de caracteres: letra, palavra, frase, etc
<u>Integer</u>	Números inteiros
<u>Float</u>	Números decimais
<u>Boolean</u>	Opção binomial – True/False



Entrada de Dados

- A entrada de dados consiste em processos de receber informações do utilizador ou de uma fonte externa.
- Para pedir informação ao utilizador na consola recorremos à função: `input()` – colocando a mensagem / pergunta a apresentar ao utilizador entre parênteses
- A informação obtida poderá ser armazenada numa variável.

```
nome = input("Como te chamas?")
```

```
idade = input("Quantos anos tens?")
```

Saída de Dados

- A saída de dados consiste em processos a fim de exibir informações ao utilizador ou grava-las num ficheiro.
- Como já sabemos, para mostrar informações na consola recorremos à função: `print()` – colocando a informação a exibir entre parênteses.
- É possível exibir as informações de uma ou mais variáveis:

```
print("O meu nome é Maria")
```

```
nome = "Maria"  
print("O meu nome é " + nome)
```

Ambas estas instruções vão “imprimir” a mesma informação!

Saída de Dados

Quando queremos mostrar informações na consola, é possível “juntar” informação armazenada em uma ou mais variáveis definidas anteriormente:

- Recorrendo ao sinal **“+”** caso todas as informações sejam do tipo String – é feita uma concatenação das diferentes strings.
- Através de vírgulas **(,)**, caso as informações sejam de diferentes tipos de dados.

```
nome = "Maria"  
apelido = "Santos"  
idade = 25  
  
print("O meu nome é " + nome + " " + apelido)  
  
print("O meu nome é", nome, "e tenho", idade, "anos")
```



Averigue o que é
impresso na consola !



Tarefa:

Elabore um programa de saída e entrada de dados, que pergunta ao utilizador o seu nome e cor favorita, e imprime na consola as informações obtidas.

→ Não se esqueça de armazenar estas informações em variáveis para simplificar o seu código.



Operações Relacionais

Qual é a maçã maior?

A maçã A é maior que a maçã B?

Qual é a maçã menor?

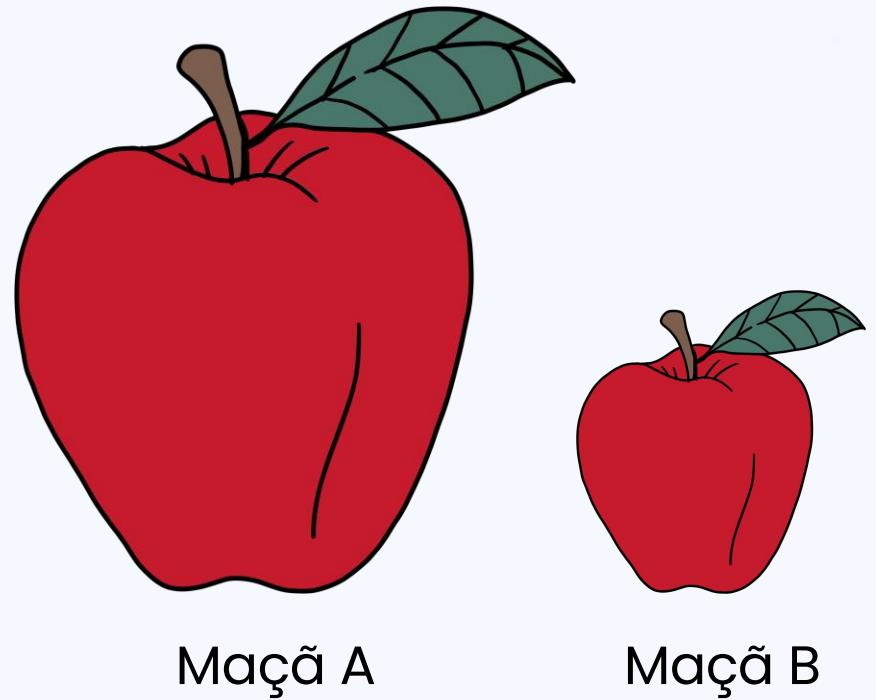
A maçã A é menor que a maçã B?

As maçãs são do mesmo tamanho?

A maçã A é igual à maçã B?

As maçãs são de tamanhos diferentes?

A maçã A é diferente da maçã B?



Operações Relacionais

- As operações relacionais permitem relacionar variáveis e, consequentemente comparar estas.
- Operações relacionais interessam para tomadas de decisão, que serão abordadas na próxima aula.

>	Maior
<	Menor
==	Igual
!=	Diferente

>=	Maior ou Igual
<=	Menor ou Igual

Operações Relacionais: Exemplo

```
tamanhoMacaA = 9  
tamanhoMacaB= 3
```

Qual é a maçã maior?

A maçã A é maior que a maçã B?

```
print(tamanhoMacaA > tamanhoMacaB)
```

Qual é a maçã menor?

A maçã A é menor que a maçã B?

```
print(tamanhoMacaA < tamanhoMacaB)
```

As maçãs são do mesmo tamanho?

A maçã A é igual à maçã B?

```
print(tamanhoMacaA == tamanhoMacaB)
```

As maçãs são de tamanhos diferentes?

A maçã A é diferente da maçã B?

```
print (tamanhoMacaA != tamanhoMacaB)
```

→ Copie o código apresentado para um ficheiro python e responda às perguntas.

Operações Aritméticas

As operações aritméticas são instruções que permitem fazer cálculos aritméticos:

- Somas: +
- Subtrações: -
- Divisões: /
- Multiplicações: *
- Potências: **



→ Mais à frente, iremos aplicar cálculos matemáticos ainda mais complexos!

Operações Aritméticas : Soma

Soma de números

Estamos a fazer uma conta matemática:

```
resultado = 10 + 3 + 7
```



A variável **resultado** vai armazenar o valor da soma. O efeito é o mesmo de: **resultado = 20**

```
num1 = 6  
num2 = 12.5
```



```
soma = num1 + num2
```

A variável **soma** vai armazenar o valor da soma dos valores contidos nas variáveis **num1** e **num2**.

Operações Aritméticas : Soma

Soma de Strings

Estamos a concatenar Strings, isto é a juntar conjuntos de caracteres num só:

```
nome = "Quim "
apelido = "Barreiros"

nomeCompleto = nome + apelido
```



A variável **nomeCompleto** vai armazenar a junção das variáveis **nome** e **apelido**!

Operações Aritméticas

Soma, Subtração, Multiplicação, Divisão e Potência

→ A utilização das restantes operações é semelhante à soma de números, mudando apenas o símbolo da operação:

Operação	Em Python
Soma	soma = num1 + num2
Subtração	subtracao = num1 - num2
Multiplicação	multiplicacao = num1 * num2
Divisão	divisao = num1 / num2
Potência	potencia = num1 ** num2

Tarefa:

Elabore um programa de saída e entrada de dados, que pergunta ao utilizador a sua idade. E imprime na consola a idade atual, a idade que o utilizador terá daqui a 5 anos e a idade que tinha à 5 anos atrás.

→ Recorra às operações aritméticas.



Operações Lógicas



and

Representa uma
conjunção – “E”



or

Representa uma
disjunção – “OU”



not

Representa uma
negação – “NÂO”

- Estas operações são úteis quando queremos comparar várias variáveis, especialmente em estruturas de decisão

Dúvidas?

