

Programar em Python

#01 - Instalação

- Instalar o Python 3.*;
- Instalar o Sublime Text 3 (opcional);
- <https://www.youtube.com/watch?v=fkG6Kxh-mWQ>

#02 - Função print() e Strings

- A função print() imprime todo o seu conteúdo na linha de comandos;
- Uma string é um conjunto de caracteres (letras, números e/ou símbolos);
- Todas as strings são inicializadas e finalizadas com apóstrofes ('string') ou com aspas ("string");
- O caractere de escape é o '\ ' e tem como objetivo fazer com que o caractere que esteja a seguir faça parte da string;
- https://www.youtube.com/watch?v=6_nEyXlayA0

#03 - Operadores Aritméticos

- + -> Adição;
- - -> Subtração;
- * -> Multiplicação;
- / -> Divisão;
- % -> Resto;
- ** -> Exponente
- https://www.youtube.com/watch?v=Gj_liwOOVSs

#04 - Variáveis

- Uma variável é um objeto (uma posição localizada na memória) que representa um valor ou uma expressão;
- Em algumas das linguagens de programação existentes é necessário definir o tipo de variável mas no Python não;
- No Python o tipo de variável é detetado automaticamente consoante o seu conteúdo;
- Tipos de variáveis: inteiro, decimal (float), string, booleano, listas, dicionários e mais alguns;
- É de referir que vamos falar de listas e de dicionários num dos próximos vídeos;
- <https://www.youtube.com/watch?v=LIBRw9IO1sk>

#05 - Operadores de Atribuição

- `a += 3 -> a = a + 3;`
- `b -= 2 -> b = b - 2;`
- `c *= 2 -> c = c * 2;`
- `d /= 2 -> d = d / 2;`
- `e %= 2 -> e = e % 2;`
- `f **= 2 -> f = f ** 2;`
- <https://www.youtube.com/watch?v=ASC94zP4oY4>

#06 - Mais sobre as Strings

- `\n -> Nova linha;`
- `\t -> Nova tab (indentação);`
- `len(variavel) -> Retorna o número de caracteres (no caso da string);`
- `variavel.upper() -> Transforma todas as letras presentes na string em letras maiúsculas;`
- `variavel.lower() -> Transforma todas as letras presentes na string em letras minúsculas;`
- <https://www.youtube.com/watch?v=2a2XVYyRjhY>

#07 - Comentários

- Os comentários (de uma ou de várias linhas) servem para comentar tudo o que seja necessário para que uma outra pessoa compreenda o nosso código;
- <https://www.youtube.com/watch?v=KBGHi3IV7t8>

#08 - Função input()

- A função `input()` tem como principal objetivo "extrair" informações dadas pelo utilizador;
- Resumindo, esta mesma função serve para aumentar a interação com o mesmo;
- <https://www.youtube.com/watch?v=dmuc2yimrig>

#09 - Listas

- Uma lista é um conjunto ordenado de valores, onde cada valor é identificado por um índice;
- Os valores que compõem uma lista são chamados de elementos;
- https://www.youtube.com/watch?v=2G-BrOX4t_w

#10 - if, elif e else

- (if - se) condição for verdadeira:
 - o código dentro do if é executado
- (elif - senão se) condição for verdadeira (só ocorre o elif caso a condição de if seja falsa):
 - o código dentro do elif é executado
- (else - senão) sem condição (só ocorre caso a condição de if e de elif sejam falsas):
 - o código dentro do else é executado
- https://www.youtube.com/watch?v=_ORLNVFJ0nw

#11 - Operadores de Comparação

- == -> Igual;
- != -> Diferente;
- > -> Maior;
- < -> Menor;
- >= -> Maior e igual;
- <= -> Menor e igual;
- and -> E;
- or -> Ou;
- <https://www.youtube.com/watch?v=WeFaMzHDKbg>

#12 - Ciclo for

- Um ciclo permite executar uma instrução ou um conjunto de instruções várias vezes;
- O ciclo for dá voltas (itera) consoante os itens de uma sequência (uma lista, uma string ou um dado range);
- <https://www.youtube.com/watch?v=9RvHP0U1DFw>

#13 - Ciclo while

- Um ciclo permite executar uma instrução ou um conjunto de instruções várias vezes;
- O ciclo while dá voltas (itera) consoante uma determinada condição (tendo ela que ser verdadeira);
- <https://www.youtube.com/watch?v=vztxu4oZGnY>

#14 - break e continue

- O break faz com que a vida do ciclo termine, ou seja, faz com que não se itere por mais nada;
- O continue ignora todo o conteúdo da volta que está a seguir dele mesmo e passa automaticamente para a próxima iteração;
- <https://www.youtube.com/watch?v=H9uqfvXpruc>

#15 - Funções

- Uma função é um pedaço de código organizado e reutilizável que é usado para executar uma única ação relacionada;
- As funções proporcionam uma melhor modularidade e um alto grau de reutilização de código;
- <https://www.youtube.com/watch?v=QqixLxrAF0E>

#16 - return

- O return tem como objetivo retornar algo de uma função para que ao se chamar essa mesma se tenha acesso ao valor retornado;
- <https://www.youtube.com/watch?v=3UASRXS6VQI>

#17 - Argumentos

- Os argumentos são como se fossem variáveis que recebem os seus valores (o seu conteúdo) quando a função é chamada;
- Esses mesmos argumentos são considerados de variáveis locais porque só existem dentro da função;
- As variáveis locais são descartadas quando se retorna ao ponto de chamada;
- <https://www.youtube.com/watch?v=pdvM1ogkyp8>

#18 - Argumentos Padrão

- Os argumentos padrão têm como objetivo usar o seu valor padrão quando o utilizador não envia nenhum valor para esse mesmo argumento;
- <https://www.youtube.com/watch?v=b2NhrDw671w>

#19 - Argumentos Chave

- Os argumentos chave têm como objetivo fazer com que não seja necessário chamar os argumentos da função ordenadamente;
- <https://www.youtube.com/watch?v=IH-yaWXxDkY>

#20 - Argumentos Flexíveis

- Os argumentos flexíveis têm como objetivo fazer com que não seja necessário definir todos os argumentos de uma função quando forem indefinidos;
- Resumindo, os argumentos flexíveis não têm um tamanho definido;
- É necessário colocar o símbolo '*' antes de um argumento flexível ser definido na função;
- <https://www.youtube.com/watch?v=JdnAumrXclE>

#21 - Variáveis Globais e Locais

- As variáveis globais são todas as que são definidas no exterior de uma função (na "raiz" do nosso código) e têm um alcance global;
- As variáveis locais são todas as que são definidas no interior de uma função;
- Resumindo, nem todas as variáveis são acessíveis a partir de todos os locais do nosso código;
- <https://www.youtube.com/watch?v=wD6hwbm11zQ>

#22 - Erros Comuns

- NameError: name 'variavel' is not defined -> Quando a variável apresentada não estiver definida;
- IndentationError: expected an indented block -> Quando a indentação não for feita corretamente;
- SyntaxError: EOL while scanning string literal -> Quando uma string não for finalizada (fechada);
- SyntaxError: invalid syntax -> Quando uma função não for fechada corretamente e não só;
- <https://www.youtube.com/watch?v=Mt6KFW8OjJQ>

#23 - Dicionários

- Os dicionários são estruturas de dados que implementam mapeamentos (coleção de associações entre pares de valores);
- O primeiro elemento do par é chamado de chave (identificador) e o outro é chamado de conteúdo;
- <https://www.youtube.com/watch?v=kbcmRyVMujk>

#24 - Módulos

- Os módulos são basicamente objetos com atributos que se podem ligar através de uma referência;
- Estes mesmos permitem agrupar o seu código, o que torna tudo mais fácil de entender e de usar;
- `dir(modulo)` -> Retorna uma lista ordenada de strings contendo os nomes definidos no módulo em questão;
- Essa mesma lista contém os nomes de todas as variáveis e funções que foram definidas nesse mesmo módulo;
- <https://www.youtube.com/watch?v=GhqKMU7KOQ0>

#25 - Funções Matemáticas

- `sin(variavel)`, `cos(variavel)` e `tan(variavel)` -> Calcular o seno, o cosseno e a tangente;
- `floor(variavel)` -> Remove o valor decimal do dado número;
- `ceil(variavel)` -> Arredonda o dado número para o próximo;
- `abs(variavel)` -> Retorna o valor absoluto do dado número;
- `fsum(variavel)` -> Faz a soma de todos os números dentro de uma lista (por exemplo);
- `pow(variavelA, variavelB)` -> Multiplica o seu valor inicial múltiplas vezes (consoante o número de vezes);
- https://www.youtube.com/watch?v=AQx-_A36bCk

#26 - Data e Hora

- `calendar(ano)` -> Retorna um calendário de um certo ano em formato de leitura;
- `calendar(ano, mes)` -> Retorna o mês escolhido de um certo ano em formato de leitura;
- `time()` -> Retorna a hora do sistema atual em ticks desde a meia-noite do dia 1 de Janeiro de 1970;
- `localtime(time())` -> Retorna a data e a hora atual em formato estruturado;
- `asctime(localtime(time()))` -> Retorna a data e a hora atual em formato de leitura;
- <https://www.youtube.com/watch?v=LMTefEEImNg>

#27 - Ficheiros (Ler e Escrever)

- `open(ficheiro, modo)` -> Abrir o ficheiro;
- `write(texto)` -> Escrever no ficheiro;
- `read()` -> Ler o conteúdo do ficheiro;
- `close(ficheiro)` -> Fechar o ficheiro;
- `'w'` -> Modo de escrita (write);
- `'r'` -> Modo de leitura (read);
- É necessário fechar sempre o ficheiro;
- <https://www.youtube.com/watch?v=B3xJYB2962M>

#28 - Exceções

- Uma exceção é um evento que ocorre durante a execução de um pedaço de código que interrompe o seu fluxo normal de instruções;
- Resumindo, uma exceção representa um erro;
- `IOError` -> quando um erro ocorre relacionado com o tratamento de ficheiros;
- <https://www.youtube.com/watch?v=zL206ukh9yg>

#29 - Classes e Objetos

- Uma classe representa um conjunto de objetos, sendo que, define o comportamento desses mesmos através de métodos e quais estados ele é capaz de manter através dos seus atributos;
- Um objeto é uma instância de uma classe e é capaz de armazenar estados através dos seus atributos e de reagir a mensagens enviadas para ele;
- <https://www.youtube.com/watch?v=kuT-cHH7wvM>

#30 - `__init__`

- O `__init__` tem como função inicializar tudo o que seja necessário ao se criar um determinado objeto;
- <https://www.youtube.com/watch?v=FrpRyp9gwrk>

#31 - Herança

- A herança é um mecanismo pelo qual uma classe (subclasse) pode estender outra classe (superclasse) para tirar proveito das suas possíveis variáveis (atributos) e do seu comportamento (métodos);
- <https://www.youtube.com/watch?v=J7vBPHR1aVk>

#32 - Múltipla Herança

- A múltipla herança faz com que seja possível herdar de várias classes;
- <https://www.youtube.com/watch?v=G2lopvQQKrE>

#33 - Lambda

- O lambda é usado na criação de funções anónimas e estas são assim chamadas porque não são declaradas com o termo "def";
- Resumindo, o lambda tem como principal objetivo simplificar certas e determinadas funções, sendo que estas, não utilizam o termo "return" e sim uma expressão;
- <https://www.youtube.com/watch?v=AgFQ9iuHj4E>

#34 - map

- A função map tem como objetivo mapear vários dados de uma só vez, retornando uma lista referente a esse mesmo mapeamento;
- <https://www.youtube.com/watch?v=kZbglXwZES0>

#35 - Operadores Bit a Bit

- Os operadores bit a bit operam um bit de cada vez e ter a noção do que é o código binário é importante;
- 1 -> Verdadeiro;
- 0 -> Falso;
- & -> Apenas é verdadeiro quando ambos os números corresponderem a 1 e 1;
- | -> Apenas é falso quando ambos os números corresponderem a 0 e 0;
- << -> Deslocam os bits de acordo com o número dado para a esquerda;
- >> -> Deslocam os bits de acordo com o número dado para a direita;
- Depois temos também os operadores ^ e ~;
- <https://www.youtube.com/watch?v=JVdBjzdnQLw>