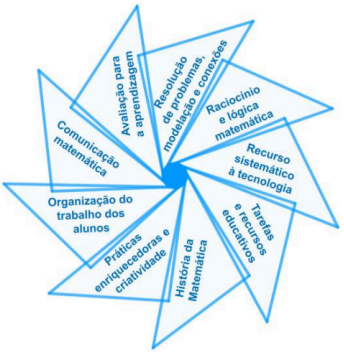
APRENDIZAGENS ESSENCIAIS (AE) - Ensino Secundário

1. Matemática Escolar Orientada para o Futuro
2. Ideias Inovadoras do Currículo
   - Matemática para a Cidadania
   - **Pensamento Computacional**
   - Diversificação de temas no currículo
   - Matemática para todos
3. Ideias chave das Aprendizagens Essenciais



<u>O PENSAMENTO COMPUTACIONAL (PC) NAS AE</u>:

"Os **aspetos comuns entre o Pensamento Matemático e o Pensamento Computacional**, bem como **a relevância atual do Pensamento Computacional na ciência e na sociedade**, justificam que <u>o currículo de Matemática valorize esta abordagem conceptual na resolução de problemas</u>."

"Para cada tema são incluídas notas clarificadoras, nomeadamente no que se refere à sugestão de: **atividades para o desenvolvimento do Pensamento Computacional**, com recurso a exemplos; propostas de possíveis aprofundamentos de alguns temas ou de abordagens alternativas; referências bibliográficas que incluem documentos e recursos para apoio ao trabalho do professor."

"**As atividades de programação devem ser integradas com uma complexidade progressiva**, sendo relevantes para o desenvolvimento de processos algorítmicos, de um pensamento estruturado e do raciocínio lógico, proporcionando um vasto campo de aplicação da Matemática e envolvendo genuinamente a formulação e a resolução de problemas, além de promover o desenvolvimento do pensamento computacional."

PEDRAS ANGULARES DO PENSAMENTO COMPUTACIONAL:

Abstração	Extrair a informação essencial de um problema.
Decomposição	Estruturar a resolução de problemas por etapas de menor complexidade de modo a reduzir a dificuldade do problema.
Reconhecimento de padrões	Reconhecer ou identificar padrões no processo de resolução de um problema e aplicar os que se revelam eficazes na resolução de outros problemas semelhantes.
Algoritmia	Desenvolver um procedimento passo a passo (algoritmo) para solucionar um problema de modo a que este possa ser implementado em recursos tecnológicos, sem necessariamente o ser.
Depuração	Procurar e corrigir erros, testar, refinar e otimizar uma dada resolução apresentada.

A CALCULADORA GRÁFICA CASIO FX-CG50

"O Python é uma linguagem de programação de alto nível, interpretada, para programação de âmbito geral. Criada por Guido van Rossum e inicialmente lançada em 1991, o Python tem uma filosofia de design que dá ênfase à legibilidade do código, notavelmente usando espaços em branco com significado. O Python possui construções que permitem uma programação clara em pequena e grande escala.

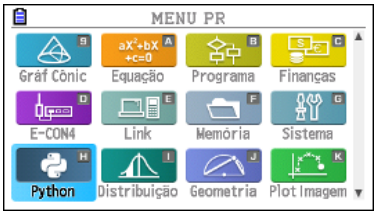
O Python deixa-o trabalhar rapidamente e integrar sistemas de forma mais eficiente.

O Python é utilizado por milhares de pessoas em todo o mundo."

(Fonte: Python Portugal. Disponível em https://python.pt/acerca/)

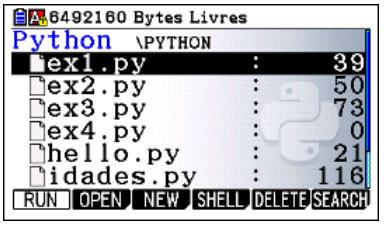
MENU PYTHON:

O modo Python oferece um ambiente de tempo de execução para a linguagem de programação Python. Permite criar, salvar, editar e executar arquivos Python.

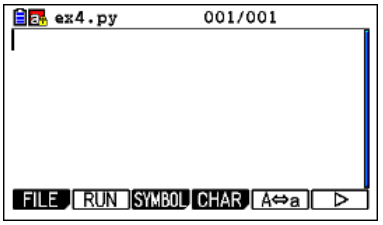


O ambiente do menu Python é constituído por três áreas:

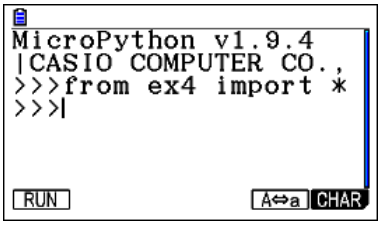
- a Lista de arquivos, em que são exibidos os nomes dos programas com a extensão *.py;
- o Editor, onde se escreve o script (sequência de instruções executadas pelo programa);
- o Shell, onde se podem escrever expressões em linguagem Python e executar scripts.



Lista de arquivos



Editor

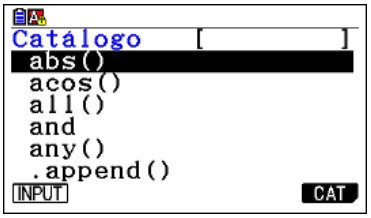


Shell

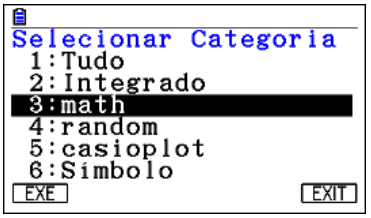
INSERIR TEXTO E COMANDOS:

Existem três formas de inserir texto e comandos no menu Python:

- usando o teclado da calculadora/emulador (pressione [ALPHA] para inserir um caractere alfabético assinalado a vermelho no teclado ou [SHIFT] [ALPHA] para permanecer em modo alfabético);
- usando as funções do Editor (SYMBOL, CHAR, COMMAND, etc.);
- usando o catálogo ([SHIFT] [4]).



Catálogo



Categorias do catálogo

Os operadores aritméticos da linguagem Python são:

Operação	Tecla	Descrição
$x + y$	⊞	Adição
$x - y$	⊟	Subtração
$x * y$	⊠	Multiplicação
x / y	÷	Divisão
$- x$	⊟	Simétrico de x
$x ** n$	^ ou ⊠ ⊠	Potenciação (x elevado a n)
$x // y$	÷ ÷	Divisão inteira de x por n
$x \% y$	F4 (CHAR)	Resto da divisão inteira de x por y

Outras expressões, tais como π e $\sqrt{\ }$, requerem a importação do módulo "math" (from math import *), disponível no catálogo.

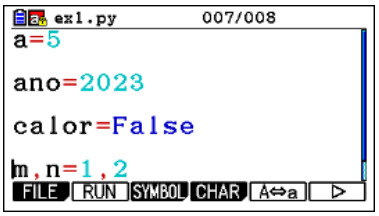
VARIÁVEIS E DADOS:

➟ O Python pode manipular informações de diferentes tipos:

- inteiro (int) – número sem parte decimal;
- real (float) – número com parte decimal;
- lógico (bool) que apenas pode assumir dois valores, True (verdadeiro) e False (falso);
- cadeia de caracteres (str) – sequência de caracteres entre aspas;
- lista (list) – sequência de elementos.

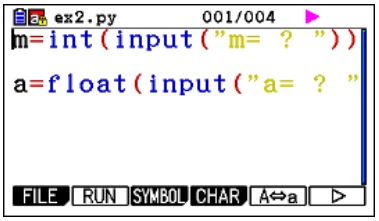
As variáveis são utilizadas para armazenar valores em memória.

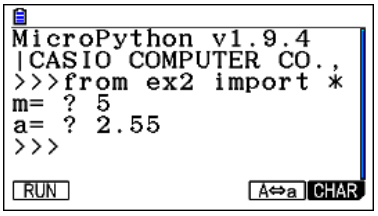
Para atribuir um valor a uma variável, utiliza-se o sinal igual (=).



➟ A entrada de dados (do Shell para o Editor) efetua-se com a função *input()*. Esta função importa dados do tipo cadeia de caracteres (*str*) que podem ser convertidos em números (*int* ou *float*).

Exemplo:

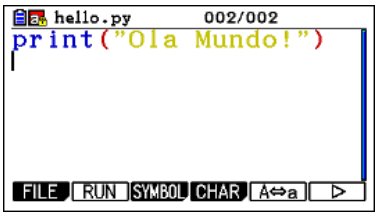


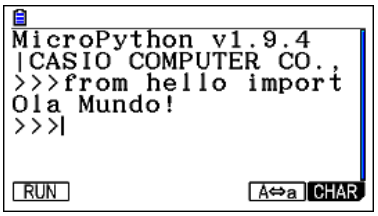


- Pedir um número inteiro (*int*) que será atribuído à variável *m*;
- Pedir um número real (*float*) que será atribuído à variável *a*.

➞ A saída de dados (do Editor para o Shell) efetua-se com a função *print()*.

Exemplo:





ESTRUTURAS DE CONDIÇÃO:

➞ IF - ELSE

A estrutura condicional *if – else* permite a seleção entre duas alternativas.

A sintaxe é a seguinte:

```
if condição :
    instruções 1
else :
    instruções 2
```

A condição começa por ser avaliada: se o seu valor lógico for True, são executas apenas as instruções 1; se o valor lógico for False, são executas apenas as instruções 2.

A utilização dos dois pontos (:) no final da primeira linha faz com que as instruções fiquem indentadas, com dois espaços para a direita.

➞ IF - ELIF - ELSE

A estrutura condicional *if – elif - else* permite a seleção entre três ou mais alternativas.

A sintaxe é a seguinte:

```
if condição :
    instruções 1
elif condição 2 :
    instruções 2
else :
    instruções 3
```

Os operadores relacionais da linguagem Python são:

Operação	Tecla	Descrição
$x == y$	SHIFT • SHIFT •	Igual
$x! = y$	F4 (CHAR)	Diferente
$x > y$	F4 (CHAR)	Maior que
$x < y$	F4 (CHAR)	Menor que
$x >= y$	F4 (CHAR)	Maior ou igual que
$x <= y$	F4 (CHAR)	Menor ou igual que

Os operadores relacionais também estão disponíveis pressionando F6 (▷) F2 (OPERAT).

Nota: Não confundir "x = 1" (o valor 1 é atribuído à variável x) com "x == 1" (condição verdadeira se x for igual a 1 e falsa se x for diferente de 1).

ESTRUTURAS DE REPETIÇÃO:

Em programação, uma sequência de instruções executada repetitivamente é um ciclo.

Um ciclo é constituído por uma instrução inicial, que controla a sua execução, e por um conjunto de instruções designado por corpo do ciclo.

A instrução for *i in range* permite repetir um bloco de instruções, um número predeterminado de vezes.

A sintaxe é a seguinte:

```
for i in range(a,b) :
    corpo do ciclo
```

A utilização dos dois pontos (:) no final da primeira linha faz com que as instruções do corpo do ciclo for fiquem indentadas (com um avanço à esquerda).

Numa instrução for i in range(), a variável i percorre um conjunto de valores de acordo com a especificação range:

- range(n) designa a sequência dos números inteiros 0, 1, …, n – 1;
- range(a,b) designa a sequência de a até b – 1, ou seja, os números inteiros i tais que a $\leq$ i < b;
- range(a,b,c) corresponde ao mesmo intervalo, mas com passo igual a c ("de c em c").

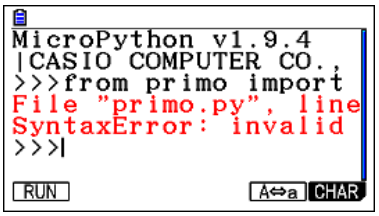
MENSAGEM DE ERRO:

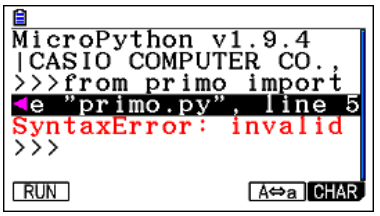
Se um programa não for executado da forma esperada, a causa pode ser um erro (bug) no script.

Neste caso, é exibida uma mensagem de erro no Shell, em vermelho, com indicação da linha onde está o erro e do tipo de erro (por exemplo, um erro de sintaxe).

Use o cursor (◀ ▶) para consultar os detalhes da mensagem de erro.

Pressione EXIT regressar ao Editor e corrigir o erro.





A CALCULADORA GRÁFICA CASIO FX-CG50

- https://www.dge.mec.pt/sites/default/files/Curriculo/Aprendizagens_Essenciais/mat_a_10_-_vf.pdf (página 16 e 17 - Maioria Absoluta)
- https://www.online-python.com/ (compilador online de linguagem python)
- https://colab.research.google.com/ (compilador online de linguagem python)
    - https://youtu.be/TjfBFqaW6-8 (vídeo explicativo)

Tarefa

1. Considere o seguinte exemplo de programa em *Python* para calcular a capitalização anual (juros simples), dado um capital inicial:

Programa em *Python*

```
ci=1000      #valor inicial
r=0.03       #taxa de juro anual
cf=ci+ci*r
print('O capital final é ',cf,'€')
```

1.1. Copie e execute o programa.

1.2. Altere os valores para calcular o juro com um capital de 550 euros e uma taxa de 2%.

1.3. Modifique o programa anterior, para calcular o capital acumulado final ao fim de n anos.

2. Considere o seguinte exemplo de programa em *Python* para calcular o capital acumulado final ao fim de n anos (juros simples):

Programa em *Python*

```
ci=1000      #valor inicial
r=0.05       #taxa de juro anual
t=20         #duração em anos
for ano in range (t+1):
   cf=ci+ci*r*ano
   print ("Ano ", ano," AC_JSim ", cf)
```

2.1. Copie o programa e execute-o.

2.2. Altere o programa anterior, para determinar o valor do capital acumulado, comparando o investimento com juro simples e juro composto, num prazo de 10 anos, à taxa de juro de 3%, com um investimento de capital inicial de 2500€.