

1.1 Executar um primeiro programa *Python*.

- (a) Utilizando o Pyzo (ou um outro editor de texto à sua escolha) escreva um pequeno programa em *Python*:

```
ola.py  
print("Ola, Python!")
```

Grave este programa num ficheiro com o nome `ola.py` no seu directório casa.

- (b) Vamos agora executar o programa que escreveu; execute o interpretador de comandos UNIX (*Terminal*) e introduza a linha de comando:

```
python3 ola.py
```

Se tudo correr bem, deve ver escrita a mensagem `Ola, mundo!`. Parabéns, executou o seu primeiro programa em *Python*!

- (c) Pode também executar o programa diretamente no ambiente de desenvolvimento (IDLE/Pyzo); peça ajuda ao docente para saber como o fazer.

1.2 Vamos agora obter alguma documentação sobre programação em *Python*. Na página da disciplina tem disponível uma ligação para o livro *Think Python* em formato PDF; este livro é de distribuição livre e gratuita. Experimente gravá-lo na sua área com o nome `thinkpython.pdf`. Pode ainda copiá-lo para estudar em casa, imprimir, ou distribuir aos seus amigos (leia a licença nas primeiras páginas).

Usando o gestor de ficheiros gráfico, abra uma janela com o seu directório casa e localize o ficheiro PDF. Clicando duas vezes deverá visualizar o documento.

1.3 Use o interpretador *Python* no ambiente de programação (recorrendo utilizar um IDE, por exemplo, o *Pyzo*) para calcular as seguintes expressões:

- | | |
|---------------------------|--------------------------|
| (a) <code>45 + 27</code> | (e) <code>2.5 * 4</code> |
| (b) <code>2**3</code> | (f) <code>10/3</code> |
| (c) <code>9 % 4</code> | (g) <code>10//3</code> |
| (d) <code>"2"+"34"</code> | (h) <code>10%3</code> |

1.4 Para cada uma das alíneas do exercício anterior, indique qual o tipo do resultado; pode confirmar as suas resposta utilizando a função `type()`:

```
>>> type("abc")  
<class 'str'>
```

1.5 Usando o interpretador de *Python* calcule cada uma das seguintes expressões. Indique se o resultado é um número inteiro ou em vírgula-flutuante.

- | | |
|--------------------|-------------------|
| (a) $(10-7)*(4-3)$ | (g) $(3+5)/(2*3)$ |
| (b) $10-7*4-3$ | (h) $1 + 1/3$ |
| (c) $17 / 3$ | (i) $1 + 1/3.0$ |
| (d) $17 // 3$ | (j) $1 + 1//3$ |
| (e) $17 \% 3$ | (k) $2**3$ |
| (f) $(3+5)/2*3$ | (l) $2.0**3$ |

1.6 Simule a execução passo-a-passo dos seguintes programas e indique os resultados finais das variáveis; use o *Python Tutor* (<http://pythontutor.com>) para conferir os resultados.

- | | | |
|---|--|--|
| (a) <code>a = 121</code>
<code>b = 45</code>
<code>t = a</code>
<code>a = b</code>
<code>b = t</code> | (c) <code>N = 1</code>
<code>N = N*10 + 2</code>
<code>N = N*10 + 3</code>
<code>N = N*10 + 4</code> | (e) <code>a = 54</code>
<code>b = 24</code>
<code>r = a%b</code>
<code>a = b</code>
<code>b = r</code>
<code>r = a%b</code>
<code>a = b</code>
<code>b = r</code> |
| (b) <code>p = 1</code>
<code>p = p * 2</code>
<code>p = p * 3</code>
<code>p = p * 4</code> | (d) <code>x = 2.0</code>
<code>r = x</code>
<code>r = 0.5*(r + x/r)</code>
<code>r = 0.5*(r + x/r)</code>
<code>r = 0.5*(r + x/r)</code> | |

1.7 Simule a execução dos seguintes programas indicando os valores das variáveis após cada passo.

- | | | |
|---|--|--|
| (a) <code>s = 0</code>
<code>s = s**2 + 1</code>
<code>s = s**2 + 2</code>
<code>s = s**2 + 3</code> | (c) <code>n = 1</code>
<code>s = 0</code>
<code>s = s + n</code>
<code>n = n + 1</code>
<code>s = s + n</code>
<code>n = n + 1</code>
<code>s = s + n</code>
<code>n = n + 1</code> | (d) <code>x = 3</code>
<code>y = 1</code>
<code>y = x*y + 1</code>
<code>y = x*y + 1</code>
<code>y = x*y + 1</code> |
| (b) <code>s = 0</code>
<code>s = (s + 1)**2</code>
<code>s = (s + 2)**2</code>
<code>s = (s + 3)**2</code> | | (e) <code>x = 3</code>
<code>y = x**3+x**2+x+1</code> |

1.8 Escreva os programas que se seguem no seu editor preferido:

- | | |
|---|---|
| (a) <code>x = input()</code>
<code>print(x)</code> | (b) <code>x = input("Please write something: ")</code>
<code>print(x)</code> |
|---|---|

Execute cada um deles no terminal com (por exemplo):

```
$ python3 test.py
[nesta linha escreva o seu nome seguido de ENTER]
```

Em alternativa, poderá recorrer interpretador de Python associado ao IDE da sua preferência.

Descreva o que observou.

1.9 Utilize a função `input` para escrever um programa que pergunta "What's your name? " e imprime a mensagem "Hello, [NOME]", onde [NOME] é a sequência de caracteres introduzida quando se executa o programa.

1.10 Traduza cada uma das seguintes expressões matemáticas para *Python* e execute-as no interpretador. Pode usar variáveis auxiliares para guardar valores intermédios.

- | | |
|---|--|
| (a) $(1+x)(-1+2x)$ para $x = 2$ | (e) $1 + 1/(2 + 1/x)$ para $x = 2$ |
| (b) $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4}$ | (f) $\sqrt{x^2 + y^2}$ para $x = 2$ e $y = 0.5$ |
| (c) 2^{2^3} | (g) $\sqrt{b^2 - 4ac}$ para $a = 1$, $b = 1$ e $c = -1/3$. |
| (d) $(1 - x^2)/(2x)$ quando $x = 5$ | (h) $\sin(\pi - \alpha)$ para $\alpha = \pi/4$ |

As funções e constantes matemáticas estão definidas no módulo `math`: `math.sqrt()`, `math.exp()`, `math.sin()`, `math.cos()`, `math.pi`, etc.

1.11 Execute as expressões seguintes no interpretador de *Python*. Se der um erro, indique se é erro sintático ou semântico. Se não der erro, verifique qual o seu tipo usando `type(...)`.

- | | |
|---------------------------------|--|
| (a) <code>97+555</code> | (i) <code>"97"+555</code> |
| (b) <code>97\$+555\$</code> | (j) <code>97 == "97"</code> |
| (c) <code>math.sqrt(2)</code> | (k) <code>97 == int("97")</code> |
| (d) <code>math.sqrt(-2)</code> | (l) <code>102 <= 97</code> |
| (e) <code>2(math.pi)</code> | (m) <code>102 <= "97"</code> |
| (f) <code>2*math.pi</code> | (n) <code>"102" <= "97"</code> |
| (g) <code>str(2*math.pi)</code> | (o) <code>"valor de pi é " + math.pi</code> |
| (h) <code>int(2*math.pi)</code> | (p) <code>"valor de pi é " + str(math.pi)</code> |

1.12 Escreva um programa que pergunte ao utilizador os quilómetros percorridos e o número de litros de combustível que um automóvel gastou, e imprima o consumo em litros gastos aos 100 quilómetros.

1.13 Implemente uma função `area_circ(r)` que calcule a área de um círculo com raio `r`.

1.14 A conversão entre medidas de temperatura em graus Fahrenheit e Celsius pode ser efectuada pela fórmula $C = \frac{5}{9}(F - 32)$, onde F é a temperatura em graus Fahrenheit e C em celsius. Implemente uma função `celsius(F)` que efetue a conversão de uma temperatura de graus Fahrenheit para Celsius.

1.15 Escreva uma função `hipotenusa(a,b)` que, dadas as dimensões a e b dos dois lados de um triângulo retângulo, calcula o comprimento da hipotenusa.