Pensamento Computacional

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Objetivos

- Conceitos Básicos de Programação:
  - Funções.
  - Strings e funções matemáticas.
  - Expressões lógicas e instruções condicionais.

Funções

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

- **Definição:**
  - Sequência nomeada de declarações que executam uma tarefa ou procedimento;
  - Aceita zero ou mais argumentos;
  - Retorna um resultado e/ou produz algum efeito (tal como apresentar ou guardar o resultado de um cálculo).

- É possível definir uma função (próximas aulas).

Funções

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

- Chamar (invocar) uma função pelo nome:

```
>>> rect_area(2, 4, "m")
8 m
```

  - Nome da função: `rect_area`
  - argumentos (3): `(2, 4, "m")`
  - resultado: `8 m`

Strings – relembrar!

- *Strings* são sequências de carateres.
- Literais string são delimitados por aspas simples ou duplas..

```
fruit = 'orange'
```

- Função que retorna o número de carateres:

```
len(fruit)                    #-> 6
```

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Strings – relembrar!

- Podemos aceder a um conjunto de carateres através da sua posição (**a primeira posição tem indíce 0**)

```
fruit[1]                    #-> 'r'
fruit[3:5]                  #-> 'ng'
```

- Podemos concatenar e repetir *strings*

```
name = 'tom' + 'cat'
#-> 'tomcat'
gps = 2 * 'tom'
#-> 'tomtom'
```

Strings

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

- As *strings* são **imutáveis**. Os métodos que implicam alteração apenas retornam um novo objeto *string*.

```
fruit.upper()             #-> 'ORANGE'
fruit.replace('a', 'A') #-> 'orAnge'
fruit                     #-> 'orange' (not changed)
```

- A classe `str` tem vários métodos já definidos: para ver todos, usar `help(str)`

String methods

U.PORTO
FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

- As *strings* têm muitos métodos úteis:

Play ▶

`str.isalpha()`
`str.isdigit()`
`str.is...`
	True se todos os carateres são alfabéticos.
	True se todos os carateres são digitos.
	...

`str.upper()`
`str.lower()`
`...`
	Converte para maiúsculas.
	Converte para minúsculas.
	...

`str.strip()`
`str.lstrip()`
`str.rstrip()`
	Remove o espaço branco à esquerda e à direita.
	Remove o espaço branco à esquerda.
	Remove o espaço branco à direita.

`s1.find(s2)`	Encontra a string s2 na string s1
`str.split()` `str.split(sep)`	Divide str pelos carateres espaço branco. Divide a str usando sep como delimitador.

Carateres Especiais

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

- A barra invertida (\) é usada para introduzir um carater especial:

```
# Introduces new line (\n)
>>> print ("Liliana Ferreira,\nFEUP")
Liliana Ferreira,
FEUP

# Introduces tab (\t)
>>> print("Liliana Ferreira,\tlsferreira@fe.up.pt");
Liliana Ferreira,   lsferreira@fe.up.pt
```

Escape sequence	Meaning
\\	Backslash
\'	Single quote
\"	Double quote
\n	Newline
\t	Tab

Verificar o conteúdo de uma *string*

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

- **in** keyword
  - Verifica se uma dada frase ou carater está presente na string.
  - Devolve True ou False
  - Pode ser usado numa declaração *if* (próximas aulas)

```
>>> print ('city' in S)
True
# Check if NOT
>>> print ('city' not in S)
False
```

Método str.format()

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

- Permite substituições de variáveis e formatação de valores:

```
<template>.format(<positional_argument(s)>, <keyword_argument(s)>)
```

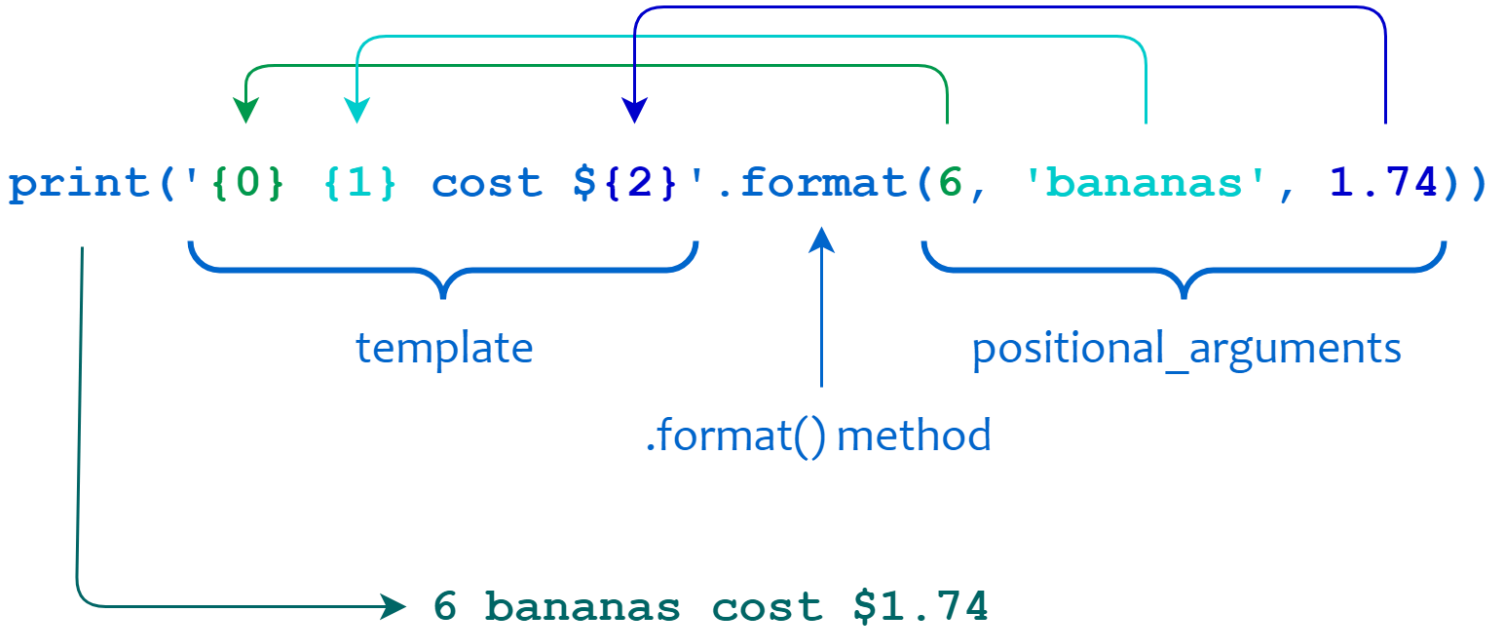
  - As strings de formatação contêm “campos de substituição” delimitados por {}.
  - Tudo o que não estiver dentro das chavetas é considerado texto literal.

Método str.format()

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

- Exemplo

```
>>> print('{0} {1} cost ${2}'.format(6, 'bananas', 1.74))
6 bananas cost $1.74
```



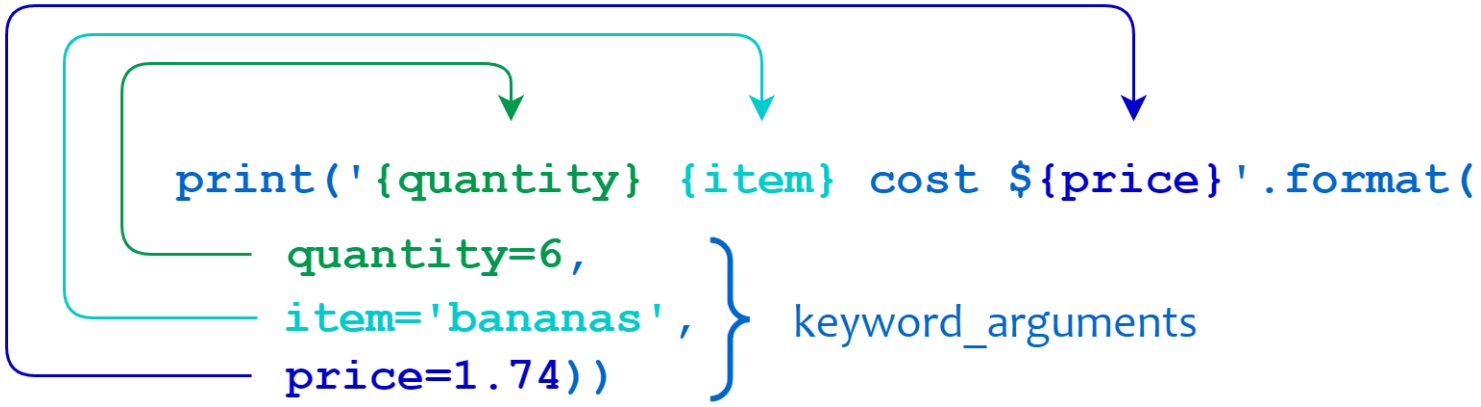
Método str.format() (cont.)

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

- Usando keywords em vez de parâmetros de posicionamento

- Exemplo

```
>>> print('{quantity} {item} cost ${price}'.format(quantity=6,
item='bananas', price=1.74))
6 bananas cost $1.74
```



Formatação de *strings*: str.format()

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

```
  '{} {}'.format('one', 'two’)          #-> 'one two'
  '{:*>10}'.format('test')              #-> '******test'
  '{:04d}'.format(42)                   #-> '0042'
  '{:5.2f}'.format(3.2451)              #-> ' 3.25'

print(1,2,3)                            #-> 1 2 3
print(10,20,30)                         #-> 10 20 30

print('{:4d}{:4d}{:4d}'.format(1,2,3))        #->    1   2   3
print('{:4d}{:4d}{:4d}'.format(10,20,30))     #->   10  20  30
```

f-String

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

- Nova e melhorada forma de formatar *strings*.

- Sintaxe é semelhante à de str.format() mas menos verbosa

  f”...”, F”...”, f’...’, F’...’

- Exemplo

```
>>> name = “Maria"
>>> age = 39
>>> f"Hello, {name}. You are {age}."
'Hello, Maria. You are 39.'
```

String Formatting: f-strings

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

```
f"{'one'} {'two'}"                  #-> 'one two'
f"{'test':*>10}"                    #-> '******test'
f'{42:04d}'                         #-> '0042'
f'{3.2451:5.2f}'                    #-> ' 3.25'


print(1,2,3)                        #-> 1 2 3
print(10,20,30)                     #-> 10 20 30

print(f'{1:4d}{2:4d}{3:4d}')        #->    1   2   3
print(f'{10:4d}{20:4d}{30:4d}')     #->   10  20  30
```

Funções Matemáticas

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

- *Python* tem um módulo **math** que implementa as funções matemáticas mais conhecidas.
- Existe também um modulo **cmath** equivalente para trabalhar com números complexos.

- Um modulo – *module* – é um ficheiro Python que definie a coleção de funções e objetos relacionados.

- Antes de usar um dado módulo, devemos importá-lo:

```
>>> import math
>>> import cmath
```

Algumas funções matemáticas

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

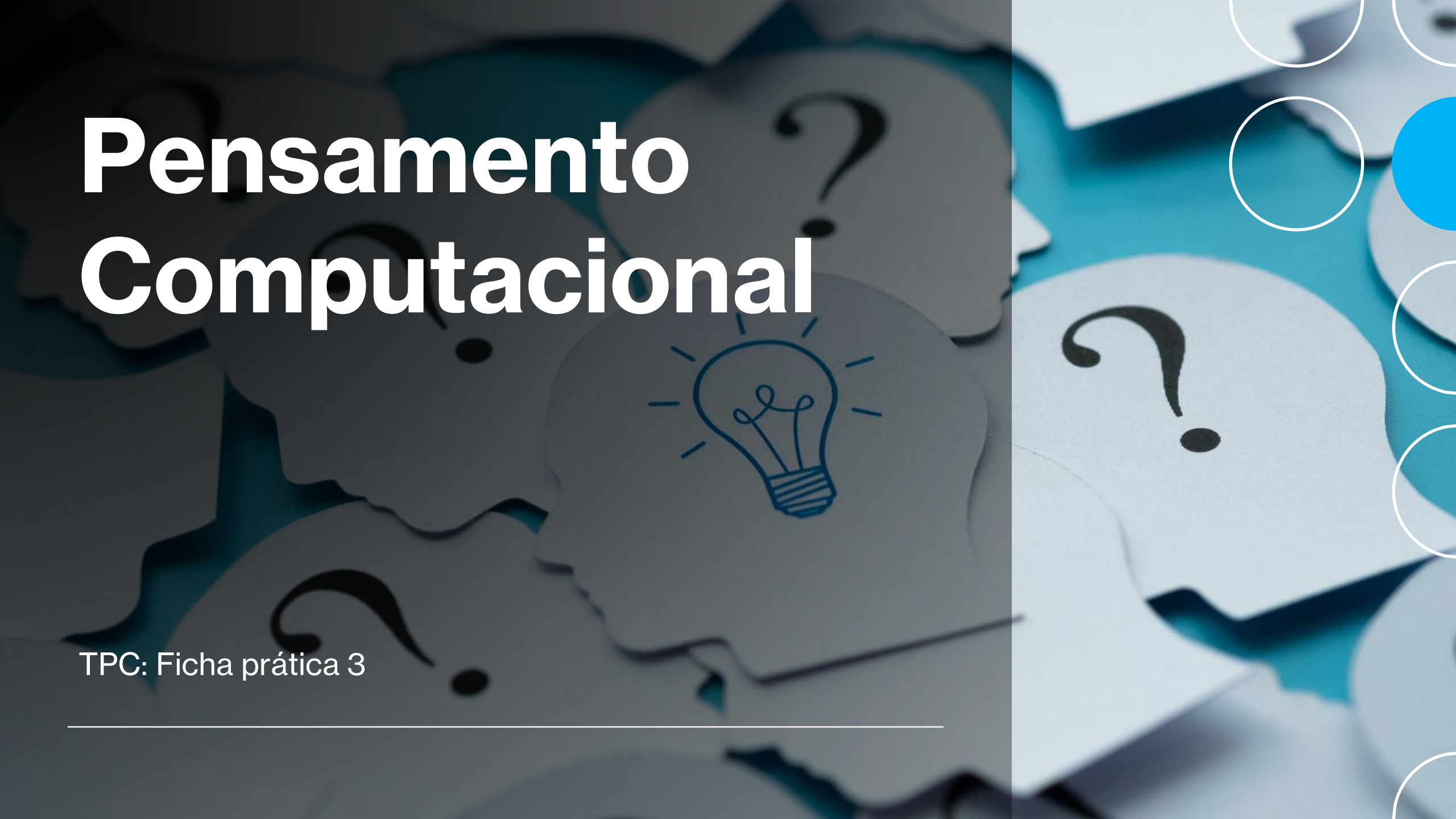
math.	**Description**
pi	An approximation of pi.
e	An approximation of e.
sqrt(x)	The square root of x.
sin(x)	The sine of x.
cos(x)	The cosine of x.
tan(x)	The tangent of x.
asin(x)	The inverse of sine x.
acos(x)	The inverse of cosine x.
atan(x)	The inverse of tangent x.
log(x)	The natural (base e) logarithm of x.
log10(x)	The common (base 10) logarithm of x.
exp(x)	The exponential of x.
ceil(x)	The smallest whole number>= x.
floor(x)	The largest whole number <= x.
degrees(x)	Convert angle x from radians to degrees.
radians(x)	Convert angle x from degrees to radians.
help(math)	**Shows all math functions in the console**

Funções matemáticas - exemplos

```
>>> degrees = 45

>>> radians = degrees * math.pi / 180

# Angle arguments are always in radians!
>>> math.sin(radians)
0.707106781187

>>> cmath.sqrt(-1)
1j
```

Para usar as funções, especificar o nome do módulo e o nome da função, separados com um ponto.

Pensamento Computacional



TPC: Ficha prática 3