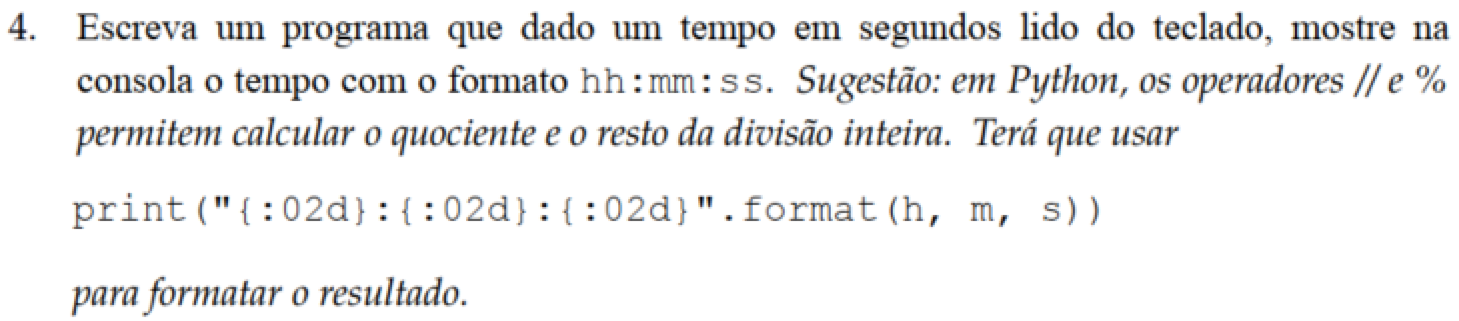
Week 02 – Support notes

Aula prática nº 2 – Introdução à linguagem Python



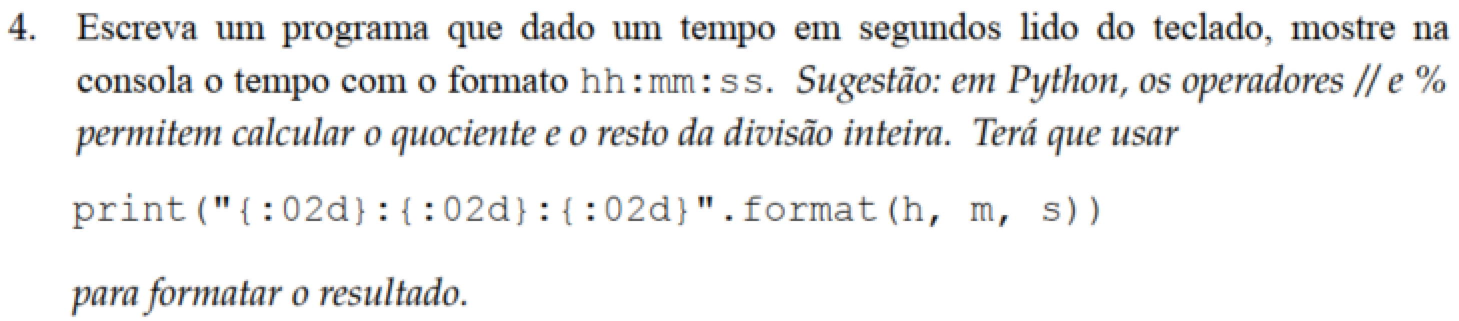
4. Escreva um programa que dado um tempo em segundos lido do teclado, mostre na consola o tempo com o formato `hh:mm:ss`. *Sugestão: em Python, os operadores // e % permitem calcular o quociente e o resto da divisão inteira. Terá que usar*

```
print("{:02d}:{:02d}:{:02d}".format(h, m, s))
```

para formatar o resultado.

Sugestão: similar a encontrar termo geral de uma sucessão
Começa pelos casos mais simples (e.g. 1, 61s) até chegares à regra
Ou seja o programa/algoritmo para resolver

Experimenta os casos para 1s, 61s, 3660 s, 3661 s , 3666 s e vê que resultado esperas e como chegas a eles



4. Escreva um programa que dado um tempo em segundos lido do teclado, mostre na consola o tempo com o formato `hh:mm:ss`. *Sugestão: em Python, os operadores // e % permitem calcular o quociente e o resto da divisão inteira. Terá que usar*

```
print("{:02d}:{:02d}:{:02d}".format(h, m, s))
```

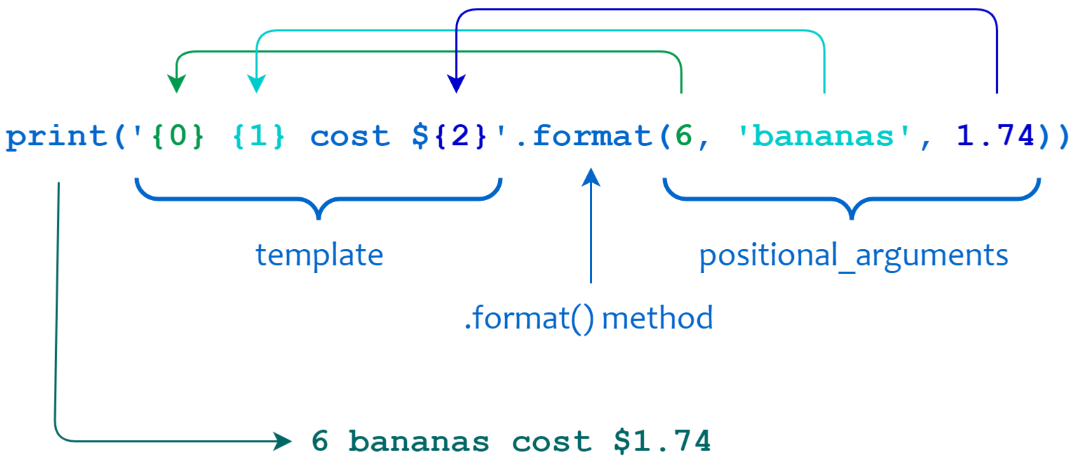
para formatar o resultado.

61 s = = 60 s + 1 s =
= 1 min + 1 s → 01:01:0

3600s= 3600 s =
= 1 hora → 01:00:00

3661s = 3600 s + 60 s + 1 s =
= 1 hora + 1 min + 1 s → 01:01:01

3666s = 3600 s + 60 s + 6 s =
= 1 hora + 1 min + 6 s → 01:01:06



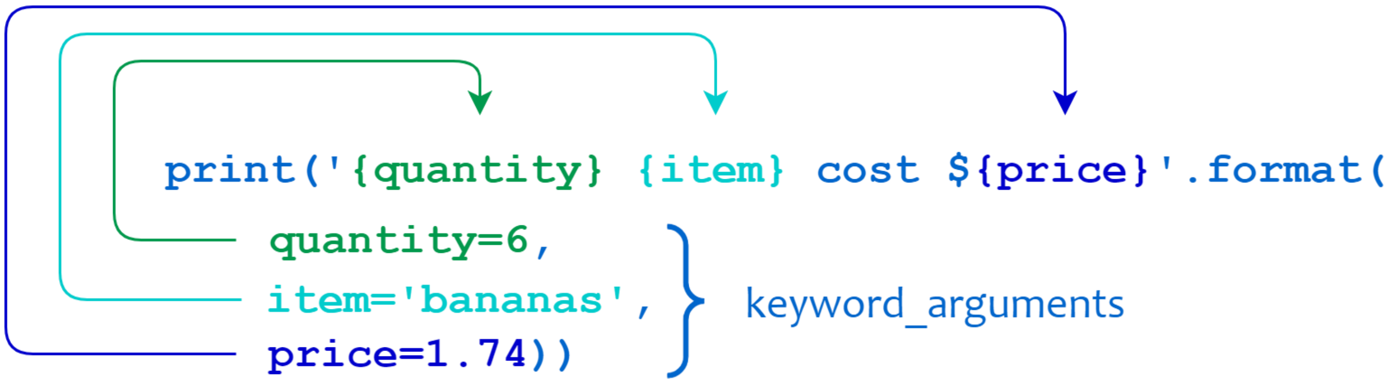
```
print('{0} {1} cost ${2}'.format(6, 'bananas', 1.74))
```

template

.format() method

positional_arguments

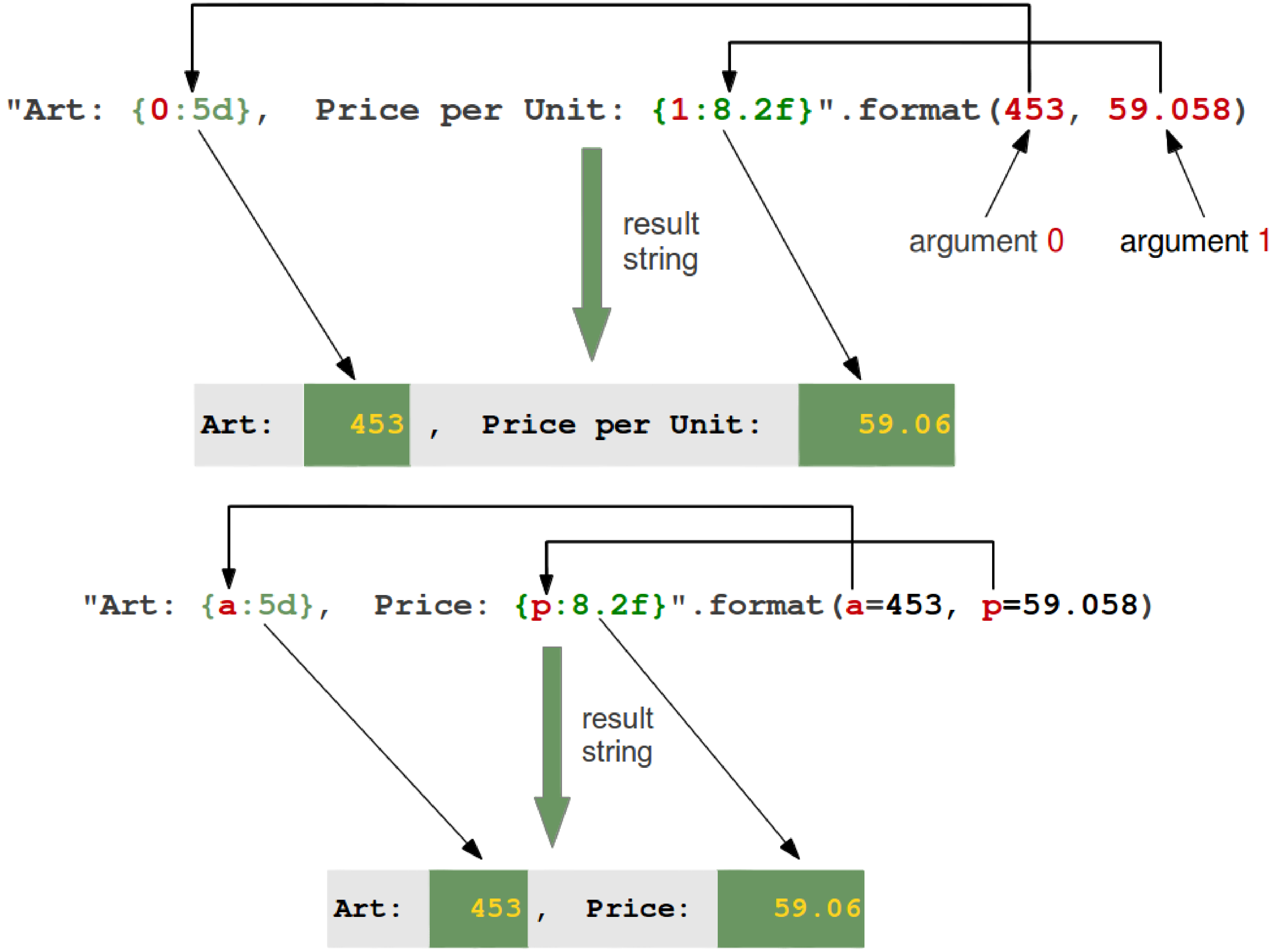
```
6 bananas cost $1.74
```



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

keyword_arguments

https://realpython.com/python-formatted-output/



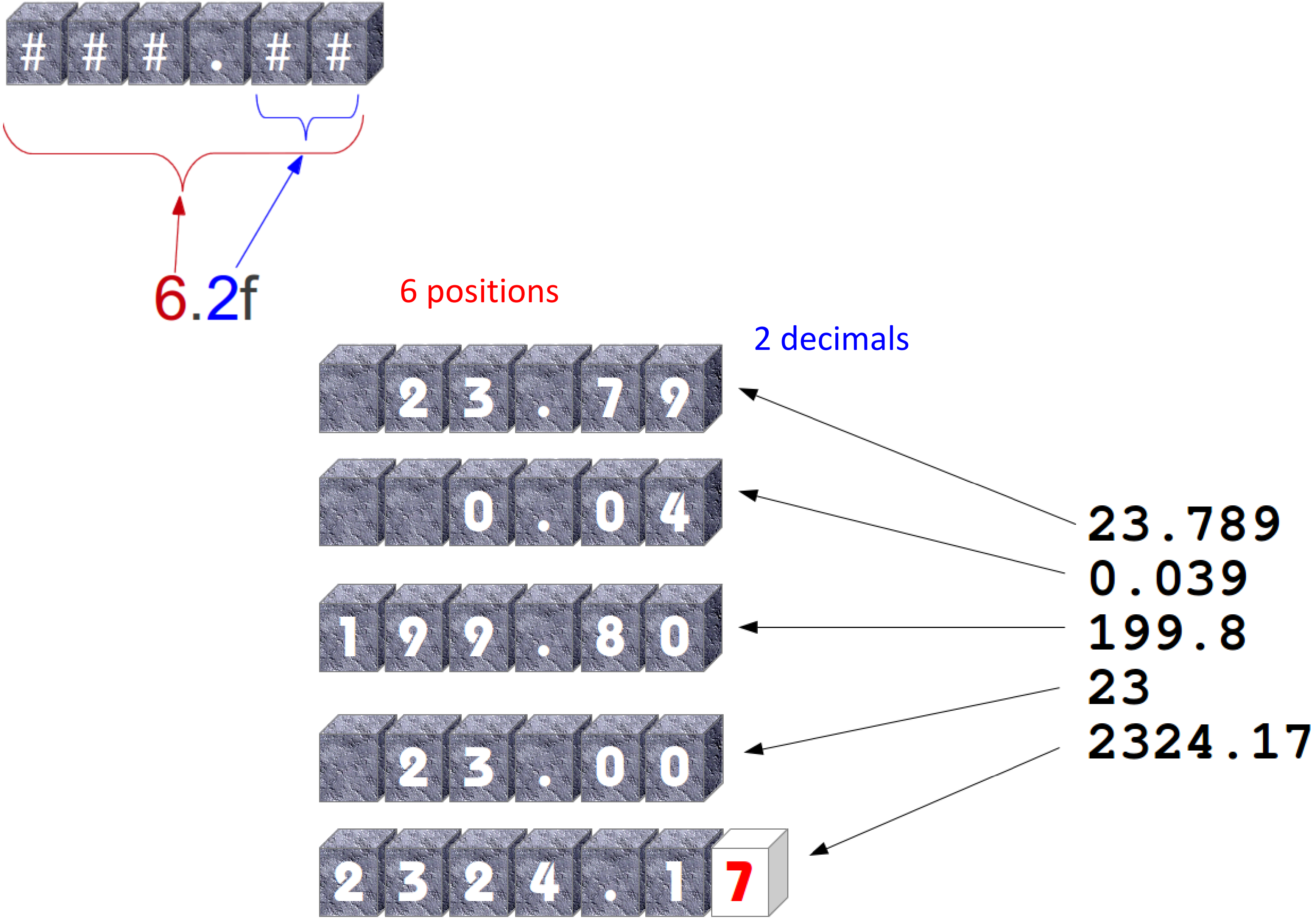
```
"Art: {0:5d}, Price per Unit: {1:8.2f}".format(453, 59.058)
```

```
Art:   453, Price per Unit:    59.06
```

```
"Art: {a:5d}, Price: {p:8.2f}".format(a=453, p=59.058)
```

```
Art:   453, Price:    59.06
```

https://www.python-course.eu/python3_formatted_output.php



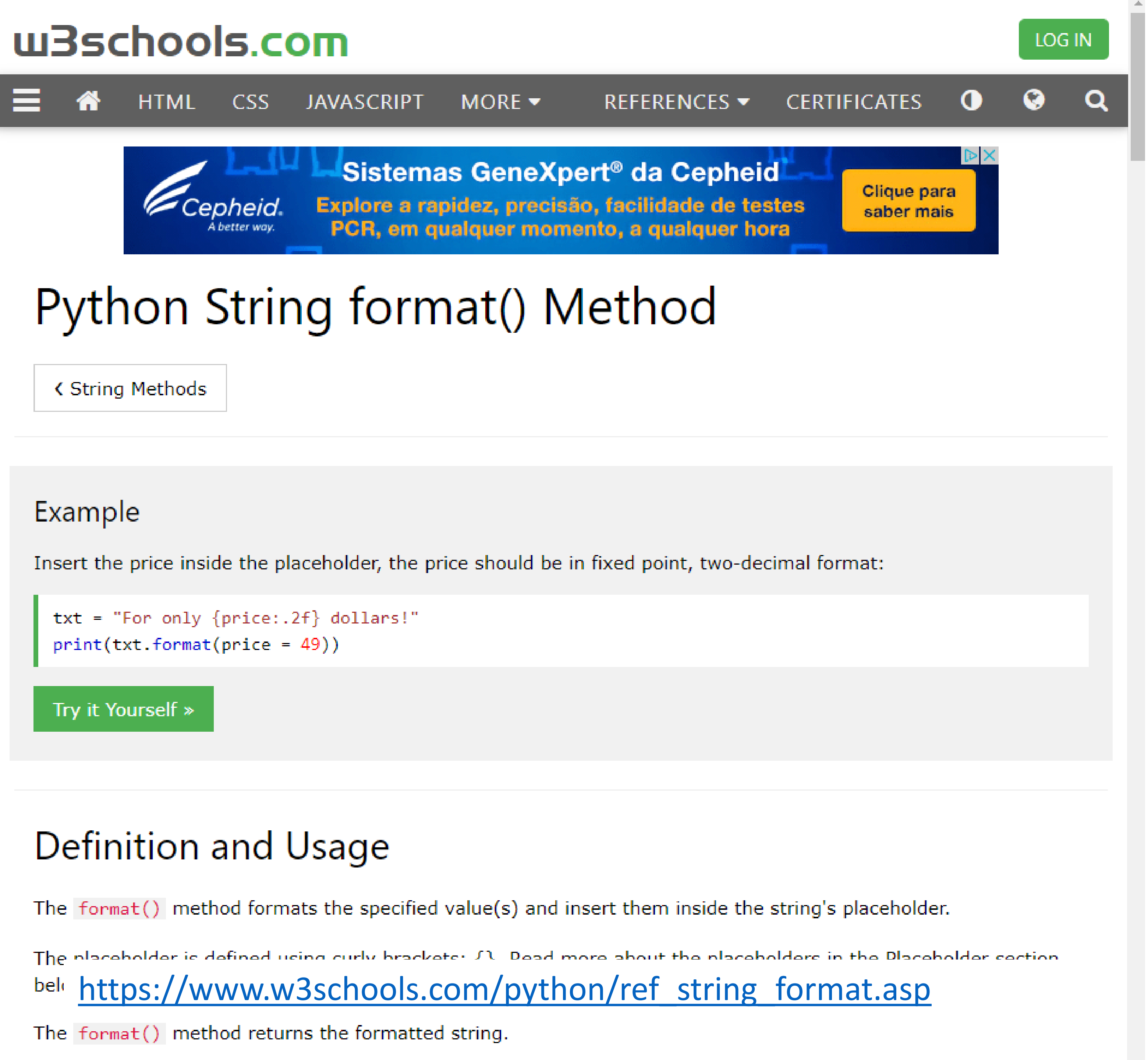
6.2f

6 positions

2 decimals

```
23.789
0.039
199.8
23
2324.17
```

https://www.python-course.eu/python3_formatted_output.php



Python String format() Method

‹ String Methods

Example

Insert the price inside the placeholder, the price should be in fixed point, two-decimal format:

```
txt = "For only {price:.2f} dollars!"
print(txt.format(price = 49))
```

Try it Yourself »

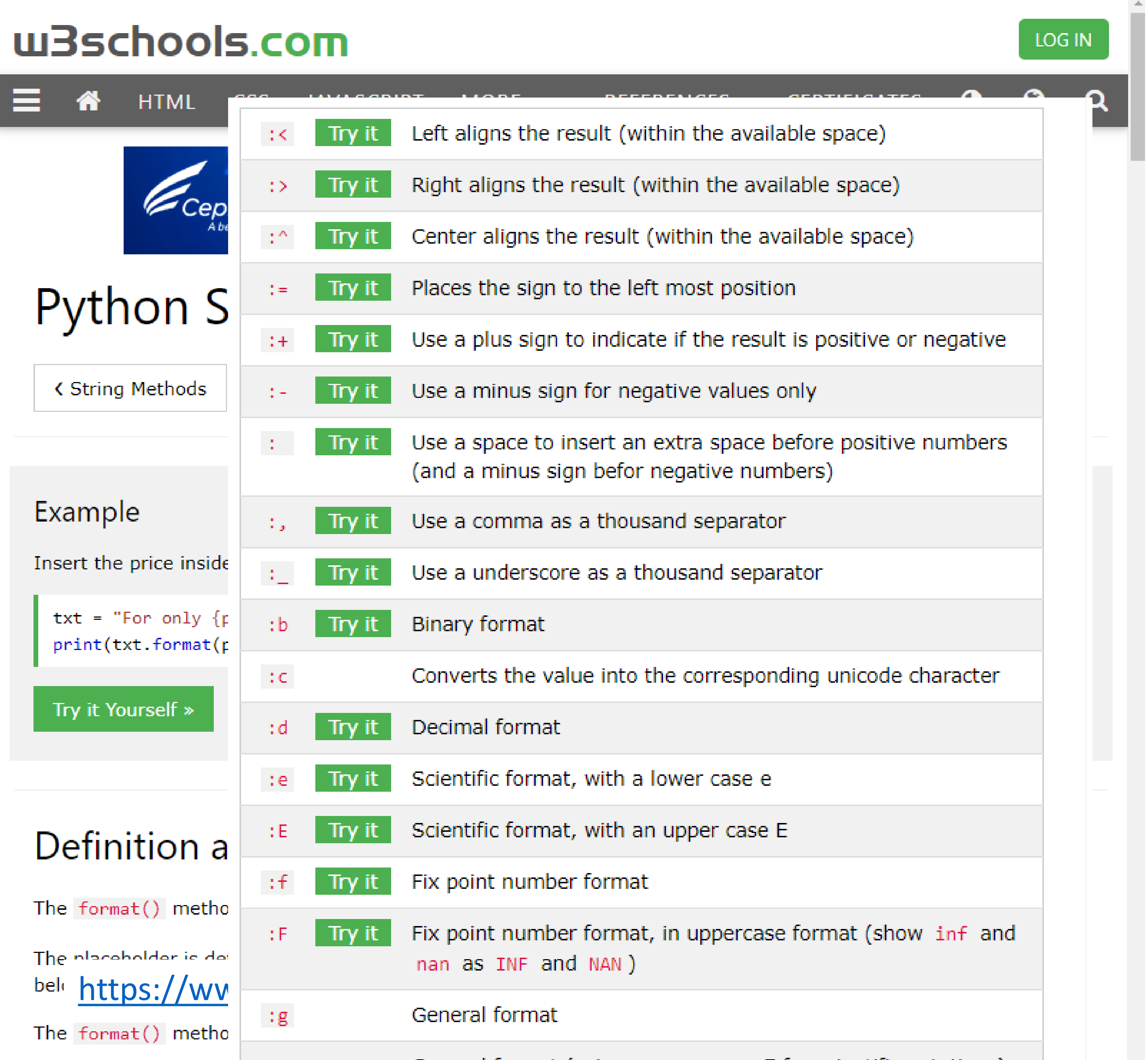
Definition and Usage

The `format()` method formats the specified value(s) and insert them inside the string's placeholder.

The placeholder is defined using curly brackets: {}. Read more about the placeholders in the Placeholder section bel

https://www.w3schools.com/python/ref_string_format.asp

The `format()` method returns the formatted string.



:<	Try it	Left aligns the result (within the available space)
:>	Try it	Right aligns the result (within the available space)
:^	Try it	Center aligns the result (within the available space)
:=	Try it	Places the sign to the left most position
:+	Try it	Use a plus sign to indicate if the result is positive or negative
:-	Try it	Use a minus sign for negative values only
:	Try it	Use a space to insert an extra space before positive numbers (and a minus sign befor negative numbers)
:,	Try it	Use a comma as a thousand separator
:_	Try it	Use a underscore as a thousand separator
:b	Try it	Binary format
:c		Converts the value into the corresponding unicode character
:d	Try it	Decimal format
:e	Try it	Scientific format, with a lower case e
:E	Try it	Scientific format, with an upper case E
:f	Try it	Fix point number format
:F	Try it	Fix point number format, in uppercase format (show `inf` and `nan` as `INF` and `NAN`)
:g		General format

Python S

‹ String Methods

Example

Insert the price inside

```
txt = "For only {p
print(txt.format(p
```

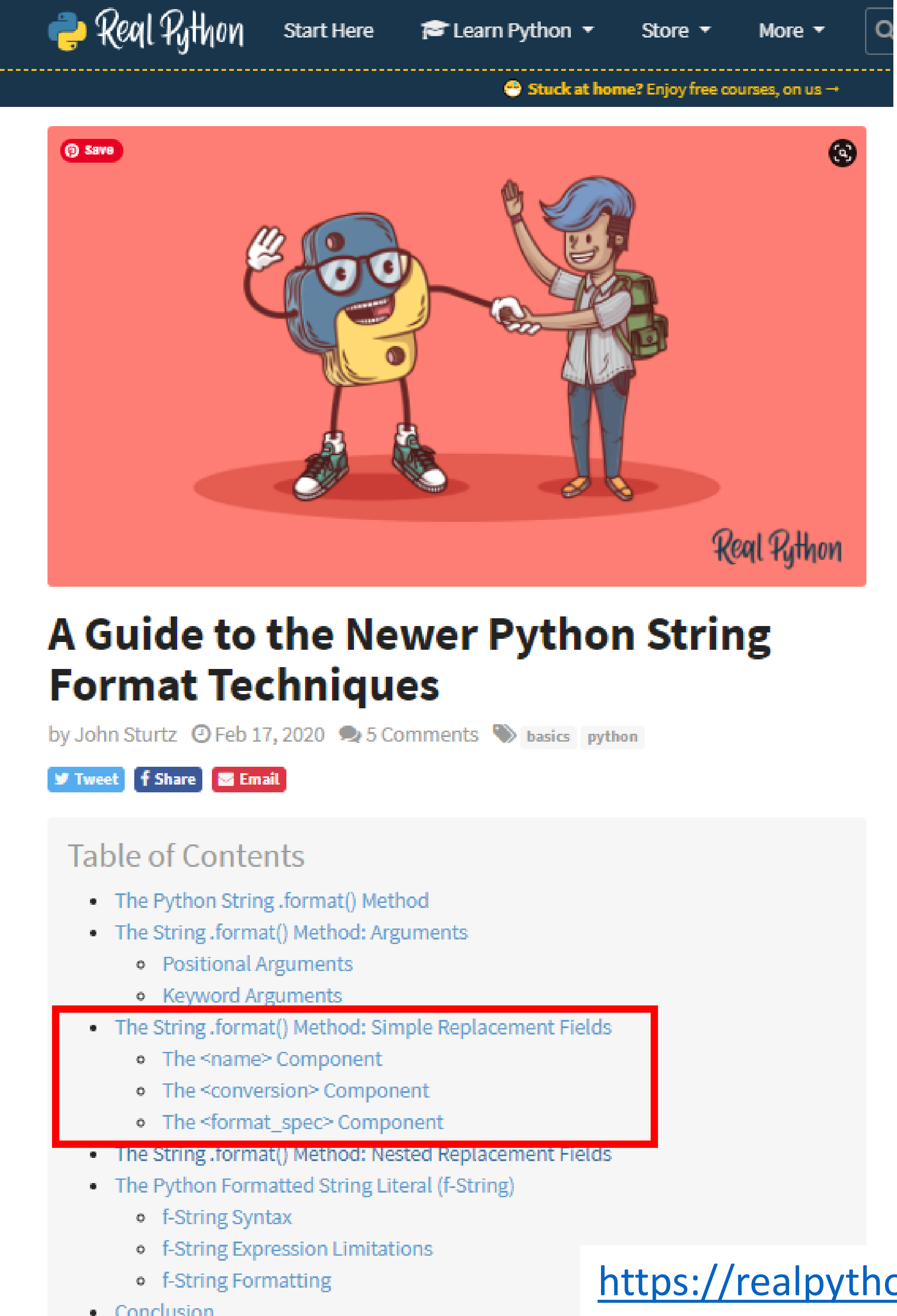
Try it Yourself »

Definition a

The `format()` metho

The placeholder is de
bel https://wv

The `format()` metho

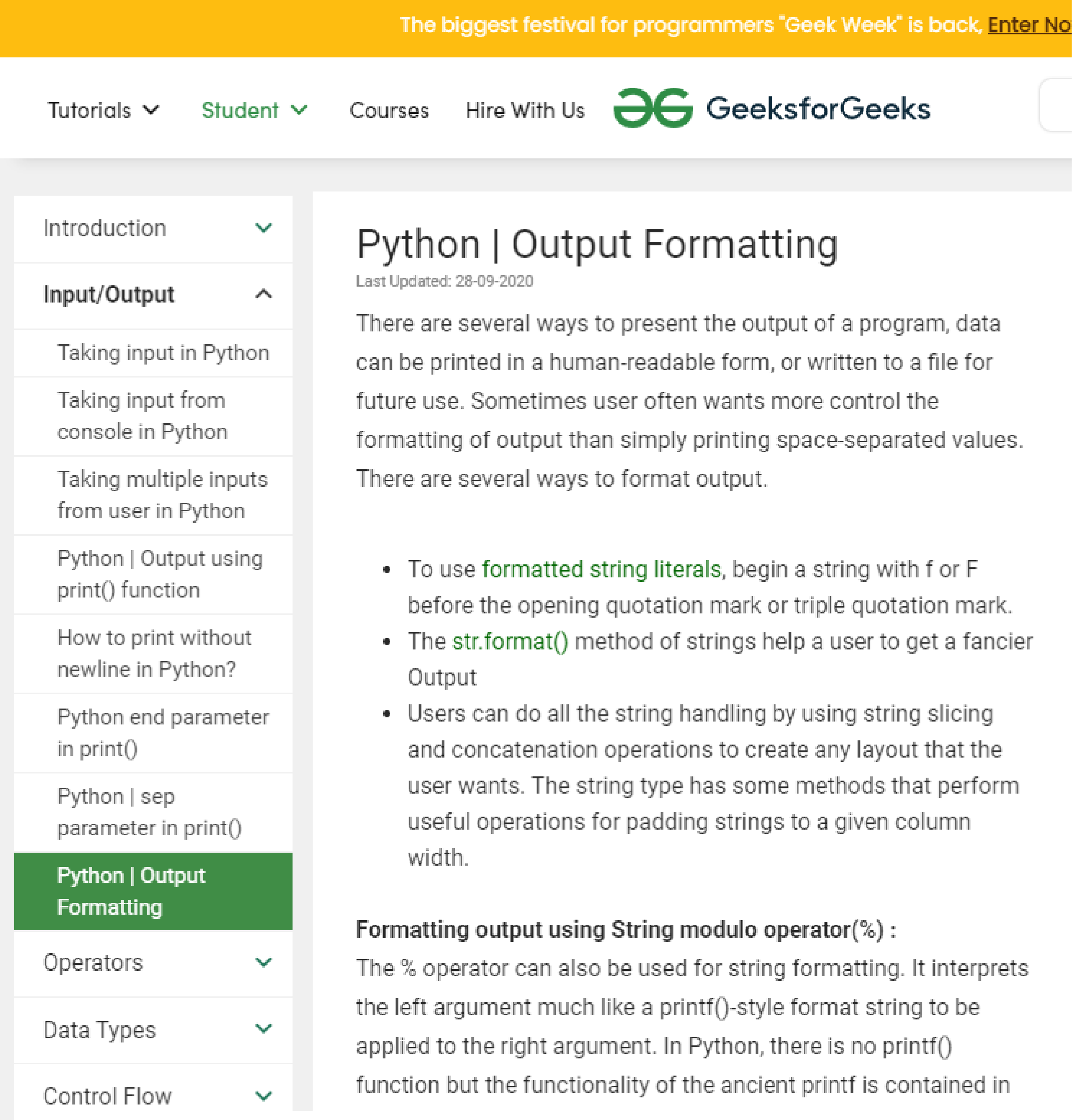


A Guide to the Newer Python String Format Techniques

by John Sturtz Feb 17, 2020 5 Comments basics python

Table of Contents

https://realpython.com/python-formatted-output/


The biggest festival for programmers "Geek Week" is back, Enter No
Tutorials
Student
Courses
Hire With Us
GeeksforGeeks
Introduction
Input/Output
Taking input in Python
Taking input from console in Python
Taking multiple inputs from user in Python
Python | Output using print() function
How to print without newline in Python?
Python end parameter in print()
Python | sep parameter in print()
Python | Output Formatting
Operators
Data Types
Control Flow

Python | Output Formatting

Last Updated: 28-09-2020

There are several ways to present the output of a program, data can be printed in a human-readable form, or written to a file for future use. Sometimes user often wants more control the formatting of output than simply printing space-separated values. There are several ways to format output.

- To use formatted string literals, begin a string with f or F before the opening quotation mark or triple quotation mark.
- The str.format() method of strings help a user to get a fancier Output
- Users can do all the string handling by using string slicing and concatenation operations to create any layout that the user wants. The string type has some methods that perform useful operations for padding strings to a given column width.

Formatting output using String modulo operator(%) :

The % operator can also be used for string formatting. It interprets the left argument much like a printf()-style format string to be applied to the right argument. In Python, there is no printf() function but the functionality of the ancient printf is contained in

https://www.geeksforgeeks.org/python-output-formatting/

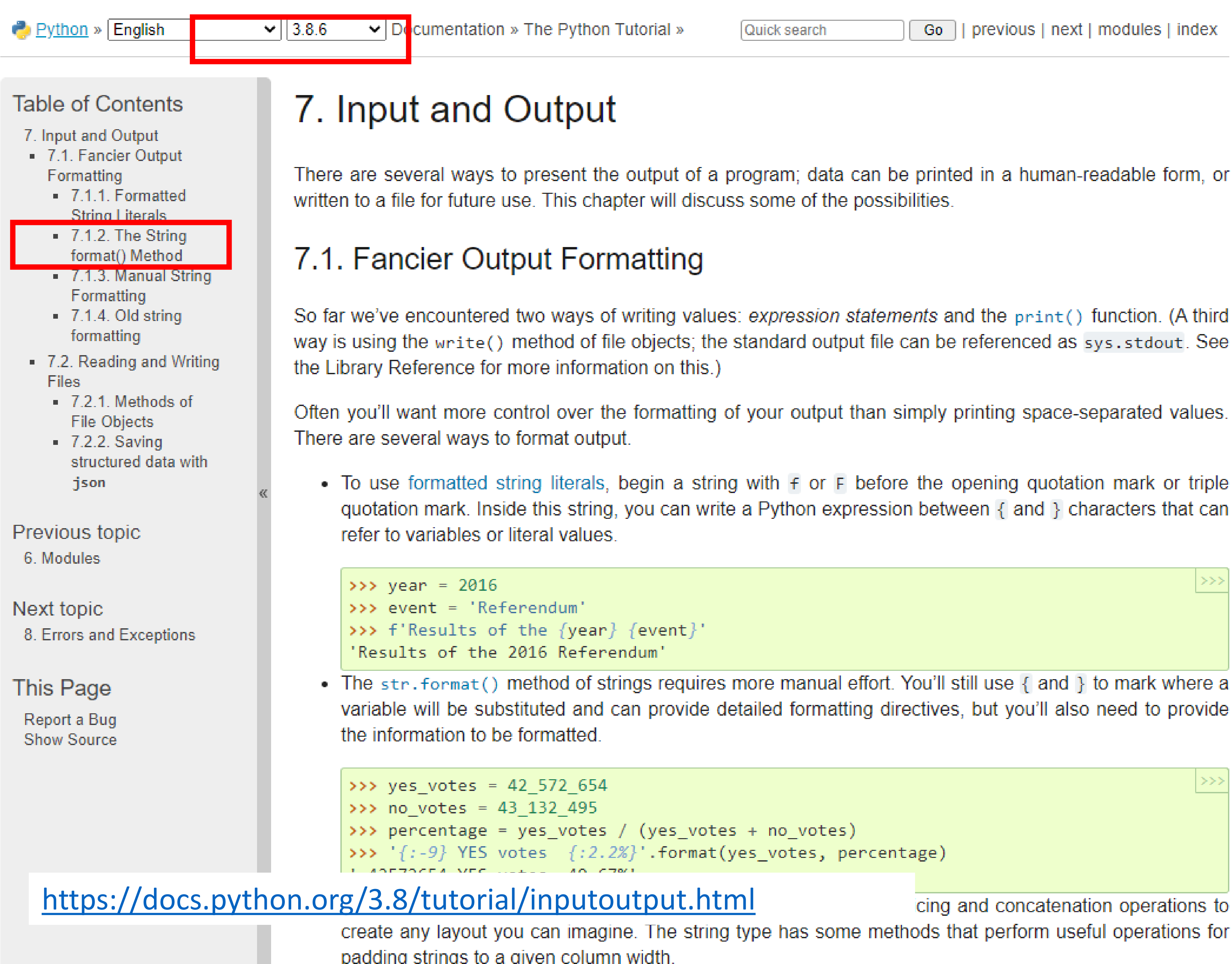

Python » English 3.8.6 Documentation » The Python Tutorial »
Quick search
Go
| previous | next | modules | index

Table of Contents

Previous topic

6. Modules

Next topic

8. Errors and Exceptions

This Page

Report a Bug
Show Source

7. Input and Output

There are several ways to present the output of a program; data can be printed in a human-readable form, or written to a file for future use. This chapter will discuss some of the possibilities.

7.1. Fancier Output Formatting

So far we've encountered two ways of writing values: *expression statements* and the `print()` function. (A third way is using the `write()` method of file objects; the standard output file can be referenced as `sys.stdout`. See the Library Reference for more information on this.)

Often you'll want more control over the formatting of your output than simply printing space-separated values. There are several ways to format output.

- To use formatted string literals, begin a string with `f` or `F` before the opening quotation mark or triple quotation mark. Inside this string, you can write a Python expression between `{` and `}` characters that can refer to variables or literal values.

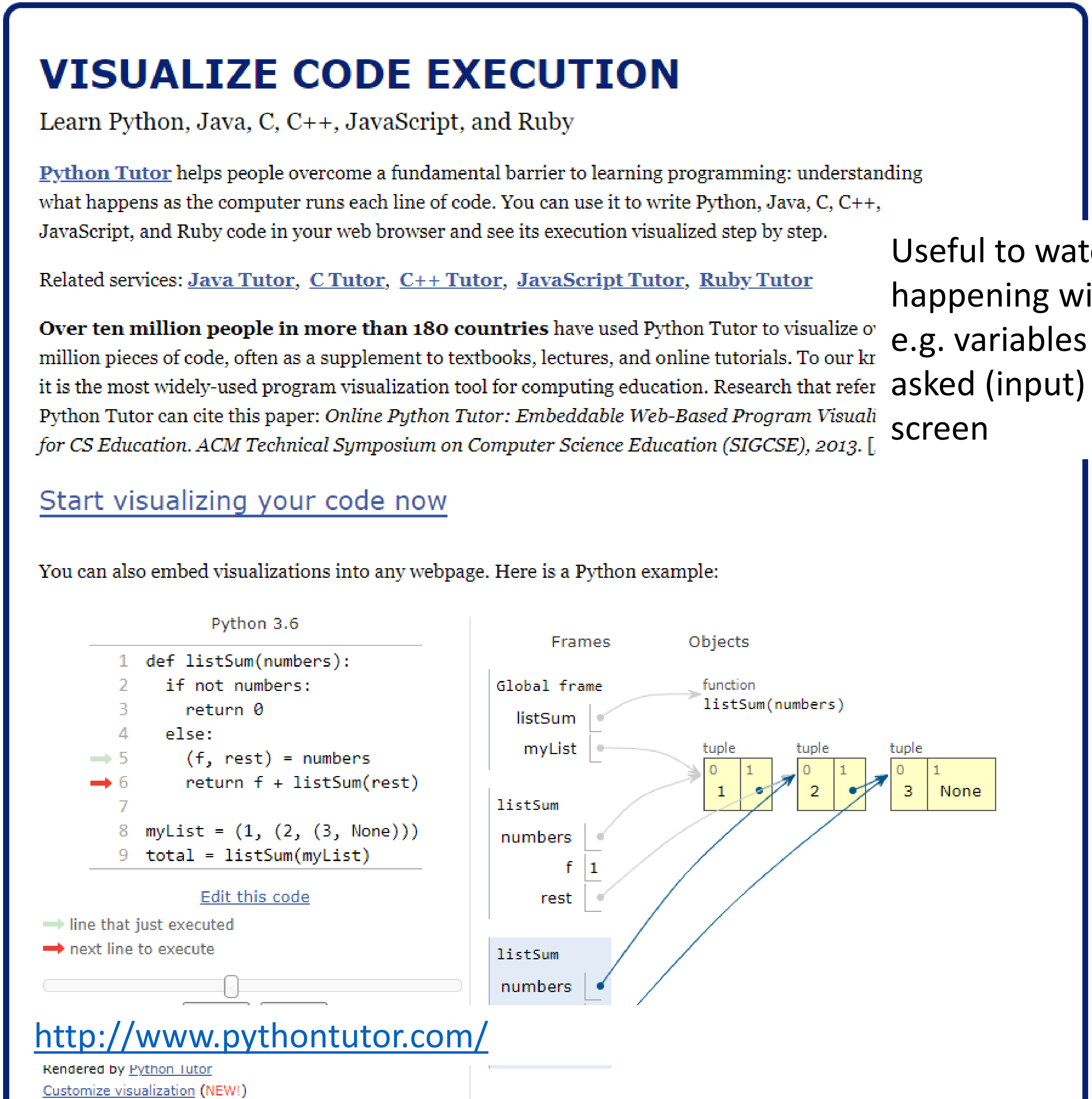
```
>>> year = 2016
>>> event = 'Referendum'
>>> f'Results of the {year} {event}'
'Results of the 2016 Referendum'
```

- The `str.format()` method of strings requires more manual effort. You'll still use `{` and `}` to mark where a variable will be substituted and can provide detailed formatting directives, but you'll also need to provide the information to be formatted.

```
>>> yes_votes = 42_572_654
>>> no_votes = 43_132_495
>>> percentage = yes_votes / (yes_votes + no_votes)
>>> '{:-9} YES votes  {:2.2%}'.format(yes_votes, percentage)
' 42572654 YES votes  49.67%'
```

https://docs.python.org/3.8/tutorial/inputoutput.html

cing and concatenation operations to create any layout you can imagine. The string type has some methods that perform useful operations for padding strings to a given column width.



VISUALIZE CODE EXECUTION

Learn Python, Java, C, C++, JavaScript, and Ruby

Python Tutor helps people overcome a fundamental barrier to learning programming: understanding what happens as the computer runs each line of code. You can use it to write Python, Java, C, C++, JavaScript, and Ruby code in your web browser and see its execution visualized step by step.

Related services: Java Tutor, C Tutor, C++ Tutor, JavaScript Tutor, Ruby Tutor

Over ten million people in more than 180 countries have used Python Tutor to visualize o million pieces of code, often as a supplement to textbooks, lectures, and online tutorials. To our kr it is the most widely-used program visualization tool for computing education. Research that refer Python Tutor can cite this paper: *Online Python Tutor: Embeddable Web-Based Program Visuali for CS Education. ACM Technical Symposium on Computer Science Education (SIGCSE), 2013.* [

Useful to watch what is happening within your program e.g. variables contents, what is asked (input) and printed in screen

Start visualizing your code now

You can also embed visualizations into any webpage. Here is a Python example:

Python 3.6

```
def listSum(numbers):
  if not numbers:
    return 0
  else:
    (f, rest) = numbers
    return f + listSum(rest)

myList = (1, (2, (3, None)))
total = listSum(myList)
```

Edit this code

→ line that just executed

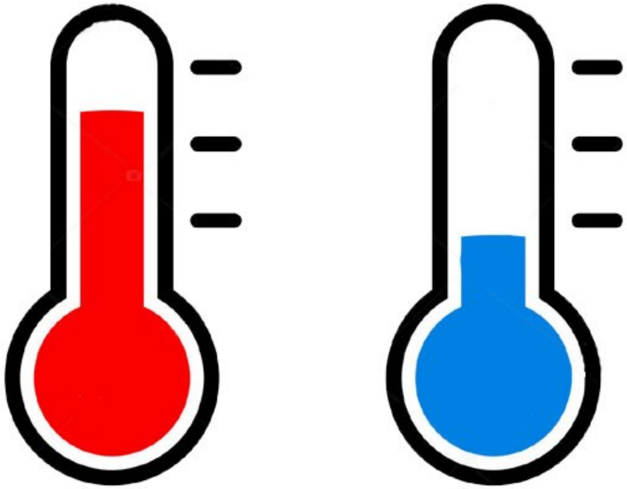
→ next line to execute

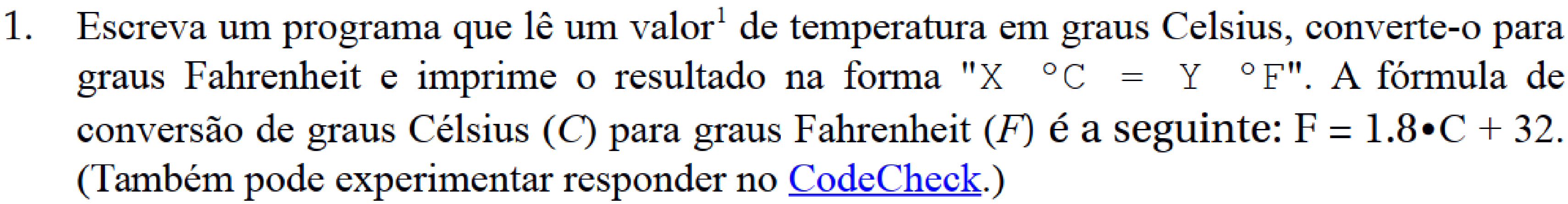
http://www.pythontutor.com/

Rendered by Python Tutor

Customize visualization (NEW!)

Fahrenheit to Celsius (°F to °C) Converter



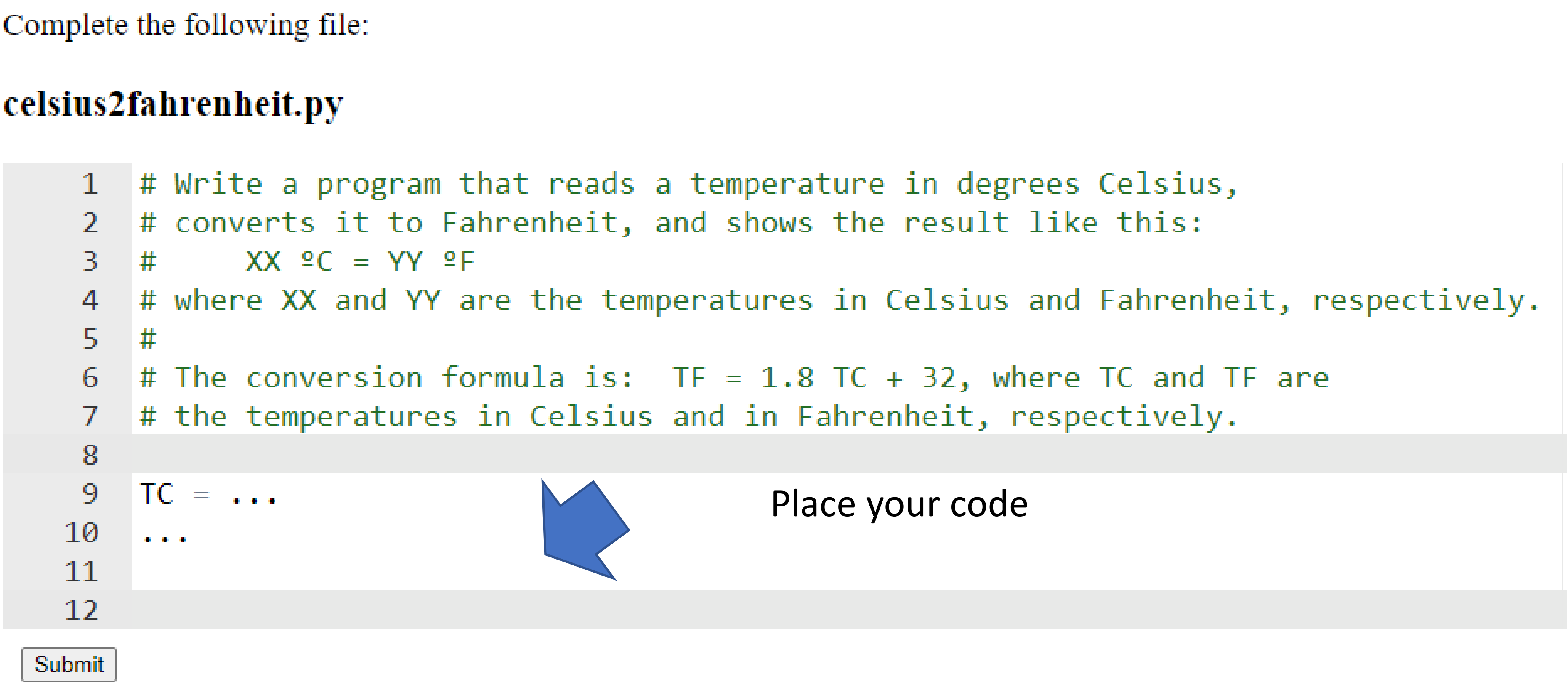


1. Escreva um programa que lê um valor[1] de temperatura em graus Celsius, converte-o para graus Fahrenheit e imprime o resultado na forma "X °C = Y °F". A fórmula de conversão de graus Célsius (*C*) para graus Fahrenheit (*F*) é a seguinte: F = 1.8•C + 32. (Também pode experimentar responder no CodeCheck.)

NOTE: this is prepared by someone who provides
- the correct program,
- the input to test (20,0.36.7 and -40 in this case)

The focus is on the results (output) presented and not on
Your specific program

If formats and messages not well specified is NATURAL to fail
It is sensible to extra spaces and the text must be the same



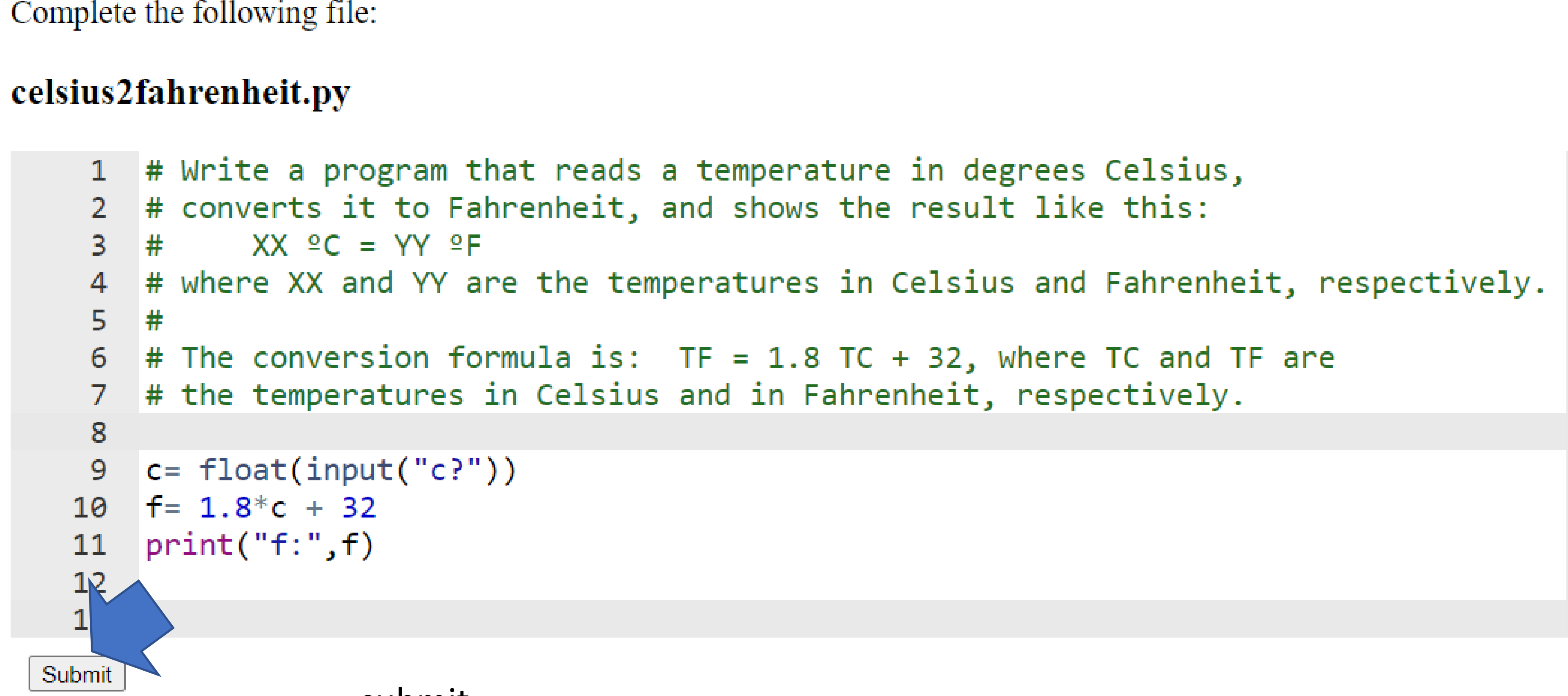
Complete the following file:

celsius2fahrenheit.py

```
# Write a program that reads a temperature in degrees Celsius,
# converts it to Fahrenheit, and shows the result like this:
#     XX ºC = YY ºF
# where XX and YY are the temperatures in Celsius and Fahrenheit, respectively.
#
# The conversion formula is:  TF = 1.8 TC + 32, where TC and TF are
# the temperatures in Celsius and in Fahrenheit, respectively.

TC = ...
...

```

Place your code

http://codecheck.it/files/19092609527ewwt28htujsacpmu6xr0fkue

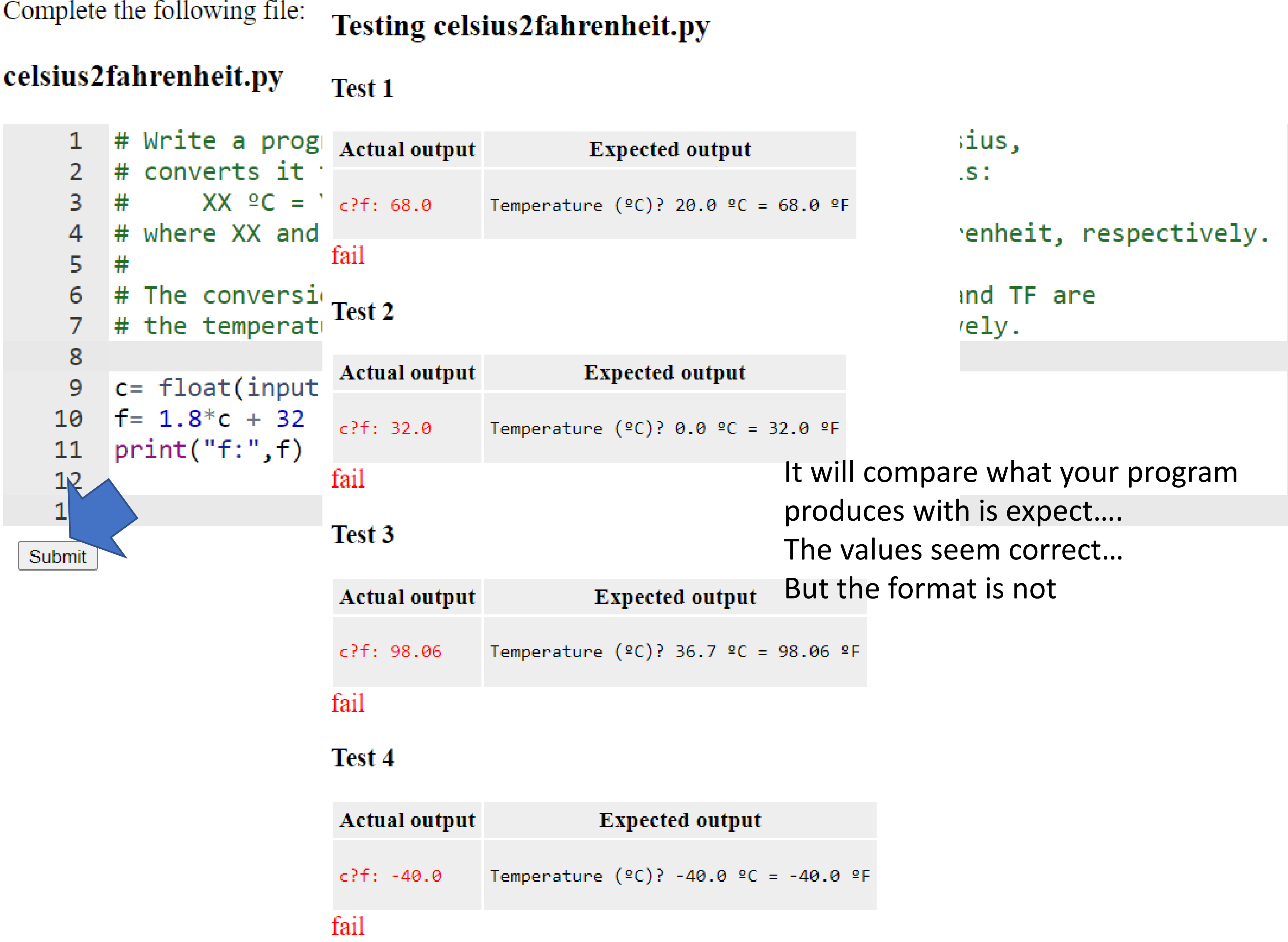


Complete the following file:

celsius2fahrenheit.py

```python
# Write a program that reads a temperature in degrees Celsius,
# converts it to Fahrenheit, and shows the result like this:
#     XX ºC = YY ºF
# where XX and YY are the temperatures in Celsius and Fahrenheit, respectively.
#
# The conversion formula is:  TF = 1.8 TC + 32, where TC and TF are
# the temperatures in Celsius and in Fahrenheit, respectively.

c= float(input("c?"))
f= 1.8*c + 32
print("f:",f)

```

submit



Complete the following file:

celsius2fahrenheit.py

```
# Write a progr                                  ius,
# converts it                                    s:
#      XX ºC = '
# where XX and                                   enheit, respectively.
#
# The conversi                                   nd TF are
# the temperat                                   ely.

c= float(input
f= 1.8*c + 32
print("f:",f)
```

Testing celsius2fahrenheit.py

Test 1

Actual output	Expected output
c?f: 68.0	Temperature (ºC)? 20.0 ºC = 68.0 ºF

fail

Test 2

Actual output	Expected output
c?f: 32.0	Temperature (ºC)? 0.0 ºC = 32.0 ºF

fail

Test 3

Actual output	Expected output
c?f: 98.06	Temperature (ºC)? 36.7 ºC = 98.06 ºF

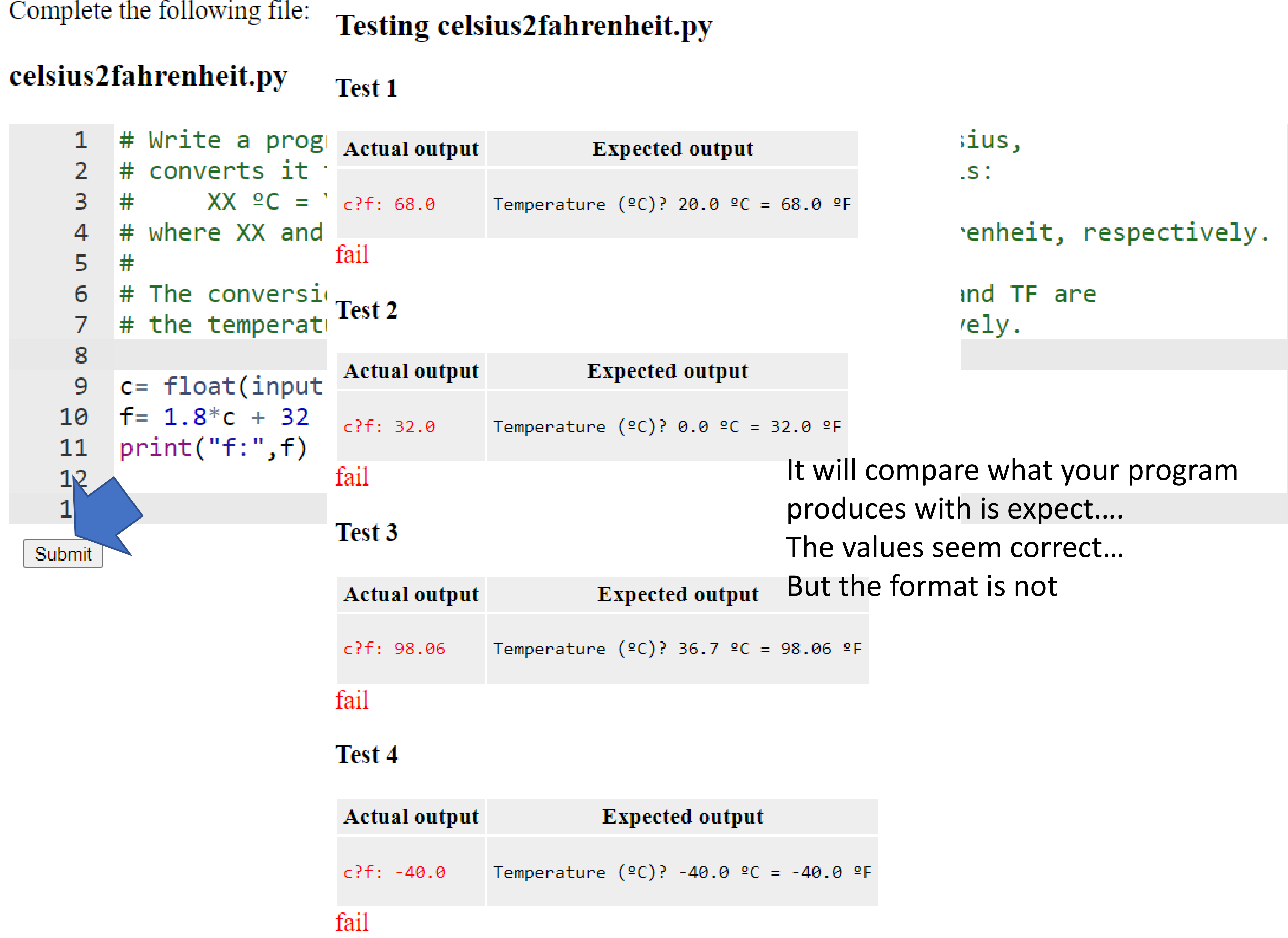
fail

Test 4

Actual output	Expected output
c?f: -40.0	Temperature (ºC)? -40.0 ºC = -40.0 ºF

fail

It will compare what your program produces with is expect….
The values seem correct…
But the format is not



Complete the following file:

celsius2fahrenheit.py

```
# Write a prog                                        ius,
# converts it                                         s:
#      XX ºC = '
# where XX and                                        enheit, respectively.
#
# The conversi                                        nd TF are
# the temperat                                        ely.

c= float(input
f= 1.8*c + 32
print("f:",f)
```

Testing celsius2fahrenheit.py

Test 1

Actual output	Expected output
c?f: 68.0	Temperature (ºC)? 20.0 ºC = 68.0 ºF

fail

Test 2

Actual output	Expected output
c?f: 32.0	Temperature (ºC)? 0.0 ºC = 32.0 ºF

fail

Test 3

Actual output	Expected output
c?f: 98.06	Temperature (ºC)? 36.7 ºC = 98.06 ºF

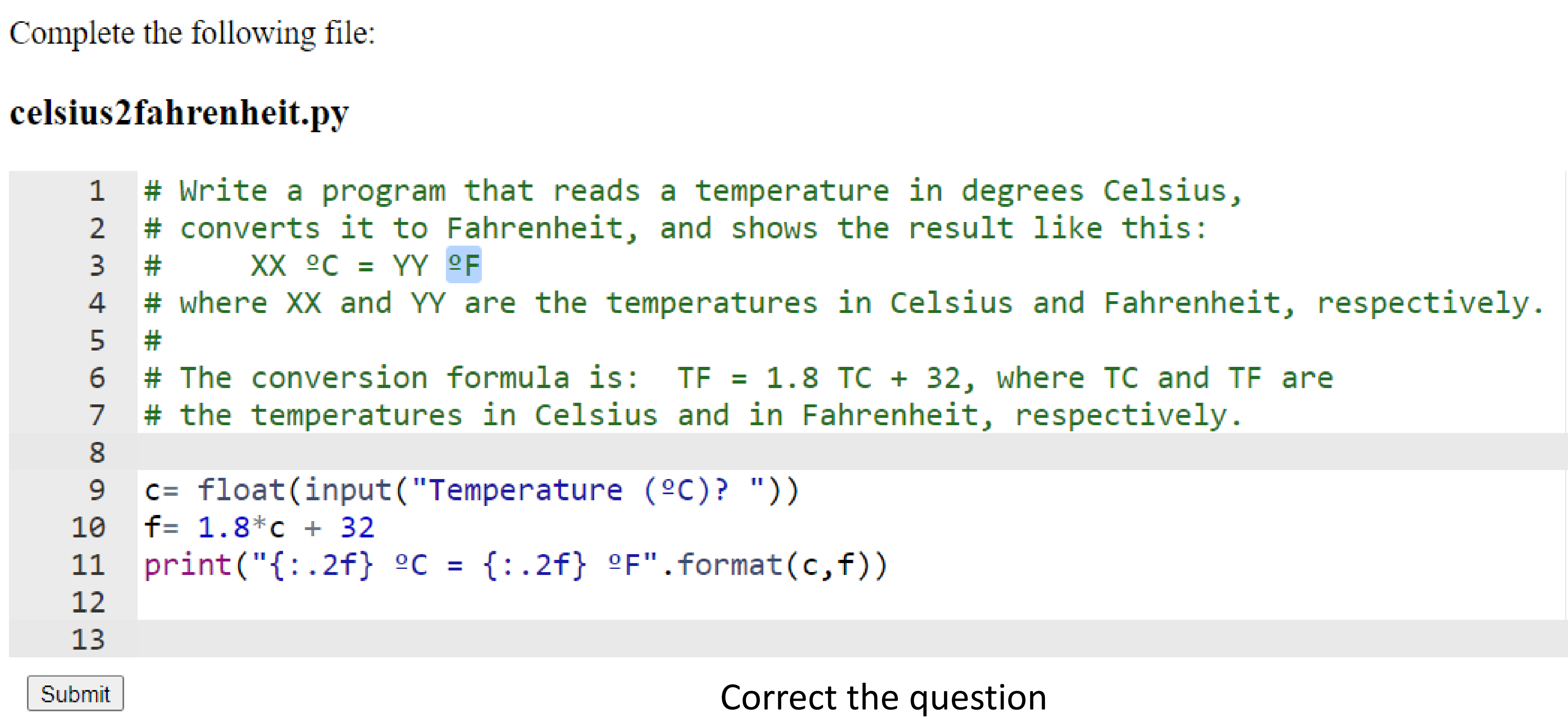
fail

Test 4

Actual output	Expected output
c?f: -40.0	Temperature (ºC)? -40.0 ºC = -40.0 ºF

fail

It will compare what your program produces with is expect….
The values seem correct…
But the format is not

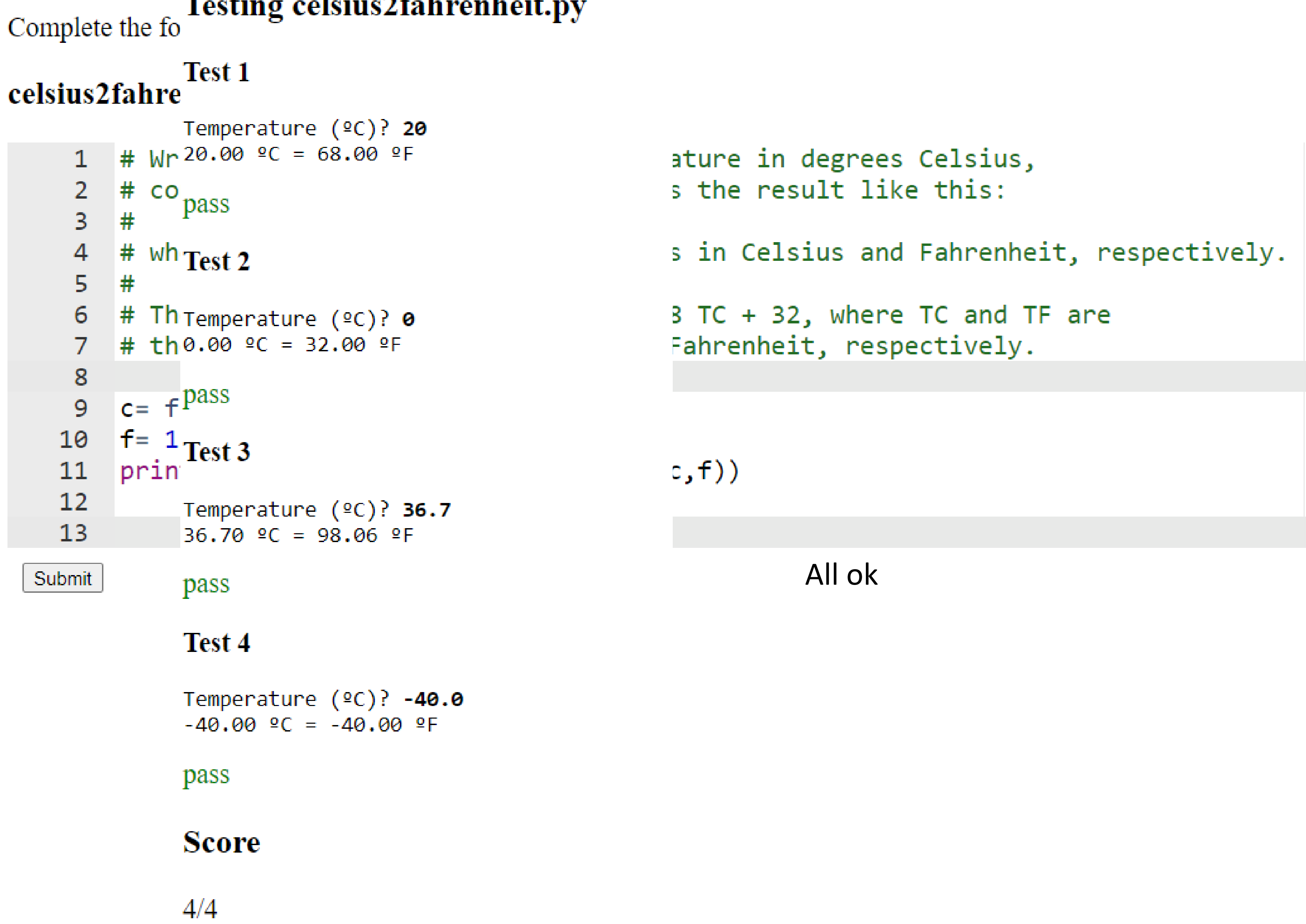


Complete the following file:

celsius2fahrenheit.py

```
# Write a program that reads a temperature in degrees Celsius,
# converts it to Fahrenheit, and shows the result like this:
#     XX ºC = YY ºF
# where XX and YY are the temperatures in Celsius and Fahrenheit, respectively.
#
# The conversion formula is:  TF = 1.8 TC + 32, where TC and TF are
# the temperatures in Celsius and in Fahrenheit, respectively.

c= float(input("Temperature (ºC)? "))
f= 1.8*c + 32
print("{:.2f} ºC = {:.2f} ºF".format(c,f))


```

Correct the question
And the output format



Complete the fo

celsius2fahre

```
# Wr                                          ature in degrees Celsius,
# co                                          s the result like this:
#
# wh                                          s in Celsius and Fahrenheit, respectively.
#
# Th                                          3 TC + 32, where TC and TF are
# th                                          Fahrenheit, respectively.

c= f
f= 1
prin                                          c,f))
```

Testing celsius2fahrenheit.py

Test 1

```
Temperature (ºC)? 20
20.00 ºC = 68.00 ºF
```

pass

Test 2

```
Temperature (ºC)? 0
0.00 ºC = 32.00 ºF
```

pass

Test 3

```
Temperature (ºC)? 36.7
36.70 ºC = 98.06 ºF
```

pass

Test 4

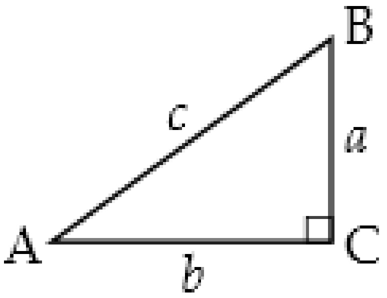
```
Temperature (ºC)? -40.0
-40.00 ºC = -40.00 ºF
```

pass

Score

4/4

All ok



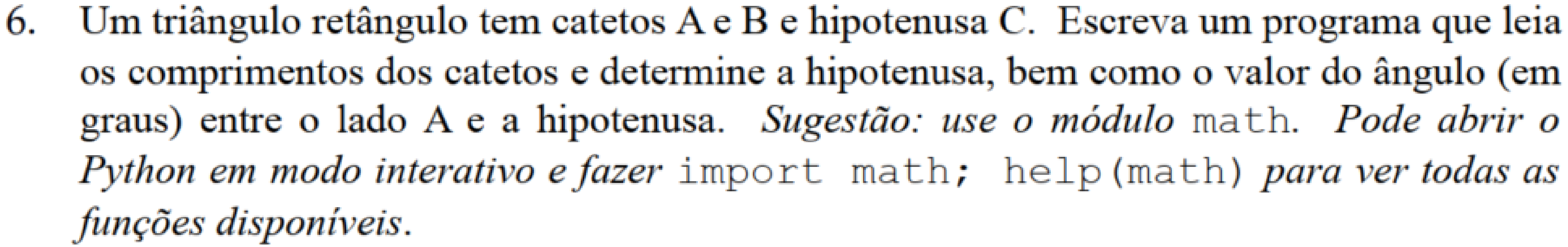
Pythagorean Theorem:

$$a^2 + b^2 = c^2$$

$$\sin A = \frac{a}{c} = \left(\frac{\text{opposite}}{\text{hypotenuse}}\right)$$

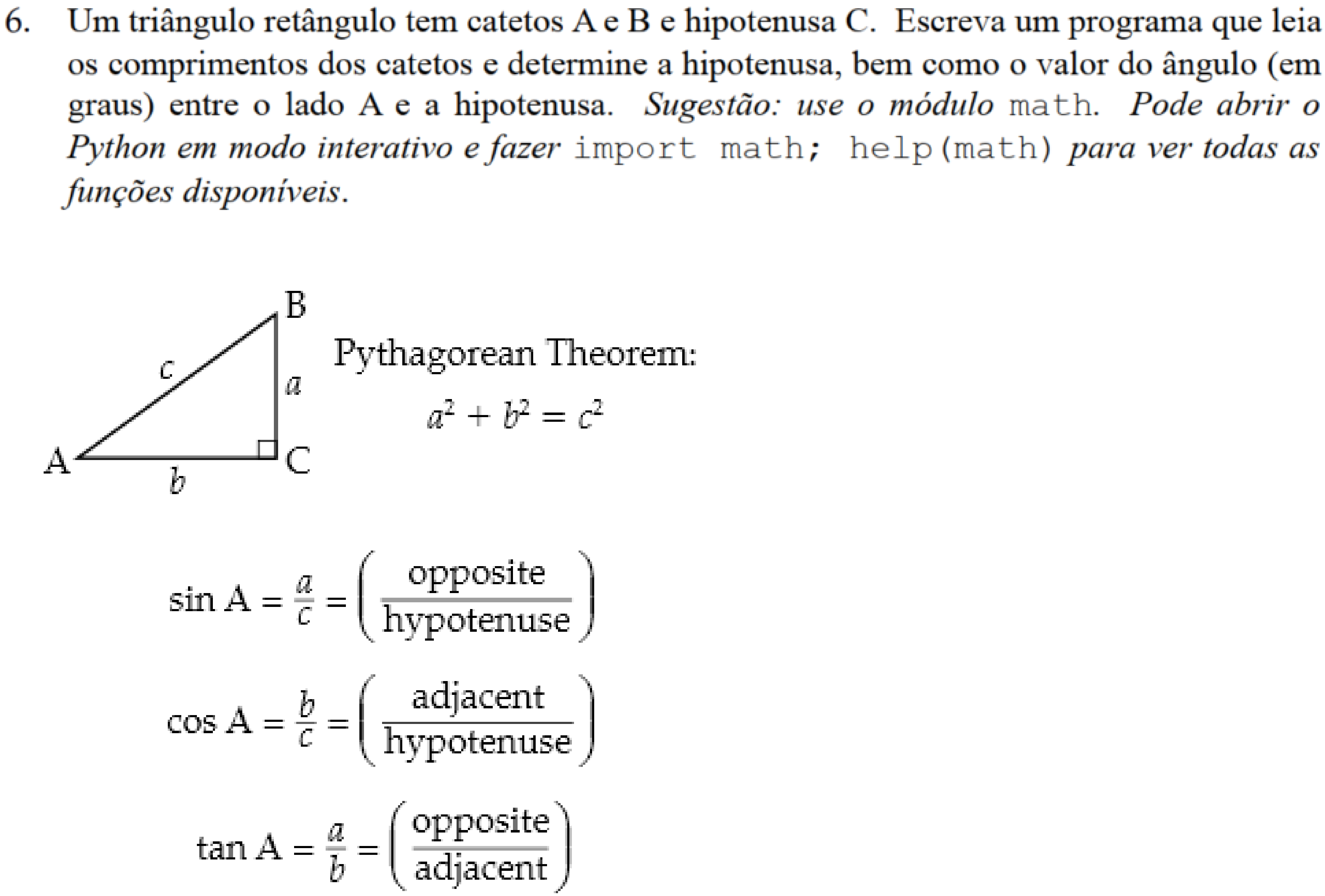
$$\cos A = \frac{b}{c} = \left(\frac{\text{adjacent}}{\text{hypotenuse}}\right)$$

$$\tan A = \frac{a}{b} = \left(\frac{\text{opposite}}{\text{adjacent}}\right)$$



6. Um triângulo retângulo tem catetos A e B e hipotenusa C. Escreva um programa que leia os comprimentos dos catetos e determine a hipotenusa, bem como o valor do ângulo (em graus) entre o lado A e a hipotenusa. *Sugestão: use o módulo* `math`. *Pode abrir o Python em modo interativo e fazer* `import math; help(math)` *para ver todas as funções disponíveis.*

Note: can find these formulas on a quick search on the web



6. Um triângulo retângulo tem catetos A e B e hipotenusa C. Escreva um programa que leia os comprimentos dos catetos e determine a hipotenusa, bem como o valor do ângulo (em graus) entre o lado A e a hipotenusa. *Sugestão: use o módulo* `math`. *Pode abrir o Python em modo interativo e fazer* `import math; help(math)` *para ver todas as funções disponíveis.*

Pythagorean Theorem:

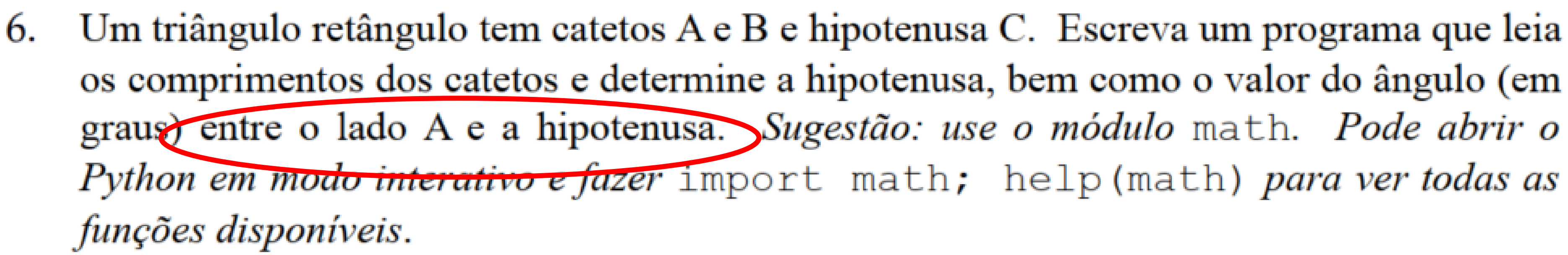
$$a^2 + b^2 = c^2$$

$$\sin A = \frac{a}{c} = \left(\frac{\text{opposite}}{\text{hypotenuse}}\right)$$

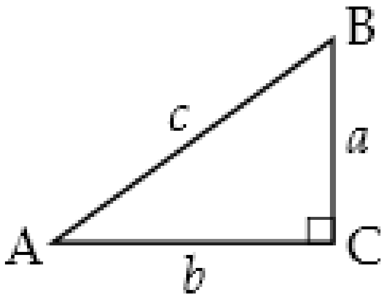
$$\cos A = \frac{b}{c} = \left(\frac{\text{adjacent}}{\text{hypotenuse}}\right)$$

$$\tan A = \frac{a}{b} = \left(\frac{\text{opposite}}{\text{adjacent}}\right)$$

Note: can find these formulas on a quick search on the web



6. Um triângulo retângulo tem catetos A e B e hipotenusa C. Escreva um programa que leia os comprimentos dos catetos e determine a hipotenusa, bem como o valor do ângulo (em graus) entre o lado A e a hipotenusa. *Sugestão: use o módulo* `math`. *Pode abrir o Python em modo interativo e fazer* `import math; help(math)` *para ver todas as funções disponíveis.*



Pythagorean Theorem:

$$a^2 + b^2 = c^2$$

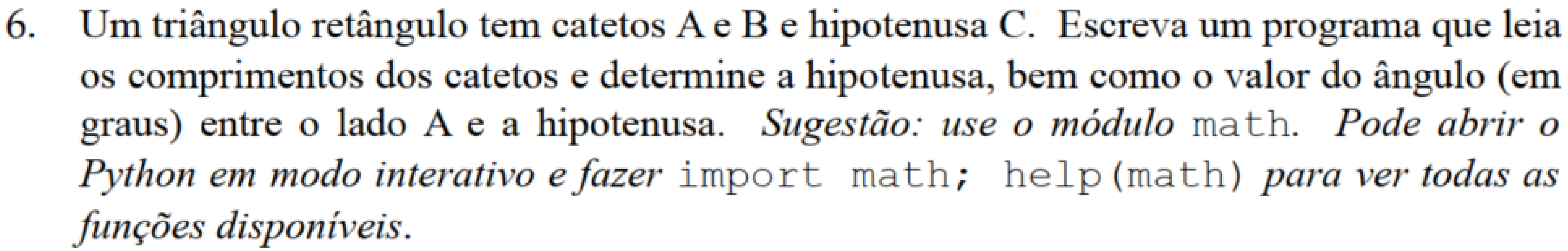
What is the side? Adjacent or opposite ?

$$\sin A = \frac{a}{c} = \left(\frac{\text{opposite}}{\text{hypotenuse}}\right)$$

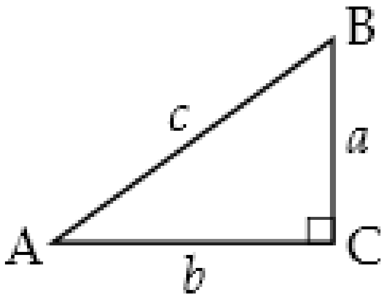
$$\cos A = \frac{b}{c} = \left(\frac{\text{adjacent}}{\text{hypotenuse}}\right)$$

$$\tan A = \frac{a}{b} = \left(\frac{\text{opposite}}{\text{adjacent}}\right)$$

Note: can find these formulas on a quick search on the web



6. Um triângulo retângulo tem catetos A e B e hipotenusa C. Escreva um programa que leia os comprimentos dos catetos e determine a hipotenusa, bem como o valor do ângulo (em graus) entre o lado A e a hipotenusa. *Sugestão: use o módulo* `math`. *Pode abrir o Python em modo interativo e fazer* `import math; help(math)` *para ver todas as funções disponíveis.*



Pythagorean Theorem:

$$a^2 + b^2 = c^2$$

$$\sin A = \frac{a}{c} = \left(\frac{\text{opposite}}{\text{hypotenuse}}\right)$$

$$\cos A = \frac{b}{c} = \left(\frac{\text{adjacent}}{\text{hypotenuse}}\right)$$

$$\tan A = \frac{a}{b} = \left(\frac{\text{opposite}}{\text{adjacent}}\right)$$

Select the model
Deduce the formula
Solve in order to hypotenuse
i.e.

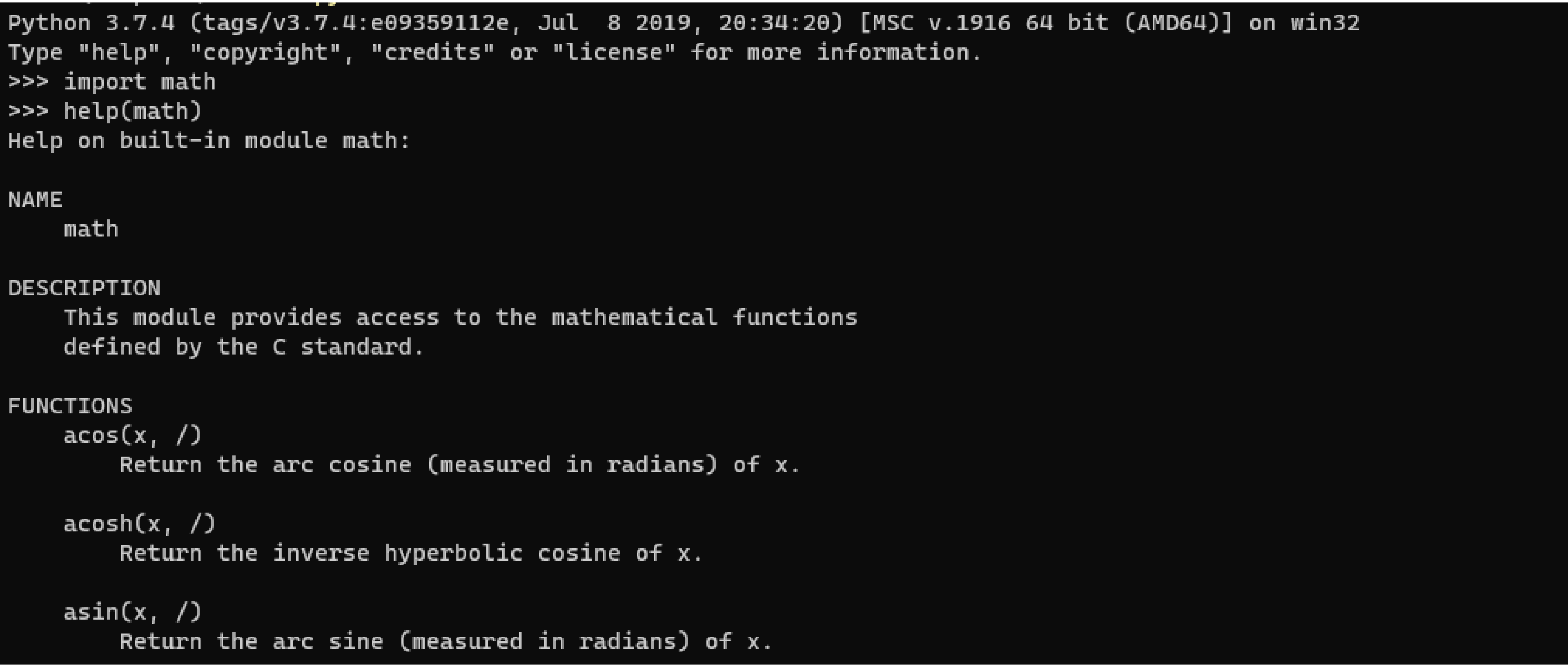
Hypotenuse = ... Something…

Build program
- Ask for values needed
- Apply the formula
- Print out the result

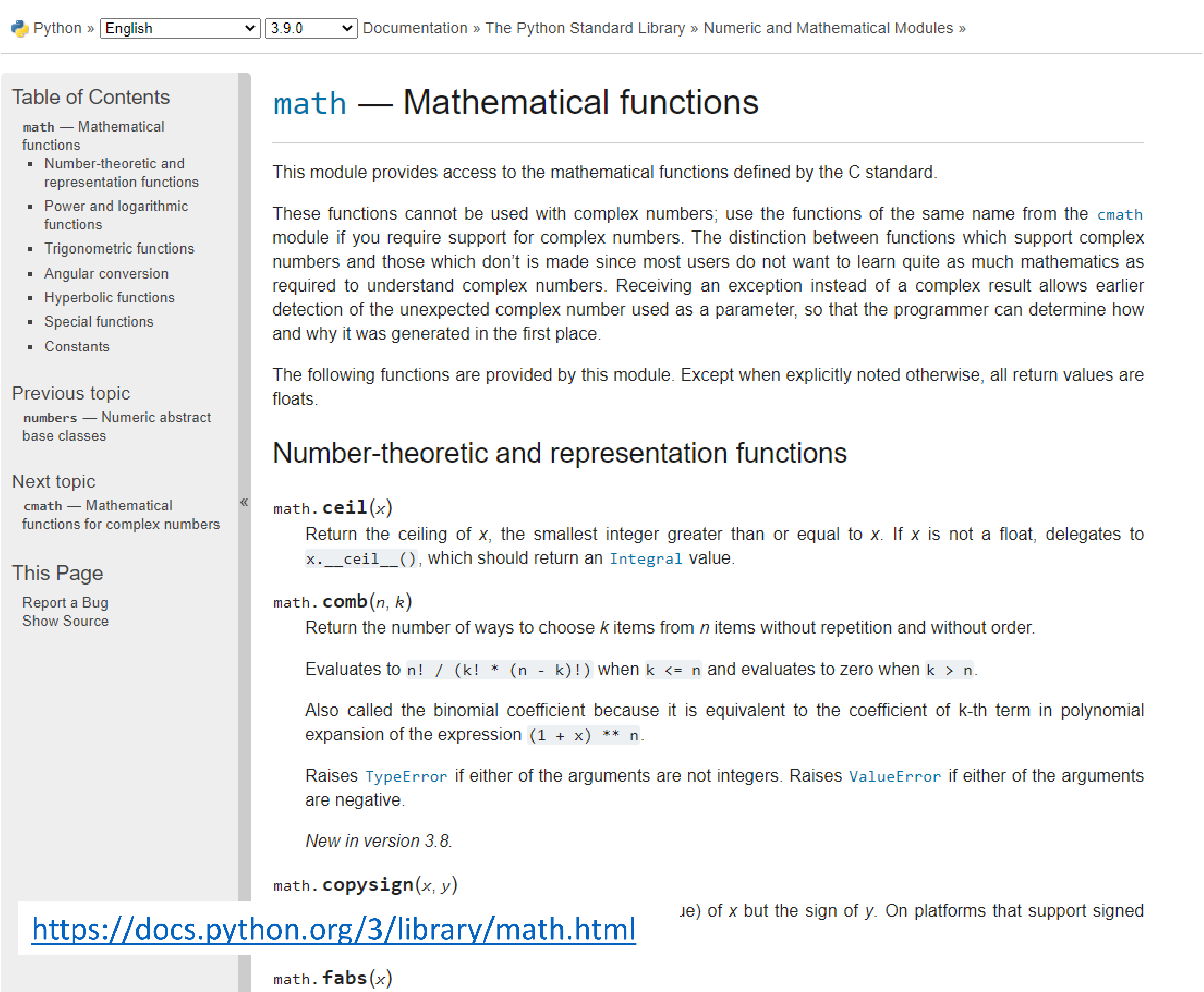
Remember for trigonometric functions
look into math library

Math library

You can find the trigonometric and other functions in library math



```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import math
>>> help(math)
Help on built-in module math:

NAME
    math

DESCRIPTION
    This module provides access to the mathematical functions
    defined by the C standard.

FUNCTIONS
    acos(x, /)
        Return the arc cosine (measured in radians) of x.

    acosh(x, /)
        Return the inverse hyperbolic cosine of x.

    asin(x, /)
        Return the arc sine (measured in radians) of x.
```



Python » English 3.9.0 Documentation » The Python Standard Library » Numeric and Mathematical Modules »

Previous topic

`numbers` — Numeric abstract base classes

Next topic

`cmath` — Mathematical functions for complex numbers

This Page

Report a Bug
Show Source

`math` — Mathematical functions

This module provides access to the mathematical functions defined by the C standard.

These functions cannot be used with complex numbers; use the functions of the same name from the `cmath` module if you require support for complex numbers. The distinction between functions which support complex numbers and those which don't is made since most users do not want to learn quite as much mathematics as required to understand complex numbers. Receiving an exception instead of a complex result allows earlier detection of the unexpected complex number used as a parameter, so that the programmer can determine how and why it was generated in the first place.

The following functions are provided by this module. Except when explicitly noted otherwise, all return values are floats.

Number-theoretic and representation functions

`math.`**`ceil`**(*x*)

Return the ceiling of *x*, the smallest integer greater than or equal to *x*. If *x* is not a float, delegates to `x.__ceil__()`, which should return an `Integral` value.

`math.`**`comb`**(*n*, *k*)

Return the number of ways to choose *k* items from *n* items without repetition and without order.

Evaluates to `n! / (k! * (n - k)!)` when `k <= n` and evaluates to zero when `k > n`.

Also called the binomial coefficient because it is equivalent to the coefficient of k-th term in polynomial expansion of the expression `(1 + x) ** n`.

Raises `TypeError` if either of the arguments are not integers. Raises `ValueError` if either of the arguments are negative.

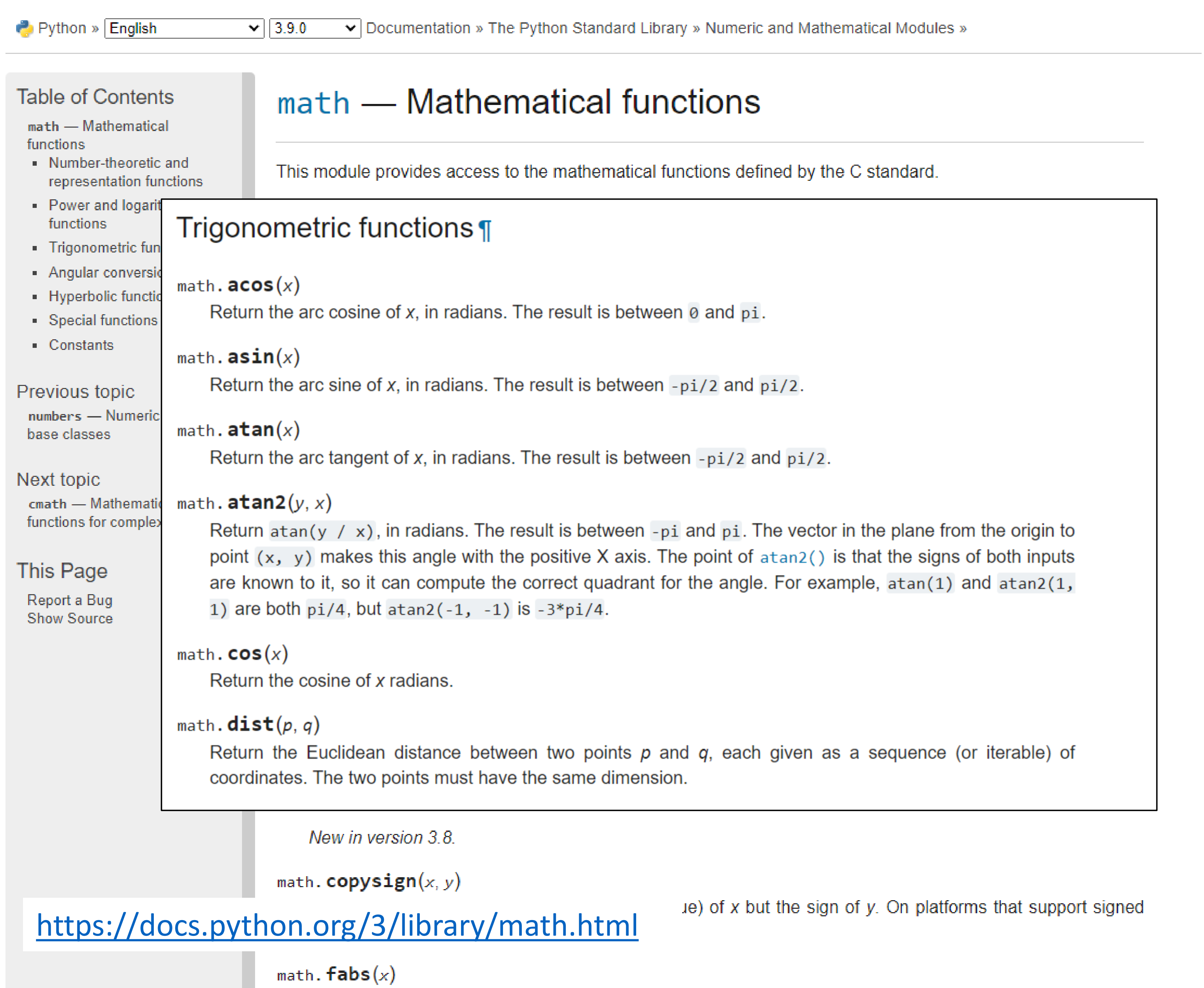
New in version 3.8.

`math.`**`copysign`**(*x*, *y*)

ue) of *x* but the sign of *y*. On platforms that support signed

https://docs.python.org/3/library/math.html

`math.`**`fabs`**(*x*)



Python » English 3.9.0 Documentation » The Python Standard Library » Numeric and Mathematical Modules »

Previous topic

`numbers` — Numeric
base classes

Next topic

`cmath` — Mathemati
functions for complex

This Page

Report a Bug
Show Source

`math` — Mathematical functions

This module provides access to the mathematical functions defined by the C standard.

Trigonometric functions ¶

`math.acos(x)`

Return the arc cosine of *x*, in radians. The result is between `0` and `pi`.

`math.asin(x)`

Return the arc sine of *x*, in radians. The result is between `-pi/2` and `pi/2`.

`math.atan(x)`

Return the arc tangent of *x*, in radians. The result is between `-pi/2` and `pi/2`.

`math.atan2(y, x)`

Return `atan(y / x)`, in radians. The result is between `-pi` and `pi`. The vector in the plane from the origin to point `(x, y)` makes this angle with the positive X axis. The point of `atan2()` is that the signs of both inputs are known to it, so it can compute the correct quadrant for the angle. For example, `atan(1)` and `atan2(1, 1)` are both `pi/4`, but `atan2(-1, -1)` is `-3*pi/4`.

`math.cos(x)`

Return the cosine of *x* radians.

`math.dist(p, q)`

Return the Euclidean distance between two points *p* and *q*, each given as a sequence (or iterable) of coordinates. The two points must have the same dimension.

New in version 3.8.

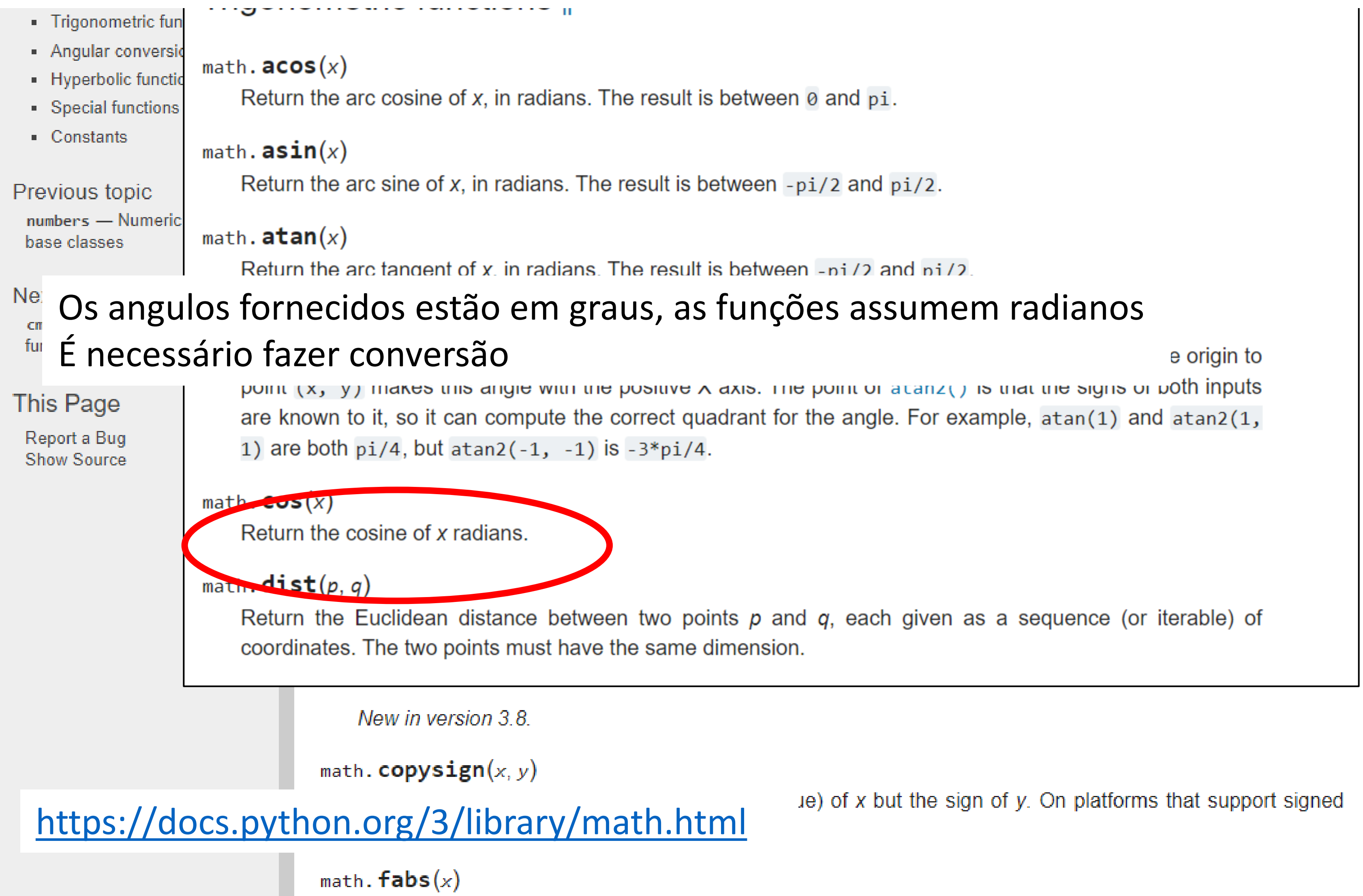
`math.copysign(x, y)`

ue) of *x* but the sign of *y*. On platforms that support signed

`math.fabs(x)`

https://docs.python.org/3/library/math.html

6. Um triângulo retângulo tem catetos A e B e hipotenusa C. Escreva um programa que leia os comprimentos dos catetos e determine a hipotenusa, bem como o valor do ângulo (em graus) entre o lado A e a hipotenusa. *Sugestão: use o módulo* `math`. *Pode abrir o Python em modo interativo e fazer* `import math; help(math)` *para ver todas as funções disponíveis.*



- Trigonometric fun
- Angular conversi
- Hyperbolic functi
- Special functions
- Constants

Previous topic

numbers — Numeric base classes

Ne

cm
fu

This Page

Report a Bug
Show Source

Trigonometric functions

`math.acos(x)`
Return the arc cosine of *x*, in radians. The result is between `0` and `pi`.

`math.asin(x)`
Return the arc sine of *x*, in radians. The result is between `-pi/2` and `pi/2`.

`math.atan(x)`
Return the arc tangent of *x*, in radians. The result is between `-pi/2` and `pi/2`.

e origin to point `(x, y)` makes this angle with the positive X axis. The point of `atan2()` is that the signs of both inputs are known to it, so it can compute the correct quadrant for the angle. For example, `atan(1)` and `atan2(1, 1)` are both `pi/4`, but `atan2(-1, -1)` is `-3*pi/4`.

`math.cos(x)`
Return the cosine of *x* radians.

`math.dist(p, q)`
Return the Euclidean distance between two points *p* and *q*, each given as a sequence (or iterable) of coordinates. The two points must have the same dimension.

New in version 3.8.

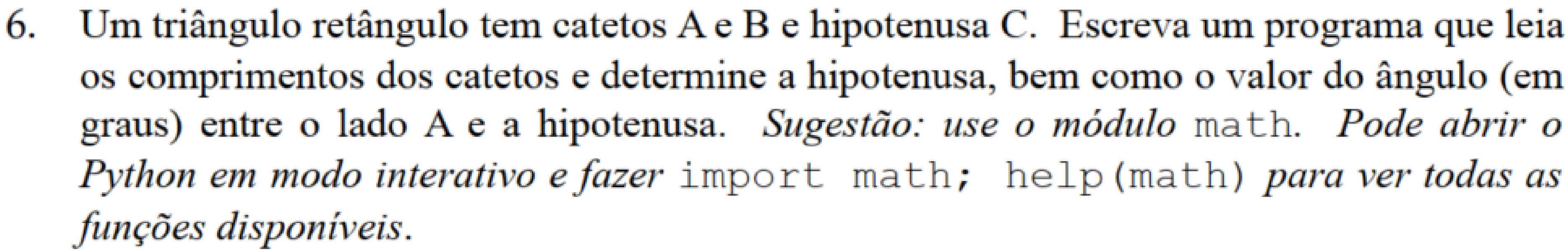
`math.copysign(x, y)`

ue) of *x* but the sign of *y*. On platforms that support signed

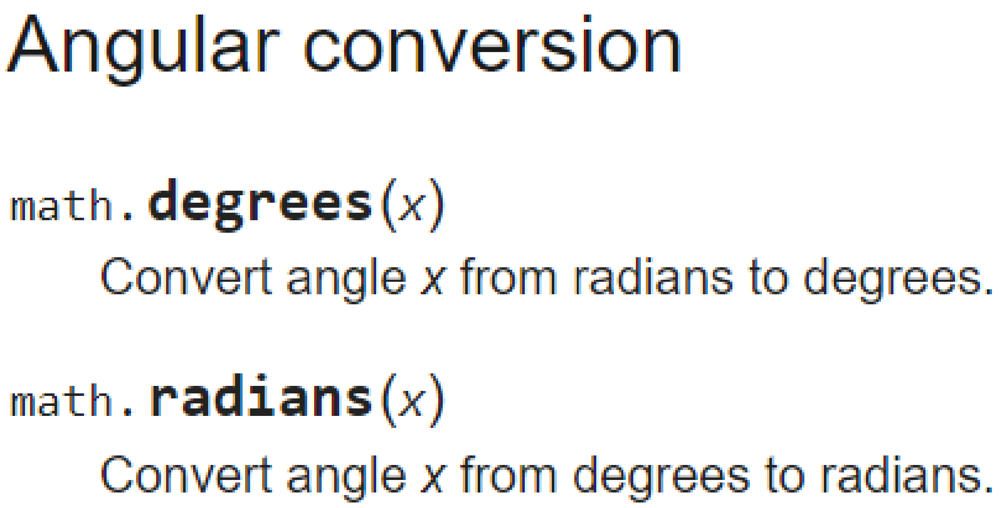
`math.fabs(x)`

Os angulos fornecidos estão em graus, as funções assumem radianos
É necessário fazer conversão

https://docs.python.org/3/library/math.html



6. Um triângulo retângulo tem catetos A e B e hipotenusa C. Escreva um programa que leia os comprimentos dos catetos e determine a hipotenusa, bem como o valor do ângulo (em graus) entre o lado A e a hipotenusa. *Sugestão: use o módulo* `math`. *Pode abrir o Python em modo interativo e fazer* `import math; help(math)` *para ver todas as funções disponíveis.*



Angular conversion

`math.`**`degrees`**`(`*x*`)`

Convert angle *x* from radians to degrees.

`math.`**`radians`**`(`*x*`)`

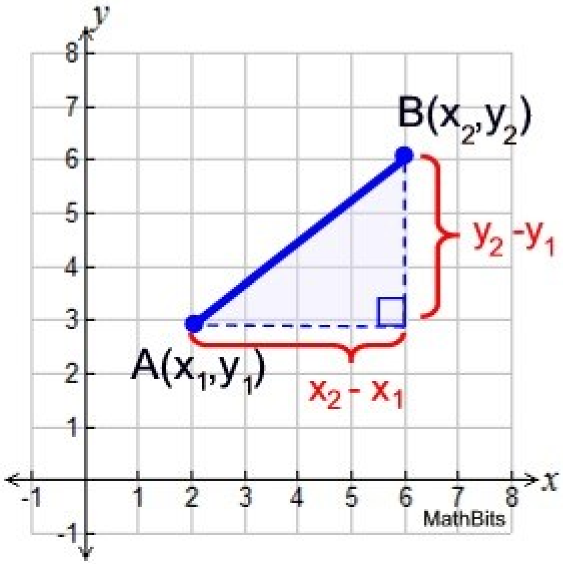
Convert angle *x* from degrees to radians.

```
Ag = float( input(“Angulo em graus?”))  # Angulo em graus
Ar = math.radians( ag )   # Angulo em radianos
… math.cos( ar ) ….   # posso usar cos pois ar já é em radianos
```

https://docs.python.org/3/library/math.html

7. O programa `points.py` lê as coordenadas cartesianas de dois pontos (x1,y1) e (x2,y2). Complete-o para calcular e imprimir a distância entre os pontos.

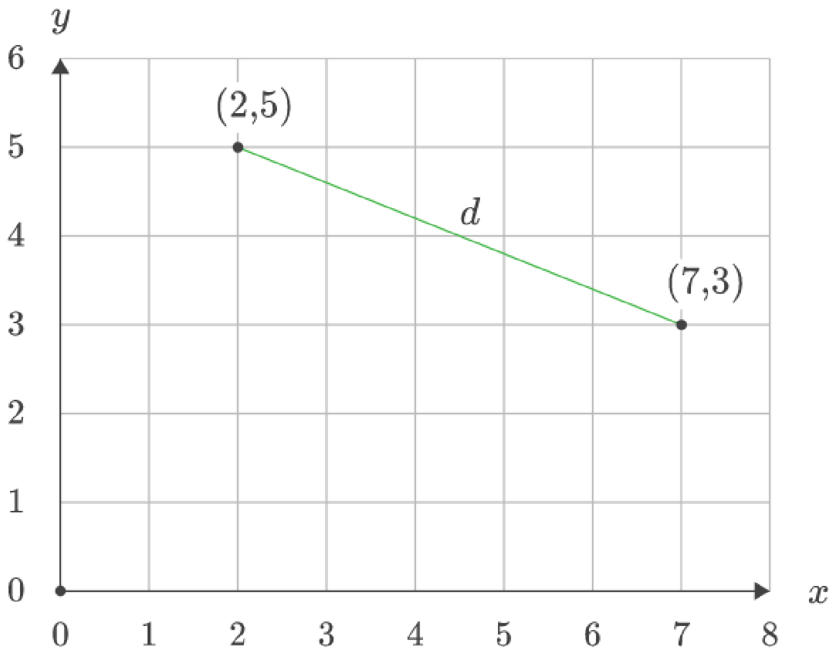
Como representar os pontos? Com inteiros? floats?
Como ler?



Deriving the Distance Formula

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Alguns exemplos



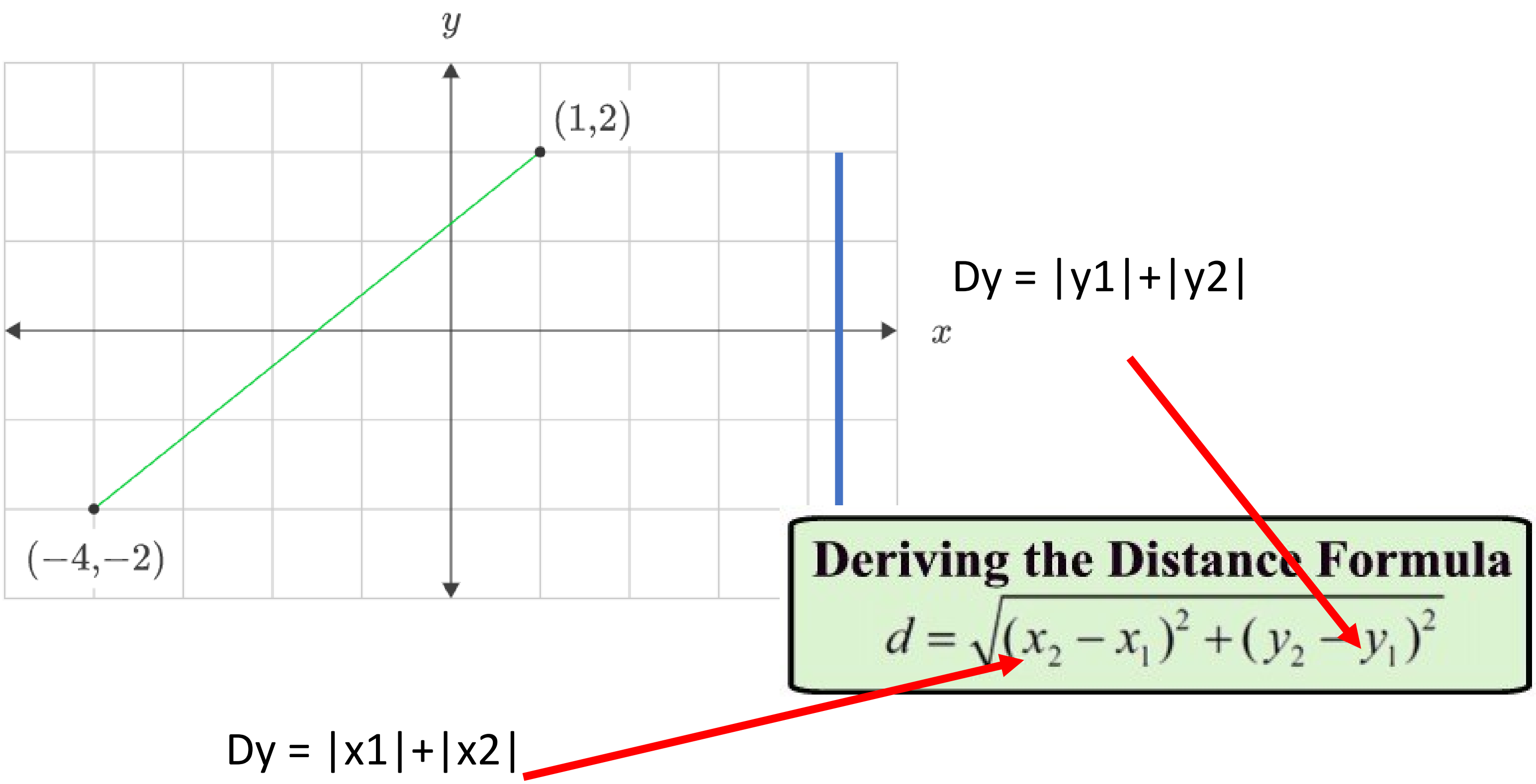
Dy = |y1-y2|

Deriving the Distance Formula

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Dy = |x1-x2|

Alguns exemplos



Dy = |y1|+|y2|

Deriving the Distance Formula

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Dy = |x1|+|x2|

resumindo

Se x1*x2 <0 (diferentes lados do eixos dos y’s)

    Dx = |x1|+|x2|

Senão

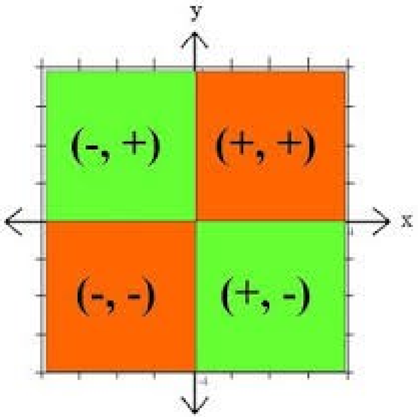
    Dx = |x1-x2|

Se y1*y2 <0 (diferentes lados do eixos dos x’s)

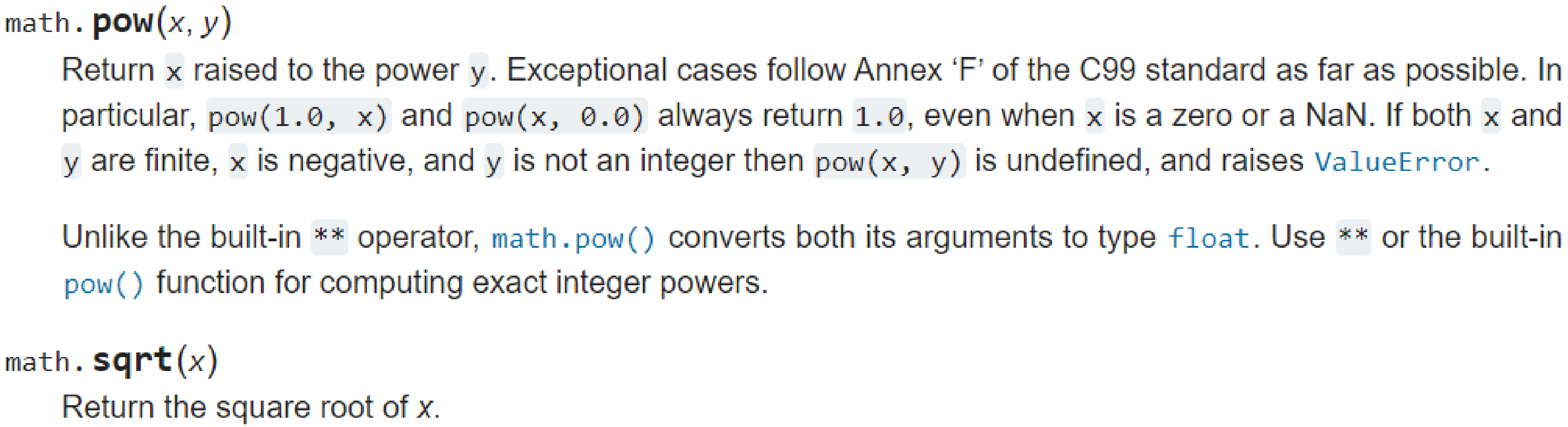
    Dy = |y1|+|y2|

Senão

    Dyx = |y1-y2|



Funções que podem úteis



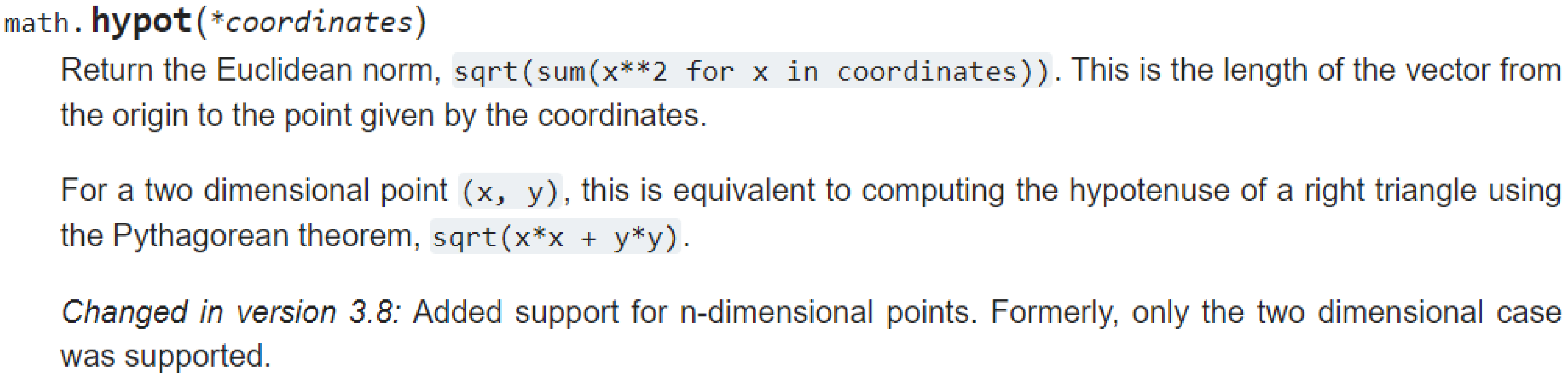
math.**pow**(*x*, *y*)

Return `x` raised to the power `y`. Exceptional cases follow Annex ‘F’ of the C99 standard as far as possible. In particular, `pow(1.0, x)` and `pow(x, 0.0)` always return `1.0`, even when `x` is a zero or a NaN. If both `x` and `y` are finite, `x` is negative, and `y` is not an integer then `pow(x, y)` is undefined, and raises `ValueError`.

Unlike the built-in `**` operator, `math.pow()` converts both its arguments to type `float`. Use `**` or the built-in `pow()` function for computing exact integer powers.

math.**sqrt**(*x*)

Return the square root of *x*.

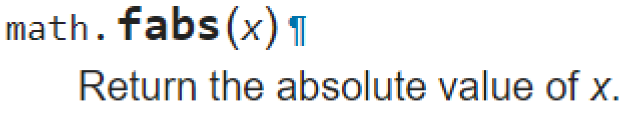


math.**hypot**(**coordinates*)

Return the Euclidean norm, `sqrt(sum(x**2 for x in coordinates))`. This is the length of the vector from the origin to the point given by the coordinates.

For a two dimensional point `(x, y)`, this is equivalent to computing the hypotenuse of a right triangle using the Pythagorean theorem, `sqrt(x*x + y*y)`.

Changed in version 3.8: Added support for n-dimensional points. Formerly, only the two dimensional case was supported.



math.**fabs**(*x*)¶

Return the absolute value of *x*.

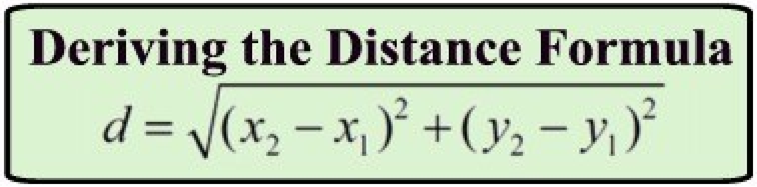
Traduzir para python

- Eventualmente usar algumas das funções anteriores

```
if x1*x2 > 0 :    dx = math.fabs( x1-x2)else:
dx= math.fabs( x1) + math.fabs( x2)
```

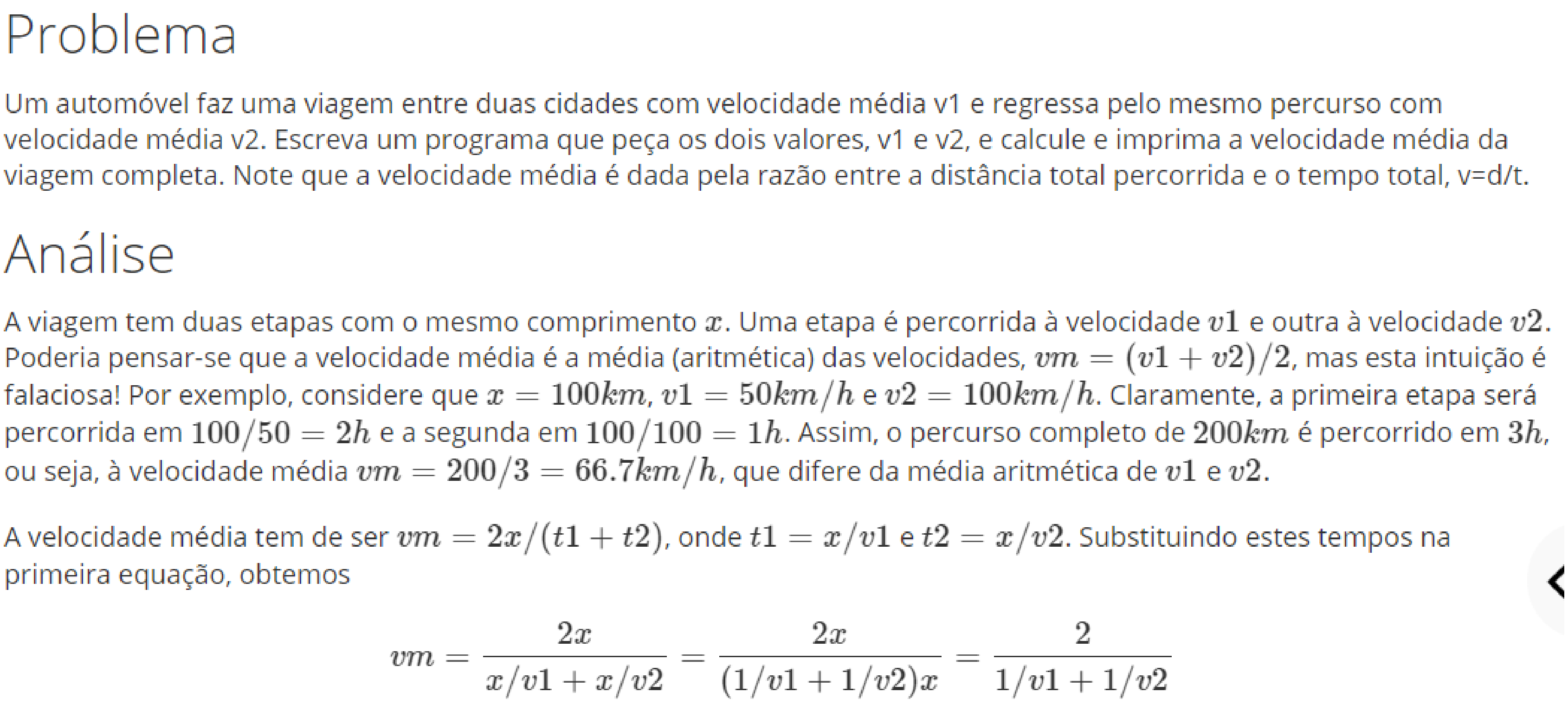
```
Se x1*x2 <0 ( diferentes lados do
eixos dos y’s)
    Dx = |x1|+|x2|
Senão
    Dx = |x1-x2|
```

```
d= math.hypot(  dx ,dy)
d= math.sqrt( dx*dx + dy*dy )
d= math.sqrt( dx**2 + dy**2 )
d= math.pow( dx**2 + dy**2 ,0.5 )
```



Deriving the Distance Formula

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$



Problema

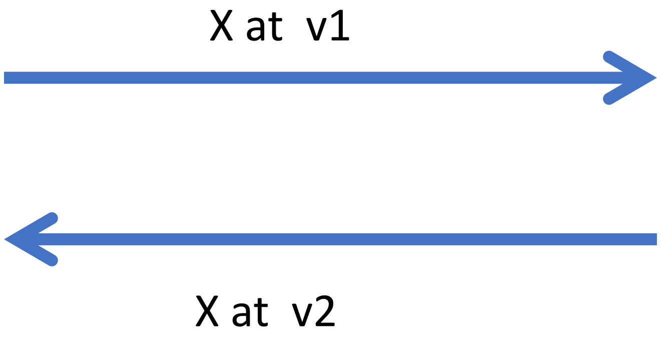
Um automóvel faz uma viagem entre duas cidades com velocidade média v1 e regressa pelo mesmo percurso com velocidade média v2. Escreva um programa que peça os dois valores, v1 e v2, e calcule e imprima a velocidade média da viagem completa. Note que a velocidade média é dada pela razão entre a distância total percorrida e o tempo total, v=d/t.

Análise

A viagem tem duas etapas com o mesmo comprimento x. Uma etapa é percorrida à velocidade $v1$ e outra à velocidade $v2$. Poderia pensar-se que a velocidade média é a média (aritmética) das velocidades, $vm = (v1 + v2)/2$, mas esta intuição é falaciosa! Por exemplo, considere que $x = 100km$, $v1 = 50km/h$ e $v2 = 100km/h$. Claramente, a primeira etapa será percorrida em $100/50 = 2h$ e a segunda em $100/100 = 1h$. Assim, o percurso completo de $200km$ é percorrido em $3h$, ou seja, à velocidade média $vm = 200/3 = 66.7km/h$, que difere da média aritmética de $v1$ e $v2$.

A velocidade média tem de ser $vm = 2x/(t1 + t2)$, onde $t1 = x/v1$ e $t2 = x/v2$. Substituindo estes tempos na primeira equação, obtemos

$$vm = \frac{2x}{x/v1 + x/v2} = \frac{2x}{(1/v1 + 1/v2)x} = \frac{2}{1/v1 + 1/v2}$$



X at v1

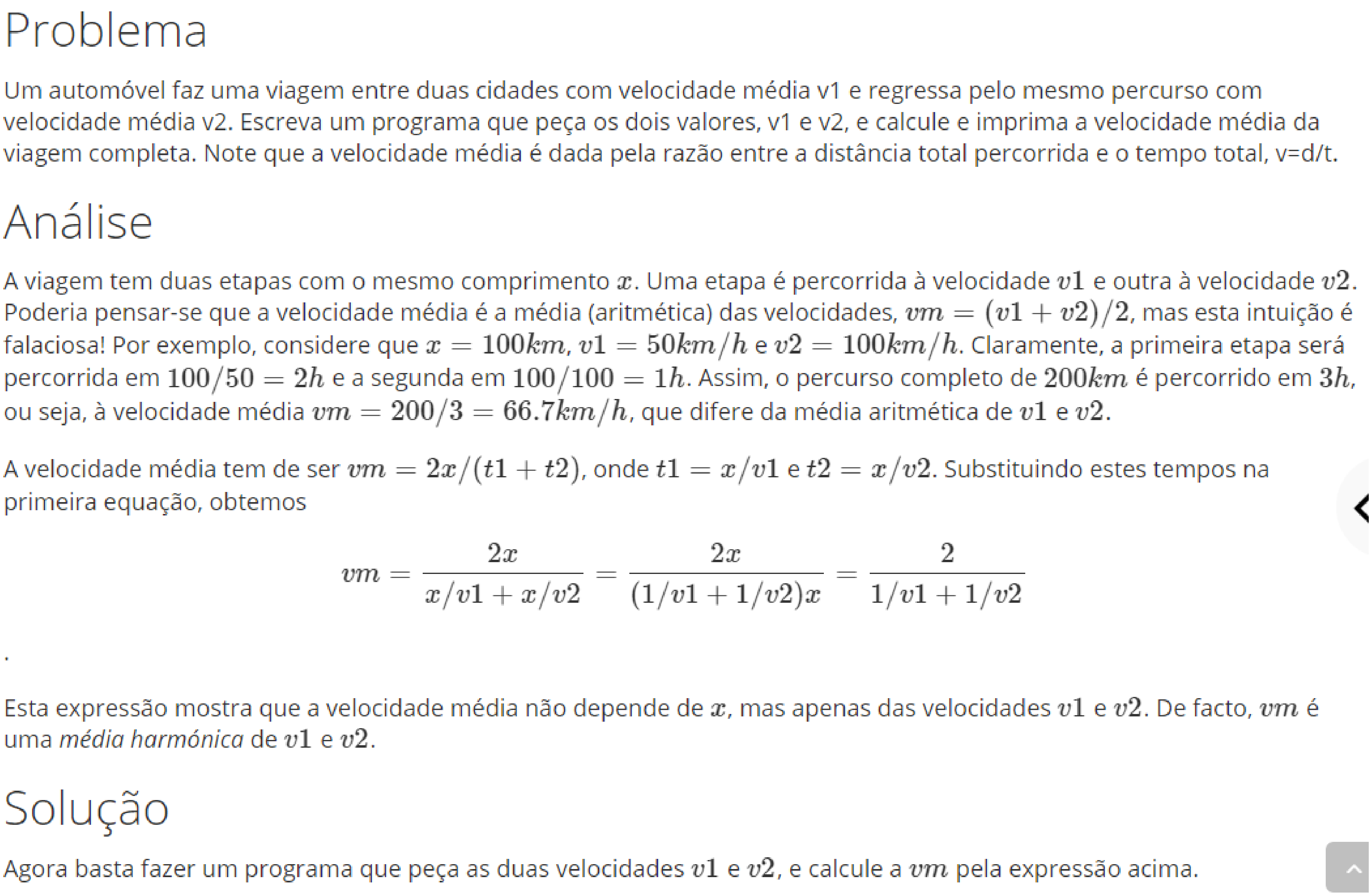
X at v2

V1 = x / t1

V2 = x / v2

Vm = dist / time = 2x / (t1 + t2)

https://elearning.ua.pt/mod/page/view.php?id=506629



Problema

Um automóvel faz uma viagem entre duas cidades com velocidade média v1 e regressa pelo mesmo percurso com velocidade média v2. Escreva um programa que peça os dois valores, v1 e v2, e calcule e imprima a velocidade média da viagem completa. Note que a velocidade média é dada pela razão entre a distância total percorrida e o tempo total, v=d/t.

Análise

A viagem tem duas etapas com o mesmo comprimento x. Uma etapa é percorrida à velocidade $v1$ e outra à velocidade $v2$. Poderia pensar-se que a velocidade média é a média (aritmética) das velocidades, $vm = (v1 + v2)/2$, mas esta intuição é falaciosa! Por exemplo, considere que $x = 100km$, $v1 = 50km/h$ e $v2 = 100km/h$. Claramente, a primeira etapa será percorrida em $100/50 = 2h$ e a segunda em $100/100 = 1h$. Assim, o percurso completo de $200km$ é percorrido em $3h$, ou seja, à velocidade média $vm = 200/3 = 66.7km/h$, que difere da média aritmética de $v1$ e $v2$.

A velocidade média tem de ser $vm = 2x/(t1 + t2)$, onde $t1 = x/v1$ e $t2 = x/v2$. Substituindo estes tempos na primeira equação, obtemos

$$vm = \frac{2x}{x/v1 + x/v2} = \frac{2x}{(1/v1 + 1/v2)x} = \frac{2}{1/v1 + 1/v2}$$

.

Esta expressão mostra que a velocidade média não depende de x, mas apenas das velocidades $v1$ e $v2$. De facto, vm é uma *média harmónica* de $v1$ e $v2$.

Solução

Agora basta fazer um programa que peça as duas velocidades $v1$ e $v2$, e calcule a vm pela expressão acima.

https://elearning.ua.pt/mod/page/view.php?id=506629