



# AMPATRÒNIC

23 DE FEBRER 2019



# python.org

Welcome to Python.org - Mozilla Firefox

Welcome to Python.org x +

Python Software Foundation (US) | https://www.python.org | 140%

Python PSF Docs PyPI Jobs Community

python™

Donate Search GO Socialize

About Downloads Documentation Community Success Stories News Events

```
# Python 3: List comprehensions
>>> fruits = ['Banana', 'Apple', 'Lime']
>>> loud_fruits = [fruit.upper() for fruit in fruits]
>>> print(loud_fruits)
['BANANA', 'APPLE', 'LIME']

# List and the enumerate function
>>> list(enumerate(fruits))
[(0, 'Banana'), (1, 'Apple'), (2, 'Lime')]
```

### Compound Data Types

Lists (known as arrays in other languages) are one of the compound data types that Python understands. Lists can be indexed, sliced and manipulated with other built-in functions.

[More about lists in Python 3](#)

1 2 3 4 5

Python is a programming language that lets you work quickly and integrate systems more effectively. [>>> Learn More](#)

**Get Started**  
Whether you're new to programming or an experienced developer, it's

**Download**  
Python source code and installers are available for download for all

**Docs**  
Documentation for Python's standard library, along with tutorials

**Jobs**  
Looking for work or have a Python related position that you're trying to

# Python for Beginners

<https://youtu.be/yE9v9rt6ziw>

- 0:00:00** Python Crash Cours
- 0:00:19** The world's fastest growing programming language
- 0:00:34** Ideal programming language to learn first (kids)
- 0:00:38** 6 reasons making Python so special: Solve complex problems in less time with fewer lines of code; Python is multi-purpose (data analysis, **AI/ML**, mobile apps, web apps...); and 4 more: High-level, cross-platform, huge community, large ecosystem
- 0:02:38** Versions: Python 2 (until 2020), **Python 3**
- 0:03:23** Download and install: **<http://python.org>** **Python 3.7**
- 0:05:17** Python Interpreter

- 0:07:14** Code Editors: Editor or IDE (+autocompletion, debugging, testing)
- 0:07:50** Tools: Editors: **VSCode**, Atom, Sublime; IDE: PyCharm
- 0:08:27** VSCode installation: **code.visualstudio.com**
- 0:08:34** Your First Python Program
- 0:12:13** Python Extension (to VS Code) : linting, debugging, autocompletion, code formatting, unit testing and code snippets
- 0:15:07** Linting Code
- 0:19:23** Formattig Code: Python Enhancement Proposals (PEPs), PEP8
- 0:23:19** Running Python Code: Code Runner
- 0:26:20** Quiz: What is an expression? What is a syntax error? What does a linter do?

**0:28:00** *Complete PYTHON course*

**0:28:10** Real-world scenarios: data analysis and visualization, files/folders, spreadsheets, PDFs, data compression, sending emails, web crawling, consuming APIs, ...

**0:28:45** Variables

**0:31:51** Variable names (descriptive, meaningful, ?)

**0:34:56** Strings

**0:40:27** Escape Sequences

**0:43:10** Formatted Strings

**0:45:20** String Methods

**0:51:17** Numbers

**0:54:06** Working with Numbers

**0:56:17** Type Conversion

**1:01:23** Quiz: What are the primitive types in Python? ...

- 1:04:09** Comparison Operators
- 1:06:15** Conditional Statements
- 1:10:26** Ternary Operator
- 1:12:37** Logical Operators
- 1:16:42** Short-circuit Evaluation
- 1:18:50** Chaining Comparison Operators
- 1:20:14** Quiz
- 1:22:00** For Loops
- 1:25:40** For..Else
- 1:28:28** Nested Loops
- 1:31:14** Iterables
- 1:34:24** While Loops
- 1:39:25** Infinite Loops
- 1:41:05** Exercise

- 1:43:12** Functions
- 1:45:38** Arguments
- 1:48:00** Types of Functions
- 1:52:05** Keyword Arguments
- 1:54:08** Default Arguments
- 1:55:45** \*args, wait, what?
- 2:00:00** \*\*args
- 2:02:24** Scope
- 2:07:35** Debugging
- 2:14:28** VSCode Tricks (Windows)
- 2:16:51** VSCode Tricks (Mac)
- 2:18:42** Exercise: FizzBuzz

# Exercicis interactius

```
> python3
2+2
4
# coixinet = inici de comentari
# + i - s'avaluen d'esq a dreta
1-2+3
2
# parèntesis per a altre ordre
1-(2+3)
-4
# multiplicació:
2*3
6
# divisió:
3/2
1.5
# divisió sencera:
3//2
1
# resta divisió sencera (mòdul):
3%2
1
# exponenciació:
3**2
9
# * i / més prioritaris q + i -
1+2*3
7
# parèntesis si cal o dubtem
(1+2)*3
9
```

```
# tipus sencer
2+3
5
# tipus real
2.0+3
5.0
# tipus booleà: False i True
# operador not
not False
True
# operador and
True and False
False
# operador or
True or False
True
# prioritat: not > and > or
# parèntesis si cal o dubtem
not True and False
False
not (True and False)
True
# == i !=
2==3
False
2!=3
True
# <, <=, > i >=
2<3
True
```



```

# variables i assignacions
x=2
x+2
4

# cadenes
'Alfons'
'Alfons'
"David"
'David'

# algunes funcions predefinides:
abs(-3)
3
float(3)
3.0
int(1.9)
1
str(1.9)
'1.9'
round(1.9)
2
ord('a')
97
ord('b')
98
'a' < 'b'
True
'aa' < 'ab'
True
chr(97)
'a'

```

```

# mòdul math
from math import *
pi
3.141592653589793
sin(pi/2)
1.0
cos(pi)
-1.0
tan(pi/4)
0.9999999999999999
e
2.718281828459045
exp(1.0)
2.718281828459045
ceil(1.1)
2
floor(1.9)
1
log(2.7183)
1.0000066849139877
log10(100)
2.0
sqrt(36)
6.0
# mètodes (funcions especials)
cadena='Escola Gavina'
cadena.lower()
'escola gavina'
cadena.upper()
'ESCOLA GAVINA'
# format
'pi és igual a {0:.4f}'.format(pi)
'pi és igual a 3.1416'

```

# Programes

```
holamon.py  
print('Hola món!')
```

```
> python3 holamon.py  
Hola món!
```

```
perimetre_cercle.py  
from math import pi  
cadena = input('Radi?: ')  
r = float(cadena)  
p = 2*pi*r  
print('Perímetre = {0:.2f}'.format(p))
```

```
> python3 perimetre_cercle.py  
Radi?: 1  
Perímetre = 6.28
```

```
# if cond: accions [else: accions]
```

```
senar.py  
n=int(input('Nombre?: '))  
if n%2==1:  
    print(f'{n} és senar')  
else:  
    print(f'{n} no és senar')
```

```
> python3 senar.py  
Nombre?: 1  
1 és senar  
> python3 senar.py  
Nombre?: 2  
2 no és senar
```

```
# while cond: accions
```

```
while.py  
i=0  
while i<3:  
    print(i)  
    i=i+1  
print('Au')
```

```
> python3 while.py  
0  
1  
2  
Au
```

```
# for variable in llista: accions
```

```
for.py  
for n in ['Manel', 'Nora', 'David']:  
    print(f'Hola {n}!')
```

```
> python3 for.py  
Hola Manel!  
Hola Nora!  
Hola David!
```