

# Відмінності між python2 та python3

---

Марія Любарська

19 листопада 2020

# Коротка історія python2 та python3

- python 2.0 випущений у 2000
- остання версія, python 2.7, була випущена у 2010
- **1 січня 2020 р. зупинено підтримку python2**
  - фінальний реліз 2.7.18 вийшов у 20 квітня 2020 р.
- python 3.0 випущений у 2008
- остання випущена версія - python 3.9

[https://en.wikipedia.org/wiki/History\\_of\\_Python](https://en.wikipedia.org/wiki/History_of_Python)

3.5	3.5.10 <sup>[55]</sup>	2015-09-13 <sup>[55]</sup>	2017-08-08 <sup>[56]</sup>	2020-09-30 <sup>[55]</sup>
3.6	3.6.12 <sup>[57]</sup>	2016-12-23 <sup>[57]</sup>	2018-12-24 <sup>[b][57]</sup>	2021-12 <sup>[57]</sup>
3.7	3.7.9 <sup>[58]</sup>	2018-06-27 <sup>[58]</sup>	2020-06-27 <sup>[b][58]</sup>	2023-06 <sup>[58]</sup>
3.8	3.8.6 <sup>[59]</sup>	2019-10-14 <sup>[59]</sup>	2021-04 <sup>[59]</sup>	2024-10 <sup>[59]</sup>
3.9	3.9.0 <sup>[60]</sup>	2020-10-05 <sup>[60]</sup>	2022-05 <sup>[61]</sup>	2025-10 <sup>[60][61]</sup>
3.10		2021-10-25 <sup>[62]</sup>	2023-05 <sup>[62]</sup>	2026-10 <sup>[62]</sup>

**Legend:** ■ Old version ■ Older version, still maintained ■ Latest version ■ Latest preview version ■ Future release

# Основні (найбільш помітні) відмінності

Basis of comparison	Python 3	Python 2
Release Date	2008	2000
Function print	<code>print("hello")</code>	<code>print "hello"</code>
Division of Integers	Whenever two integers are divided, you get a float value	When two integers are divided, you always provide integer value.
Unicode	In Python 3, default storing of strings is Unicode.	To store Unicode string value, you require to define them with "u".
Syntax	The syntax is simpler and easily understandable.	The syntax of Python 2 was comparatively difficult to understand.
Rules of ordering Comparisons	In this version, Rules of ordering comparisons have been simplified.	Rules of ordering comparison are very complex.

Basis of comparison	Python 3	Python 2
Iteration	The new Range() function introduced to perform iterations.	In Python 2, the xrange() is used for iterations.
Exceptions	It should be enclosed in parenthesis.	It should be enclosed in notations.
Leak of variables	The value of variables never changes.	The value of the global variable will change while using it inside for-loop.
Backward compatibility	Not difficult to port python 2 to python 3 but it is never reliable.	Python version 3 is not backwardly compatible with Python 2.
Library	Many recent developers are creating libraries which you can only use with Python 3.	Many older libraries created for Python 2 is not forward-compatible.

<https://www.guru99.com/python-2-vs-python-3.html>

# print

## python2

print це директива, працює з дужками та без дужок:

```
print 'Hello, World!'
```

```
print('Hello, World!')
```

## python3

print це функція, працює тільки з дужками:

```
print('Hello, World!')
```

# Ділення для типу int

python2

ділення інтеджерів - це ділення націло

$$3 / 2 = 1$$

$$3 / 2.0 = 1.5$$

python3

результат ділення інтеджерів - float:

$$3 / 2 = 1.5$$

$$3 / 2.0 = 1.5$$

# Кодування стрічок

## python2

за замовчуванням ASCII, потрібно  
явно вказувати кодування у Unicode

```
'this string is ASCII'
```

```
u'this string is Unicode'
```

```
unicode('this string is also Unicode')
```

тип `byte` еквівалентний типу `str`,  
але `bytearray` - окремий тип

## python3

за замовчуванням Unicode

```
'this string is unicode'
```

два класа `byte`: `byte` та `bytearray`

# xrange та range

## python2

xrange() - генерація чисел “на ходу”, швидше ніж range коли потрібно пройти цикл тільки один раз

range() - створюється у пам’яті один раз

## python3

range() імплементовано як xrange() у python2

xrange() не існує

# Виключення (exceptions)

## python2

створення виключення працює з дужками або без дужок

```
raise IOError, "file error"
```

```
raise IOError("file error")
```

## python3

створення виключення працює тільки з дужками

```
raise IOError("file error")
```



# next() функція та .next() метод

## python2

можна використовувати next() як функцію або як метод

```
my_generator = (letter for letter in 'abcdefg')
```

```
next(my_generator)
```

```
my_generator.next()
```

## python3

next() можна використовувати лише як функцію

```
my_generator = (letter for letter in 'abcdefg')
```

```
next(my_generator)
```

# Локальність змінних у циклі for

## python2

змінні циклу могли потрапляти в простір глобальних змінних

```
print 'before: i =', i  
print 'comprehension: ', [i for i in range(5)]  
print 'after: i =', i
```

```
before: i = 1
```

```
comprehension: [0, 1, 2, 3, 4]
```

```
after: i = 4
```

## python3

змінні циклу більше не можуть потрапляти в простір глобальних імен

```
print 'before: i =', i  
print 'comprehension: ', [i for i in range(5)]  
print 'after: i =', i
```

```
before: i = 1
```

```
comprehension: [0, 1, 2, 3, 4]
```

```
after: i = 1
```

# input() та raw\_input()

## python2

input() не обов'язково повертає  
значення типу str

потрібно використовувати raw\_input()  
щоб уникнути такої поведінки

## python3

input() завжди повертає значення типу  
str

# Правила округлення

## python2

використовує стандартне  
математичне округлення:

неціле число  $X.5$  округлюється до  
найближчого більшого цілого числа

## python3

використовує “банківське  
округлення”:

неціле число  $X.5$  округлюється до  
найближчого парного числа

дозволяє уникнути накопичення  
похибок унаслідок “зсуву” до більших  
чисел

# Keyword only arguments

python2

python3

можна створювати аргументи, які  
можна змінити тільки за ім'ям:

```
def f(a, b, *args, option=True):
```

```
...
```

або

```
def f(a, b, *, option=True):
```

```
...
```

# Advanced unpacking

## python2

```
a, b = range(2)
```

## python3

```
a, b = range(2)
```

```
a, *range, b = range(10)
```

```
a, b, *range = range(10)
```

```
*range, a, b = range(10)
```

# Джерела

1. [https://sebastianraschka.com/Articles/2014\\_python\\_2\\_3\\_key\\_diff.html](https://sebastianraschka.com/Articles/2014_python_2_3_key_diff.html)
2. <https://docs.python.org/3/whatsnew/3.0.html>
3. <https://www.asmeurer.com/python3-presentation/slides.html>
4. <https://learntocodewith.me/programming/python/python-2-vs-python-3/#history-of-python2-vs-3>