

Основные отличия Python2 и Python3

Шестакова Анна Николаевна

МГУ имени М. В. Ломоносова, факультет ВМК, кафедра ММП

21 ноября 2017 г.

Хронология

- 1991: вышла первая версия Python
- 2000: выпуск Python 2.0
- 2008: выпуск Python 3.0
- 2010: выпуск Python 2.7
- 2017: выпуск Python 3.7
- 2020: окончание поддержки Python 2.7

Print — Функция

```
Python2: print "The answer is", 2*2  
Python3: print("The answer is", 2*2)
```

```
Python2: print x,  
Python3: print(x, end=" ")
```

```
Python2: print  
Python3: print()
```

```
Python2: print >>sys.stderr, "fatal error"  
Python3: print("fatal error", file=sys.stderr)
```

```
Python2: print (x, y)  
Python3: print((x, y))
```

Print — Функция

Также в Python3 возможно настроить разделитель между элементами, например:

```
>>> print("There are <", 2**32, "> possibilities!", sep="")  
There are <4294967296> possibilities!
```

Функция `print()` не поддерживает особенность «программный пробел» («softspace») старого оператора `print`:

Python 2:

```
>>> print "A\n", "B"
```

A

B

Python 3:

```
>>> print("A\n", "B")
```

A

B

Целые числа

- В Python3 тип `long` переименован в `int`.
- В Python3 выражение вида $1/2$ возвращает `float`. Для отсечения дробной части используется $1//2$.
- Константа `sys.maxint` была удалена, так как не существует предела значений целых чисел.
- Восьмеричные литералы имеют формы вида `0720` в Python2 и `0o720` в Python3.

Unicode

- По умолчанию Python2 использует алфавит ASCII, а Python3 – Unicode.
- В Python3 любая попытка комбинировать текст (тип `str`) и данные (тип `bytes`) в Python 3.0 поднимает `TypeError`, в то время как в Python2 комбинирование возможно, если 8-битная строка содержала только 7-битные (ASCII) символы, но вы получите `UnicodeDecodeError`, если она содержит не-ASCII символы.
- `str.encode()` и `bytes.decode()` позволяют перейти от `str` к `bytes` и обратно;
- `str` и `bytes` — неизменяемые типы. Существует отдельный изменяемый тип для двоичных данных `bytearray`.
- `"\u20ac"` это строка из 6 символов в Python3, в то время как в Python2 это один символ.

Еще несколько отличий

- Python2: функция `next(generator)` и метод `generator.next()`
Python3: только функция `next(generator)`
- В Python3 переменные цикла `for` больше не просачиваются в глобальное пространство имен:

Python2:

```
>>> i = 1
>>> print'before: i =', i
>>> print[i for i in range(5)]
>>> print'after: i =', i
```

```
before: i = 1
[0, 1, 2, 3, 4]
after: i = 4
```

Python3:

```
>>> i = 1
>>> print('before: i =', i)
>>> print([i for i in range(5)])
>>> print('after: i =', i)
```

```
before: i = 1
[0, 1, 2, 3, 4]
after: i = 1
```

Еще несколько отличий

- Многие методы возвращают отображения (Views) и итераторы вместо списков: `dict.keys()`, `dict.items()`, `map()`, `filter()`, `zip()` и др.
- В Python3 операторы сравнения (`<`, `<=`, `>=`, `>`) поднимают исключение `TypeError`, когда операнды не упорядочиваемы.
- Обработка исключений:

Python2:

```
try:
    let_us_cause_a_NameError
except NameError, err:
    ...
```

Python3:

```
try:
    let_us_cause_a_NameError
except NameError as err:
    ...
```


Ссылки

<https://docs.python.org/3/whatsnew/3.0.html>

<https://pythonworld.ru/osnovy/python2-vs-python3-razlichiya-sintaksisa.html>