

# Python 3 vs Python 2

A common guide to the differences

by

Jerry Davis

# Objects

- Old style classes are gone
- New style classes inherit from **object** class
- `__cmp__()` and `cmp()` are gone
  - use `functools.total_ordering` instead
- `.next()` method is gone, use builtin `next()` instead

# dictionaries

- the python 2 methods: `iterkeys()`, `itervalues()` and `iteritems()` are gone.
  - use `keys()`, `values()` and `items()`
- the `has_key()` method on dictionaries is gone.
  - Use the `in` operator instead.

# lists

- xrange() does not exist
  - use range()
- functions in python 2 that returned lists may well return views or iterators instead, so beware.
- for instance:
  - range(10) returns an iterator instead of a list.
  - Use **list(range(10))** instead, to get the same behavior

# exceptions

- python 2:
  - `except (Exception1, Exception2), target:`
- python 3:
  - `except (Exception1, Exception2) as target:`

# various

- The `execfile` statement is gone
  - use: `exec(open(afile).read())` instead
- `raw_input()` is gone.
  - use: `input()`
- `long` is gone, just use `int`
- integer division is now different
  - python 2:  $1/2 \Rightarrow 0$ , python 3:  $1/2 \Rightarrow 0.5$
  - python 3:  $1//2 =$  python 2's behavior

# the big ones that bite you

- print statement is gone, use print() function instead
- Python 2 had two string types; str and unicode. Python 3 has only one: str, but instead it also has a bytes type made to handle binary data.
- see:
  - <http://python3porting.com/problems.html#binary-section>
  - <http://python3porting.com/noconv.html#unicode-section>