

# 2to3

porting summarized - issues, workflows and solutions

# intro / me

- Anime-Lover
- Monaco-Lover
- concurrency,  
version control,  
information  
security
- been working on  
python for a  
little over an  
year



# today we`ll be discussing

- how different is py3k?
- d2, configobj
- suitable workflow
- how to contribute
- few changes in detail





how different are they???

# LOTS OF SMALL DIFFERENCES!!!

- <http://bit.ly/djYOVa>



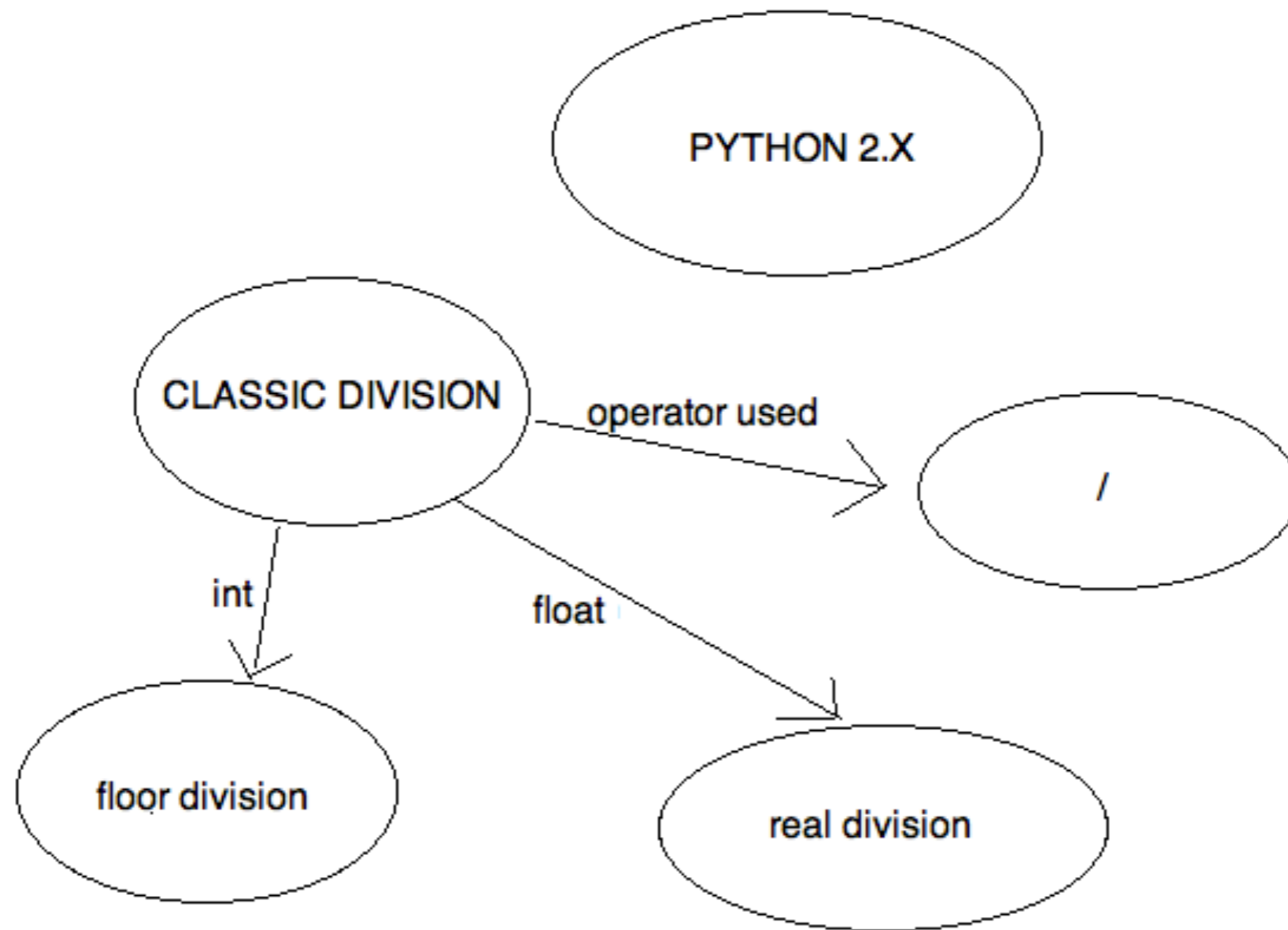
# a `_few_` of them ...

- `dict.keys()`, `dict.values()`, `dict.items()` return views -> not lists
  - trivially solved using `list(dict.keys())`
  - not thread-safe in python3k
  - use `sorted(d)`

# a `_few_` of them ...

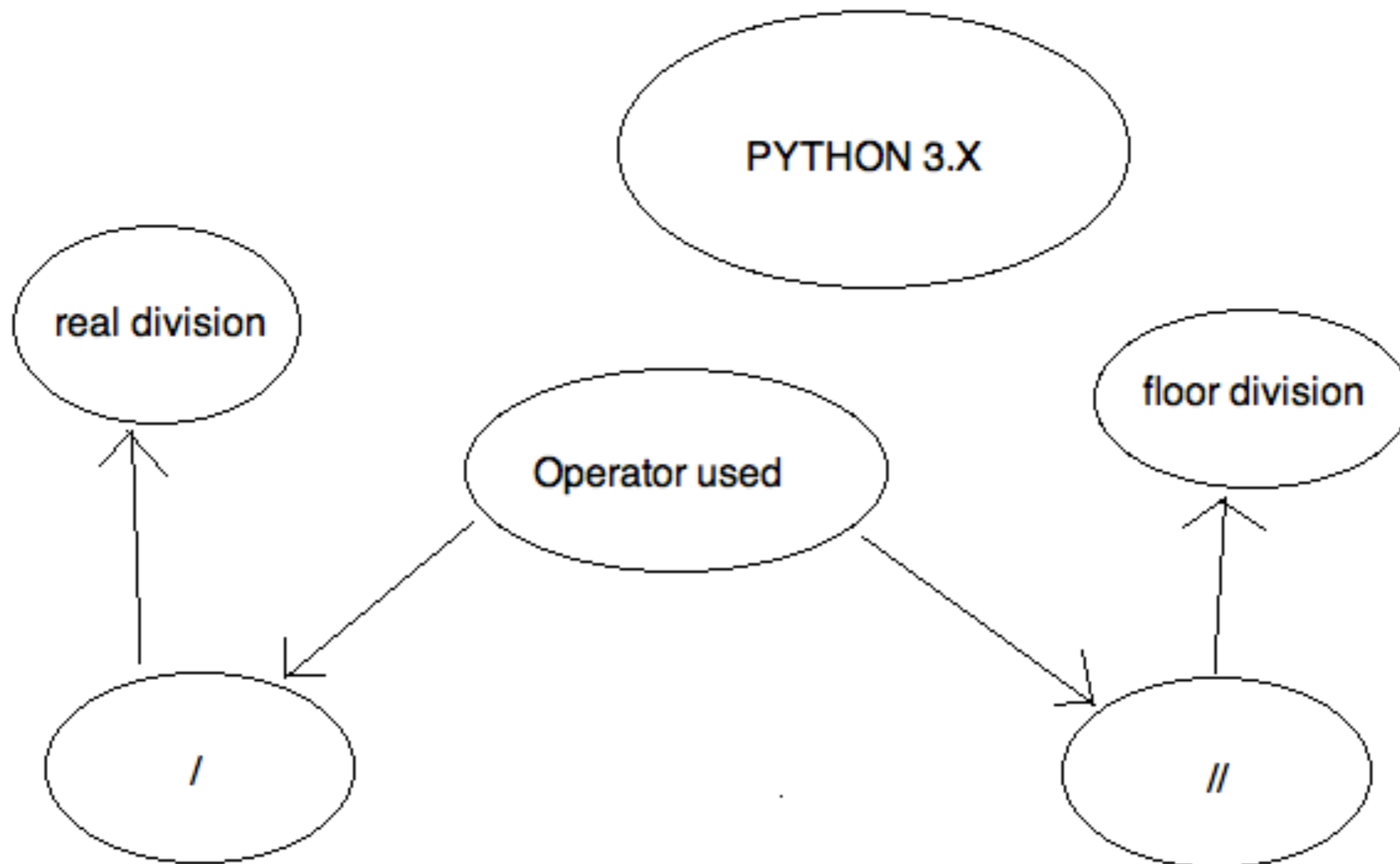
- `print` becomes `print()`  
[PEP 3105]
- `map()`, `filter()`, `zip()` return iterators
- `range()` behaves like `xrange()`

# one big change





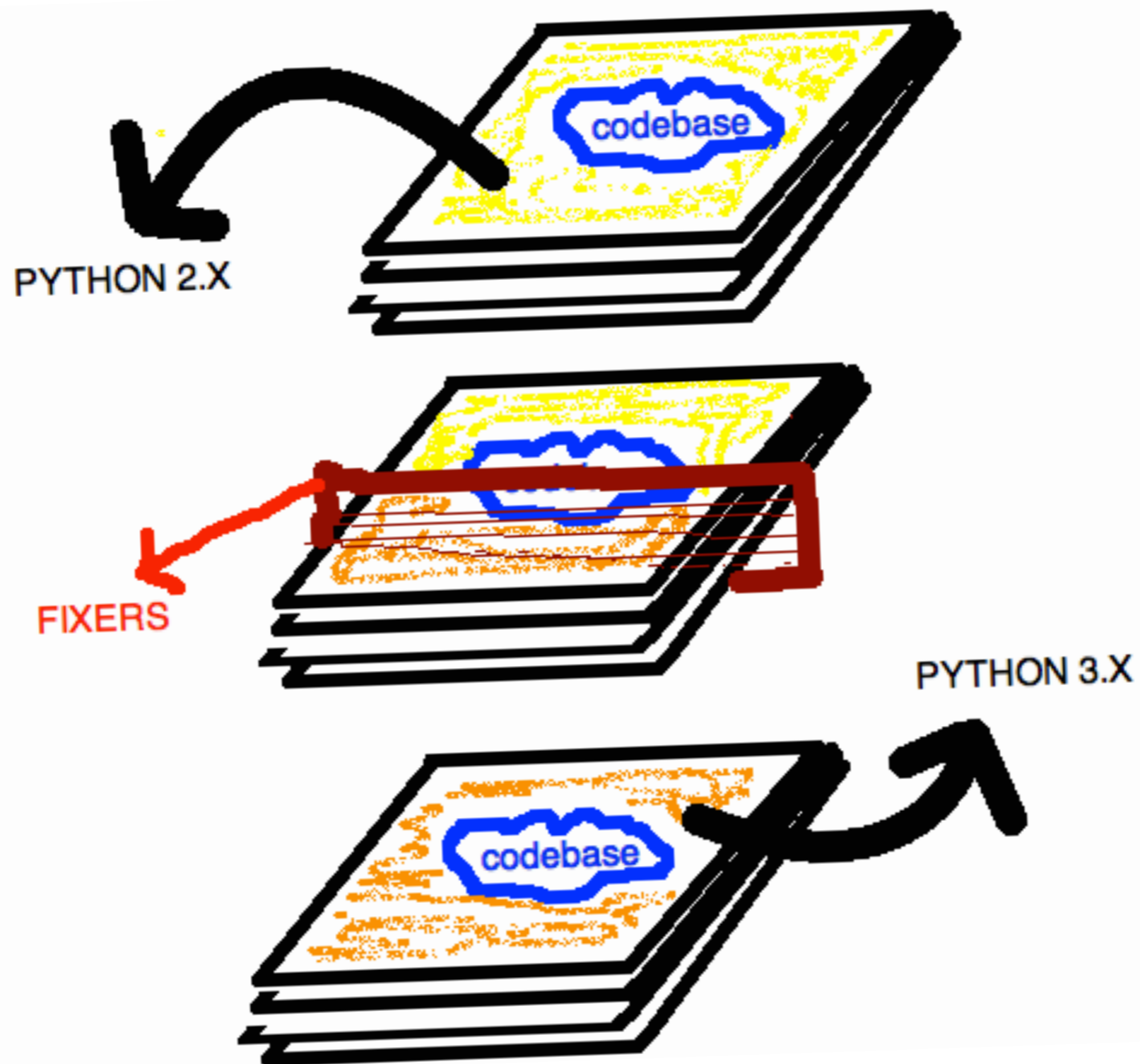
# division in py3



# and a few major ones ...

- All strings are UNICODE
- (int, long) -> int

# 2to3 - the tool



# using 2to3

- `$ 2to3 -w myfile.py`

# using 2to3 (in ur code)

- ```
>>> from lib2to3.refactor import  
RefactoringTool, get_fixers_from_package
```
- ```
>>> std_fixers = get_fixers_from_package  
(‘lib2to3.fixes’)
```
- ```
>>> r = RefactoringTool(std_fixers)
```
- ```
>>> r.refactor([list_of_files], write=True)
```

porting experiences so  
far

# porting d2

- more than 60 files -> 7200 lines approx.
- d2 has the same codebase for 2.x and 3.x(as there is some major dev going on)
- run 2to3 as part of the build process
- make sure the tests work. fix stuff if they don`t.



# porting configobj

- converted the repo over to mercurial(from svn)
- ran tests (which failed)
  - work not complete! help is appreciated!
  - <http://bitbucket.org/zubin71/configobj>

# single vs. multiple codebases

- Single codebase
  - apply bug-fixes once
  - codebase becomes messier as you try to increase compatibility with different version

# single vs. multiple codebases

- Multiple codebase
  - neater codebase
  - apply your bug-fixes on two branches
  - you probably have to do this if you intend to re-design your APIs

# single vs. multiple codebases

- expecting rapid development soon?
- use a single branch for now, use another branch when you feel you have enough time to apply patches to 2 branches

# Unicode and Py3k

long  $\Rightarrow$  integer

# helping the community



- stdlib/3rd party library
- web-frameworks
- python-porting  
ML
- PSF sponsored sprints



# AACK! warning!

- DO NOT use python3.0
  - use python3.1
  - python3.2 alpha is out. Final release scheduled for Jan 15, 2011

doubts, anyone?

Q & A ?

no question off limits

well, c u around then!

Have Fun at  
Pycon! :-)