

Week 03M: Modifying & Rewriting Software

Thuy Vu

- Office Hours
 - Tuesdays and Thursdays, 9:00 AM - 9:50 AM at BH 2432
- Assignment 3
 - Modifying and rewriting software
 - web.cs.ucla.edu/classes/winter17/cs35L/assign/assign2.html
 - Time due 23:55 next Saturday, January 28
- Assignment 10 Sign-up

- Syntax:

```
target: dependencies
      system command
```

- Example:

```
all:  hello

hello: main.o factorial.o hello.o
      g++ main.o factorial.o hello.o -o hello

main.o:  main.cpp
      g++ -c main.cpp

factorial.o:  factorial.cpp
      g++ -c factorial.cpp

hello.o:  hello.cpp
      g++ -c hello.cpp

clean:

      rm *o hello
```

- Syntax:

```
target:  dependencies
        system command
```

- Example:

```
CC=g++
all:  hello
hello: main.o factorial.o hello.o
        $(CC) main.o factorial.o hello.o -o hello
main.o:  main.cpp
        $(CC) -c main.cpp
factorial.o:  factorial.cpp
        $(CC) -c factorial.cpp
hello.o:  hello.cpp
        $(CC) -c hello.cpp
clean:
        rm *.o hello
```

- 1 **dynamic type** system
 - duck-typing
- 2 **indentation** — spacing characters make all the difference (no braces!)
- 3 string: `*`, `+`, ... (docs.python.org/2/library/string.html)
- 4 list: `[2, 3.5, 'a', "h"]` (docs.python.org/2/tutorial/datastructures.html)
 - **dynamic & heterogeneous**
- 5 comment: `#` or `''' ... '''`
- 6 **tuple**: `(1,2,3,4)`
- 7 set: `set("abcbca")`, (docs.python.org/2/library/sets.html)
 - `len(list(set(['a', 'b', 'd', 'a', 'd', 'c'])))`
- 8 dictionary: `dict={'a':10, 'z':50}`; `values()`, `keys()`, `items()`, ...
- 9 control-flow: `if`, `for`, ... (docs.python.org/2/tutorial/controlflow.html)
- 10 functions
- 11 main: https://docs.python.org/2/library/__main__.html

- Python 2

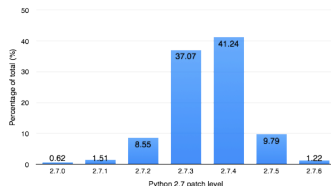
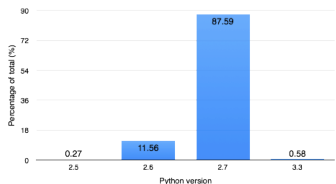
- 1 classic **classes** (up to Python 2.1)
 - old: `type(x)` is always `<type 'instance'>`
 - new: `type(x)` is typically the same as `x.__class__`
- 2 new-type classes (since Python 2.2)
 - new-style helps to subclass most built-in types
 - “descriptors” call `get`, `set`, `delete` upon access an attribute
- 3 **integer division** ($3/2 \rightarrow 1$)
- 4 confusing text model (`str` and `unicode`)

- “Python 3 by default” means

- 1 $3/2 \rightarrow 1.5$

- 2 no old-type

- 3 `str` and `byte`



still in progress due to high backward incompatibility

- ❶ drop support for Python 2 completely – **much easier!**
 - learn the difference; then use **2to3** translator
- ❷ single-source Python 2.7 and Python 3 – **straightforward!**
 - prepare a good test coverage
 - update your code with
 - Modernize makes Py-3 possible in Py-2 (back port bytes)
 - Futurize targets a subset of Py-3 and Py-2 (conservative)
 - Pylint to keep supporting Py-3 (using `--py3k`)
 - `pypi.python.org/pypi/caniusepython3`
- ❸ single-source Python 2.x and Python 3 – **...it depends**
 - `six` might be useful (`six = 2 × 3`)

<https://docs.python.org/3/howto/pyporting.html>

❶ randline.py returns n random lines from a_file

- `./randline.py -n 3 a_file`
- `python randline.py -n 3 a_file`

```
parser = OptionParser(version=version_msg, usage=usage_msg)
parser.add_option("-n", "--numlines",
                  action="store", dest="numlines", default=1,
                  help="output NUMLINES lines (default 1)")
```