

CSE 446 -- Python Review

John Kaltenbach, Patrick Spieker
Deric Pang, Jane Zhang, Andrew Yu

A Note About Windows

It's going to be tough for us to support Windows Python issues since most of the TAs don't have experience with it. If you have a Windows machine, try to use the lab computers as much as possible.

```
ssh username@umnak.cs.washington.edu
```

To run python3 on umnak, use: `python3` instead of `python`

Ask us about packages and libraries that aren't available on umnak.

Java vs. Python

Java

Statically typed

Compiled

Object oriented

Non-functional (Java 8 kind of)

Python

Dynamically (duck) typed

Interpreted

Object oriented scripting language

Functional

Hello, world!

Write a Python script in a plain text file.

Run it with `python script_name.py`

Style recommendations:

- 4 spaces for indentation
- Variables, filenames, and methods should be `lower_case_separated_by_underscores`
- Class names should be `CapsCase`
- Main method should be contained in conditional
- See: <https://www.python.org/dev/peps/pep-0008/>

Data Structures

- List: [1, 1, 2, 3], list() E.x. `a = [1, 2, 3], a.append(4), a.remove(4)`
- Set: {1, 2, 3}, set() E.x `a = {1, 2, 3}, a.add(4)`
- Tuple: ("a", "ab"), (1, 2, 3), (1, 3, 5, ..., 101) E.x `a = (1, 2, 3)`
`a[0] + a[1] + a[2] // 6`
- Dictionary: {"k1": "v1", "k2": "v2"}, dict() E.x `a = {"k1": "v1", "k2": "v2"}`
`a["k1"], a.k2`

Basic Syntax

```
a = []
```

```
for i in range(100):
```

```
    a.append(i)
```

```
print(a) // [0, 1, ..., 99]
```

```
b = [i + 100 for i in a]
```

```
print(b) // [100, 101, ..., 199]
```

```
c = {7, 3, 13, 7, 11, 2, 14}
```

```
for p in c:
```

```
    if p % 2 == 0 and p < 3:
```

```
        print(p)
```

```
// output 2
```

Installing Python Packages

- pip - Package manager
- Updating pip (If you have Python 2 \geq 2.7.9 or Python 3 \geq 3.4):
 - `pip install -U pip setuptools`
- Otherwise:
 - Download get-pip.py: <https://packaging.python.org/tutorials/installing-packages/>
 - Install pip with `python get-pip.py`
- Install package with pip using command line interface
 - `pip install <package-name>`
 - `pip install -r requirements.txt`
 - `pip install --upgrade <package-name>`
 - Read more: <https://pip.pypa.io/en/stable/>

Virtual Environments

Create isolated Python environments.

```
pip install virtualenv  
virtualenv <DIR>  
source <DIR>/bin/activate
```

On Windows

```
pip install virtualenv  
pip install virtualenvwrapper-win  
mkvirtualenv <DIR>
```


NumPy Array Creation Routines

Ones and zeros

<code>empty</code> (shape[, dtype, order])	Return a new array of given shape and type, without initializing entries.
<code>empty_like</code> (a[, dtype, order, subok])	Return a new array with the same shape and type as a given array.
<code>eye</code> (N[, M, k, dtype])	Return a 2-D array with ones on the diagonal and zeros elsewhere.
<code>identity</code> (n[, dtype])	Return the identity array.
<code>ones</code> (shape[, dtype, order])	Return a new array of given shape and type, filled with ones.
<code>ones_like</code> (a[, dtype, order, subok])	Return an array of ones with the same shape and type as a given array.
<code>zeros</code> (shape[, dtype, order])	Return a new array of given shape and type, filled with zeros.
<code>zeros_like</code> (a[, dtype, order, subok])	Return an array of zeros with the same shape and type as a given array.
<code>full</code> (shape, fill_value[, dtype, order])	Return a new array of given shape and type, filled with <i>fill_value</i> .
<code>full_like</code> (a, fill_value[, dtype, order, subok])	Return a full array with the same shape and type as a given array.

From existing data

<code>array</code> (object[, dtype, copy, order, subok, ndmin])	Create an array.
<code>asarray</code> (a[, dtype, order])	Convert the input to an array.
<code>asanyarray</code> (a[, dtype, order])	Convert the input to an ndarray, but pass ndarray subclasses through.
<code>ascontiguousarray</code> (a[, dtype])	Return a contiguous array in memory (C order).
<code>asmatrix</code> (data[, dtype])	Interpret the input as a matrix.
<code>copy</code> (a[, order])	Return an array copy of the given object.
<code>frombuffer</code> (buffer[, dtype, count, offset])	Interpret a buffer as a 1-dimensional array.
<code>fromfile</code> (file[, dtype, count, sep])	Construct an array from data in a text or binary file.
<code>fromfunction</code> (function, shape, **kwargs)	Construct an array by executing a function over each coordinate.
<code>fromiter</code> (iterable, dtype[, count])	Create a new 1-dimensional array from an iterable object.
<code>fromstring</code> (string[, dtype, count, sep])	A new 1-D array initialized from raw binary or text data in a string.
<code>loadtxt</code> (fname[, dtype, comments, delimiter, ...])	Load data from a text file.