PyLadies

“Python for Intermediates” Workshop

http://www.meetup.com/NYC-PyLadies/events/213064682/
Metis New York, 20141017

David Branner, Hacker School

We assume you have already seen

- Python functions
- basic Python syntax
- basic Python datatypes

and have practiced them.

Today we will introduce

- comprehensions
- timing
- generators

- exceptions
- try-except blocks
- testing with Pytest

Goals for the workshop:

- to be exposed to these six Python tools
- to play with them with your own hands
- to have a document containing these examples that you can consult later

We assume you have

- Python 3
- Ipython 2

and are able to play with them during the workshop.

List comprehension pattern:

```
lst = []
<for loop>:
    lst.append(<something>)
```

```
lst = [something <for loop>]
```

Nested list comprehension pattern:

```
lst = []
<for loop 1>:
    <for loop 2>:
        lst.append(<something>)
```

```
lst = [something
        <for loop 1>
        <for loop 2>]
```

Filtered list comprehension pattern (`if` only):

```
lst = []
<for loop 1>:
    <for loop 2>:
        if <condition>:
            lst.append(<something>)
```

```
lst = [something
        <for loop 1> <for loop 2>
        if <condition>]
```

Filtered list comprehension pattern (`if-else`):

```
lst = []
<for loop 1>:
    <for loop 2>:
        if <condition>:
            lst.append(<something>)
        else:
            lst.append(<something else>)
```

```
lst = [something
        if <condition>
        else <something else>
        <for loop 1> <for loop 2>]
```

Timeit pattern:

`timeit <one-liner>`
`timeit <one-liner> # in Ipython`

Generator pattern:

```
g = (something <for loop>)
next(g)
```

Exception pattern:

```
raise <Exception>

raise <Exception>('message')
```

Try-Except basic pattern:

```
try:
    <intended code block>
except <Exception>:
    <alternate code block>
```

Try-Except full pattern:

```
try:
    <intended code block>
except <Exception>:
    <alternate code block>
else:
    <do this if "try" block succeeds>
finally:
    <do this last in any case>
```

Basic testing pattern:

```
assert <condition>
```

Pytest basic pattern:

```
cd test
import pytest # install, perhaps with pip
pytest.main()
```

Other things to remember:

1. To get information about a module or built-in in Ipython:

```
<module>?
<built-in>?
```

(“object introspection”)

Other things to remember:

2. To view source-code for a function in Ipython:

```
<function>??
```

Other things to remember:

3. The Python documentation should be your regular reading:

- On line: http://docs.python.org/3.4/
- On Android: (free download) https://play.google.com/store/apps/details?id=com.jubad.pydocs330
- On iOS: (99¢) https://itunes.apple.com/us/app/python-documentation/id371454248?mt=8

劇終