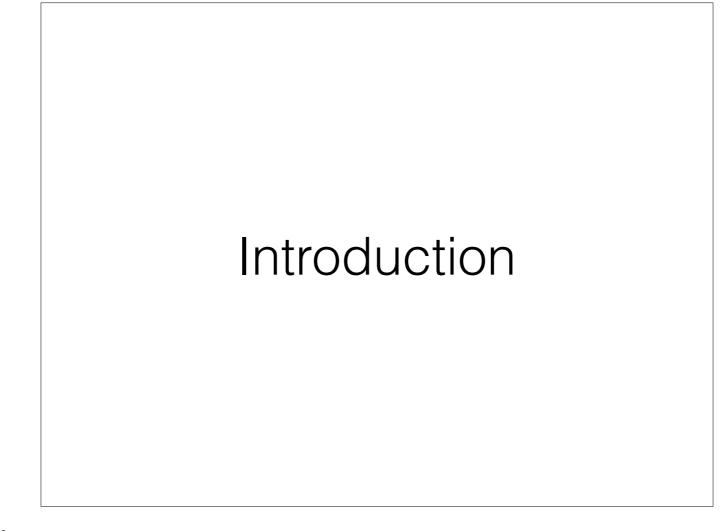
## Python Workshop

**DUCSS**, Trinity College Dublin

Allen Thomas Varghese 16-Feb-2017



- \* Brief introduction about yourself
- \* Introduction about what is going to be covered in the workshop

#### Setting up Workspace

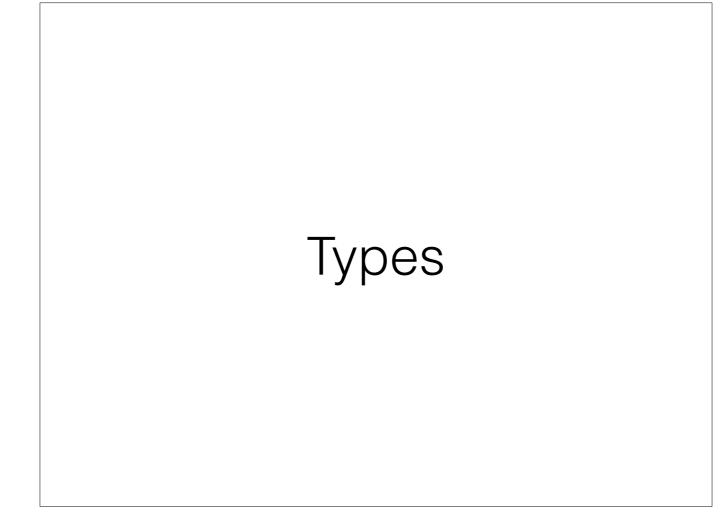
- \* Check installation of Py3.x
- \* Check Python in PATH
- \* Distribute installers in USB key
- \* IDE : Vim, Emacs, Atom, Sublime, PyCharm, Anaconda, Eclipse, IntellijIDEA

Hello World!

- \* Create a python file
- \* Add print command
- \* Modify the file to run using \_\_name\_\_ == '\_\_main\_\_'

#### Variables ····►Indentation Comments

- \* Variables
- \* White spacing
- \* Comments
  - \* Single line
  - \* Multi-line



- \* String concatenation, multiplication, accessing elements, splicing, split
- \* Integer
- \* Boolean
- \* Float
- \* Decimal from decimal import Decimal
- \* None

#### Operators

- \* Arithmetic => +, -, /, \*, \*\*, +=, -=, /=, \*=
- \* Logical operators => not, and, or
- \* Identity => is
- \* Comparison : >, <, >=, <=, ==, !=

#### Error Handling

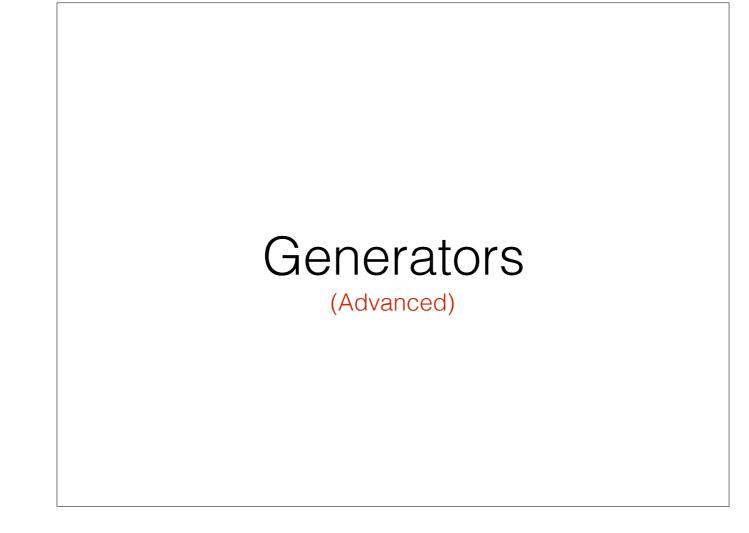
- \* try except
- \* try except finally
- \* try except else
- \* try except else finally

#### Control Flow

- \* if
- \* if else
- \* if elif else
- \* while
- \* for

#### **Functions**

- \* Single return value
- \* Multiple return values
- \* Anonymous functions : Lambda
- \* Default value arguments
- \* Arbitrary arguments



\* Simple function generator

#### Decorators

(Advanced)

- \* Simple decorator
- \* Decorator with arguments

#### Data Structures

- \* List len, append, pop, insert, extend, splicing, reverse, sorted
- \* Tuple len
- \* Dict update, del, keys, values, get
- \* Set intersection, union, addition, subtraction
- \* Factory functions list, tuple, dict, set

## List Comprehension

(Advanced)

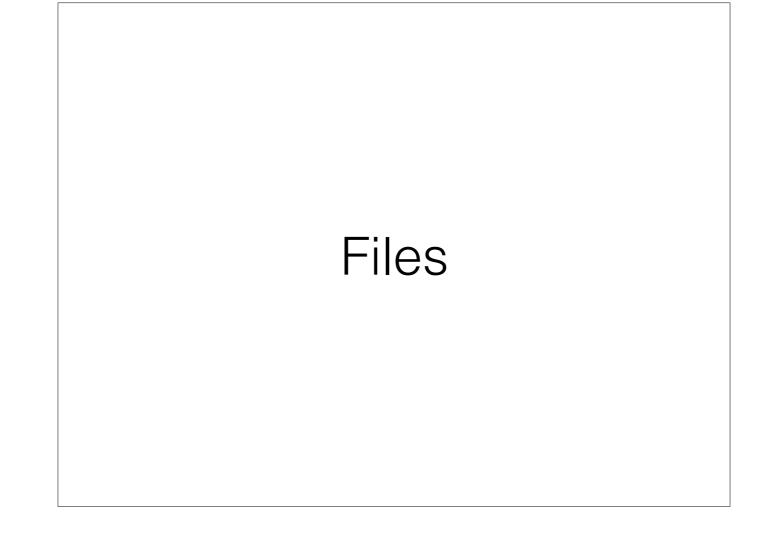
- \* Using for loop
- \* Using for and if
- \* Using generators

#### Classes

- \* Definition
- \* Constructor
- \* Attributes
- \* Methods
- \* Functions
- \* Scope public & private

# Magic Methods (Advanced)

Operator methods



- \* Use open() directly
- \* Use context manager (with)

## Packages

- File organization
- \* \_\_init\_\_.py\* "import" statement

math, sys & os

#### 3rd Party Library

- \* Try pip install
- \* Import the package shell & program
- \* Do something!