

# Types, Numerics, Strings, and Polymorphism

Info 206

Niall Keleher

31 August 2017

# Today's Outline

1. Quiz
2. Overview of Python
3. Python Types and Operations
4. Exercise repository
5. Individual assignment - working with numerics and strings
6. Collaborative exercise - Exploring Github repository



Today's Quiz: <http://bit.ly/2el7USd>

# Overview of Python

# Python

- Developed in the late 1980s - led by Guido van Rossum - released in 1991

# Python

- Developed in the late 1980s - led by Guido van Rossum - released in 1991
- High-level programming language

# Python

- Developed in the late 1980s - led by Guido van Rossum - released in 1991
- High-level programming language
- Interpreted language

# Python

- Developed in the late 1980s - led by Guido van Rossum - released in 1991
- High-level programming language
- Interpreted language
- Focus on readable code



# Python

- Developed in the late 1980s - led by Guido van Rossum - released in 1991
- High-level programming language
- Interpreted language
- Focus on readable code
- Object-oriented and Functional - good for learning core programming paradigms

# Python

- Developed in the late 1980s - led by Guido van Rossum - released in 1991
- High-level programming language
- Interpreted language
- Focus on readable code
- Object-oriented and Functional - good for learning core programming paradigms
- Dynamically typed - no need to declare object types

# Python

- Developed in the late 1980s - led by Guido van Rossum - released in 1991
- High-level programming language
- Interpreted language
- Focus on readable code
- Object-oriented and Functional - good for learning core programming paradigms
- Dynamically typed - no need to declare object types
- Efficient memory usage - "garbage collection", no need to declare the object type, less code

# Python

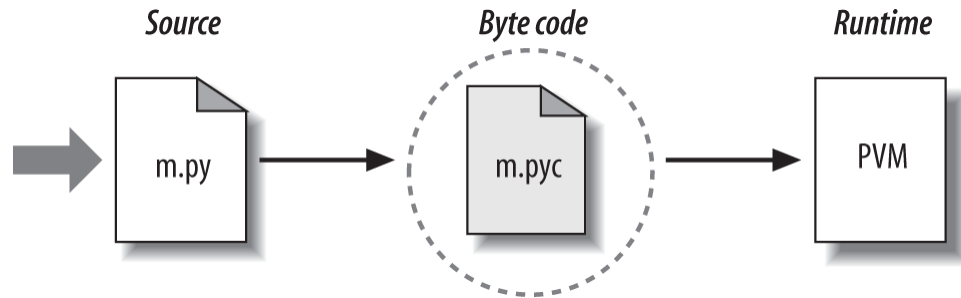
- Developed in the late 1980s - led by Guido van Rossum - released in 1991
- High-level programming language
- Interpreted language
- Focus on readable code
- Object-oriented and Functional - good for learning core programming paradigms
- Dynamically typed - no need to declare object types
- Efficient memory usage - "garbage collection", no need to declare the object type, less code
- Portability - works across platforms

# Python

- Developed in the late 1980s - led by Guido van Rossum - released in 1991
- High-level programming language
- Interpreted language
- Focus on readable code
- Object-oriented and Functional - good for learning core programming paradigms
- Dynamically typed - no need to declare object types
- Efficient memory usage - "garbage collection", no need to declare the object type, less code
- Portability - works across platforms
- Extensive libraries and support

# Python

- Developed in the late 1980s - led by Guido van Rossum - released in 1991
- High-level programming language
- Interpreted language
- Focus on readable code
- Object-oriented and Functional - good for learning core programming paradigms
- Dynamically typed - no need to declare object types
- Efficient memory usage - "garbage collection", no need to declare the object type, less code
- Portability - works across platforms
- Extensive libraries and support
- Python 2.x vs. 3.x



# How Python runs a program

# Running code in Python

- running python interactively
- writing (module) scripts to run in python
- Working within IDLE or Notebook (e.g. Jupyter Notebook)



# Built-in Python object types

- ***Numerics***
  - Float
  - Int
  - Fraction
  - Decimal
- ***Strings***
- Lists
- Dictionaries
- Tuples
- Files

# Dynamic Typing and Polymorphism

# Dynamic Typing

- types are determined automatically at runtime
- No need to declare variables ahead of time
- variables are generic, types are associated with the object
- variables (names) are created at time of assignment and become objects
- using the variable (name) later in the code references the object

# Variables, Objects, References

**Variables** - entries in a system table, with spaces for links to objects

**Objects** - pieces of allocated memory with enough space to represent the values for which they stand

**References** - pointers from variables to objects

# Types are associated with objects, *not* variables

```
a = 3  
a = "info206"  
a = 3.141592653589793
```

# Shared References

```
a = 3  
b = a  
a = "Thursday"
```

# Polymorphism

The meaning of the operation depends on the objects being operated on

```
3*10
```

```
3*"Huge! "
```

```
"Hello " + "World!"
```

```
3.50 + 17
```

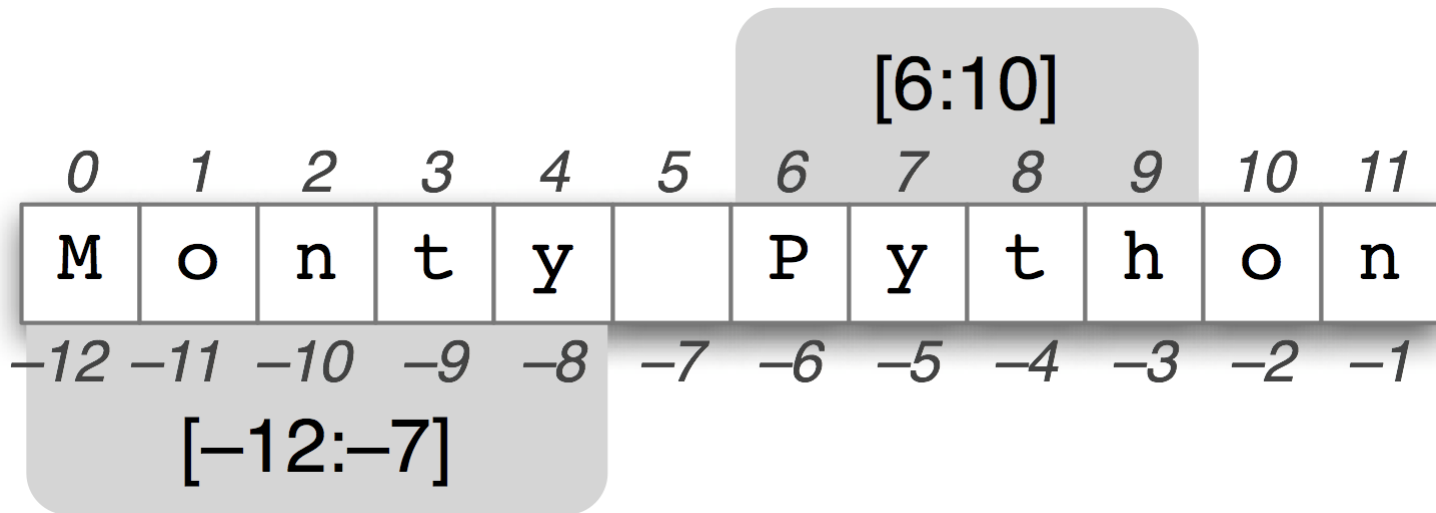
```
print("-" * 50) # Commonly used form in logging
```

# Brief note on mutability

## Strings are immutable sequences:

- The characters in a string have a left-to-right positional order
- They cannot be changed in place after they are created





## Indexing and Slicing

# Python demo - pygame

# Working on and submitting individual assignments

# Individual assignment - Initial python scripts

**course-exercises repo:**  
**meeting3/meeting3\_exercise.md**

# Collaborative exercise: Exploring Github repository

# Work in groups of 2-3

- Explore the following public Github repository that uses python
- With your group members, discuss what you see in the repositories.
  - How are files organized?
  - Who contributes to the repository?
  - Does the repositories have any branches? Has it been forked?
  - Are the repositories well documented?
  - Are there any commonalities across multiple repositories?
- Write up your notes as a group and submit on the bCourses site.
- Repos:
  - <https://github.com/fogleman/Minecraft>
  - <https://github.com/facebook/pyaib>
  - <https://github.com/cherrypy/cherrypy>

# Group Projects

<https://bcourses.berkeley.edu/courses/1465709/pages/project-overview>



End of Meeting #3

# For next meeting

- Videos:

1. Sequences (11 mins)
2. Lists (12 mins)
3. Lists and Mutability (4 mins)
4. Mutability, Part 1 (8 mins)
5. Mutability, Part 2 (8 mins)
6. Tuples (7 mins)
7. Ranges (5 mins)
8. Dictionaries (21 mins)
9. Encoding Text (5 mins)
10. Unicode Strings (10 mins)
11. Encoding (16 mins)

- Readings:

- Lutz Chapter 8: Lists and Dictionaries
- Lutz Chapter 9: Tuples, Files, and Everything Else