Python Bootcamp

Workshop 1

8/2

Youth In Code

Justin Liu

Roadmap

- A Gentle Intro to Programming
- What is Python?
- Printing - Your first program
- Variables and Data Types
- Assignments
- Numerical Operations
- Strings and String Concatenation
- Comments
- User Input

“Software is at the core of so many of the tools we use today: nearly everyone uses social networks to communicate, many people have internet-connected computers in their phones, and most office jobs involve interacting with a computer to get work done. As a result, the demand for people who can code has skyrocketed.” - Al Sweigart (renowned software developer and teacher)

Movies and paradigms in general often depict hackers & programmers furiously clanking on keyboards, typing cryptic streams of 1s and 0s on glowing green screens — modern programming isn't nearly that mysterious.

Demystifying Programming

- Common myths, misconceptions, and preconceived notions:
  - “I can’t be a programmer; I suck at math.”
    - While it is common for people with a strong foundation in math to be good at programming, the converse isn’t always true.
  - “I’m too old to learn how to code.” or “I’m too young to learn how to code.”
    - It’s never too late to start learning/picking up a new skill. But that’s not to say that you have to start coding at the age of 5 to be a successful SWE.
  - “Programming is for nerds.”
    - Anyone can and should enjoy the art of programming
    - A counterargument: “Be nice to nerds. Chances are you’ll end up working for one.” - Bill Gates

Approach programming with a “growth mindset” and be creative with it

What is Programming?

To put it simply: *Programming is the act of telling what a computer to do.*

What is Programming?

- Program instructions might perform some numerical calculations, modify text, search for information in files, or communicate with other computers over the internet.
  - Machine-styled instructions in English:
    - “Do this; then do that.”
    - “If that’s true, perform this action; otherwise, do that action.”
    - “Repeat this action exactly 5 times.”
    - “Keep doing that until this condition is broken.”

What is Programming?

- Combine the aforementioned building blocks to create more intricate decisions
- On the next slide are the programming instructions, or the source code, for a simple Python program. The computer program runs each line of code from top to bottom.

An example of a Python Script

```
❶ passwordFile = open('SecretPasswordFile.txt')
❷ secretPassword = passwordFile.read()
❸ print('Enter your password.')
  typedPassword = input()
❹ if typedPassword == secretPassword:
  ❺ print('Access granted')
  ❻ if typedPassword == '12345':
     ❼ print('Hey genius, that’s a really strong password you have there.')
 else:
  ❽ print('Access denied')
```

An example of a Python Script

You might not know anything about programming, but you could probably guess what the previous code does with a cursory read:

- ◉ The file SecretPasswordFile.txt is opened ❶
- ◉ The secret password in it is read ❷
- ◉ User is prompted to input a password ❸
- ◉ Compare passwords ❹
- ◉ Grant access based on whether or not the passwords match ❺ - ❽

What is Python?

- The name stems from a British comedy group Monty Python, not from the snake.
- Python is a programming language
- A Python *interpreter* reads *source code* and performs its encoded instructions.
- If we weren’t using Replit, you would download the Python interpreter for free at python.org

Python: The Language of the Future

- High-level
- Straightforward Syntax
- Human-interpretable
- Built-in functions
- Easy to use
- Powerful libraries
- Speed...

Your first program: Hello World

- Replit (our coding environment) → New repo (Python)
- print("Hello World")
- The print() function prints whatever string value is inside its parentheses
- () indicates that you are making a *function call*, and the stuff inside the parentheses are the *parameters*
- Note you can also use this function to put a blank line on the screen
  - just call print() with nothing in between the parentheses

Variables and Data Types

- Strings (no chars!)
- Ints
- Floats
- Booleans
- Python is completely object oriented
- You do not need to declare variables types when *instantiating* them
- You must define a variable before using it
- Every variable in Python is an object (more on objects in Workshop 5)
- Building block for programming

Variables and Data Types

```
mystring = 'hello'
print(mystring)
mystring = "hello"
print(mystring)
```

- Note:
  - Both single and double quotes are interpreted the same way
  - You don’t need semicolons to indicate the end of a line

Assignments

- For simplicity's purpose, think of a variable as a box
  - A more formal definition involves addresses/references & computer memory allocations
- Variables allow you to store information and modify/use it later
- The "=" denotes an *assignment*
- To the left of the "=" is the variable name
- To the right of the "=" is the value being stored

Assignments

```
a, b = 3, 4
print(a + b)

# This will not work!
one = 1
two = 2
hello = "hello"
print(one + two + hello)
```

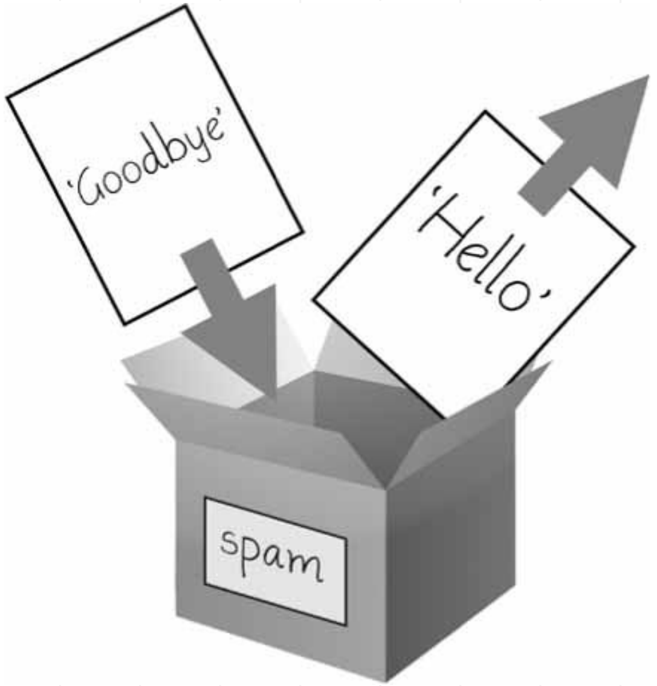
Assignments

```
# This will work...
age1 =1
age2 = 2

print(“Hello Josh, you are “ + str(age1 + age2) + “ years old”)
```

Assignments

```
>>> spam = 'Hello'
>>> spam
'Hello'
>>> spam = 'Goodbye'
>>> spam
'Goodbye'
```



What’s in a Name?

- A good variable name describes the data it contains while maintaining brevity.
- It can be only one word with no spaces.
- It can use only letters, numbers, and the underscore (_) character.
- It can’t begin with a number.
- Convention in Python is *snake case*, not *camelcase*
    - Ex. my_var = ‘data’
- Case-sensitive
    - Use Caps for Class names and Constants (all caps)

String Concatenation

- str(), int(), and float() functions will evaluate to the string, integer, and floating-point forms of the value you pass, respectively
- The meaning of an operator may change based on the data types of the values next to it → context matters
  - + is the addition operator when it operates on numbers
  - But when + is used on two string values, it joins/concatenates the strings
- Some other string functions: capitalize(), format(), index(), split(), strip(), partition(), upper(), lower()

```
name = 'Ralph'

print('My name is ' + name + '.')

Output: My name is Ralph.
```

float()

```
myfloat = 7.0
print(myfloat)
myfloat = float(7)
print(myfloat)
```

- Both versions of myfloat are equivalent
- Note that float(7) and 7, or int(7), are not equivalent

Operations

- The * operator multiplies two integer or floating-point values
- when the * operator is used on one string value and one integer value, it becomes the string *replication* operator
- 2 consecutive asterisks indicate an exponential
- % = modulo (remainder)

Operators

- Arithmetic operations (addition, subtraction, division, multiplication)
  - number = 1 + 2 * 3 / 4.0
- Modulo
  - remainder = 23 % 3
  - Useful for determining if a number is even or odd
- Exponents
  - squared = 7 ** 2
  - cubed = 2 ** 3
  - sqrt = 4 ** 0.5
    - Math.sqrt()
- Operations with Strings
  - hello_space_world = "hello" + " " + "world"
  - lots_of_hellos = "hello" * 10

A Quick Note on *Comments*

- For programmers to describe what a specific portion of their code does
- Readability purposes
- Debugging
- Code organization
- The compiler ignores it → comments aren’t really run
- Denoted by “#” in Python

```
# This is a comment
```

Taking in user input

- The input() function allows you to take in information from the user

```
userName = input()
print(‘Hi ’ + userName + ‘, hope you’re having a good day!’)
```

A Good Resource

http://pythontutor.com/visualize.html#mode=edit allows you to visualize your code in real-time, line by line.

Thank You