Intro to Python Class 1

(cc) BY-NC

Welcome!

Girl Develop It is here to provide affordable and accessible programs to learn software through mentorship and hands-on instruction.

Some "rules"

- We are here for you!
- Every question is important
- Help each other
- Have fun

Welcome!

Tell us about yourself.

- Who are you?
- What do you hope to get out of the class?
- What is your favorite or dream vacation destination?

What we will cover this morning

- Why Python?
- What is programming?
- Variables and arithmetic
- Statements and Error Messages
- Development Environment Setup

Girl Develop It
don't be shy. develop it.

Why Python?

- Suitable for beginners, yet used by professionals
- Readable, maintainable code
- Rapid rate of development
- Variety of applications
- Vast, welcoming community

What is Python used for?

- System Administration (Fabric, Salt, Ansible)
- 3D animation and image editing (Maya, Blender, Gimp)
- Data Science (pandas, sklearn, nltk, tensorflow)
- Web development (Django, Flask)
- Game Development (Civilization 4, EVE Online)
- Web scraping (selenium, scrapy, beautifulsoup)
- Test Automation (Robot Framework)

Who is using Python?

- Disney
- Dropbox
- Buzzfeed
- Google
- NASA
- Condé Nast
- (many, many others)

What is programming?

- Teaching the computer to do a task
- A program is made of one or more files of code, each of which solve part of the overall task.
- The code that you program is human readable but translates into a form that the computer can understand and run.
- Don't focus on what's "under the hood" for now. We will "drive the car" first.

NOTE: Computers need simple, clear instructions

Thinking like a programmer

Computers are great at *processing*. They are bad at *understanding*.

When you write a program, you must break down every step into simple pieces.

Example: Draw a Square

1. Find a whiteboard and a dry erase marker.
2. Uncap the dry erase maker.
3. Hold the marker in your hand.
4. Place the marker against the whiteboard.
5. Move your hand 1 foot to the right.
6. Stop.
7. Move your hand 1 foot down...

Example: Make a Sandwich

What are the steps to making a sandwhich?

Basically, computers can be hard to work with!

How does Python work?

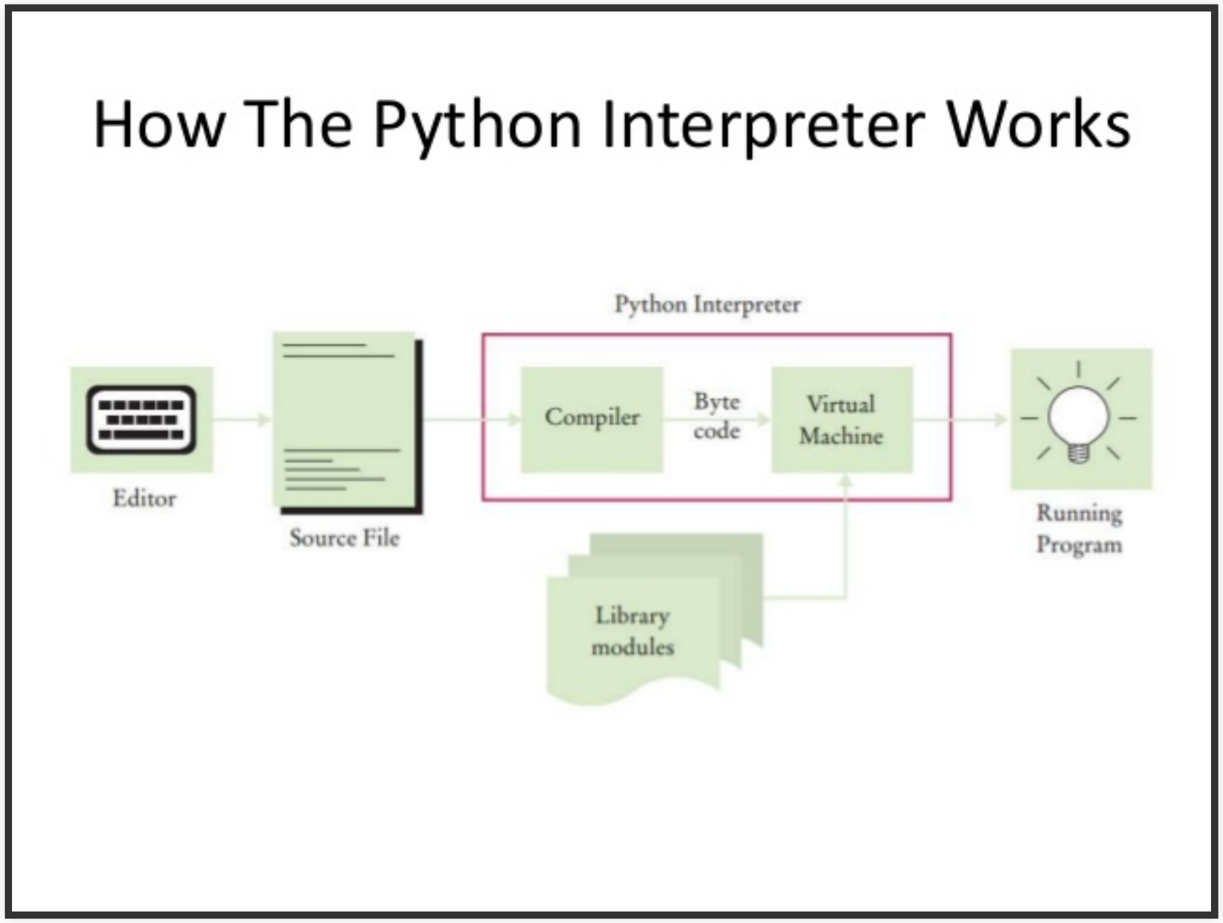

Photo credit: Open Source for Geeks

If the interpreter finds code it doesn't understand, it stops running and creates an error message.

Command line, Python Shell, Text Editors, Notebooks

Terminal	A program that has a command line interface and issues commands to the operating system.
Python Shell	A command line program that runs inside of the terminal, takes Python code as input, interprets it, and prints out any results.
Text Editor	A program that opens text files and allows the user to edit and save them. (Different than a word processor).

Example Text Editors

Linux	Gedit, Jedit, Kate
MacOSX	Textmate
Windows	Notepad++
All	Vim, Emacs, PyCharm, Atom

Let's Develop It!

Let's setup our computer for Python programming

- Install Atom - A free text editor
- Install Python 3.6
  (This step is for Windows users only. GNU/Linux and Mac OS X come with Python installed! -- maybe, check the version)
- Windows Users: Make sure you add to path
- Locate and run the terminal program.
- More setup instructions are available: here

Let's Develop It!

Working in the Python Shell

Open up your terminal and type `python`

Follow along with the examples in the upcoming slides.
Just type them right in!

Feel free to explore, as well.
You will not accidentally break things!

Variables and Arithmetic

```
3 + 4
2 * 4
6 - 2
4 / 2
```

```
>>> a = 2
>>> b = 3
>>> print(a + b)
5
>>> c = a + b
>>> print(c * 2)
10
```

```
>>> a = 0
>>> a = a + .5
>>> print(a)
0.5
>>> a
0.5
```

Data Type: Strings

Python provides some tools to manipulate different data types

For the data type string, we can print words in all uppercase, lowercase, or capitalize the first letter.

```
>>>'hello'.upper()
'HELLO'
>>>'welcome'.capitalize()
'Welcome'
>>>'GOOD BYE'.lower()
'good bye'
```

Operators for strings

Strings can also be manipulated with a set of operators

A "string" can be used with the operators: `+`, `*`

```
a = 'Hello '
b = 'World'
c = a + b
print(c)
```

```
a = "Spam "
b = a * 4
print(b)
```

```
a = 'spam '
b = 'eggs'
c = a * 4 + b
print(c)
```

Girl Develop It
don't be shy. develop it.

Data Types

- Variables are used to store data for future use
- Data always has a "type"
- The type of a piece of data helps define what it can do
- The type can be found using: `type()`
- `type()` is a function. We *call* it by using parenthesis and pass it an object by placing the object inside the parenthesis

```
a = 4
print(type(a))
print(type(4))

print(type(3.14))

b = 'spam, again'
print(type(b))
print(type("But I don't like spam"))
```

Data types - continued ...

- Each data type has a specific set of operators
- We saw strings use the `+` and `*` operators
- An "int" or "float" can be used with any of: `+`, `-`, `*`, `/`
- What happens if we try to use division or subtraction with a string?

```
>>> print("Spam" / "eggs")
Traceback (most recent call last):
File "", line 1, in
TypeError: unsupported operand type(s) for /: 'str' and 'str'
```

Errors

- There are different kinds of errors that can occur.
  We've seen a few already.

- A "runtime error" results in an Exception, which has several types.
  Each type gives us some information about the nature of the error and how to correct it

- One type of exception is a SyntaxError. This results when our code can not be evaluated because it is incorrect at a syntactic level.
  In other words, we are not following the "rules" of the language.

- Some other examples are the TypeError and NameError exceptions.

Errors - continued ...

```python
# SyntaxError - Doesn't conform to the rules of Python.
# This statement isn't meaningful to the computer
4spam)eggs(garbage) + 10

# NameError - Using a name that hasn't been defined yet
a = 5
print(b)
b = 10

# TypeError - Using an object in a way that its type does not support
'string1' - 'string2'
```

There are also semantic errors.

These are harder to catch because the computer can't catch them for us.

Girl Develop It
don't be shy. develop it.

Comments

You can leave comments in your code—notes that people and future you (!) can read (and appreciate) but computers will ignore.

```
# This is a sample comment.
```

Let's Develop It

We'll practice what we've learned in the shell.

Practice commands you didn't fully understand before.

Play with strings, arithmetic, variables, type.

Ask the teacher, TAs, and students around you for help!

Using the Terminal or Console

Command	Short for	Description
`pwd`	Print working directory	Displays what folder you are in. Windows the prompt always shows you.
`ls`	List (dir for windows)	Lists the files and folders in the current folder
`cd`	Change directory	Change to another folder. Takes the folder name as an argument. 'cd ..' goes up a directory
`mkdir`	Make directory	Creates a folder with specified name in the current folder

Creating a folder

We need a folder to save and run our code from.

($ shows the shell prompt where commands are entered. Lines without $ are output)

```
$ cd
$ pwd
/home/username ( or /User/username or similar)
$ mkdir gdipython
$ cd gdipython
$ pwd
/home/username/gdipython
```

Now that the folders are made, we only have to use `$ cd ~/gdipython` in the future.

The Text Editor

- Open Atom.
- Click "File", then "Open". Navigate to the `gdi-intro-python` folder we just created and click "Open"
- In the text editor, enter the following:

```
print('Hello World!')
```

- Click "File", then "Save As...". Type "class1.py" and click "Save".
- Open a terminal and navigate to the `gdi-intro-python` folder.
  If you don't already have this folder open in a terminal.
- Type `python class1.py`
- You should see the terminal print "Hello World!"

User Input

To obtain user input, use `input()`

Change the class1.py text to the following and run it again

```
name = input("What is your name?")
age = input("How old are you?")
print("Your name is" + name + " and you are " age + " years old.")
```

The input function returns a value that is always a string. Keep that in mind when working on other problems!

Let's Develop It!

Write your own program that uses input
and does something else with the value

You can use float() to treat the input as a number if you need a number,
or use the input directly as a string.
If you want to print a integer, use str(int_value) to convert it to a string.

Questions?

Girl Develop It
don't be shy. develop it.

Exercises

Write a program that could replace the initial conversation you have at the checkout line. It should do the following:

```
>> Did you find everything ok?
(wait for input)

>> What's your name?
(read in a name from the user)

>> Hi, (User's Name) do you want paper or plastic?
(wait for input)

>> What is your total before tax?
(read in a total)

>> Your total with sales tax is: (total with sales tax)
(wait for input before exiting)
```