Intro to Python

Class 1

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

What we will cover today

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

Why Python?

- Suitable for beginners, yet used by professionals
- Readable, maintainable code
- Rapid rate of development
- Few "magical" side-effects
- Variety of applications

What is Python used for?

- System Administration (Fabric, Salt, Ansible)
- 3D animation and image editing (Maya, Blender, Gimp)
- Scientific computing (numpy, scipy)
- Web development (Django, Flask)
- Game Development (Civilization 4, EVE Online)

Who is using Python?

- Disney
- Dropbox
- Canonical and Red Hat
- Google
- NASA

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
- Programming code is human readable but also needs a form that the computer can run directly. This form is not human readable.
- To create the form of code the computer can use, we use the Python **interpreter**. Other languages use other interpreters or a **compiler**
- Don't focus on what's "under the hood" for now. We will "drive the

car" first
- In other words, there are many layers to the onion. We start at one layer and slowly move toward layers that are beneath or above us

Command line, Python Shell, Text Editors

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, TextWrangler
Windows	Notepad++
All	Sublime Text, Vim, Emacs

Let's Develop It

Let's setup our computer for Python programming

- Let's install a text editor - **Install Sublime Text 2**
- **Install Python** (This step is for Windows users only. GNU/Linux and MacOSX come with Python installed)
- (Windows only): After installing Python, open the "powershell" program and type:

```
[Environment]::SetEnvironmentVariable("Path", "$env:Path;C:\Python27", "User")
```

- Locate and run the terminal program. Type 'python' and hit enter.
- More setup instructions are available: **here** (You can, but don't have to install the other text editors recommended in this guide)

Working in the Python Shell

Open up your terminal and type 'python'

Follow along with the examples in the slides. Type them 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
c = a + b
print c * 2
```

```
a = 0
a = a + .5
print a
```

Strings

```
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
```

Data types

- Variables are names of objects
- Among other things, variables are used to represent something that can't be known until the program is run
- Objects always have a "type"
- The type of an object 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 ...

- Objects can be used with a set of **operators**
- An int or float can be used with any of: +, -, *, /
- A string can be used with any of: +, *
- What happens if we try to use division or subtraction with a string?

```
print "Spam" - "am"
a = 'Spam and eggs'
print a / 'hashbrowns'
print a / 6
```

Errors

- There are different kinds of errors that can occur. We've seen a few of these so far
- A runtime error results in an **Exception** of which there are 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 ...

```
# 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.

Let's Develop It

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

Review the slides on your computer and practice entering any commands you didn't fully understand before

Ask the teacher or a TA for any help

Using the Terminal

Try each of the following commands in turn:

Command	Short for	Description
pwd	Print working directory	Displays what folder you are in.
ls	List	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
cat	Concatenate	Prints the contents of a file. Takes a filename as an argument

Creating a folder

We need a folder to save our work in.

The ->'s below indicate the expected output of the previous command.

```
pwd
-> /home/username
mkdir Projects
cd Projects
mkdir gdi-intro-python
cd gdi-intro-python
pwd
-> /home/username/Projects/gdi-intro-python
```

Now that the folders are made, we only have to use 'cd Projects/gdi-intro-python' in the future.

The Text Editor

Open sublime text.

- Click file, open folder. Navigate to the gdi-intro-python folder we created and click "open"
- In the text editor, enter the following:

```
print 'I am a Python program'
```

- Click file, save as. Type 'class1.py' and click ok.
- Open a terminal and navigate to the gdi-intro-python folder (if not left open from before)
- Type 'python class1.py'
- You should see the terminal print "I am a Python program"

User Input

To obtain user input, use 'raw_input()'

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

```
input_value = raw_input("Enter a radius:")
radius = float(input_value)
area = 3.14159 * radius ** 2
print "The area of a circle with radius " + input_value + " is:"
print area
```

N.B. - The user's input is a string so we use float() to make it a number

Let's Develop It

Write your own program that uses raw_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

Questions?