



Lecture 1

basic Python programs,
defining functions

Lecture notes modified from CS Washington CS 142
Except where otherwise noted, this work is licensed under:
<http://creativecommons.org/licenses/by-nc-sa/3.0>

Python!

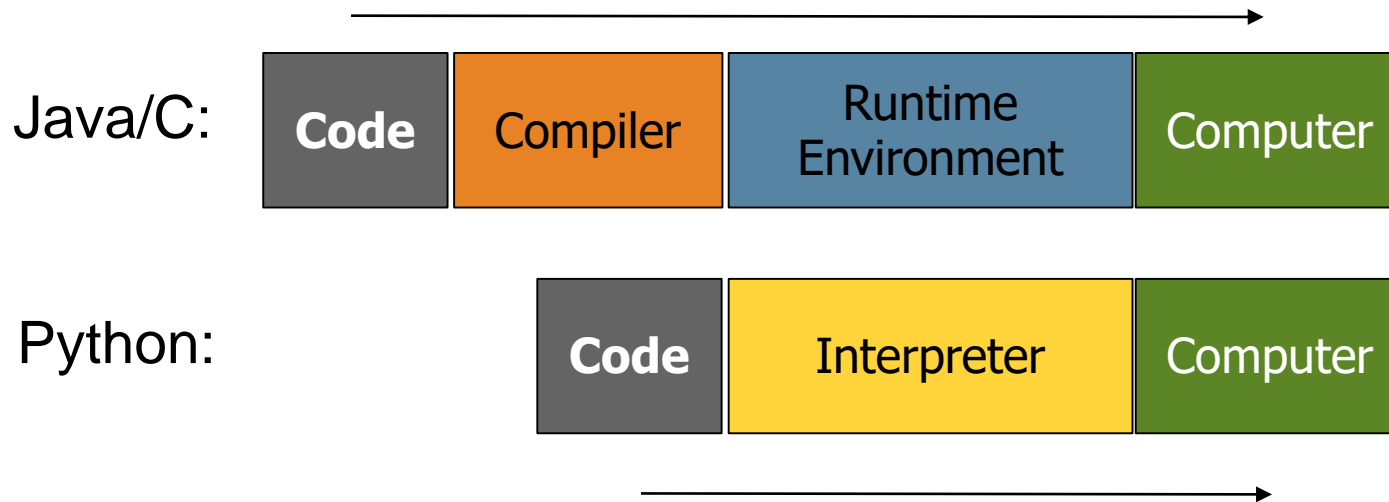
- Created in 1991 by Guido van Rossum (now at Google)
 - Named for Monty Python
- Useful as a **scripting language**
 - **script**: A small program meant for one-time use
 - Targeted towards small to medium sized projects
- Used by:
 - Google, Yahoo!, Youtube
 - Many Linux distributions
 - Games and apps (e.g. Eve Online)



Interpreted Languages

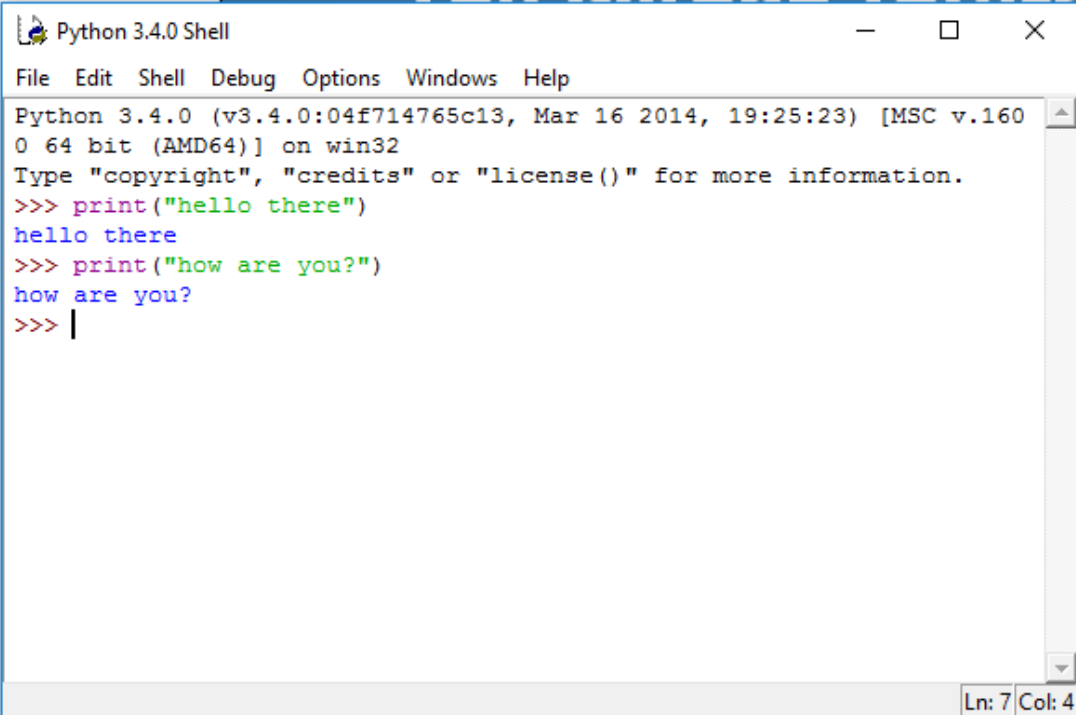
- **interpreted**

- Not compiled like Java
- Code is written and then directly executed by an **interpreter**
- Type commands into interpreter and see immediate results



The Python Interpreter

- Allows you to type commands one-at-a-time and see results
- A great way to explore Python's syntax
 - Repeat previous command: Alt+P
 - Next command: Alt+N



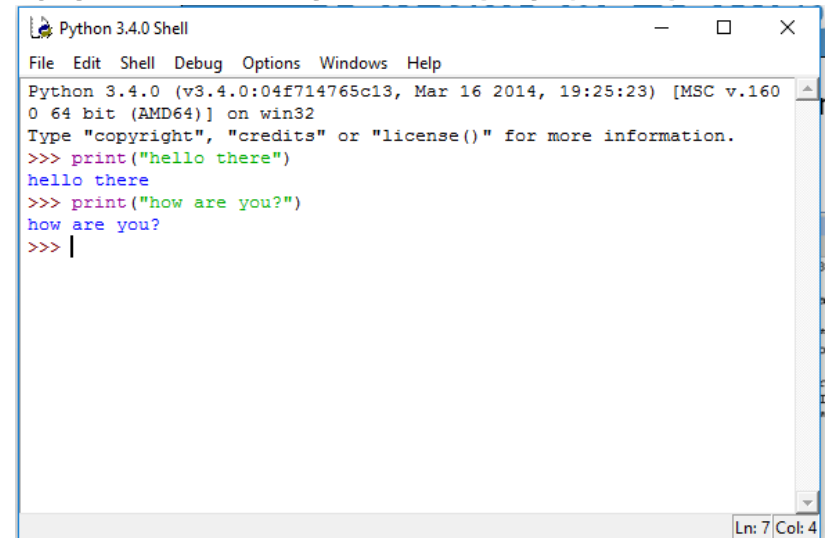
```
Python 3.4.0 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.0 (v3.4.0:04f714765c13, Mar 16 2014, 19:25:23) [MSC v.160
0 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print("hello there")
hello there
>>> print("how are you?")
how are you?
>>> |
```

Running Python Program

- Python does not have a `main` method like Java or C
 - The program's main code is just written directly in the file
- Python statements do not end with semicolons
- Python two running modes:
 - Interactive Python Shell mode
 - IDLE Python source file – create, edit, and running

Python Interactive Shell

- When you start up IDLE, a window with an interactive Python shell will pop up:



```
Python 3.4.0 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.0 (v3.4.0:04f714765c13, Mar 16 2014, 19:25:23) [MSC v.160
0 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print("hello there")
hello there
>>> print("how are you?")
how are you?
>>> |
```

- You can type Python code directly into this shell, at the '>>>' prompt.
- Whenever you enter a complete code fragment, it will be executed.

Python Interactive Shell

Python Interactive Shell Practice:

The print command outputs text directly to the Python shell

```
>>> print('hello there')
```

```
hello there
```

```
>>> print( )
```

```
>>> print(4+6)
```

```
10
```

```
>>> print(8**3)
```

```
512
```

Number vs String

- Number: Numeric value can be added, subtracted, multiplied, divided, etc
- String: a string value is a value that can contain any combinations of characters, it'll normally get used for an address or similar, you can put what you want there really.
In Python, single quote, 'hello' or double quotes "hello"

In Python Shell prompt, try

```
>>> print(123+53)
```

```
176
```

```
>>> print('123+53')
```

```
123+53
```

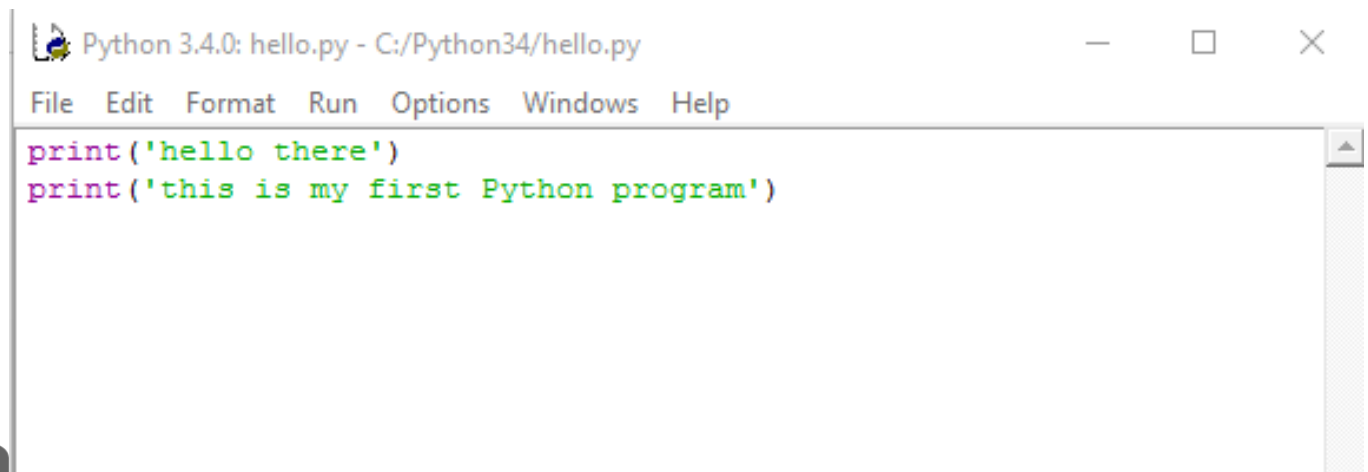
More on these will be covered later.



Python IDLE File

- In IDLE, go to FILE → New → Create a new python file (Open -> open an existing Python file)
- You can edit/modify/create a Python file and run it (IDLE->RUN-> RUN MODULE)

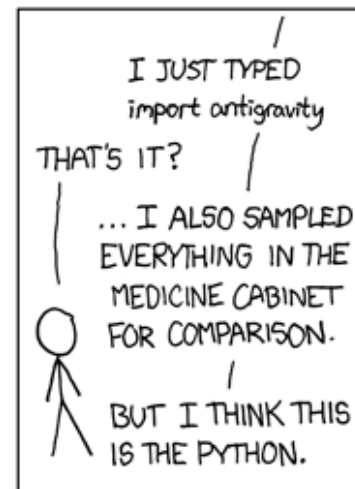
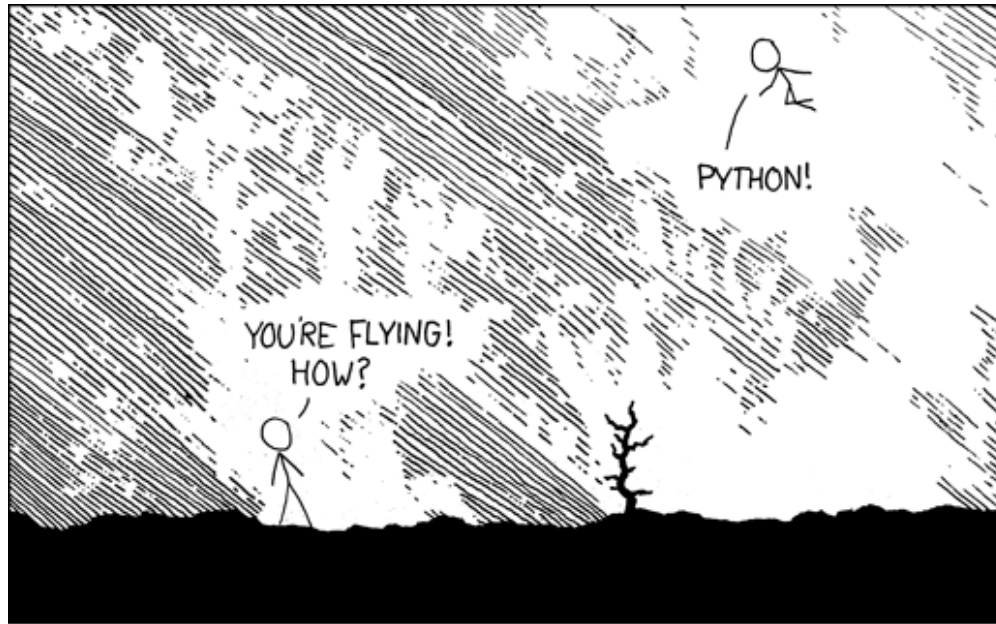
Demonstrate, and create a new hello.py file with these python commands.



The screenshot shows a window titled "Python 3.4.0: hello.py - C:/Python34/hello.py". The window contains a menu bar with "File", "Edit", "Format", "Run", "Options", "Windows", and "Help". The main text area contains the following Python code:

```
print('hello there')  
print('this is my first Python program')
```

A Brief Review



The `print` Statement

```
print("text")  
print()          (a blank line)
```

- Certain special characters must be prefixed with a backslash
- Strings can also start/end with `'`

swallows.py

```
1 print("Hello, world!")  
2 print()  
3 print("Suppose two swallows \"carry\" it together.")  
4 print('African or "European" swallows?')
```

Comments

- Comments are ignored
- Syntax:
 - # **comment text** (anything after the # on the same line is ignored)

swallows2.py

```
1 # Suzy Student, Raspberry Pi Camp, Summer 2016
2 # This program prints important messages.
3 print("Hello, world!")
4 print()                # blank line
5 print("Suppose two swallows \"carry\" it together.")
6 print('African or "European" swallows?')
```

Functions

- **Function:** A block of Python commands executed together.
- **Syntax:**

```
def name () :  
    statement  
    statement  
    ...  
    statement
```

hello2.py

```
1  # Prints a helpful message.  
2  def hello():  
3      print("Hello, world!")  
4  
5  # main (calls hello twice)  
6  hello()  
7  hello()
```

- Must be declared above the 'main' code
- Statements inside the function must be indented

Whitespace Significance

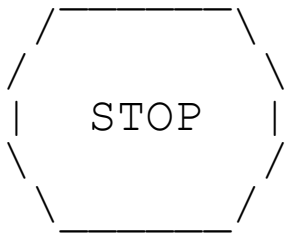
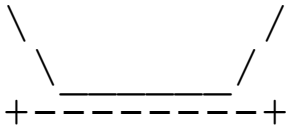
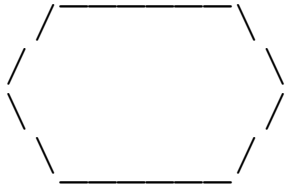
- Python uses indentation to indicate blocks
 - Makes the code simpler and more readable
 - In Java/C, indenting is optional. In Python, you **must** indent.

hello3.py

```
1 # Prints a helpful message.
2 def hello():
3     print("Hello, world!")
4     print("How are you?")
5
6 # main (calls hello twice)
7 hello()
8 hello()
```

Exercise

- Rewrite the figures.py lecture program in Python, so it outputs:



Exercise Solution

```
def egg():  
    top()  
    bottom()  
    print
```

```
def cup():  
    bottom()  
    line()  
    print
```

```
def stop():  
    top()  
    print("|   STOP   |")  
    bottom()  
    print
```

```
def hat():  
    top()  
    line()  
    print
```

```
def top():  
    print("           ")  
    print("/_____\\")  
    print("/           \\")
```

```
def bottom():  
    print("\\           /")  
    print("\\_____/" )
```

```
def line():  
    print("+-----+")
```

```
# main  
egg()  
cup()  
stop()  
hat()
```