

DataScience for Development and Social Change, 2015

Basic Coding Concepts

Just enough Python

Time to learn Python

- ❖ Learning:
 - ❖ Enough Python and programming concepts to get you started
- ❖ Not learning:
 - ❖ All of Python
 - ❖ All of programming language theory

Prerequisites

- ❖ Python installed
- ❖ Terminal window
- ❖ Text editor
- ❖ Directory to put code into
- ❖ Git installed

Strings, Printing & your first program

- ❖ Create a directory to put your Python code in.
- ❖ Open a terminal window, and use “cd” to move to that directory.
- ❖ Open your editor, and type this in the editor:

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

- ❖ Save to file “helloworld.py” in your Python code directory
- ❖ Go into your terminal window again, and type:

```
python helloworld.py
```

* **NB: Beware the quote characters!** If you cut and paste the above code, you might see this error message:

“SyntaxError: Non-ASCII character '\xe2' in file helloworld.py on line 1”

- ❖ This happens because the symbols “ and ” above aren’t the same as the ones in your code editor. It’s annoying, but easily fixed: just delete them and type “ and ” in the right places in the editor. And if you’re in sublimetext, notice how the code changes color when you do this...

That was your first program...

- ❖ Python filenames end in “.py”
- ❖ **“Hello World!”** is a “string”
- ❖ **print(“Hello World!”)** prints “Hello World!” to your terminal

Comments

- ❖ Add a line starting with “#” to your helloworld.py file, e.g.

#This is a comment

- ❖ Add text starting with “#” to the end of your “print” line, e.g.

print(“Hello World!”) #English

- ❖ And type “python helloworld.py” again

Magic Comments

`#!/usr/bin/python`

`# -*- coding: utf-8 -*-`

- ❖ “Shebang” at start of every Python program
- ❖ Tells the interpreter which type of file this is (python), and which character set you used in it (utf-8 is a very common character set)

Variables

```
my_string = “Hello World!”
```

```
print(my_string)
```

```
my_boolean = True
```

```
my_number = 10
```

```
print(my_number)
```

```
my_number = 15.523
```

```
print(“My number is {}".format(my_number))
```

❖ NB beware the quote characters again!

String formatting

```
my_number = 15.523
```

```
print(“My number is {}. It’s not {}”.format(my_number,  
“nothing”))
```

see http://www.python-course.eu/python3_formatted_output.php for more complex formatting

Python Interpreter

- ❖ Type “python” in your terminal window
- ❖ Try out some Python commands
- ❖ Type “exit()” to leave the python interpreter

```
>>> print(“Give me text”)
```

Getting help

- ❖ Online, there's:
 - ❖ `docs.python.org`
- ❖ In the interpreter, you can use:
 - ❖ `help()`
 - ❖ type “q” to leave a help page
 - ❖ `dir(str)`
 - ❖ `help(str.rjust)`

Getting Input from the User

```
user_text = raw_input("Give me some text>")
```

```
lower_text = user_text.lower()
```

```
text_length = len(user_text)
```

```
print("Your text is {}, its length is {}".format(user_text,  
text_length))
```

```
print("In lowercase, that's {}".format(lower_text))
```

Collections : Dictionary

- ❖ You'll meet two types of collection in Python: lists and dictionaries
- ❖ Dictionary:

```
iso3166 = {  
    'SLE': "Sierra Leone",\  
    'NGA': "Nigeria",\  
    'LBR': "Liberia" }
```

```
iso3166['LBR']
```

```
iso3166.keys()
```

Collections: List

❖ List:

```
rowvals = [1, 3, 5, 6, 4, 7, 3, 1, 3]
```

```
rowvals[3]
```

```
max(rowvals)
```

```
rowvals.sort()
```

```
rowcols = [['a','b','c'], ['d','e',1]]
```

```
rowcols[0][2]
```

Iterators

- ❖ Iterators allow you to use every item in a list, in turn:

```
alist = [1,2,3,4]
```

```
for item in enumerate(alist):
```

```
    print(item)
```

```
for index, item in enumerate(alist):
```

```
    print(item)
```

```
iso3166 = {
```

```
    'SLE': 'Sierra Leone', \
```

```
    'NGA': 'Nigeria', \
```

```
    'LBR': 'Liberia' }
```

```
for code in iso3166:
```

```
    print('the code for {} is {}'.format(iso3166[code], code))
```

NB INDENTS! Python is an indented language - spaces mean something...

Getting input from a file

❖ Libraries are your friends!

```
import csv
```

```
csvfilename = 'data/ebola-data-db-format.csv';
```

```
fin = open(csvfilename, "rb");
```

```
csvin = csv.reader(fin);
```

❖ You just created a link to a CSV file. Now take a look at it:

```
print("Headers:");
```

```
headers = csvin.next();
```

```
for header in headers:
```

```
    print(header);
```

```
for row in csvin:
```

```
    print("New row:");
```

```
    for col in range(0,len(row)):
```

```
        print(row[col]);
```

```
fin.close();
```

Getting serious now: using libraries, loops, and indents!

Conditionals

```
import csv

csvfilename = 'data/ebola-data-db-format.csv';

fin = open(csvfilename, "rb");

csvin = csv.reader(fin);

headers = csvin.next();


#Find all the rows about Liberia

for row in csvin:

    if row[1] == "Liberia":

        print("Liberiaaaa!");

        for col in range(0,len(row)):

            print(row[col]);

fin.close();
```

NB You might need to convert from strings to other data types. Conversion methods include `str(anumber)`, `int(astring)`, `float(astring)`

Libraries

- ❖ Libraries are pieces of code that somebody else wrote to do something you need
- ❖ You already used one when you typed **import 'csv'**
- ❖ Places to look for libraries include:
 - ❖ <https://docs.python.org/2/library/> (you already have these)
 - ❖ <https://pypi.python.org/pypi>
 - ❖ github

That's enough Python for now

- ❖ For more, see sites including:
 - ❖ [Learn python the hard way](#)
 - ❖ [Writing idiomatic Python](#)
 - ❖ [IcanHaz python courses list](#)