CS50 for MBAs: Programming with Python

April 8 2016 | Limor Gultchin gultchin@college.harvard.edu

A bit of inspiration – or who is this Python Person anyway

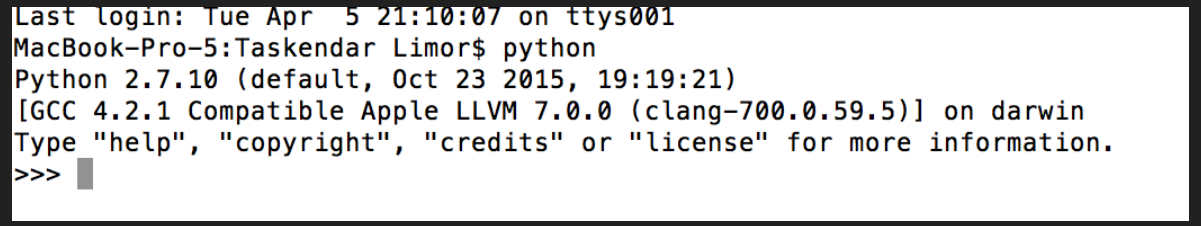
- Is this picture to the right familiar?
- The name of this popular programming language is commemorating Monty Python (those British guys from the Holy grail)
- Was developed since 1989 and released in 2000.
- Its main author is Guido van Rossum, who is till so involved in its development, that he received the flattering title "benevolent dictator for life" from the Python community

Why Python?

- If you are already here (and by the overwhelming sign up rates), I probably don't have to explain, but:
- Flexible, powerful, elegant and a slightly more forgiving language that will allow you to write programs quickly and elegantly
- Great support provided by the ever-growing python community on-line (e.g. pythontips.com)
- Incredible libraries (e.g. Requests, SQLAlchemy, Pyglet, Pygame and many more)
- For web-dev – Django
- However! Less powerful than c, takes more time to execute

Where to begin

- Make sure you have python installed on your computer (by typing python in your terminal)
- Don't see something like this?



- Go to https://wiki.python.org/moin/BeginnersGuide/Download, download and install as per instructions
- Make sure you have a compatible text editor. Most of them are, but Sublime is the one I'd recommend

Got it. Now what can we do with it?

- Anything you can imagine!
- I mean, aside from actually managing your memory and the like…
- Python "doesn't care" as much about syntax as other languges you might be familiar with (such as C), but has most of their functionality
- Plus some other interesting built in options, like the for x in range(y) loop (replacing c's for i=0; i< n; i++)
- But let's do it properly!

General syntax/guidlines

- No semi colons (like in c)
- No curly braces ({}) around functions/conditions/loops
- However, indentation super important
- Most first lines of function declarations/conditions/loops will end with : (but think about it like writing an essay – giving a title: then elaborate on it
- In general, python should overall look much more intuitive to English speakers. It almost uses the same grammar
- Don't need to make programs. Python interpreter will do it from the command line once you run the program (using python name_of_program.py)

Variables

- Just go ahead and… declare
- `var = 10`. No need for data type specification, it's all done for you
- Alternatively, `var = 10.0`
- `var = “hello, world!”`
- To print, just type: `Print var`
- Strings: `print ("You had " + answer + "! That sounds delicious!”)`
- `Useful functions to manipulate these: int(), str(), input(), len()`

Loops

```
for x in xrange(1, 11):
    for y in xrange(1, 11):
        print '%d * %d = %d' % (x, y, x*y)
```

```
for x in xrange(3):
    if x == 1:
        break
```

```
#!/usr/bin/python

count = 0
while count < 5:
   print count, " is  less than 5"
   count = count + 1
else:
   print count, " is not less than 5"
```

Conditions

Code

```
#!/usr/bin/python

count = 0
while count < 5:
   print count, " is  less than 5"
   count = count + 1
else:
   print count, " is not less than 5"
```

Output

```
1 <= 7
2 <= 7
3 <= 7
4 <= 7
5 <= 7
6 > 5
7 > 5
8 > 5
9 > 5
10 > 5
```

Touples

The way you write a touple

```
months = ('January','February','March','April','May','June',\
'July','August','September','October','November','  December')
```

Python stores as

Table 1 - tuple indicies

Index	Value
0	January
1	Feb
2	Mar
3	Apr
4	May
5	Jun
6	Jul
7	Aug
8	Sep
9	Oct
10	Nov
11	Dec

Lists

Creating a list

```
cats = ['Tom', 'Snappy', 'Kitty', 'Jessie', 'Chester']
```

Adding to a list

```
cats.append('Catherine')
```

Major differences: mutable, can append, use [] and not ()

Dicts

Made of key-value pairing

```
phonebook = {'Andrew Parson':8806336, \
'Emily Everett':6784346, 'Peter Power':7658344, \
'Lewis Lame':1122345}
```

To add to a dictionary:

```
phonebook['Gingerbread Man'] = 1234567
```

To delete

```
del phonebook['Andrew Parson']
```

Python magic

- Print ('#'*5)
  > #####
- For a in array:
  print(a)
- Manipulating lists:

```
squares = [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

```
squares = [ i**2 for i in range(1,10+1) ]
```

Let's compare C to python

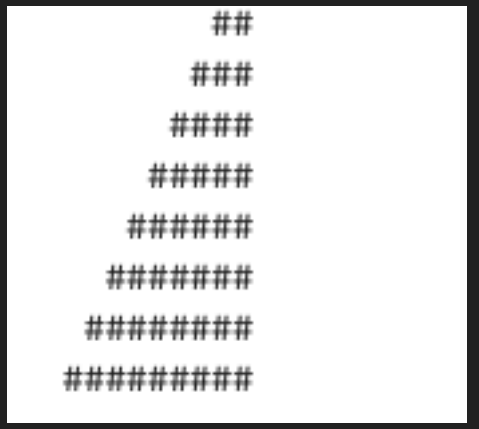
C

```c
#include <stdio.h>

int main(void)
{
    int n;
    n = 10;

    for (int row=0; row<n; row++)
    {
        for (int col=0; col<n; col++)
        {
            if (row+col < n-1)
            {
                printf(" ");
            }
            else
            {
                printf("#");
            }
        }
        printf("#\n");
    }
}
```

Python

```python
def mario(n):
    for row in range(n):
        print(('#'*(row+2)).rjust(n+1))


if __name__ == '__main__':
    mario(10)
```

Output



Some fucntions, some methods

```
object.method(argument, arg2)

function(object, argument, arg2)
```

Remember

- Most of the syntax in Python makes sense. Try writing almost in psuedo code and many times it would work
- If not – Google. There's so much out there
- For most things you'd want to do, someone has already written a useful library. First look it up, then think of how you'd design it yourself