Robert
March 2012

WTF is programming?

“Ugly programs are like ugly suspension bridges: they're much more liable to collapse than pretty ones, because the way humans (especially engineer-humans) perceive beauty is intimately related to our ability to process and understand complexity.”
-*Why Python?* Eric Ramond

Some helpful websites:

http://docs.python.org/tutorial/appetite.html
The official python tutorial, updated with each version of python.
http://www.reddit.com/r/Python
The reddit community for python programming, go ask for help! Or see what other python programmers are up to.
http://codingbat.com/python
A great website that has python projects to work through, you can code on the site and it will test your code. It eve provides solutions.

What is Python?

“Python is a general-purpose, high-level programming language whose design philosophy emphasizes code readability. Python claims to combine "remarkable power with very clear syntax", and its standard library is large and comprehensive.” - Wikipedia

Thinking like a programmer

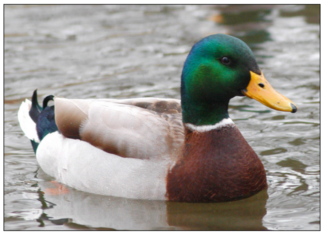
Programmers have to explicitly tell a computer *every* step in whatever process they want the computer to do. As a fledgling programmer yourself (YAY!), you need to start thinking in this manner.

Below, work with those around you to write all of the steps needed to make a peanut butter and jelly sandwich.

Translate your protocol into a flow chart, using the standard elements Robert babbled about.

Data types

Python's type system



Python is 'duck typed', what does this mean? (No Lucia, it has nothing to do with duck-rape.) Some languages, like Java and C, are 'static typed', this means that when you make a variable you define its type. Duck types languages, on the other hand, use the philosophy “if it looks like a duck, walks like a duck, and sounds like a duck, it's a duck”. This means that you never need to tell python what kind of data a variable holds, and the kind of data a variable can hold changes dynamically. List some of advantages and disadvantages of duck typing.

Numbers

What are the two types of numbers that Python can store?

Here are some basic functions that all numbers can use, what does each of them do? *, +, -, /, %.

When would we use an int? When would we use a float?

What functions can we use to compare two numbers? Can we compare an int and a float?

Strings

What is a string, give an example of one. How do we define a string in Python?

What do the functions + and * do to a string?

What does the function **split()** do to a string? What about **len()**?

What functions can we use to compare two strings? What happens if we try to compare a string to an int?

When would you use a string?

Lists

What is a list? How are lists defined in Python?

How do you access, or subscript, a particular item in a list?

What happened if we try to use a negative number to access a list?

What are some important functions we can use on lists?

What similarities are there between lists and strings in Python?

What is a *slice* and how to we get one from a list?

If we want to make a copy of a list how do we do this?

What does the keyword 'in' do to a list?

How do we compare two lists?

In what situations would you use a list?

Dictionary

What is a dictionary? How do we define a dictionary in python?

How do we access items with a dictionary?

How do we add items to a dictionary?

How can we tell what *keys* or *values* a dictionary has?

When would you use a dictionary?

Booleans

What is a Boolean? How do we define them?

How do we compare two Booleans?

What values other than False, will test as False?

None or NoneType

What is None?

How would None compare to a Boolean?

When would we use None as a value?

Type casting

Sometimes you have a variable of one type, but you want it to be something else. For example, you may have a variable defined as:

```
myVariable = “12”
```

But you want that variable to act like the number 12. How do we fix this? We can cast the item using the functions that are named the same as the data type we want.

How would we cast the above variable to an int?

What if we had some variable “myNumber” that held a number, but we wanted it to be a string?

What other type casting functions can we use?

This second case is very common. If you try to concatenate (i.e. +) a string and a number, python errors, but if you cast it to a string first it will work:

```
    print(“I have a ”+str(12)+“ cats”)
```

WARNING: some types cannot be cast to one another, you cannot, for example, cast a string like “apple” to an int, that will crash your program. So be careful with this. As a general rule any item can be safely cast to a string for printing though.

Operators

Comparison Operators

Operator	Effect
A == B	returns True if A is equal to B
A != B	returns True if A is not equal to B
A < B	returns True if A is less than B
A > B	returns True is A is greater than B
A <= B	returns True if A is less than or equal to B
A >= B	returns True if A is greater than or equal to B
A or B	returns True if either A or B is True
A and B	returns True if both A and B are True
not A	returns True if A is False

Remember, the number 0, the empty list [], and the None value all evaluate as False. So if you try 0 and A, since 0 is equal to False this is equivalent to saying False and A.

Basic operators

Operator	Effect on Numbers	Effect on Strings	Effect on Lists
A + B	Adds A and b and returns the result. (2 + 2 returns 4)	Concatenates two strings together. (“apple” + “sauce” returns “applesauce”)	Appends the list B to the end of list A. ([1,2,3] + [4,5,6] returns [1,2,3,4,5,6])
A – B	Subtracts B from A returns the result. (3 – 2 returns 1)	-	-
A * B	Multiplies A and B returns the result. (4 * 3 returns 12)	If A is a string and B is a number returns a new string containing B copies of A. (“dog” * 3 retruns “dogdogdog”)	If A is a list and B is a number returns a new list containing B copies of A. ([1] * 3 returns [1,1,1])
A / B	Divides A and B returns the result. (6 / 3 returns 2)*	-	-
A % B	Returns the modulus of A and B. This is the remainder from dividing them. (5 / 2 returns 1)	-	-
A ** B	Returns A raised to the Bth power. (3 ** 2 returns 9)	-	-
A in B	-	Returns True if A is a substring of B. (“app” in “apple” returns True)	Returns True if A is equivalent to an item in B. (1 in [2,3,1] returns True)
A not in B	-	Returns True if A is not a substring of B. (“dog” not in “apple” returns True)	Returns True if A is not equivalent to any item in B. (4 not in [1,3,2] return True)

Any cell with a – indicates that that operator does not work with that data type. (e.g. The in operator does nothing to numbers)

*Note that in some versions of python two integers being divided always return an integer (5/2 returns 2), however is at least one of the numbers is a float then it will return a float (5.0/2 returns 2.5)

Functions

Write the basic form of a function below.

How does python know where the function 'stops'?

How do we make a function return a value when it is done?

What happens to any code after a return statement in a function?

How can we make a function return more than one value?

How do functions help us maintain code? (Maintaining code means to keep it up to date.)

Control flow

Write the basic formula for an if, if-else, and elif construct below.

How does Python know what goes within the if statement? i.e. what code to execute if the test evaluates to True?

What does the 'not' operator do?

What kinds of things can we use as a 'test' in an if statement?

Loops

What is a loop? When would we use one? What element from a flow chart indicates we need a loop?

For

Write the basic form of a for loop below.

How does Python know when to stop executing a for loop?

The variable between 'for' and 'in' in a for loop is never defined (we never used an '=' to give it a value), how does it have a value inside the loop? What happens if we try to use that variable after the loop is done executing?

What are loops for? When would you use them?

While

Write the basic formula for a while loop below.

What is the difference between a while and a for loop? When would you use one versus the other?

What kinds of tests can we use in a while loop?

How does Python know what code gets executed inside the loop versus after the loop?

What happens if we make a while loop like:

```
while True:
    something()
```

Robert
March 2012

Input and Output

What function will open a file for reading input?

How do we read a single line from an input file?

How do we read all of the lines in an input file?

What function opens a file for writing output?

How do we write a line to an output file?