Computer Science Notes

Programming

A program is a sequence of instructions. So create a sequence of instructions in language, turn the program into binary. Give the binary code to the OS, to give it to the processor. OS tells processor to run program. When its finished (or dies) OS cleans it up.

Python is an interpreted language. Meaning, when running the program, it is also translating. Because we can test one line of code at a time, its nice.

# Calculate area and circumference of a circle from its radius.

# Step 1: Prompt for a radius.

# Step 2: Apple the circumference and area formulae.

# Step 3: Print out the results.

**import** math

radius\_str = input (“*Enter the radius of your circle:* ”)

radius\_int = *int*(radius\_str)

circumference = 2 \* math.pi \* radius\_int

area = math.pi \* (radius\_int \*\* 2)

**print** (“*The circumference is:*” , circumference, \

“, and the area is:” , area)

So, we can import math for uses. And conversion.

At the core of all programming languages.

Control flow of the program

Construct and access data elements

Construct functions

Construct classes

Libraries and built-in classes

Save as a “module”

When you save a file, and place a .py suffix on it, it becomes a python module.

Modules are files with python statements.

**Statements** are commands in Python. They perform some action, often called a side effect, but they do not return any values.

**Expressions** perform some operation and return a value. Can act as statements, not vice versa. Expressions typically do not modify values in the interpreter.

Tabbing is special, just avoid it.

Comments begin with #

Python name conventions, must begin with letters or underscore. Names starting with underscore are different

1/28/16

Variables

Represents an object (number, data structure, function)

Use names to make programs more readable.

Python maintains a list of pairs for every variable (name and value)

A variable is created when a value is assigned the first time.

Subsequent assignments update the associated value.

Namespace, table that contains the associations between variables and values.

The = sign doesn’t mean equal.

So, when you see [variable = variable + 7], the variable is now assigned a different number in the namespace, which was the original variable + 7.

So the result on the right side of the = is assigned to the variable on the left.

You can do this with two variables, where you set one equal to the other.

Types:

Integers: 5

Floats: 1.2

Booleans: True or False

Strings: “anything” or ‘something’

Lists: [,] [‘a’,1,1.3]

Types define what it contains, and also what operations can be performed on it. Like with a string, you can capitalize it.

Converting types:

So “123” is not the same as 123

int(“123”) goes to 123

Examples

int(2.1) goes to 2, but int(‘2.1’) fails

float(2) goes to 2.0

str(2) goes to ‘2’

Operators

* Integers: +, -, \*, /, // is integer quotient, % is remainder
* Floating points: +, -, \*, /

// or integer division,

Augmented assignment

Shortcut for operations.

my\_int += 2 is the equivalent of my\_int = my\_int + 2

**Developing an Algorithm**

Algorithm- a sequence of steps to solve a problem

Ex. If one inch of rain falls on an acre of land, how many gallons of water have accumulated on that acre?

1. Find the volume in cubic ft of water.
2. Convert the volume to gallons

1 acre = 43,560 square feet

1 cubic foot = 7.48051945 gallons

1 inch = 1/12 foot

volume = depth \* area = (1/12 \* 43560)

# calculate rainfall

inches\_str = input(“How many inches of rain have fallen: “)

inches\_float = float(inches\_str)

volume = (inches\_float / 12) \* 43560

gallons = volume \* 7.48051945

print(inches\_float, “ inches of rain on 1 acre is”, gallons, “gallons”)

Rule: Test your code, often and thoroughly.

2/2/16

Chapter 2

Selection: How programs make choices, and it is the process of making choices that provides a lot of the power of computing.

If Boolean\_expression: True then it will do a series of statements. False it will do a different. However will complete and finish the program regardless.

Boolean operators: <, >, <= (less than or equal to), >=, == (equal to), != (not equal to)

If Boolean expression:

suite

Program will evaluate the Boolean (true or false) if true, execute all statements in the suite.

Elements of the suite must be indented the same distance (4 spaces).

If Boolean expression:

suite1

else:

suite2

If Boolean expression true, run suite1, if false, run suite2.

Basketball algorithm example.

Boolean Expressions. True: any nonzero number or nonempty object. False: a zero number or empty object.

Two senses of equality, two variables referring to different objects.

Boolean, not, and, or.

2/4/16

Project 1: gotta be saved as Proj1.py

Gotta download Proj1.zip

Can do multiple assignments

a\_int, b\_int = 2, 3

print(a\_int, b\_int) # prints 2 3

Traditional swap

a\_int = 2, b\_int = 3

have to introduce extra variable temp

temp = a\_int

a\_int = b\_int

b\_int = temp

Chaining for assignment

a\_int = b\_int = 5

print(a\_int, b\_int) # prints 5 5

More Control: Selection

Compound statements

With Booleans, if, and then elif if the thing is mutually exclusive.

If 90 <= percent\_float:

print(“you received and A”)

elif 80 <= percent\_float < 90

print(“you received a B”)

A perfect number

6 because it is 1+2+3

Design. Prompt for a number, collect factors, sum up factors,

Repeating statements ARE THINGS.

Loop, do test on Boolean

2/9/16

Perfect number example cont.

Work with a range of numbers

For each number, collect the factors, sum up the factors, compare the sum and the number and respond accordingly.

Top\_num\_str = input (“upper number for range: “)

Top\_num = str(topnumstr)

Number = 2

While number <= top\_num:

(picture on phone)

you have to do the steps, and the += 1 at the bottom does it for each number up to upper limit at 30. It’s a loop.

Developing a while loop

* Initialize the variable, outside of the loop
* If the loop never starts, the control variable isn’t initizlized
* If the loop never ends, the control variable is not modified during the loop.

Loops are confusing: read dat shit in the textbook.

While loop can have an associated else suite. Else suite is executed when the loop finishes under normal conditions. (Last thing it does as it exits)

Can’t use this or points off: Break statement. Exits loop immediately.

Random is a module.

Continue statement

A continue statement, in a loop, means to immediately jump back to top of loop and reevaluate the conditional.

While loops are easiest to read when the conditions of exit are clear. Don’t use break though. And pretty much don’t use continue.

Seriously, figure out wtf loops are (while).

For and iteration

Different kind of loop: for loop.

The for statement iterates through each element of a collection (list, etc.)

for element in collection:

suite

Range function

* The range function represents a sequence of integers
* Takes 3 arguments: beginning of range, end of range, and the step of range.

Iterating through the sequence

for num in range(1,5)

print(num)

2/11/16

fuckin algorithms

bangin

2/16/16

sequence of characters. Strings.

“”” triple quotes preserve vertivcal and horizontal formatting of str “””

non-printing characters

string representation:

every character is “mapped” with an integer

2/18/16

functions

-funtion: a program that peforms an operation. Its details are hidden, only its interface is provided. It takes some number of inputs and returns a value based on the arguments nad the function’s operation.

Stirng method

-a method is a variation on a function. Represents a

ex. Upper. Upper is name of a method. Generates a new string all uppercase

my\_str = “Python Rules!”

my\_str.upper() #makes it “PYTHON RULES!”

dot notation

my\_str

string formatting

{} in a string .format(,)

2/23/16

Chapter 5

Files and Exceptions I

When opening a file, you create a file object or file stream that is a connection between the file information on disk and the program.

Buffering, Computer reads data from a file, contains a copy of the file called a cache.

File Reads

fileObject.readLines()

for line in fileObject: #iterator to go through the lines of a file.

How to deal with problems

Exceptional stuff, includes errors. Opening nonexistent files, stuff like dat.

Also includes events. Ending conditions.

Non-local control, when it sees exceptions, catches is and handle it.

Try command.

2/25/16

Functions. Does a thing and returns one value. Ex. sqrt()

3/1/16

data structures

-built in data structures, or user-defined data structures

lists, string, tuples, etc.

lists, have constructors

indexing the [] operator

[1, 2, 3] + [4] #gives [1, 2, 3, 4]

[1, 2, 3] \* 2 #gives [1, 2, 3, 1, 2, 3]

lists values can be changed

my\_list = [1, 2, 3]

my\_list[0] = 127

print(my\_list) #prints [127, 2, 3]

lists have methods like .append which adds another thing to the end of the list