**List, Strings, Dictionary**

* If you SLICE either a string or list, it will return the same way!A list stays alist,
* If I SLICE it cuts off at END POSITION [:3] prints up to 2!
* Look out for identifying what is what! “ = str, [] = lst {}= dict
* THERE ARE NO QUOTES IN OUTPUT OF STRING OR PRINT UNLESS LIST
* Watch LIST PLACEMENT always START WITH
* KEYS ARE LEFT VALUES ARE RIGHT
* t = (20, 30, 40) to x, y, z = t
* APPEND is for a LIST + is for a STR
* **DON’T FORGET TO SPLIT STR IN DICTIONARY ACCUMULATION**
* Str1>str2 evaluates letter to integer in order a= 1b=2
* Insert places it before the letter
* Using the IN method for DICT defaults to KEYS

**Integers, methods, and Boolean**

* FLOAT is always dominant
* == can produce a BOOLEAN response
* SPLIT always makes it into a LIST
* FIND makes an INTEGER
* If I SPLIT with [] it will CUT OFF AT THE SPLIT INVOCATION
* s = "<published>2009-01-23T20:04:53Z</published>"
* INDEX cause an ERROR find returns -1
* Pf.find(word) == 0 for prefix!

print (s.split('T')[0].split('2'))= ['<published>', '009-01-', '3']

* ELIF does not require an output
* IF and ELSE MUST HAVE OUTPUT

**Entropy, Accumulation, Default values**

* LESS REPETITIVE ALWAYS HAS MORE INFO MENTION IT LESS REPETITIVE
* ENTROPY IS 50/50 surprisal is more like 5%. SMALLER ODDS VS LESS PREDICTABLE
* If we ITERATE and PRINT IS INDENTED IT WILL PRINT = len(elements)
* It will just print whatever the output is as many times as list, string, dict(keys
* Always ACCUMULATE a WHILE LOOP
* DEFAULT VALUE can be defined outside of function
* DEFAULT VALUES GO IN ORDER OF ( )
* To print nested value: for e in d: (tab) print e[position you want]
* .keys method produces a LIST
* WE APPEND TO A LIST ADD TO A STRING
* Accum: a = 0

while a <= 21:

x = int(raw\_input(‘Give me a number))

a = x + a(AND THEN PRInt

*To count WORDS:* for words in a:

time = words.split()

accum = accum + len(time)

Sandwich accumulation: def determine\_price(sandwich, prices):

total = 0

ingridients = sandwich.split(', ')

for a in prices:

if a in ingridients:

total = prices[a] + total

return total

BACKWARDS + b = ‘’ then iterate, iterate variable to b = a + b

a = open('studentdata.txt', 'r')

b = a.readlines()

for c in b:

**d = c.split()**

if len(d[1:]) >= 6:

print d

TRANSLATION a = {'hello': 'avast', 'my': 'me', 'is': 'be'}

b = raw\_input('Pirate')

d = b.split()

b = ""

for p in d:

if p in a:

b = b + a[p]+ ' '

else:

b = b + p + ' '

print b

def grep(str, filename)

f=open(filename, 'r')

    lines = f.readlines()

    for lin in lines:

if str in lin:

print lin

lossy compression: what you don't know. With lossless compression, every single bit of data that was originally in the file remains after the file is uncompressed.