413-513 Notes- Week 6

[ Review validations ]

user defined vs. system defined

**built-in exceptions**

<https://docs.python.org/3/library/exceptions.html>

[ List processing ]

Processing a list into sections:

**Split** method

The method **split()** returns a list of all the words in the string, using *str* as the separator (splits on all whitespace if left unspecified), optionally limiting the number of splits to *num*.

str.split(str="", num=string.count(str)).

**str** -- This is any delimeter, by default it is space.

**num** -- this is number of lines to be made

str = "Line1-abcdef \nLine2-abc \nLine4-abcd";

print str.split( );

print str.split(' ', 1 );

result:

['Line1-abcdef', 'Line2-abc', 'Line4-abcd']

['Line1-abcdef', '\nLine2-abc \nLine4-abcd']

**x = 'blue,red,green'**

**x.split(",")**

**['blue', 'red', 'green']**

**>>>**

**>>> a,b,c = x.split(",")**

**>>> a**

**'blue'**

**>>> b**

**'red'**

**>>> c**

**'green'**

**>>> word = "This is some random text"**

**>>> words2 = word.split(" ")**

**>>> words2**

**['This', 'is', 'some', 'random', 'text']**

**strip** method

The method **strip()** returns a copy of the string in which all chars have been stripped from the beginning and the end of the string (default whitespace characters).

str.strip([chars]);

str = "0000000this is string example....wow!!!0000000";

print str.strip( '0' );

result:

this is string example....wow!!!

Special Concepts: lambdas / list comprehensions

Lambdas

Lambdas enable shorter and cleaner code, thus making code quicker to write and more readable.

Consider the following examples, first without lambdas:

squares = [x \*\* 2 for x in range(1,11)]

def f(x):

if x > 30 and x < 70:

return True

else:

return False

print (filter(f, squares))

Syntax for filter() is as follows:

filter(function, list)

*Use*: It checks every item in the list against the function. If the function evaluates to True, that item is included, if it evaluates to False the item is not included.

...now with lambdas:

squares = [x \*\* 2 for x in range(1,11)]

print filter(lambda x : x > 30 and x < 70, squares)

These two pieces of code do the exact same thing, but the second is shorter and thus there is no need to go hunting in the code for the function definition to see what it does.

List Comprehension

List comprehension is also very useful and provides a quick way of creating a list that contains exactly whats needed. Consider the following two examples, first using a for loop:

squares = []

for i in range(1,11):

if i \*\* 2 % 2 == 0:

squares.append(i \*\* 2)

And now using list comprehension:

squares = [x \*\* 2 for x in range(1,11) if x % 2 == 0]

Both pieces of code output the squares of the numbers 1-10 if the square of that number is an even number (divisible by 2).

**Next class**:

dictionaries & sets!