more python power: list indexing, dictionaries, strings!

---

tinyurl.com/CLPS0950Specialization

warmup: list indexing

```
[1] # in python, list indexing will be start:stop:step for list indexing (the same as the order in the range function).
    # remember! the stop is non-inclusive
    # so if we want the first 3 elements of the list, we can do mylist[0:3]
    groceries = ['apples','bananas','carrots','dragonfruit','eggplant']
    print(groceries[0:3])
```

['apples', 'bananas', 'carrots']

```
[2] # something nice in python is that you can ignore an input and python will assume you mean the first / last element.
    print(groceries[:3])
    print(groceries[3:])
```

['apples', 'bananas', 'carrots']

['dragonfruit', 'eggplant']

warmup: list indexing

```
[4]  # the same thing is true for the step. what if we want every other element?
     print(groceries[::2])
     # and starting at the first element...
     print(groceries[1::2])
     # now stop by the third one...
     print(groceries[:3:2])
```

[‘apples’, ‘carrots’, ‘eggplant]
[‘bananas’, ‘dragonfruit’,]
[‘apples’, ‘carrots’]

```
# you can make lists making multiplication if it's the same element!
a = ['hello']
print(a * 10)
b = ['world']
print((a+b) * 10) # we can use the + combination from last time too!
```

[‘hello’, ‘hello’, ‘hello’ …]
[‘hello’, ‘world’,’hello’,’world’…]

warmup: list indexing

```
# strings in python work like lists! you can use the same [start:stop:step] indexing for both.
myword = '0a1b2c3d4e5f6g7'
print(myword[0:3])
print(myword[::2])
```

0a1
01234567

```
# those math operations also work for strings, since they are lists of characters!
a = 'hello' # see that this doesn't have [] to say it's just the word, not a list with the word
print(a+a)
print(a*3)
```

hellohello
hellohellohello

now you!

bit.ly/clps950_lect44

```
# make a function which takes in a string and returns it backwards.

def turnaround(x):
      return x[::-1]
```

list comprehension

```
# list indexing
# what if you want to do some function on every element in a list?
# we use 'variable for variable in list' to loop through everything. it's just like doing a for loop on one line!
x = [1,2,3,4,5]
y = [val*2 for val in x]
print(y)
```

```
[2, 4, 6, 8, 10]
```

```
# you don't have to use the variable you are iterating through!
# just like you could do a for loop for a range, and then just print hello on each iteration.
zz = ['hello' for val in x]
print(zz)
```

```
['hello', 'hello', 'hello', 'hello', 'hello']
```

list comprehension

```
val # BUT unlike lists, it doesn't hold onto that value!
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-14-22e33626e689> in <module>()
----> 1 val

NameError: name 'val' is not defined
```

SEARCH STACK OVERFLOW

```
# but any variable works
z = [egg**2 for egg in x]
print(z)
```

```
[1, 4, 9, 16, 25]
```

list comprehension

```
# let's use the grocery list but I want to put the words backwards! use the function you did above to change every word to backwards, in one line!
groceries = ['apples','bananas','carrots','dragonfruit','eggplant']
flipped_groceries = [val[::-1] for val in groceries] # or [turnaround(val) for val in groceries]
print(flipped_groceries)
```

```
['selppa', 'sananab', 'storrac', 'tiurfnogard', 'tnalpgge']
```

```
# write a function that takes in 2 strings (assume they are the same length) and returns them mixed together every other
# example:
input1 = 'abc'
input2 = '123'
output = 'a1b2c3'
# now try writing a function which takes in 2 strings (assume the same length) and mix them one forwards and one backwards
# example:
input1 = 'abc'
input2 = '123'
output = 'a3b2c1'
```

```
def mixup(in1,in2):
    outp = ''
    for index in range(len(in1)):
        outp += in1[index] + in2[index]
    return outp
input1 = 'abc'
input2 = '123'
output = mixup(input1,input2)
print(output)
```

```
def mixup2(in1,in2):
    outp = ''
    for index in range(len(in1)):
        outp += in1[index] + in2[len(in2) - index - 1]
    return outp
input1 = 'abc'
input2 = '123'
output = mixup2(input1,input2)
print(output)
```