

L15 Functions Scope Enumeration Mutability

Friday, April 29, 2022 3:04 PM

Flapdoodle demo

Friday, April 29, 2022 1:38 PM

```
#TODO C1: add words to each list so that they're at least 10 long. You can make them as long as you like!
nouns = ["cat", "dog", "chocolate", "phone", "money", "secret"]
emotions_I_am = ["angry", "frustrated", "curious", "furious", "gloomy", "cheerful"]
verbs_ing = ["fetching", "eating", "drinking", "singing", "hacking", "breaking"]
names = ["Zelda", "Don", "Mary", "John", "Ross", "Michelle"]
```

String split demo

Wednesday, March 3, 2021 12:56 PM

Without running, can you predict?
Breakout rooms : until poll ready

The screenshot shows a Python Scratchpad window with the following code:

```
PythonScratchpad.py
1 a = "This is a sentence."
2 b = "pink, green, blue"
3 c = """This is a string with \t weird
4
5
6
7 whitespace"""
8
9 s1 = a.split()      #on all whitespace
10 s2 = c.split()     #on all whitespace
11 s3 = c.split(" ") #on spaces only
12 s4 = b.split()     #commas remain
13 s5 = b.split(",") #spaces remain
14 s6 = b.split(", ")#just values remain
```

And a Variables panel showing the state of variables:

Name	Value
a	'This is a sentence.'
b	'pink, green, blue'
c	'This is a string with \t weird\n\n\nwhitespace'
s1	['This', 'is', 'a', 'sentence.']}
s2	['This', 'is', 'a', 'string', 'with', 'weird', 'whitespace']
s3	['This', 'is', 'a', 'string', 'with', '\t', 'weird\n\n\nwhitespace']
s4	['pink', 'green', 'blue']
s5	['pink', 'green', 'blue']
s6	['pink', 'green', 'blue']

Ready to Move on Poll
(Paper demo)

split "eats" Separators

List() and split()

Wednesday, April 27, 2022 1:28 PM

We can turn a string into a list:

```
letters = "abcde"  
list_letters = list(letters)
```

List is one of the sequence types, which come with a bunch of built in methods of their own

See:

<https://docs.python.org/3/library/stdtypes.html#typesseq>

Operation	Result
<code>x in s</code>	<code>True</code> if an item of <code>s</code> is equal to <code>x</code> , else <code>False</code>
<code>x not in s</code>	<code>False</code> if an item of <code>s</code> is equal to <code>x</code> , else <code>True</code>
<code>s + t</code>	the concatenation of <code>s</code> and <code>t</code>
<code>s * n</code> or <code>n * s</code>	equivalent to adding <code>s</code> to itself <code>n</code> times
<code>s[i]</code>	<i>i</i> th item of <code>s</code> , origin 0
<code>s[i:j]</code>	slice of <code>s</code> from <i>i</i> to <i>j</i>
<code>s[i:j:k]</code>	slice of <code>s</code> from <i>i</i> to <i>j</i> with step <i>k</i>
<code>len(s)</code>	length of <code>s</code>
<code>min(s)</code>	smallest item of <code>s</code>
<code>max(s)</code>	largest item of <code>s</code>
<code>s.index(x[, i[, j]])</code>	index of the first occurrence of <code>x</code> in <code>s</code> (at or after index <i>i</i> and before index <i>j</i>)
<code>s.count(x)</code>	total number of occurrences of <code>x</code> in <code>s</code>

You'll probably want to use many of these for A3. One you will need is

```
" ".split()
```

What does this do?

<https://docs.python.org/3/library/stdtypes.html?highlight=split#str.rsplit>

It can split a sentence into a list of each word.

```
Sen = "this is a sentence"
```

```
Sen_list = sen.split(" ")
```

There are many string methods you'll find useful in this class. We'll go over ones required for the HW and exams, but you can use any of the others; often they'll make your life easier.

Try it!

```
>>> %Run bingo.py
```

```
There was a farmer had a dog, and Bingo was his name-o  
B i n g o B i n g o B i n g o and Bingo was his name-o  
There was a farmer had a dog, and Bingo was his name-o  
* i n g o * i n g o * i n g o and Bingo was his name-o  
There was a farmer had a dog, and Bingo was his name-o  
* * n g o * * n g o * * n g o and Bingo was his name-o  
There was a farmer had a dog, and Bingo was his name-o  
* * * g o * * * g o * * * g o and Bingo was his name-o  
There was a farmer had a dog, and Bingo was his name-o  
* * * * o * * * * o * * * * o and Bingo was his name-o  
There was a farmer had a dog, and Bingo was his name-o  
* * * * * * * * * * * * * * * * and Bingo was his name-o
```

```
1 dog = "Bingo"  
2 dog_as_list = list(dog)  
3  
4 for i in range(6):  
5     print("There was a farmer had a dog, and Bingo was his name-o")  
6     for k in range(3):  
7         for j in range(5):  
8             print(dog_as_list[j], end = " ")  
9         dog_as_list[i%len(dog_as_list)] = "*"  
10        print(" and Bingo was his name-o")
```

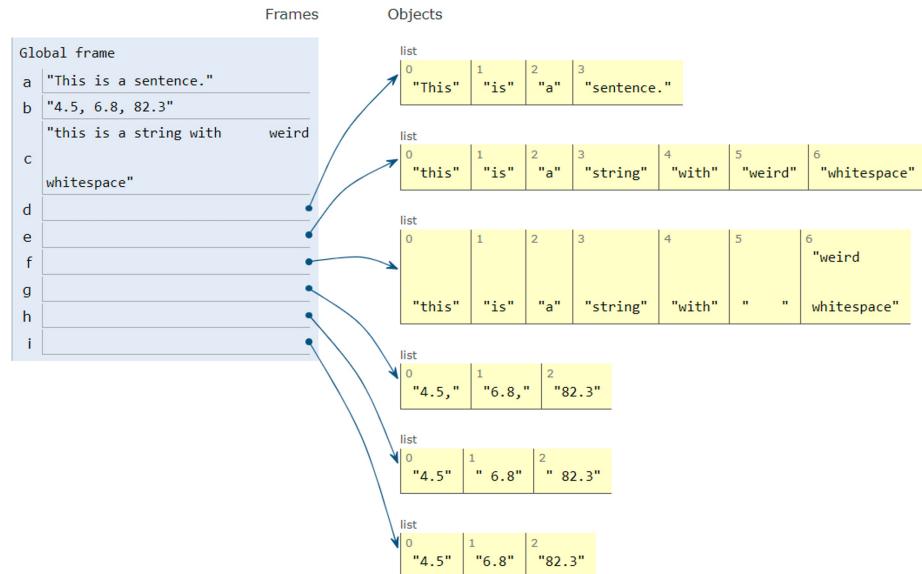
String split details

Thursday, February 24, 2022 11:44 PM

```
1 a = "This is a sentence."
2 b = "4.5, 6.8, 82.3"
3 c = """this is a string with \t weird
4
5
6 whitespace"""
7
8 #we can break strings up into lists
9 #which is really useful for reading in
10 #comma separated data from a file
11 #and putting it into a list for using|
12
13 d = a.split() #on all whitespace
14 e = c.split() #on all whitespace
15 f = c.split(" ") #on spaces only
16 g = b.split() #commas remain
17 h = b.split(",") #spaces remain
18 i = b.split(", ") #just the values
```

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>



Lists mutability and functions

Wednesday, March 3, 2021 12:28 PM

what if:

def Z1(A_list):
 A_list[0] = 0

A = [0, 1, 1]

Z1(A)
Print(A)

I min: think (therdemo)

What if

def Z2(A_list):
 A_list = []

A = [1, 1, 1]

Z2(A)

Print(A)

Think: draw

A → [1, 1, 1]

A_list → []

ABCD

Mutable Objects and Functions

11-

Mutable Objects and Functions

```
def z0(y):  
    y[0] = 4  
    return y  
  
b = [5, 6]  
c = z0(b)  
print(b[0], c[0])
```

What does this code print?

A. 4 4
B. 4 5
C. 5 4
D. 5 5

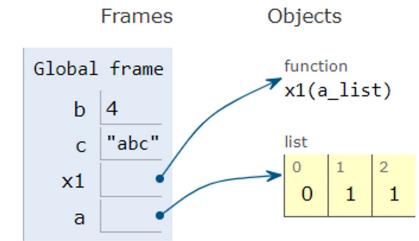


(in beatout from slices)

Passing lists

Monday, October 17, 2022 9:13 PM

```
1 b = 4
2 c= "abc"
3
4 def x1(a_list): #a_list is the parameter which receives the argument
5     a_list[0] = 0
6
7 a = [1,1,1]
8 x1(a) # a is the argument
9 print(a)
10
11 #but if you pass a list into a function , and change
12 #the list in the fuction, you've changed the list
13 #for everyone|
```



Bingo example of strings

Saturday, April 30, 2022 12:30 PM

```
2 dog = "Bingo"
3
4 dog_as_list = list(dog)
5
6
7 for i in range(6):
8     print("There was a farmer had a dog, and Bingo was his name-o")
9     # The letters repeat 3 times
10    for k in range(3):
11
12        #each time through it does one less letter
13        for j in range(5):
14            print(dog_as_list[j], end = " ")
15            dog_as_list[i] = "*"
16
17    print(" and Bingo was his name-o")
```

Lists in lists

Friday, October 28, 2022 12:45 PM

```
tops = ["shirts", "blouse", "jacket"]
bottoms = ["pants", "skirt", "skort", "kilt"]
shoes = ["boots", "wedges", "sneakers"]
```

```
clothing = [tops, bottoms, shoes]
```

```
#print out all the different types of clothing
#print(clothing[0][2])
```

```
for i in range(len(clothing)):
    #print(clothing[i])
    for j in range(len(clothing[i])):
        print(clothing[i][j])
    print(i,j)
```