16 tuples

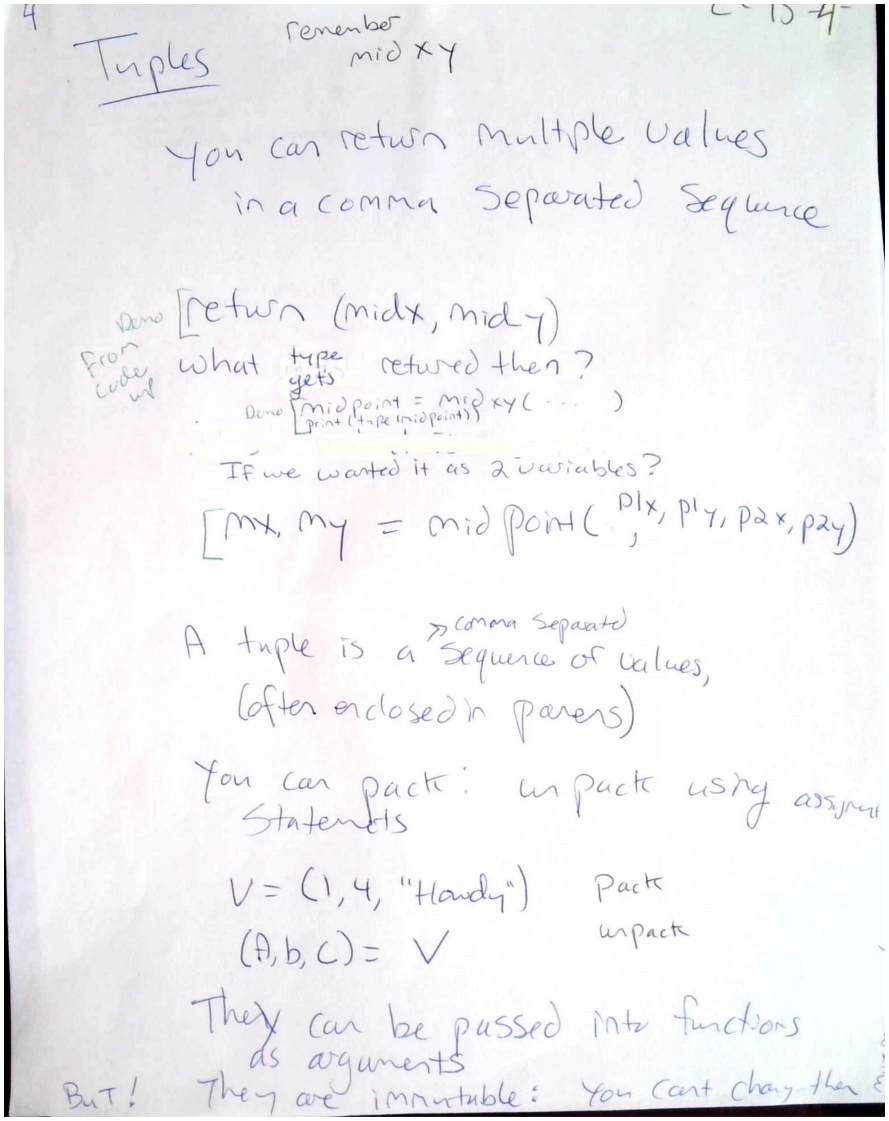
Monday, May 9, 2022 10:03 PM

Tuples 1 motivation

Tuesday, April 26, 2022 10:15 PM

```
drinks = ["2% milk", "Diet coke, large", "strawberry milkshake"]
food = ["burger", "veggie burger"]
condiments = ["Ranch dressing", "ketchup", "wasabi"]

bag = [drinks, food, condiments]

#          0                    1                          2
# 0 [ "2% milk",          "Diet coke, large", "strawberry milkshake"]
# 1 ["burger",            "veggie burger"]
# 2 ["Ranch dressing",  "ketchup",          "wasabi"]

```

```
'''
looks for the food(string) in the bag(list of lists) and returns the loc
parameters : food(string), bag(list of lists)
return coordinates which list and at which index
'''
def find_my_food(bag, food):

    #do a 'for' loop over each list
    # and a 'for' loop within each list, looking for a match

    loc1 = 0 #todo temp value : replace once 'for' loops are done
    loc2 = 2 #todo temp value : replace once 'for' loops are done

    location = "?? we want to return two things, loc1 and loc2...."

    return location

found = find_my_food(bag, "strawberry milkshake")
```




```
26      location = (loc1, loc2) # loc 1, loc2 bundled together
27      print(type(location))
28
29      return location
30
31  found = find_my_food(bag, "strawberry milkshake")
32
33  f1, f2 = found #unpack the tuple
34
35  print(bag[f1][f2])
```

Tuples packing and unpacking

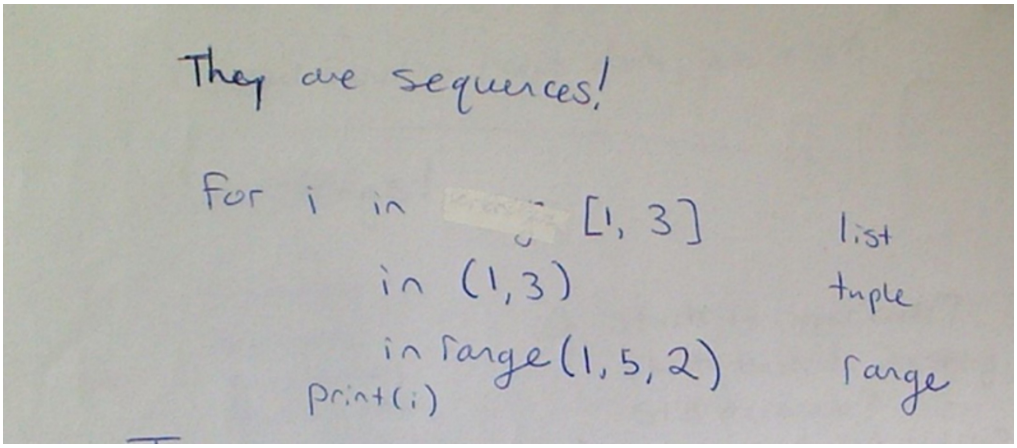
Tuesday, April 26, 2022 10:20 PM

```
my_tuple = (1,2,3) #packing a tuple

a,b,c = my_tuple #unpacking a tuple

#d,e = my_tuple #this won't work: need same num of variables
            #as values in the tuple!

(f,g,h) = my_tuple #still just three variables
        #this did not make a new tuple
```

```
a = 1
b = 2
c = 3

v = a, a, c

print(v)

# What does this print?
# A: 1 2 3
# B: 1 1 3
# C: (1, 2, 3)
# D: (1, 1, 3)
```

(answer is D)

```
3  def a():
4      print("a")
5  def b():
6      print("b")
7  def c():
8      print("c")
9
10
11 letters = [a,b,c]
12 a=7
13 print(a)
14
15 letters[0]()
16
17
18 #this will print :
19 # 7
20 # a
```



Write down what is in memory at each step:
will any of the steps crash?

```
a,b, c = 7, 5, 3
(z,x) = c, b
w = (z,x)
print((x,z))
print(w)
v = (x, z, c, w)
print(v)
a, b = v
```

a,b, c = 7, 5, 3
(z,x) = c, b
w = (z,x)
print((x,z))
print(w)
v = (x, z, c, w)
print(v)
a, b = v

```
#lists are sequences
for letter in ["a","b"]:
    print(letter)

#tuples are sequences!
for number in (1,3):
    print(number)

#can you write a for loop that uses 'range'
#that prints the same numbers?

#A: range(1,3)
#B: range(3)
#C: range(1,3,2)
#D: range(1,5,2)

#answer:
for i in range(1,5,2):
    print(i)
```

Lists vs Tuples

Tuesday, February 23, 2021 12:45 PM

Lists vs Tuples

Tuples are immutable → can't mutate (change)

Lists are mutable → can mutate

Examples

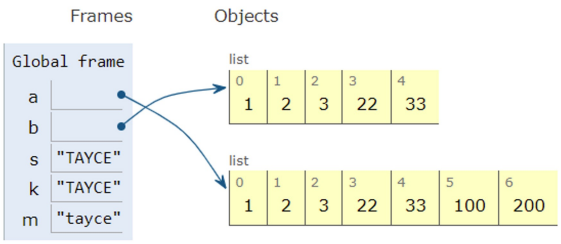
```
A_tuple = (1,2,3)
A_list = [1,2,3]
A_tuple[1] = ∅   ✗ Error!
A_list[1] = ∅    ✓ ok!
```

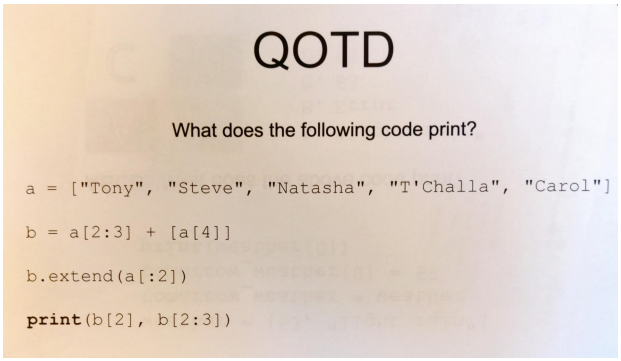
List extend vs +

Tuesday, March 1, 2022 11:12 PM

Write code in Python 3.6 (drag lower right corner to resize code editor)

```
1  a = [1,2,3]
2
3  a.extend([22,33]) #extend mutates it in place
4
5  b = a             #just points to the existing list
6                    #we'll do copies in an upcoming lecture
7
8  a = a + [100,200] # plus makes a whole new list!
9                    # and the old list is left...as long as something
10                   #is pointing to it it'll stay around
11                   # (won't get garbage collected)
12                   # more advanced CS classes go into how to know that
13                   # it's still being used!
14 s = "TAYCE"
15 k = s  #fresh copy, not pointing to the same one
16 s.lower() #this doesn't change anything, it returns something
17 m = s.lower() #this makes a new copy and saves it back
```





Return values vs effects

Wednesday, February 23, 2022 10:52 AM