Arjun Bastola

11 Followers · About Follow

Learn Python Basics while making Greek Salad

Arjun Bastola Feb 22 · 5 min read ★

So my little cousin wanted to learn basic Python programming and called me.

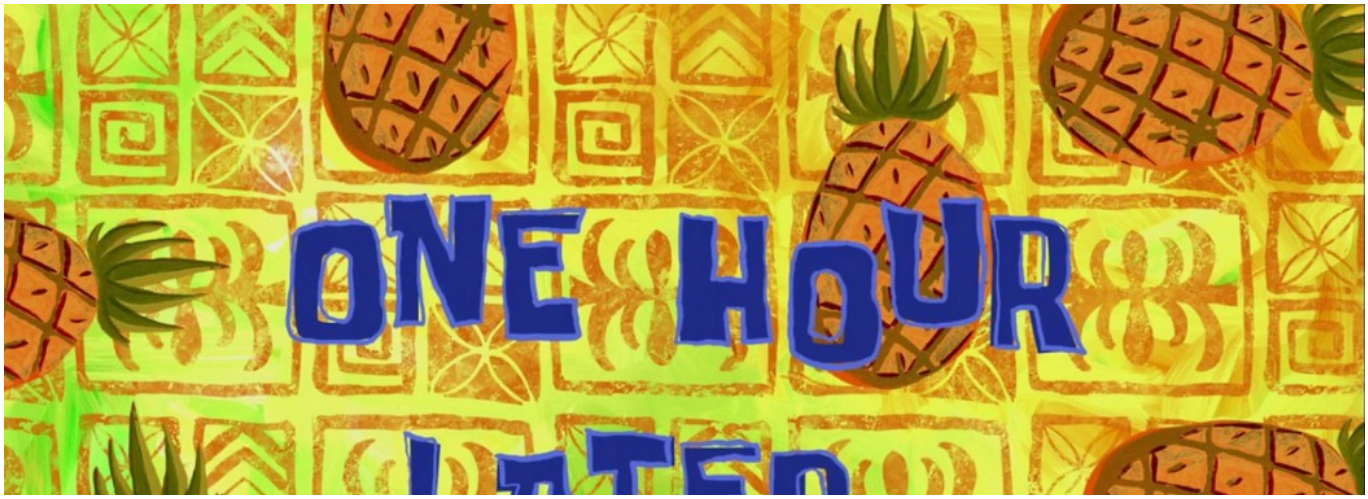
Her: “I have been having hard time learning Python from this professor. Will you be able to help?

Me: “Of course. Let’s do this. Tell me what you ate for lunch.”

Her: “Greek Salad.”

Me: “Meet me at Dunkin in an hour and I will teach you how to make Greek Salad in Python.”

Her: “*Kekeke*.” (Her version of okay. Weird but cute.)



Source: https://i.ytimg.com/vi/MXcMV-d_2Js/maxresdefault.jpg

Me and my cousin agreed that salad making process will involve 4 major steps:

Step 1: Get all the items out.
Step 2: Cut all veggies or whatever you want to put in the salad.
Step 3: Mix all the items.
Step 4: Eat.

Step 1: Getting all the items out.

Alright, so we will need a bowl to mix the salad, a chopping board, a knife and all the ingredients (lettuce, onion, cucumber, cherry tomatoes, olives, goat cheese, dressing).

Now, let's write a python statement that indicates/asserts that we took out bowl from the cabinet. We can simply use **print** statement to do this.

```
print("Bowl is out.")
```

Now, let's take out a chopping board and also a knife.

```
print("Chopping board is out.")
print("Knife is out.")
```

Now, let's take out ingredients.

```
print("Lettuce is out.")
print("Onion is out.")
...
```

is out" to "… is out now"? You will have to go back and change every single print statement.

However, we can use **functions** in Python to eliminate this problem. Let's create a list of items we need to take out first. Better yet, let's make two lists:

1. Utensils and utilities required to make salad.
2. Ingredients for salad.

You can create a Python list by simply putting all items inside square brackets '[]'. So let's go ahead and create two lists.

```
utensils = ["Chopping Board", "Knife", "Bowl"]
ingredients = ["Lettuce", "Onion", "Cucumber", "Cherry Tomatoes",
"Olives", "Goat Cheese", "Dressing"]
```

Now let's create a function that accepts the name of the item and takes that item out of the cabinets or fridge.

```
def takeItemOut(nameOfItem):
   print(nameOfItem, "is out")
```

Now let's create a **loop** that loops through the items and calls the takeItemOut function for each item. Also notice that weird # symbol? It's Python way to write a single line comment.

```
# For ingredients

for ingredient in ingredients:
   takeItemOut(ingredient)

# For utensils
```

Now that we have all of items, we can now finally start cutting the ingredients.

Step 2: Cut all veggies or whatever you want to put in the salad.

Step 2 will be very similar to step 1. We will loop through the items and pass each item to a function that cuts the item to small pieces.

Let's write the function and the loop that cuts all our ingredients.

```
def cutTheIngredient(ingredient):
   print(ingredient, "has been diced")

for ingredient in ingredients:
   cutTheIngredient(ingredient)
```

Alright so we did it. We took out items and diced the ingredients. However, not all the ingredients are diced the same way. Lettuce should be diced multiple times while olives should be diced maybe 3 times.

So, how do we incorporate this information in our program. Python has a pretty slick collection called ***tuple***. In Python tuples are written with round brackets.

```
personInfo = ("John", "Doe", "Full Stack Developer")
```

Here personInfo is a tuple that has three items: first name, last name and title.

We can access the tuple's item by using the index or by tuple unpacking.

```
# Using Index
# This will print: Doe

print(personInfo[1])
```

```
firstName, lastName, title = personInfo
print(lastName)
```

Now, back to our salad.

Let's create a list of tuples with the name of item as well as how many times we have to chop the item.

```
items = [("Lettuce", 50), ("Onion", 10), ("Cucumber", 20), ("Cherry
Tomatoes", 2), ("Olives", 3), ("Goat Cheese", 20), ("Dressing", 0)]
```

("Lettuce", 50) means we are going to cut lettuce 50 times. Also we can't cut dressing and therefore we have 0 next to it.

Now let's write a function that accepts the tuple as a parameter and cuts the item into given number of pieces. Also, let's write a loop that loops through the items and passes each item to the function.

```
# Function to cut an item for x number of times.

def cutTheItemMultipleTimes(itemData):
   nameOfItem, cutTimes = itemData # Tuple unpacking
   for i in range(0, cutTimes):
      print("Cutting", nameOfItem, ".")

# For loop to loop through all the items and call the function.

for item in items:
   cutTheItemMultipleTimes(item)
```

Sample output when tomatoes and olives are passed:

```
Cutting Cherry Tomatoes.
Cutting Cherry Tomatoes.
Cutting Olives.
```

We have all the chopped items now. Now we are ready to mix them.

Step 3: Mix all the items.

We will again use for loop and a function to mix all the items. But this time instead of using the **ingredients** list, we will be using **items** list that we created just now.

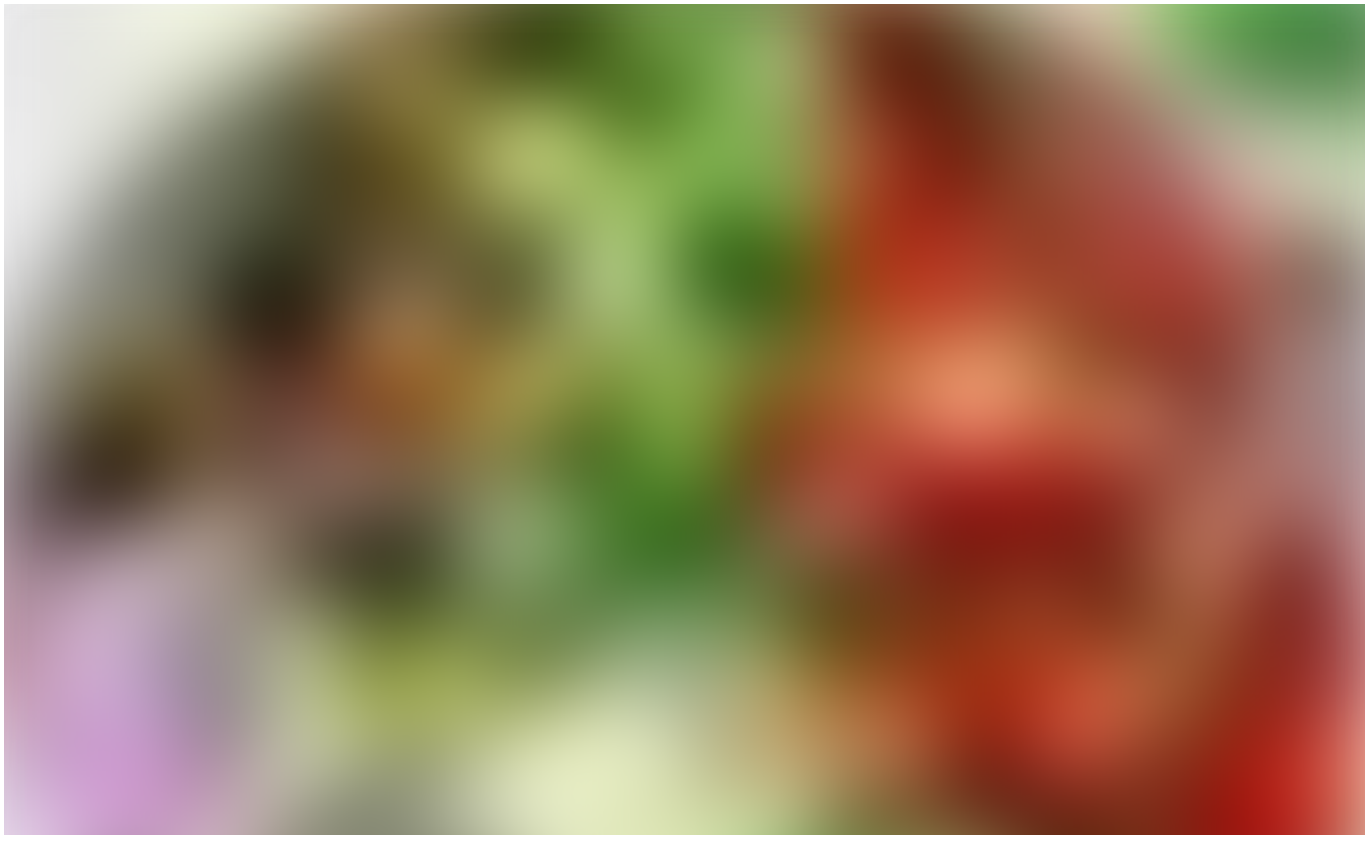
```
# Function to mix the item.

def mixTheItems(itemData):
   nameOfItem, cutTimes = itemData # Tuple unpacking
   print("Mixing", nameOfItem, ".")

# For loop to loop through all the items and call the function.

for item in items:
   mixTheItems(item)
```

We now have mixed all the items and we have a bowl of yummy greek salad.



Source: https://www.healthyseasonalrecipes.com/wp-content/uploads/2014/05/chopped-greek-salad-sq-024.jpg

Step 4: Eat.

Eating is simple. However, life isn't fun if everything goes according to plan. So, let's make this little more interesting.

My cousin is tiny and she only eats like 10 bites of salad. So we are going to track her number of bites and stop her after her 10th bite.

We will use a variable called numberOfBites to track her bites. We will use an **infinite while** loop and a condition check to stop her at 10th bite.

```
numberOfBites = 0

'''
An infinite loop: the loop will run forever because the condition is
always true.
'''

while(True):

   # Increase number of bites each time.
   numberOfBites = numberOfBites + 1
   print("Eating. Total Bites:", numberOfBites)

   # If statement to check if bites is 10 and break the loop.
   if numberOfBites == 10:
      break;
```

Arjun Bastola is a Data Analyst and Full Stack Developer based out of New York City Area. Github: abastola

Python Basic Python Tutorial Food Beginner Programmers