Learning Python #3

The Computer Science Team

Table Of Contents

- Solutions (Last week's problems)

- Learning sets (Working with items...)

- Learning functions (Making your own...)

- Practice problems

Solutions...

Problem #1

Ask the user to enter 5 types of fruits ~ Try to use a loop
Create a list that stores these 5 fruits ~ Print the list

```python
1   #create an empty list
2   types_of_fruits = []
3
4   #create an integer type variable that asks the user the # of items they would like to enter
5   number_of_fruits = int(input("Enter the number of fruits you would like to add to the list: "))
6   print() #printing empty line
7
8   #for loop that starts from the 0th value to the value the user entered for the variable "number_of_fruits"
9   for i in range (0, number_of_fruits):
10      fruits = str(input()) #gathering user input and storing the users values in a string type variable called fruits
11      types_of_fruits.append(fruits) #adding the value to the list using the append function
12
13  print() #printing empty line
14  print(types_of_fruits) #printing the list
```

Solutions...

Problem #1

Ask the user to enter 5 types of fruits ~ Try to use a loop
Create a list that stores these 5 fruits ~ Print the list

```
Enter the number of fruits you would like to add to the list: 5

Oranges
Apples
Cherries
Grapes
Dragonfruit

['Oranges', 'Apples', 'Cherries', 'Grapes', 'Dragonfruit']
```

Solutions...

Problem #2

Create a dictionary containing 5 different cuisines(Chinese, Italian, Indian etc..) name your favorite dish in each cuisine as the "value" to the "key". Then print each favorite dish one by one using a for loop.

```python
# Create a dictionary containing 5 different cuisines(Chinese, Italian, Indian etc..)
# name your favorite dish in each cuisine as the "value" to the "key". Then print each
# favorite dish one by one using a for loop.

favorite_dishes = {'Italian':'Lasagna', 'American':'Cheeseburger',
 'Chinese':'Dumplings', 'Indian':'Butter Chicken', 'Mexican':'Enchiladas'}


for val in favorite_dishes:
    print(favorite_dishes[val])
```

Solutions...

Problem #2

Create a dictionary containing 5 different cuisines(Chinese, Italian, Indian etc..) name your favorite dish in each cuisine as the "value" to the "key". Then print each favorite dish one by one using a for loop.

```
Code/Computer Club/Week3Problem2.py"
Lasagna
Cheeseburger
Dumplings
Butter Chicken
Enchiladas
PS C:\Users\LouisVuitton> []
                                                        Ln 11, Col 1    Spaces:
```

What if we wanted to output in a single line?

```
C: > Users > LouisVuitton > Desktop > Code > Computer Club > 🐍 Week3Problem2.py > ...
 1    # Create a dictionary containing 5 different cuisines(Chinese, Italian, Indian etc..)
 2    # name your favorite dish in each cuisine as the "value" to the "key". Then print each
 3    # favorite dish one by one using a for loop.
 4
 5    favorite_dishes = {'Italian':'Lasagna', 'American':'Cheeseburger',
 6     'Chinese':'Dumplings', 'Indian':'Butter Chicken', 'Mexican':'Enchiladas'}
 7
 8
 9    for val in favorite_dishes:
10        print(favorite_dishes[val], end=" ")
11
```

```
Code/Computer Club/Week3Problem2.py"
Lasagna, Cheeseburger, Dumplings, Butter Chicken, Enchiladas,
PS C:\Users\LouisVuitton> []
```

Learning Sets

↝ What are sets?

- Store multiple items in a single variable
- Considered a **_sequence/collection_** type variable
- **_Sequence/collection_** type variables (List, tuple, dictionaries)
- Collections that are unordered and unindexed and do not allow duplicates
- Written with curly brackets

Creating Sets Method #1

↝ Let's create a
set

```
16    #creating a set
17    cars = {"mazda", "honda", "lambo"}
18
19    #printing a set
20    print(cars)
```

↝ Let's evaluate the output

```
{'honda', 'lambo', 'mazda'}
```

```
{'mazda', 'lambo', 'honda'}
```

```
{'honda', 'mazda', 'lambo'}
```

Creating Sets

↝ Let's create duplicate values within a set

```
31    #creating duplicate values within sets
32    cars = {"mazda", "honda", "mazda"}
33    countries = {"Canada", "USA", "China", "China", "USA", "Mexico"}
34
35    #printing the sets
36    print(cars)
37    print()
38    print(countries)
```

↝ Let's evaluate the output

```
{'honda', 'mazda'}

{'Mexico', 'China', 'USA', 'Canada'}
```

```
{'mazda', 'honda'}

{'Canada', 'Mexico', 'USA', 'China'}
```

Creating Sets
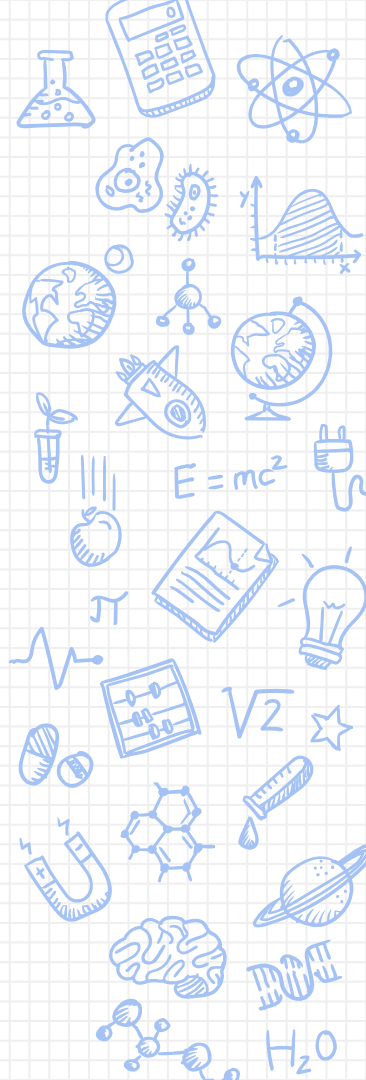
↝ Let's try to call a specific index within the set

```
43   #creating a set
44   cars = {"mazda", "honda", "mazda"}
45
46   #printing a specific index of the set
47   print(cars[1])
48   print(cars[2])
49   print(cars[3])
```

↝ Let's evaluate the output

```
Traceback (most recent call last):
                                                                          line 47, in <module>
    print(cars[1])
TypeError: 'set' object is not subscriptable
```

What does this mean?

↝ Unordered

- – Items in sets do not have a defined order
- – Can appear in a different order every time it is used/printed

↝ No Duplicates

- – Cannot have two or more items with the same EXACT value
- – Values are case sensitive!
- – When printed it will remove the duplicates and print the value only once

↝ Unindexed

- – Cannot be referred to by an index

Going In Depth!

↝ Unchangeable

- Sets are unchangeable
- Once the set has been created you cannot change the items in value
- But you can still add and remove items (just not change the

↝ Adding items

- You can add items using the "add()" function
- "add()" does not add an element to the set if it's already present in it

↝ Removing items

- You can remove items using the "remove()" function
- If the item passed in "remove()" does not exist ~ "KeyError"

Adding Items

Example 1: Add an element to a set

```python
# set of vowels
vowels = {'a', 'e', 'i', 'u'}

# adding 'o'
vowels.add('o')
print('Vowels are:', vowels)

# adding 'a' again
vowels.add('a')
print('Vowels are:', vowels)
```

Output

```
Vowels are: {'a', 'i', 'o', 'u', 'e'}
Vowels are: {'a', 'i', 'o', 'u', 'e'}
```

> **Note:** Order of the vowels can be different.

14

Removing Items

Example 1: Remove an Element From The Set

```python
# language set
language = {'English', 'French', 'German'}

# removing 'German' from language
language.remove('German')

# Updated language set
print('Updated language set:', language)
```

Output

```
Updated language set: {'English', 'French'}
```

Removing Items

Example 2: Deleting Element That Doesn't Exist

```python
# animal set
animal = {'cat', 'dog', 'rabbit', 'guinea pig'}

# Deleting 'fish' element
animal.remove('fish')

# Updated animal
print('Updated animal set:', animal)
```

Output

```
Traceback (most recent call last):
  File "<stdin>", line 5, in <module>
    animal.remove('fish')
KeyError: 'fish'
```

You can use the set discard() method if you do not want this error.

The `discard()` method removes the specified element from the set. However, if the element doesn't exist, the set remains unchanged; you will not get an error.

Different Data Types

```
89    #creating sets with different values
90    set1 = {"apple", "banana", "cherry"}
91    set2 = {1, 5, 7, 9, 3}
92    set3 = {True, False, False}
93    set4 = {"abc", 34, True, 40, "male"}
94
95    #printing a set
96    print(set1)
97    print(set2)
98    print(set3)
99    print(set4)
```

```
{'cherry', 'apple', 'banana'}
{1, 3, 5, 7, 9}
{False, True}
{True, 34, 'abc', 40, 'male'}
```

Creating Sets Method #2

↝ Let's create a set using the "set()" constructor/function

```
104    #creating sets using the set() method
105    cars = set({"mazda", "honda", "lambo"}) #uses curly brackets
106    foods = set(("Pizza", "Pasta", "Salads")) #uses round brackets
107
108    #printing the sets
109    print(cars)
110    print(foods)
```

```
{'honda', 'mazda', 'lambo'}
{'Pasta', 'Pizza', 'Salads'}
```

What are Functions?

Learning Functions

↝ What are **Functions**?

- A function is a block of code which **only runs when** it is **called**.
- You can pass **data(parameters)**, into a function.
- Functions help break our program into smaller modular chunks
- They help clean up code and allow easier usage

Example of a function

```python
def greet(name):
    """
    This function greets to
    the person passed in as
    a parameter
    """
    print("Hello, " + name + ". Good morning!")
```

Calling a Function

```
GradeCalculator.py    Week3Problem2.py    functions.py ✕

C: > Users > LouisVuitton > Desktop > Code > Computer Club > functions.py > ...
  1   def greet(name):
  2       """
  3       This function greets to
  4       the person passed in as
  5       a parameter
  6       """
  7       print("Hello, " + name + ". Good morning!")
  8
  9
 10
 11   greet('Bill')
```

```
Code/Computer Club/functions.py"
Hello, Bill. Good morning!
PS C:\Users\LouisVuitton>
```

Return


```python
def greet(name):
    """
    This function greets to
    the person passed in as
    a parameter
    """
    greet_phrase = "Hello, " + name + ". Good morning!"

    return greet_phrase



print(greet('Bill') + "   (THIS IS BEING PRINTED OUTSIDE OF THE FUNCTION!!)")
```

```
Code/Computer Club/functions.py"
Hello, Bill. Good morning!   (THIS IS BEING PRINTED OUTSIDE OF THE FUNCTION!!)
PS C:\Users\LouisVuitton> []
```
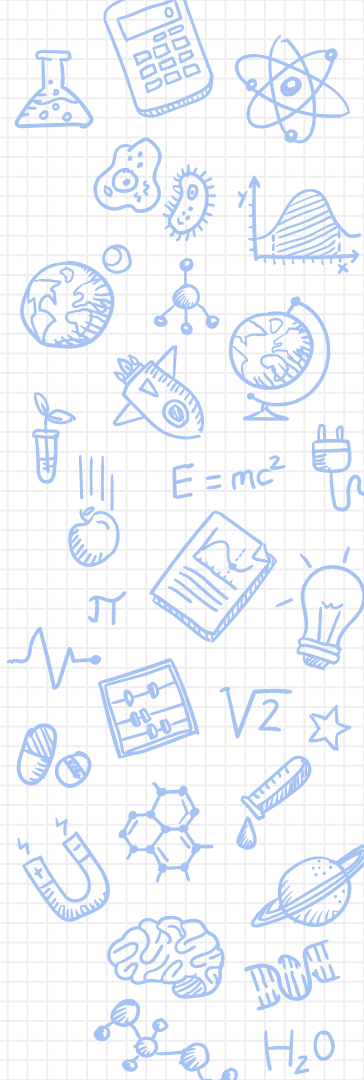
Global Variables

```python
x = "global"

def foo():
    print("x inside:", x)


foo()
print("x outside:", x)
```
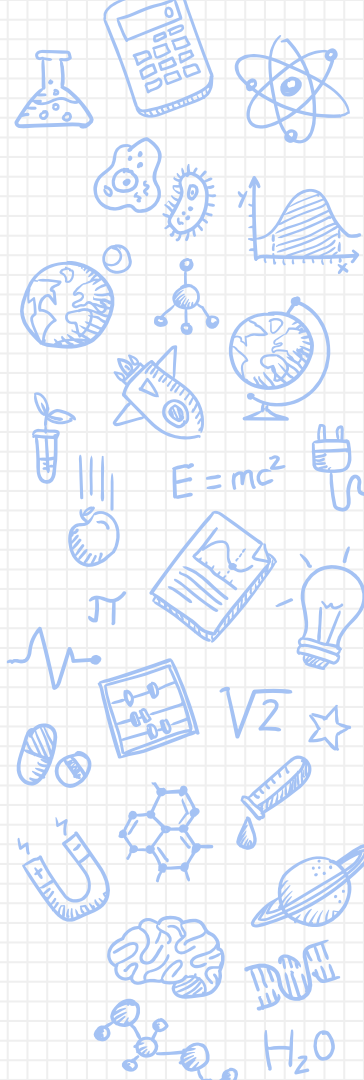
```
x inside: global
x outside: global
```

Local Variables

```python
def foo():
    y = "local"


foo()
print(y)
```

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

Practice Problems

↝ Problem #1

Ask the user to enter 5 countries.

2 of them should be the same exact value (same case and spelling).

Store these values in a set. Then iterate over the set using a loop and print it.

↝ Problem #2

Create the function **calculation()**. Allow it to accept two variables and calculate the addition and subtraction of it. Return both addition and subtraction values in one call and print them to the user.

The End!!!

Any Questions?