

# Python #4



# Revisiting 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.

## Solution

```
Lambda.py > ...
1  # set of countries (1 will repeat)
2  countries = {"Canada"}
3
4  # adding country
5  for i in range(0, 5):
6      add_country = str(input("Enter a country: "))
7      countries.add(add_country)
8
9  # print commands
10 print()
11 print(countries)
12 print()
13
14 # iterate over the set
15 for i in countries:
16     print(i)
```

```
Enter a country: Canada
Enter a country: U.S
Enter a country: Germany
Enter a country: England
Enter a country: Brazil
```

```
{'Brazil', 'Canada', 'Germany', 'England', 'U.S'}
```

```
Brazil
Canada
Germany
England
U.S
```

## Revisiting Problems

---

### → 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.

# Solution

---

Solving together on screen!

## What is Lambda?

---

- A lambda function is a small anonymous function
- Can take any number of arguments, but can only have one expression
- Syntax goes as following:

*Lambda arguments : expression*

- The expression is executed then the result is returned

## Examples of Lambda

```
20 # Section 2 - Learning lambda
21
22 # add 10 to the argument "a", and return the result
23 x = lambda a : a + 10
24 print(x(5))
25 print()
26
27 # lambda functions can take any number of arguments
28 x = lambda a, b : a + b
29 print(x(5, 6))
30 print()
31
32 x = lambda a, b, c : a * b * c
33 print(x(1, 2, 3))
34 print()
```

15

11

6



## Why use Lambda?

---

- Think of it as a shortcut to other mini functions
- Best used as an anonymous function inside another function
- Use lambda functions when an anonymous function is required for a short period of time

Example:

- Let's create a function that takes one argument, and multiplies the argument with an unknown number
- This is a perfect case of when you can use lambda

## Create the Function

---

```
36  # Section 3 - Using Lambda
37  # create a function
38  def myfunc(n):
39      return lambda a : a * n
```

## Use the Function

Let's make the function double a number we send in...

```
38 def myfunc(n):  
39     return lambda a : a * n  
40  
41 mydoubler = myfunc(2) # double it  
42  
43 print(mydoubler(11))
```

22

## Use the Function

Let's make the function triple a number we send in...

```
38 def myfunc(n):  
39     return lambda a : a * n  
40  
41     #mydoubler = myfunc(2) # double it  
42     mytripler = myfunc(3) # triple it  
43  
44     #print(mydoubler(11))  
45     print(mytripler(11))
```

33

## Use the Function

Let's make the function double and triple a number we send in...

```
38  ✓ def myfunc(n):  
39      return lambda a : a * n  
40  
41  mydoubler = myfunc(2) # double it  
42  mytripler = myfunc(3) # triple it  
43  
44  print(mydoubler(11))  
45  print(mytripler(11))
```

22

33

## Practice Problems

---

### → Problem #1

Ask the user to enter 2 numbers.

Create a lambda function to divide numbers by n.

Store 1 number in "n" and store the other number in the value that gets multiplied by n.