ERRORS, LAMBDA, AND LOOP SOLUTIONS

TASK 1: CONVERT FUNCTIONS INTO LAMBDA EXPRESSIONS

WORK WITH

1.

```
def mul(num1= 5, num2= 20, num3= 100):
        return num1 + num2 + num3
```

2.

```
bowl = ["cherries", "orange", "apple", "melon", "figs"]
       def salad(fruit):
            if fruit in bowl:
                print(fruit)
            else:
                print("not in bowl")
```

3.

```
def inside(num):
            if num in list(range(10)):
                return num**2
            else:
                print("outside")
```

DESIRED OUTPUTS

1. 125 **mul()** parameters are left empty.
2. 'not in bowl' or 'apple'
3. 16 or "outside"

HINTS?

1. For two of the **lambda** expresions, use control flow.
2. For the first **lambda** expression, set three default values.
3. Use the **in** statement.

In [1]:

```
mul = lambda num1 = 5, num2 = 20, num3= 100: num1+num2+num3
mul()
```

Out[1]:

125

In [4]:

```
bowl = ["cherries", "orange", "apple", "melon", "figs"]
salad = lambda x : "fruit" if x in bowl else "not in bowl"

salad("mango")
```

Out[4]:

```
'not in bowl'
```

In [6]:

```
inside = lambda num: num**2 if num in range(10) else "outside"
inside(20)
```

Out[6]:

```
'outside'
```

TASK 2: HANDLING ERRORS WITH LAMBDA AND WHILE LOOP

WORK WITH

```
while True:


    try:

        val = int(eval(input("Enter an integer: ")))

        g = lamda x: x*3 if x > 10 else "too low"

        ### CONTROL FLOW GOES HERE  ####

        g(val) > 20

        ##################################

    except:
        continue


    finally:
```

DESIRED OUTPUTS

Enter an integer: 1
Try again please
End code
Enter an integer: 10
Try again please

Try again please
End code
Enter an integer: 12.12345
Lambda is 36
Input value is 12
End code

HINTS?

1. Use a **break** and **continue** statement.
2. Use conditional operator where g(val) **>** 20.
3. Put the completed code in a function called **error()** and run it.

In [7]:

```
def error():

    while True:

        try:
            val = int(eval(input("Enter an integer: ")))

            g = lambda x:x*3 if x > 10 else "too low"

            if g(val) > 20:
                print("Lambda is", g(val))
                print("Input value is", val)
                break

            else:
                print("No output")

        except:
            print("Try again, please")
            continue

        finally:
            print("END CODE")
error()
```

```
Enter an integer: 1
Try again, please
END CODE
Enter an integer: 10
Try again, please
END CODE
Enter an integer: 12.12345
Lambda is 36
Input value is 12
END CODE
```

TASK 3: USE FOR LOOPS WITH LAMBDA

WORK WITH

```
g = lambda x: x**2 if x > 5 in range(10) else 0
```

DESIRED OUTPUTS

```
Where g(9):      Where g(3):

100 81               100 0
101 82               101 1
102 83               102 2
103 84               103 3
104 85               104 4
105 86               105 5
106 87               106 6
107 88               107 7
108 89               108 8
109 90               109 9
```

HINTS?

1. Use a for loop with a range 0 to 10.
2. Use print and add 100 to one of the output values.
3. Use two **+**.

In [9]:

```
g = lambda x: x**2 if x > 5 in range(10) else 0

for i in range(10):
    print(i+100, i + g(9))
```

```
100 81
101 82
102 83
103 84
104 85
105 86
106 87
107 88
108 89
109 90
```

TASK 4: FOR LOOP AND LIST COMPREHENSION

WORK WITH

```
critics = ("Mustafa", "Michael", "Callum", "George")

films = ["Akira", "Blade Runner 2049", "Mr. Robot", "The Ten
Commandments"]
```

DESIRED OUTPUTS

["Mustafa's favourite film is Akira",

"Michael's favourite film is Blade Runner 2049",
"Callum's favourite film is Mr. Robot",
"George's favourite film is The Ten Commandments"]

HINTS?

1. None required!

In [10]:

```
critics = ("Mustafa", "Michael", "Callum", "George")

films = ["Akira", "Blade Runner 2049", "Mr. Robot", "The Ten Commandments"]
```

In [11]:

```
opinion = []

for i in range(4):
    fave = critics[i] + "'s favourite film is " + films[i]
    opinion.append(fave)
opinion
```

Out[11]:

```
["Mustafa's favourite film is Akira",
 "Michael's favourite film is Blade Runner 2049",
 "Callum's favourite film is Mr. Robot",
 "George's favourite film is The Ten Commandments"]
```

In [12]:

```
[critics[i] + "'s favourite film is " + films[i] for i in range(4)]
```

Out[12]:

```
["Mustafa's favourite film is Akira",
 "Michael's favourite film is Blade Runner 2049",
 "Callum's favourite film is Mr. Robot",
 "George's favourite film is The Ten Commandments"]
```