

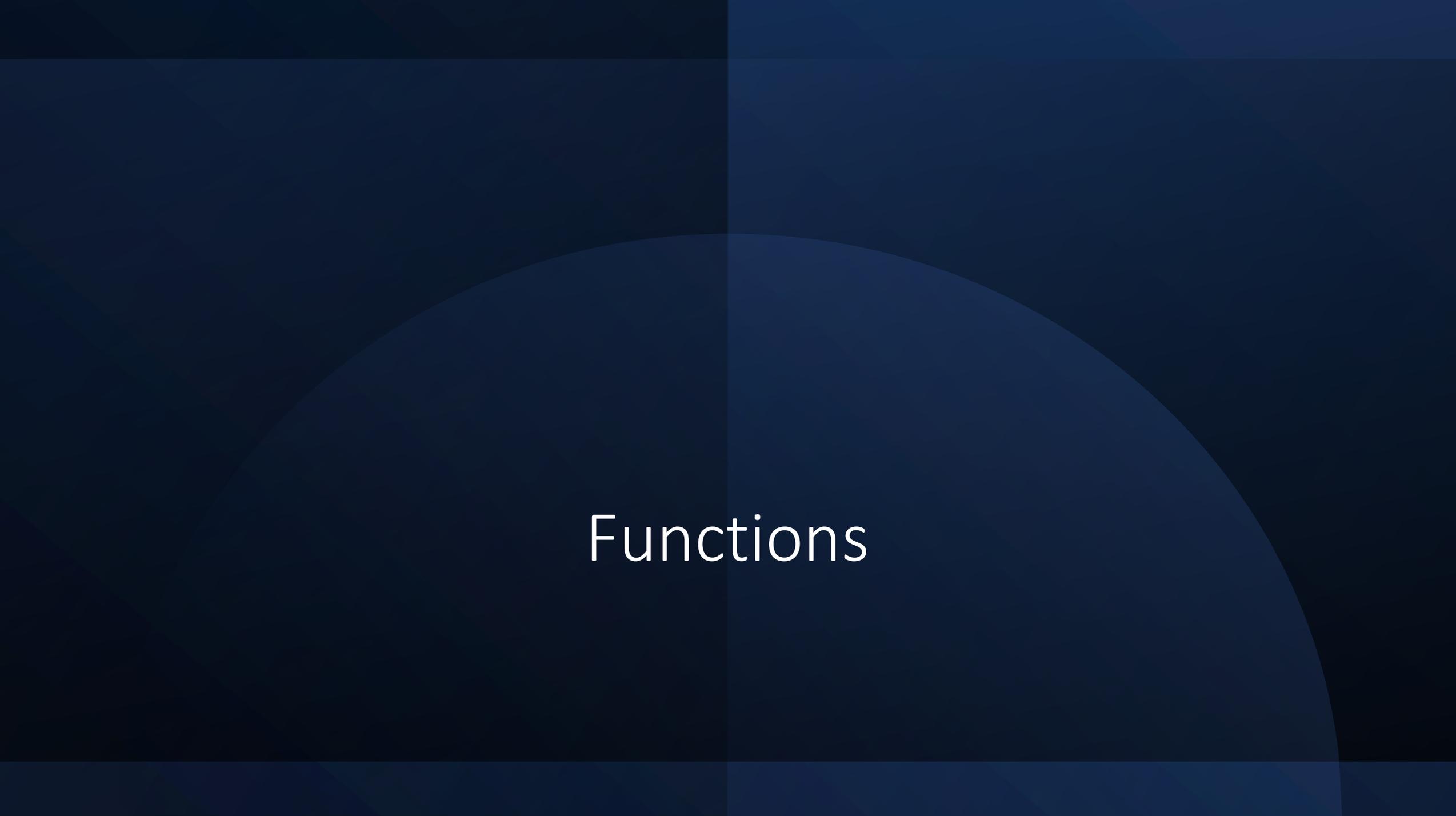
# Day 5

Functions – Parameters – Scope – Return

# Objectives

Students should be able to:

- Use functions.
- Differentiate between local variables and global variables.
- Use parameters effectively
- Use return values effectively.

The image features a dark blue background with a vertical line running down the center. A semi-circle is positioned at the bottom, centered on the vertical line. The word "Functions" is written in white, sans-serif font, centered horizontally and positioned above the semi-circle.

Functions

# Functions

- A function is a set of code that you give a name so that you can run that set of code whenever you want in your program without copying the code.
- The benefit of this is that if you need to change the code, you don't have to look for everywhere you copied it to change it, which can become very difficult in larger programs.
- It also helps the code be more organized by separating bits and pieces of code into functions that do different things.

# Example 1 – Without Function

## Source Code

```
print('Hello')
print('Welcome')
print('Please enter your name: ')
name = input()
print('Hello')
print('Welcome')
```

## Explanation

- In this code, we greet the user by saying hello and welcome. We ask them to enter their name. Then we greet them again.
- If we need to change the greeting, then we will have to change it both places.
- If we forget to change it one of the places, the program will not work as expected.
- Instead, we could put that code in a function and use the function.

# Example 1 – Function Syntax

## Source Code

```
def greet():  
    print('Hello')  
    print('Welcome')  
  
greet()  
print('Please enter your name: ')  
name = input()  
greet()
```

## Explanation

- The first three lines of code shows us how to make a function.
- We use the word “def” which means we’re creating a function
- We then put the name we want to give the function. In this case we called it greet.
- We then put parentheses and a colon.
- Then indented under that we put the code that goes in the function.

# Example 1 – Function Syntax

## Source Code

```
def greet():  
    print('Hello')  
    print('Welcome')  
  
greet()  
print('Please enter your name: ')  
name = input()  
greet()
```

## Explanation

- Note that this on its own does not run the code inside the function.
- To run the function, you “call” the function by putting its name and putting parentheses afterwards as seen on lines 5 and 8.
- When a function is called, the code inside that function is run, so this program will work exactly like the original which didn't have a function. It's like the code in the function got copied everywhere the function was called.

# Example 2 – Without Function

## Source Code

```
print('*****')  
print('*      *')  
print('*      *')  
print('*      *')  
print('*****')
```

## Explanation

- In this code, we print out a rectangle of asterisks.
- Notice that the first and last line of code are the same and should always be the same.
- So, we could put them in a function and use the function instead.

# Example 2 – With Function

## Source Code

```
def displayEdge():  
    print('*****')  
  
displayEdge()  
print('*      *')  
print('*      *')  
print('*      *')  
displayEdge()
```

## Explanation

- Here we created the displayEdge function which we use to print the top and bottom of the rectangle.
- Notice that printing out the sides uses repetitive code as well.
- These can be placed into a function as well.

# Example 2 – Without Function

## Source Code

```
def displayEdge () :  
    print ('*****')
```

```
def displaySide () :  
    print ('*      *')
```

```
displayEdge ()  
displaySide ()  
displaySide ()  
displaySide ()  
displaySide ()  
displayEdge ()
```

## Explanation

- Here we created another function called displaySide.
- This replaces the several print statements we had.
- Since we are repeating the displaySide function call, we can put it in a for loop instead.

# Example 2 – With Function and Loop

## Source Code

```
def displayEdge():  
    print('*****')  
  
def displaySide():  
    print('*      *')  
  
displayEdge()  
for i in range(3):  
    displaySide()  
displayEdge()
```

## Explanation

- Here we called the displaySide function in a for-each loop in order to run it 3 times.
- We can also put a function call inside another function.
- So, we could take all the code to display the rectangle and put it in one function.
- Then call that function.

## Example 2 – With Function Calls

### Source Code

```
def displayEdge():
    print('*****')

def displaySide():
    print('*          *')

def displayRect():
    displayEdge()
    for i in range(3):
        displaySide()
    displayEdge()

displayRect()
```

TDM	729.89	915.51	185.62	▲25.43%	FLR	660.27	745.28	85.01	▲12.88%
HUM	749.73	924.29	174.56	▲23.28%	UVD	155.59	181.57	25.98	▲16.70%
DMW	833.72	1004.01	170.29	▲20.43%	QUV	440.55	540.21	99.66	▲22.62%
YZJ	903.49	1127.46	223.97	▲24.79%	HZT	285.51	344.98	59.47	▲20.83%
GLY	982.07	1219.39	237.32	▲24.17%	PCW	811.44	1029.66	218.22	▲26.89%
VDA	113.74	143.41	29.67	▲26.09%	AIK	361.77	451.39	89.62	▲24.77%
UVV	468.08	535.41	67.33	▲14.38%	ZJJ	858.36	994.57	136.21	▲15.87%
HJS	545.49	659.05	113.56	▲20.82%	RHJ	894.79	1046.68	151.89	▲16.97%
ECC	566.96	664.69	97.73	▲17.24%	VDV	425.08	509.95	84.87	▲19.97%

PPJ	912.63	1038.36	125.73	▲13.78%	ZGK	391.59	491.48	99.89	▲25.51%
UAQ	1309.55	1655.62	346.07	▲26.43%	BNY	969.21	1130.65	161.44	▲16.66%
DAQ	1295.17	1641.66	346.49	▲26.75%	SDM	735.44	913.39	177.95	▲24.20%
PNR	654.33	775.84	121.51	▲18.57%	TQQ	1323.91	1646.42	322.51	▲24.36%
ZTM	751.99	929.53	177.54	▲23.61%	QIS	543.42	667.24	123.82	▲22.79%
					TTB	1495.17	1823.31	328.14	▲21.94%

# Local and Global Variables



# Local and Global Variables

- When a variable is created outside of a function, it is known as a global variable, and it can be used inside of any function.
- When a variable is created inside of a function, it is known as a local variable, and it can only be used inside of that function. Other functions can have a local variable with the same name, but these will count as different variables.

# Example

## Source Code

```
x = 5

def myFunction():
    y = 7
    print(x)
    print(y)

myFunction()
print(x)
print(y)
```

## Explanation

- In this code, we create a global variable called x and set it to 5.
- We create a function called myFunction which creates a local variable called y and prints out both the local and global variable.
- We then call that function causing it to run.
- If we then try to print y like on the last line we get an error since y is local.

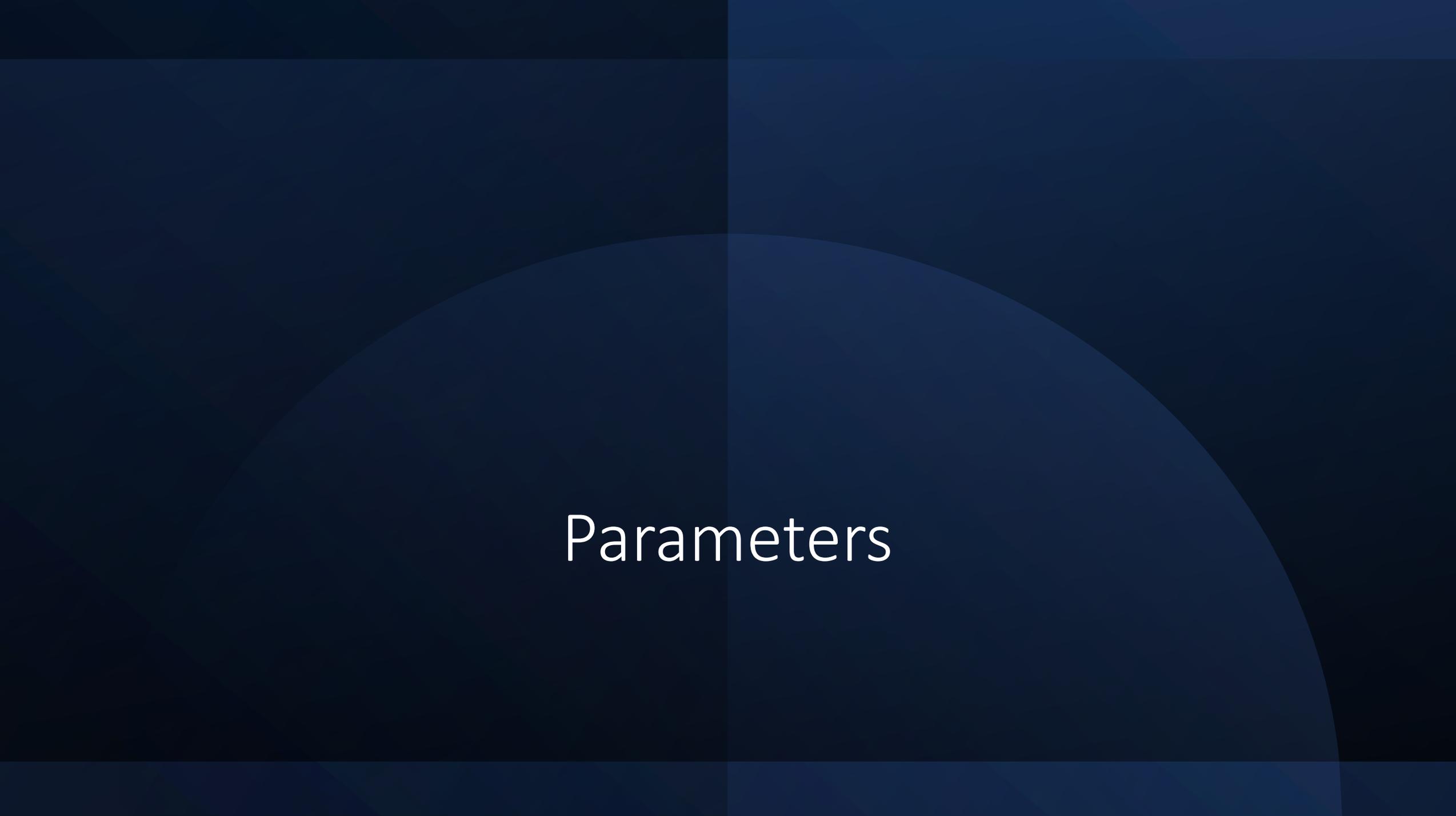
See Problem Set For Day 5

Question 1 – Name Prompt Function

Question 2 – Even Number Function

Question 3 – Sum of Range Function

Question 4 – List Insertion Function



Parameters

# Parameters

- A lot of times, we don't want a function to be too specific.
- There may be some local variables in the function that we do not want to give a specific value in the function itself.
- Instead, we want to specify the local variable's value when we call the function.
- These sort of variables are called parameters.

# Example 1

## Source Code

```
def displayRect():  
    displayEdge()  
    x = 3  
    for i in range(x):  
        displaySide()  
    displayEdge()
```

## Explanation

- Look at the displayRect function from earlier. It is modified slightly to use a variable x.
- Because x is set to 3, every time we call the function, it runs the displaySide function 3 times.
- This means that if we wanted a function which does the same thing, but instead runs the displaySide function 4 times (or any other number), we would need a separate function for that.
- This is inconvenient and the better way to do it would be to use a parameter.

# Example 1

## Source Code

```
def displayEdge():  
    print('*****')
```

```
def displaySide():  
    print('*      *')
```

```
def displayRect(x):  
    displayEdge()  
    for i in range(x):  
        displaySide()  
    displayEdge()
```

```
displayRect(3)  
displayRect(7)|
```

## Explanation

- Instead of setting x to a specific number, we place x in the brackets next to the function name.
- This indicates that x is a parameter.
- Now, whenever we call the function (like in the last two lines), we can specify what values to use for x.
- So displayRect(3) will run the displayRect function but using x as 3. and displayRect(7) will run the function using x as 7.

# Example 1

## Source Code

```
def displayEdge():
    print('*****')

def displaySide():
    print('*          *')

def displayRect(x):
    displayEdge()
    for i in range(x):
        displaySide()
    displayEdge()

displayRect(3)
displayRect(7)
```

## Output

```
*****
*          *
*          *
*          *
*****
*****
*          *
*          *
*          *
*          *
*          *
*          *
*          *
*****
```

# Multiple Parameters

- A function can have as many parameter variables as it needs.
- The parameter variables are listed out in the parentheses next to the function name.

# Example 2

## Source Code

```
def printNumbers():  
    first = 1  
    last = 100  
    for i in range(first, last+1):  
        print(i)
```

## Explanation

- This function sets the variables first to 1, and last to 100, and prints the numbers from first to last (from 1 to 100).
- So, whenever we call the function, it will always print the numbers starting at and ending at 100.
- If we want to change the first or last number, we will have to make a new function.
- We can make the function more useful by changing the variables first and last to parameters.

# Example 2

## Source Code

```
def printNumbers(first, last):  
    for i in range(first, last+1):  
        print(i)
```

```
printNumbers(5, 25)  
printNumbers(-100, 100)
```

## Explanation

- Since first and last are now parameters, whenever we call the function, we need to specify what values to use for first and last.

See Problem Set For Day 5

Question 5 – Modify the previous questions so that they use parameters instead.

Question 6 – Modify day 4's questions to use functions.



# Return Values

# Return Values

- Sometimes, we want a function to perform some sort of calculation or computation.
- The answer to this calculation could be stored in a local variable.
- However, we might want to use that answer outside of the function in the rest of the program, but since the variable is local, we can't.
- To be able to use the answer outside of the function, we use something called a return value.

# Example 1

## Source Code

```
weight = float(input('Enter weight in kg: '))
height = float(input('Enter height in m: '))
bmi = weight / height**2
print('Your bmi is %.2f' % (bmi))
```

## Explanation

- Here we are calculating a person's BMI. If this is part of a larger program where we must calculate BMI several times, we may want to use a function instead of repeating the formula

# Example 1

## Source Code

```
def calculateBMI(weight, height):  
    bmi = weight / height**2  
    print('Your bmi is %.2f' % (bmi))  
  
weight = float(input('Enter weight in kg: '))  
height = float(input('Enter height in m: '))  
calculateBMI(weight, height)
```

## Explanation

- Here we have a function which calculates the BMI using the parameters weight and height, then prints it out.
- In the program, we get the user's weight and height, then call the function to calculate and print their BMI.
- The problem is that if we don't want to print their BMI, and instead we want to use it somewhere else in the program, we can't use this function.

# Example 1

## Source Code

```
def calculateBMI(weight, height):  
    bmi = weight / height**2  
    return bmi  
  
weight = float(input('Enter weight in kg: '))  
height = float(input('Enter height in m: '))  
bmi = calculateBMI(weight, height)  
print('Your bmi is %.2f' % (bmi))
```

## Explanation

- To solve this issue, we use a return statement.
- The return statement allows us to take a calculation we made in the function and use it outside the function.
- In this function we are returning the BMI that we calculated.
- On line 7 we see how to use a return value. The function call itself gets replaced with the return value. In this case we are storing it in a variable.
- We can then do multiple things with this stored variable outside of the function.

See Problem Set For Day 5

Question 7 – Modify the previous questions so that they use return values. Modify day 4's questions to use return values.

Note that not all functions need to have return values.