Agenda

1. Functions (Examples)
2. Global Variables vs Local Variables
3. Nested Functions
4. nonlocal variables
5. Higher-Order Functions
6. Anonymous Functions

```
# Global variables vs Local Variables

def func_a():
  # local_variable
  a = 100
  print("Local Variable - a: ", a)

func_a()
print("Access Local Variable - a from outside the function: ", a)
```

```
Local Variable - a:  100
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-1-4ae613b354ff> in <cell line: 9>()
      7 
      8 func_a()
----> 9 print("Access Local Variable - a from outside the function: ", a)

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

Next steps: Explain error

```
# global variable

b = 100 # global variable

def func_b():
  print("Access Global Variable -b Inside Function: ", b)

func_b()
print("Access Global Variable - b Outside Function: ", b)

def func_new():
  print("Accessing b: ", b)

func_new()
```

```
Access Global Variable -b Inside Function:  100
Access Global Variable - b Outside Function:  100
Accessing b:  100
```

```
# global variable vs local variable

def func_c():
  c = 200 # treated as local variable
  print("What will get printed: ",c)

c = 1000
print("Global Variable c : ", c)
func_c()
print("Global Variable c: ", c)
```

```
Global Variable c :  1000
What will get printed:  200
```

```
Global Variable c:  1000
```

```python
# purpose of "global" keyword

def func_d():
  print("Global Variable d : ", d)
  d += 500
  print("Changing d inside the function: ",d)

d = 1000
print("Initializing d: -->", d)
d += 500
print("Modifying d: -->", d)
func_d()
print("Outside function, d changed to : ", d)
```

```
Initializing d: --> 1000
Modifying d: --> 1500
---------------------------------------------------------------------------
UnboundLocalError                         Traceback (most recent call last)
<ipython-input-4-dfedef53094a> in <cell line: 12>()
     10 d += 500
     11 print("Modifying d: -->", d)
---> 12 func_d()
     13 print("Outside function, d changed to : ", d)

<ipython-input-4-dfedef53094a> in func_d()
      2
      3 def func_d():
----> 4   print("Global Variable d : ", d)
      5   d += 500
      6   print("Changing d inside the function: ",d)

UnboundLocalError: local variable 'd' referenced before assignment
```

Next steps: Explain error

```python
def func_d(d):
  print("Global Variable d : ", d)
  d += 500
  print("Changing d inside the function: ",d)

d = 1000
print("Initializing d: -->", d)
d += 500
print("Modifying d: -->", d)
func_d(d)
print("Outside function, d changed to : ", d)
```

```
Initializing d: --> 1000
Modifying d: --> 1500
Global Variable d :  1500
Changing d inside the function:  2000
Outside function, d changed to :  1500
```

```
def func_d():
  global d
  print("Global Variable d : ", d)
  d += 500
  print("Changing d inside the function: ",d)

d = 1000
print("Initializing d: -->", d)
d += 500
print("Modifying d: -->", d)
func_d()
print("Outside function, d changed to : ", d)
```

```
Initializing d: --> 1000
Modifying d: --> 1500
Global Variable d :  1500
Changing d inside the function:  2000
Outside function, d changed to :  2000
```

```
def func_d():
  print("Global Variable d : ", d)
  d += 500
  print("Changing d inside the function: ",d)

d = 1000
print("Initializing d: -->", d)
d += 500
print("Modifying d: -->", d)
func_d()
print("Outside function, d changed to : ", d)
```

```
Initializing d: --> 1000
Modifying d: --> 1500
---------------------------------------------------------------------------
UnboundLocalError                         Traceback (most recent call last)
<ipython-input-7-d1c363b14b46> in <cell line: 10>()
      8 d += 500
      9 print("Modifying d: -->", d)
---> 10 func_d()
     11 print("Outside function, d changed to : ", d)

<ipython-input-7-d1c363b14b46> in func_d()
      1 def func_d():
----> 2   print("Global Variable d : ", d)
      3   d += 500
      4   print("Changing d inside the function: ",d)
      5 

UnboundLocalError: local variable 'd' referenced before assignment
```

Next steps: Explain error

```
def func_d():
  d = 1500
  print("Local Variable d : ", d)
  d += 500
  print("Changing d inside the function: ",d)

d = 1000
print("Initializing d: -->", d)
d += 500
print("Modifying d: -->", d)
func_d()
print("Outside function, d changed to : ", d)
```

```
Initializing d: --> 1000
Modifying d: --> 1500
```

```
Local Variable d :  1500
Changing d inside the function:  2000
Outside function, d changed to :  1500
```

```
def func_d():
  e = d + 500
  print("Changing d inside the function: ",d)
  print("Creating a new variable e: ", e)

d = 1000
print("Initializing d: -->", d)
d += 500
print("Modifying d: -->", d)
func_d()
print("Outside function, d changed to : ", d)
```

```
Initializing d: --> 1000
Modifying d: --> 1500
Changing d inside the function:  1500
Creating a new variable e:  2000
Outside function, d changed to :  1500
```

```
# nested functions

def func_outer():
  # code block - outer func
  x = 1000
  y = 2000
  def func_inner():
    print("Accessing outer func variables: x -> {}, y ->{}".format(x,y))
  func_inner()

func_outer()
```

```
Accessing outer func variables: x -> 1000, y ->2000
```

```
def func_outer():
  # code block - outer func
  x = 1000
  y = 2000
  def func_inner():
    print("Accessing outer func variables: x -> {}, y ->{}".format(x,y))

func_outer()
```

```
def func_outer():
  # code block - outer func
  x = 1000
  y = 2000
  def func_inner():
    x += 10000
    print("Accessing outer func variables: x -> {}, y ->{}".format(x,y))
  func_inner()

func_outer()
```

```
---------------------------------------------------------------------------
UnboundLocalError                         Traceback (most recent call last)
<ipython-input-13-64229083808c> in <cell line: 10>()
      8   func_inner()
      9 
---> 10 func_outer()

                                    1 frames
<ipython-input-13-64229083808c> in func_inner()
      4   y = 2000
      5   def func_inner():
----> 6     x += 10000
      7     print("Accessing outer func variables: x -> {}, y ->{}".format(x,y))
      8   func_inner()

UnboundLocalError: local variable 'x' referenced before assignment
```

Next steps: Explain error

```
def func_outer():
  # code block - outer func
  x = 1000
  y = 2000
  def func_inner():
    global x
    x = 5000
    x += 10000
    print("Inner func: x -> {}, y ->{}".format(x,y))
  func_inner()
  print("Outer Func: x-->", x)

func_outer()
print("x outside the outer/main function : x ->", x)
```

```
Inner func: x -> 15000, y ->2000
Outer Func: x--> 1000
x outside the outer/main function : x -> 15000
```

```
def func_outer():
  # code block - outer func
  global x
  x = 1000
  y = 2000
  def func_inner():
    global x
    x = 5000
    x += 10000
    print("Inner func: x -> {}, y ->{}".format(x,y))
  func_inner()
  print("Outer Func: x-->", x)

func_outer()
print("x outside the outer/main function : x ->", x)
```

```
Inner func: x -> 15000, y ->2000
Outer Func: x--> 15000
x outside the outer/main function : x -> 15000
```

```
def outer():
  v = 500 # local variable
  def inner():
    global rohit
    rohit = 92
    print(v) # you can access the outer function local variable from inner function
  inner()
  print("outer func, rohit: ", rohit)

outer()
print(rohit)
```

```
500
outer func, rohit:  92
92
```

```
def func_outer():
  # code block - outer func
  global x
  x = 1000
  y = 2000
  def func_inner():
    global x
    x = 5000
    x += 10000
    print("Inner func: x -> {}, y ->{}".format(x,y))
  func_inner()
  print("Outer Func: x-->", x)

func_inner()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-21-d21085717dd0> in <cell line: 14>()
     12   print("Outer Func: x-->", x)
     13
---> 14 func_inner()

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

Next steps: Explain error

```
def func_outer():
  # code block - outer func
  v1 = 1000 # local variable
  def func_inner(v1):
    v1 += 10000
    print("Inner func: v1 -> {}".format(v1))
  func_inner(v1)
  print("Outer Func: v1-->", v1) # v1 -> 11000

func_outer()
print("v1 outside the outer/main function : v1 ->", v1)
```

```
Inner func: v1 -> 11000
Outer Func: v1--> 1000
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-26-47b80018918d> in <cell line: 11>()
      9
     10 func_outer()
---> 11 print("v1 outside the outer/main function : v1 ->", v1)

NameError: name 'v1' is not defined
```

Next steps: Explain error

```
# use of "nonlocal" keyword comes into picture

def func_outer():
  # code block - outer func
  v1 = 1000 # local variable to outer function
  def func_inner():
    nonlocal v1 # local variable to inner function
    v1 += 1000
    print("Inner func: v1 -> {}".format(v1))
  func_inner()
  print("Outer Func: v1-->", v1) # v1 -> 2000

func_outer()
print("in main code: v1-->",v1)
```

```
Inner func: v1 -> 2000
Outer Func: v1--> 2000
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-29-fa61348e62bd> in <cell line: 14>()
     12
     13 func_outer()
---> 14 print("in main code: v1-->",v1)

NameError: name 'v1' is not defined
```

Next steps: Explain error

```
def func_outer():
  # code block - outer func
  v1 = 1000 # local variable to outer function
  def func_inner():
    # local variable to inner function
    v1 = 1000
    v1 += 1000
    print("Inner func: v1 -> {}".format(v1))
  func_inner()
  print("Outer Func: v1-->", v1) # v1 -> 2000

func_outer()
print("main code: v1-->", v1)
```

```
Inner func: v1 -> 2000
Outer Func: v1--> 1000
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-31-72c85661f34a> in <cell line: 13>()
     11
     12 func_outer()
---> 13 print("main code: v1-->", v1)

NameError: name 'v1' is not defined
```

Next steps: Explain error

```
def func_outer():
  # code block - outer func
  v1 = 1000 # local variable to outer function
  def func_inner():
    # local variable to inner function
    global v1
    v1 = 5000
    v1 += 1000
    print("Inner func: v1 -> {}".format(v1))
  func_inner()
  print("Outer Func: v1-->", v1) # v1 -> 2000

func_outer()
print("main code: v1-->", v1)
```

```
Inner func: v1 -> 6000
Outer Func: v1--> 1000
main code: v1--> 6000
```

```
def func_outer():
  # code block - outer func
  v2 = 1000 # local variable to outer function
  def func_inner():
    # local variable to inner function
    nonlocal v2
    v2 += 1000
    print("Inner func: v2 -> {}".format(v2))
  func_inner()
  print("Outer Func: v2-->", v2) # v1 -> 2000

func_outer()
print("main code: v1-->", v2)
```

```
Inner func: v2 -> 2000
Outer Func: v2--> 2000
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-34-bd1d1ba7c1ff> in <cell line: 13>()
     11
     12 func_outer()
---> 13 print("main code: v1-->", v2)

NameError: name 'v2' is not defined
```

Next steps: Explain error

```
def func_outer():
  # code block - outer func
  v2 = 1000 # local variable to outer function
  def func_inner():
    # local variable to inner function
    nonlocal v3
    v2 += 1000
    print("Inner func: v2 -> {}".format(v2))
  func_inner()
  print("Outer Func: v2-->", v2) # v1 -> 2000

func_outer()
print("main code: v1-->", v2)
```

```
  File "<ipython-input-35-6cfc265bfd48>", line 6
    nonlocal v3
    ^
SyntaxError: no binding for nonlocal 'v3' found
```

Next steps: Fix error

```
def func_outer():
  # code block - outer func
  v2 = 1000 # local variable to outer function
  def func_inner():
    # local variable to inner function
    v2 = 5000 # local variable to inner function
    print("Inner func: v2 -> {}".format(v2))
    def func_insideInner():
      nonlocal v2
      print("Func inside inner: -->",v2)
    func_insideInner()

  func_inner()
  print("Outer Func: v2-->", v2) # v1 -> 2000

func_outer()
```

```
Inner func: v2 -> 5000
Func inside inner: --> 5000
Outer Func: v2--> 1000
```

```
def func_outer():
  # code block - outer func
  v2 = 1000 # local variable to outer function
  def func_inner():
    # local variable to inner function
    v2 = 5000 # local variable to inner function
    print("Inner func: v2 -> {}".format(v2))
    def func_insideInner():
      nonlocal v2
      print("Func inside inner: -->",v2)
    func_insideInner()

  func_inner()
  func_insideInner()
  print("Outer Func: v2-->", v2) # v1 -> 2000

func_outer()
```

```
Inner func: v2 -> 5000
Func inside inner: --> 5000
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-38-094c28e47bd7> in <cell line: 17>()
     15   print("Outer Func: v2-->", v2) # v1 -> 2000
     16 
---> 17 func_outer()

<ipython-input-38-094c28e47bd7> in func_outer()
     12 
     13   func_inner()
---> 14   func_insideInner()
     15   print("Outer Func: v2-->", v2) # v1 -> 2000
     16 

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

Next steps: Explain error

```
def calculator(a,b):
  print("Calculations is printed below: ")
  def add(a,b):
    print("a+b: -->",a+b)

  def sub(a,b):
    print("a-b: -->",a-b)

  add(a,b)
  sub(a,b)

  global expose_add_func
  expose_add_func = add
  global expose_sub_func
```

```
def calculator(a,b):
  print("Calculations is printed below: ")
  def add(a,b):
    print("a+b: -->",a+b)

  def sub(a,b):
    print("a-b: -->",a-b)

  return add
```

```
add_fun = calculator(10,15)
```

```
Calculations is printed below:
```

```
add_fun(12,24)
```

```
a+b: --> 36
```