Function Scope

In [1]:
```
def performOperation(*args, **kwargs):
    print(args)
    print(kwargs)

performOperation(1, 2, operation='sum')
```

```
(1, 2)
{'operation': 'sum'}
```

locals()

In [3]:
```
def performOperation(num1, num2, operation='sum'):
    print(locals())

performOperation(1, 2, operation='multiply')
print(num1)
```

```
{'num1': 1, 'num2': 2, 'operation': 'multiply'}

---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Input In [3], in <module>
      2     print(locals())
      4 performOperation(1, 2, operation='multiply')
----> 5 print(num1)

NameError: name 'num1' is not defined
```

globals()

In [4]:
```
globals()
```

Out[4]:
```
{'__name__': '__main__',
 '__doc__': 'Automatically created module for IPython interactive environmen
t',
 '__package__': None,
 '__loader__': None,
 '__spec__': None,
 '__builtin__': <module 'builtins' (built-in)>,
 '__builtins__': <module 'builtins' (built-in)>,
 '_ih': ['',
  "def performOperation(*args, **kwargs):\n    print(args)\n    print(kwargs)
\n\nperformOperation(1, 2, operation='sum')",
  "def performOperation(num1, num2, operation='sum'):\n    print(locals())\n
\nperformOperation(1, 2, operation='multiply')",
  "def performOperation(num1, num2, operation='sum'):\n    print(locals())\n
\nperformOperation(1, 2, operation='multiply')\nprint(num1)",
  'globals()'],
 '_oh': {},
 '_dh': [PosixPath('/Users/ryanmitchell/Documents/GitHub/python-essential-tra
ining-2449125/exercise_files')],
 'In': ['',
  "def performOperation(*args, **kwargs):\n    print(args)\n    print(kwargs)
\n\nperformOperation(1, 2, operation='sum')",
  "def performOperation(num1, num2, operation='sum'):\n    print(locals())\n
\nperformOperation(1, 2, operation='multiply')",
  "def performOperation(num1, num2, operation='sum'):\n    print(locals())\n
\nperformOperation(1, 2, operation='multiply')\nprint(num1)",
  'globals()'],
 'Out': {},
 'get_ipython': <bound method InteractiveShell.get_ipython of <ipykernel.zmqs
hell.ZMQInteractiveShell object at 0x107176ad0>>,
 'exit': <IPython.core.autocall.ZMQExitAutocall at 0x1071d43d0>,
 'quit': <IPython.core.autocall.ZMQExitAutocall at 0x1071d43d0>,
 '_': '',
 '__': '',
 '___': '',
 '_i': "def performOperation(num1, num2, operation='sum'):\n    print(locals
())\n    \nperformOperation(1, 2, operation='multiply')\nprint(num1)",
 '_ii': "def performOperation(num1, num2, operation='sum'):\n    print(locals
())\n    \nperformOperation(1, 2, operation='multiply')",
 '_iii': "def performOperation(*args, **kwargs):\n    print(args)\n    print
(kwargs)\n\nperformOperation(1, 2, operation='sum')",
 '_i1': "def performOperation(*args, **kwargs):\n    print(args)\n    print(k
wargs)\n\nperformOperation(1, 2, operation='sum')",
 'performOperation': <function __main__.performOperation(num1, num2, operatio
n='sum')>,
 '_i2': "def performOperation(num1, num2, operation='sum'):\n    print(locals
())\n    \nperformOperation(1, 2, operation='multiply')",
 '_i3': "def performOperation(num1, num2, operation='sum'):\n    print(locals
())\n    \nperformOperation(1, 2, operation='multiply')\nprint(num1)",
 '_i4': 'globals()'}
```

Global and Local scope

In [8]:
```
message = 'Some global data'

def function1(varA, varB):
    print(message)
    print(locals())


def function2(varC, varB):
    print(message)
    print(locals())

function1(1, 2)
function2(3, 4)
```

```
Some global data
{'varA': 1, 'varB': 2}
Some global data
{'varC': 3, 'varB': 4}
```

In [10]:
```
message = 'Some global data'
varA = 2
def function1(varA, varB):
    message = 'Some local data'
    print(varA)
    print(message)
    print(locals())


def function2(varC, varB):
    print(varA)
    print(message)
    print(locals())

function1(1, 2)
function2(3, 4)
```

```
1
Some local data
{'varA': 1, 'varB': 2, 'message': 'Some local data'}
2
Some global data
{'varC': 3, 'varB': 4}
```

In [13]:
```
def function1(varA, varB):
    message = 'Some local data'
    print(varA)
    def inner_function(varA, varB):
        print(f'inner_function local scope: {locals()}')

    inner_function(123, 456)


function1(1, 2)
```

```
1
inner_function local scope: {'varA': 123, 'varB': 456}
```

In [14]:
```
def function1(varA, varB):
    message = 'Some local data'
    print(varA)
    def inner_function(varA, varB):
        print(f'inner_function local scope: {locals()}')

    print(locals())
    inner_function(123, 456)


function1(1, 2)
```

```
1
{'varA': 1, 'varB': 2, 'message': 'Some local data', 'inner_function': <funct
ion function1.<locals>.inner_function at 0x107b6af80>}
inner_function local scope: {'varA': 123, 'varB': 456}
```

In [ ]:
```

```