Sign in:
cadtx.pw/week6

Week 6

Intro to Python

Objectives

Chapter 6

Define functions with parameters

Invoke functions with parameters

Understand functions that return and do not return a value

Functions

Functions can be used to define reusable code and organize and simplify code.

Syntax:

```
def functionName(parameters):
    # Function body
```

def is a keyword in Python

Don’t forget the colon!

Function example

We can make a function that finds which of two numbers is the biggest.

```
# Assuming n1 != n2
def max(n1, n2):
    if n1 > n2:
        return n1
    else:
        return n2
```

A return statement using the keyword return is required for a value-returning function to return a result. The function terminates when a return statement is executed.

Invoking a function

By calling a function, you're executing the code contained in it.

Main function: programs often define a function named *main* that contains the main functionality for a program. If we only had max() and no main() then nothing would run.

```
# Assuming n1 != n2
def max(n1, n2):
    if n1 > n2:
        return n1
    else:
        return n2


def main():
    larger = max(5, 7)
    print(larger)

    # Or simply

    print(max(5, 7))

main()
```

Functions and Return values

Functions don't need a return value and those type of functions are called *void functions*.

```
# Return the grade for the score
def getGrade(score):
    if score >= 90.0:
        return 'A'
    elif score >= 80.0:
        return 'B'
    elif score >= 70.0:
        return 'C'
    elif score >= 60.0:
        return 'D'
    else:
        return 'F'

def main():
    score = eval(input("Enter a score: "))
    print("The grade is", getGrade(score))

main() # Call the main function
```

What happens in the following:

```
def function(x):
    print(x)
    x = 4.5
    y = 3.4
    print(y)

x = 2
y = 4
function(x)
print(x)
print(y)
```

```
def f(x, y = 1, z = 2):
    return x + y + z

print(f(1, 1, 1))
print(f(y = 1, x = 2, z = 3))
print(f(1, z = 3))
```

Is there something wrong with this code?

```
def function():
    x = 4.5
    y = 3.4
    print(x)
    print(y)

function()
print(x)
print(y)
```

Variables (global vs local)

Local variables are created in a function

Global variables are created outside all functions and can be used in all functions

```
1  globalVar = 1
2  def f1():
3      localVar = 2
4      print(globalVar)
5      print(localVar)
6
7  f1()
8  print(globalVar)
9  print(localVar) # Out of scope, so this gives an error
```

Function example with a for loop

Find the sum of integers from 1 to 10, 20 to 37, and 35 to 49