More on Functions (class slides)

CSC 110 - More on Functions in Python

Function Comments

- Every function created is required to have a function comment, including main
- Function comments are a multi-line string (as opposed to using # for other comments)

Function Comments

```
'''
Adriana Picoral
CSC110
Class Demonstration
This program has two functions: one to calculate the area of a sphere,
the other to calculate the volume of a sphere.
The main() function is called to print to the standard output the
area and volume of a sphere of radius .75
'''

def sphere_volume(radius):
  '''
  This function calculates the volume of a sphere of given radius.
  Args:
    radius: integer representing the radius of the sphere
  Returns:
    The float representing the volume of a sphere of the given radius
  '''
  volume = (1 / 3) * sphere_area(radius) * radius
  return round(volume, 2)
```

```
def sphere_area(radius):
  '''
  This function calculates the area of a sphere of given radius.
  Args:
    radius: integer representing the radius of the sphere
  Returns:
    The float representing the area of a sphere of the given radius
  '''
  area = 4 * 3.1415 * radius**2
  return round(area, 2)


def main():
  '''
  This function prints the volume and area of a sphere of radius .75.
  Args:
    None
  Returns:
    None
  '''
  r = .75
  v = sphere_volume(r)
  a = sphere_area(r)
  print(v, a)

main()
```

```
1.77 7.07
```

Global vs. Local variables (scope)

- Every variable that is created has a particular scope
- The scope of a variable is the range of coder over which that variable can be used or modified

Global vs. Local variables (scope)

- **Local Variables** have local scope – for example, a variable assigned inside a function can only be used or modified withing that function

- **Global Variables** have global scope – for example, a variable delcared outside a function can be accessed or modified acorss mulitple functions

Global vs. Local variables (scope)

- In the previous program we wrote (volume and area of sphere), `r`, `v` and `a` are local variables within the `main()` function.
- The variable `area` is also local within the `sphere_area(radius)` function scope.
- The variable `volume` is local within `sphere_volume(radius)`

Global or Local?

```
a = 10                    # What are the global and local variables?
b = 5                     # Is the output of the two programs the same
                          # or different?
def sum():
  return a + b

def main():
  print(sum())

main()
```

vs.

```
def sum(a, b):
  return a + b

def main():
  print(sum(10, 5))

main()
```

Argument vs. Parameter

- Never set variables as global variables, pass values to functions when called
- When a function is defined, the values you want to pass to the function are called **parameter variables**
- When the function is then called, the values you pass to the function are called **arguments**

Write a function

Write a Python function that does the following:

1. Its name is `hypotenuse`
2. It takes two arguments: `a` and `b` representing the length of the two non-hypotenuse sides of a right angle triangle
3. It calculate the hypotenuse (`c`) according to the Pythagorean theorem formula: $c = \sqrt(a^2 + b^2)$
4. It returns the calculated hypotenuse
5. Test cases: `hypotenuse(3, 4)` should return `5.0`, `hypotenuse(10, 10)` should return `14.14`

Write a function

```
def sqrt(n):
  '''
  This function calculates the square root of a number
  Args:
    n: integer or float
  Returns:
    The square root of n
  '''
  return n**.5

def hypotenuse(a, b):
  '''
  This function calculates the hypotenuse of a right angle triangle.
  Args:
    a: number (integer or float) representing one of the non-hypotenuse sides
    b: number (integer or float) representing one of the non-hypotenuse sides
  Returns:
    Float representing the length of the hypotenuse given a and b
  '''
  h = sqrt(a**2 + b**2)
  return round(h, 2)

def main():
  '''
  This function calls the hypotenuse function to calculate and then
  print out the hypotenuse of a right angle triangle of sides 3 and 4
```

```
    and the hypotenuse of a right angle triange of sides 10 and 10
    '''
    result = hypotenuse(3, 4)
    print(result)

    result = hypotenuse(10, 10)
    print(result)

main()
```

```
5.0
14.14
```