Functions (class slides)

CSc 110 Functions Adriana Picoral

Functions

- Functions are named operations that are available to do tasks
- Some functions are *built-in* functions that Python provides
- Programmers can also define their own functions
- Functions are **called** (or **invoked**)

Function definitions

```
def two():
  return 2
```

This function definition has many parts:

- `two` is the name of the function
- `()` is the **parameter** list (Here, it is empty)
- the body (or content) of the function is indented
- `return 2` is a statement that causes the function to cease and produce the value 2

Example of a simple function

```
def add_one(n):
  return n + 1
```

- `add_one` is the name of the function
- `(n)` is the **parameter** list
- the body (or content) of the function is indented

- `return n + 1` is a statement that causes the function to cease and produce the value n + 1

Function to calculate area of a circle

Remember this from the last set of slides?

```
# assign a radius value
radius = 3
# compute the area of a circle
area = 3.1415 * radius ** 2
```

Calculating the `area` of a circle is an `abstraction`.

In the code above, that is done by a variable *assignment* with a variable named `area`.

Let's create a function called `area`, that given a `radius` parameter, it returns the `area` of the circle.

Function to calculate the volume of a cylinder

Write a function that does the following:

1. Its name is `volume`
2. It takes two integer arguments: `radius` and `height`
3. It calculates the volume of a cylinder, based on `radius` and `height`. Volume is area multiplied by height.
4. It returns the float value for calculated volume.

Function to calculate the volume of a cylinder

```
def volume(radius, height):
  # calculate the area first
  area = 3.1415 * radius ** 2

  # multiply area by height
  vol = area * height

  # return calculated volume
  return vol
```

```
print(volume(1, 2)) # 6.283
print(volume(6, 10)) # 1130.94
print(volume(5, 5)) # 392.68750000000006
```

Submit to Gradescope

- Name your python script with the `volume` function `cylinder.py`
- Go to Gradescope and submit your `.py` file
- You can resubmit as many times as you need (up to the deadline)
- You can look at your `Submission History` and activate a previous submission

Order of Operations

PEMDAS

- What does PEMDAS stand for?
- The operator precedence:
  - **P**arentheses
  - **E**xponentiation
  - **M**ultiplication and **D**ivision (including // and %)
  - **A**ddition and **S**ubtraction

PEMDAS

What value will each of these variables take on? No computers!

PEMDAS – answer

```
a1 = 5 /  5 * 10  * 5
a2 = 5 / (5 * 10) * 5

b1 = 5 *  10 - 2
b2 = 5 * (10 - 2)

# c = (3 // (4 // 5)) + 1 ERROR -- Zero Division

print(a1)
print(a2)
print(b1)
print(b2)
```

```
50.0
0.5
48
40
```

Note that the division operator returns a float even when both numerator and denominator are integers