

Assignments

- ▶ Due the Monday the week *after* it is listed on the schedule
- ▶ Composed of a single .py file
 - ▶ When executed displays answers to all questions that were programming based
- ▶ And, sometimes, a single text document
 - ▶ Word, Open Office, PDF and plain text documents are OK
- ▶ Emailed to ethan.white@usu.edu
 - ▶ Subject line: Programming Assignment X



Assignments

assignment_1_ethan_white.py

Assignment 1

imports

Problem 1

Commands that answer problem 1

Problem 2

Commands that answer problem 2



Classroom times

- ▶ **Anytime except**
 - ▶ Thursday 10 am – Friday 1 pm
 - ▶ Tuesdays 1:30 – 2:45 pm
 - ▶ MWF 2:30 – 3:20 pm



Functions

Possibly the single most important thing I'll teach you

Function basics

- ▶ A complicated expression
 - ▶ Python command that returns a value

- ▶ We use functions all the time

```
>>> mass_kg = 2.25
```

```
>>> str(mass_kg)
```

```
'2.25'
```

```
>>> round(3.8)
```

```
4.0
```



Making your own functions

- ▶ Critical to writing manageable programs
- ▶ Basic structure

```
def function_name(inputs):
```

```
    commands
```

```
    return output_value
```

```
>>> function_name(arguments)
```

```
>>> round(3.8)
```

Also a common conceptual stumbling block



Making your own functions

▶ Example

```
def convert_mass_to_kg_from_g(mass_g):  
    mass_kg = mass_g / 1000  
    return mass_kg
```

```
>>> mass_kg = convert_mass_to_kg_from_g(10000)
```

```
>>> print mass_kg
```

```
10
```



Whitespace matters in Python

▶ **Good**

```
def convert_mass_to_g_from_kg(mass_kg):  
    mass_g = mass_kg * 1000  
    return mass_g
```

▶ **Bad**

```
def convert_mass_to_g_from_kg(mass_kg):  
mass_g = mass_kg * 1000  
return mass_g
```



You can have multiple inputs

- ▶ We fit a linear regression to some data and we want to get the predicted value for some x with the parameters of the regression

```
def predicted_value(a, b, x):  
    prediction = a + b * x  
    return prediction
```



Can be in same file as other commands

- ▶ When the lines of a function are executed that function then exists

```
>>> def predicted_value(a, b, x):  
...     prediction = a + b * x  
...     return prediction
```

```
>>> predicted_value(2, 1, 1)  
3
```



Can be in same file as other commands

- ▶ When the lines of a function are executed that function then exists

```
def predicted_value(a, b, x):  
    prediction = a + b * x  
    return prediction  
print(predicted_value(2, 1, 1))
```

3



Can be in same file as other commands

- ▶ But the function must be defined before it is called, because otherwise it doesn't exist

```
print(predicted_value(2, 1, 1))
```

```
def predicted_value(a, b, x):
```

```
    prediction = a + b * x
```

```
    return prediction
```

Traceback (most recent call last):

File "<string>", line 1, in <fragment>

NameError: name 'predicted_value' is not defined



Local variables

- ▶ Variables defined inside functions do not exist outside of those functions

Program

Input



Function



Output



Local variables

- ▶ Variables defined inside functions do not exist outside of those functions

```
def predicted_value(a, b, x):  
    prediction = a + b * x  
    return prediction
```

```
predicted_value(2,1,1)  
print(prediction)
```

Traceback (most recent call last):

File "<string>", line 5, in <fragment>

NameError: name 'prediction' is not defined

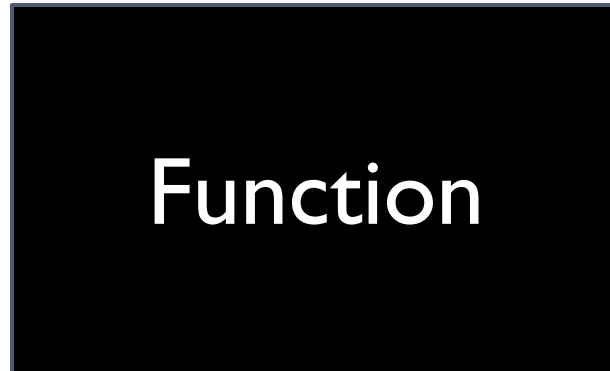


Local variables

- ▶ Variables defined inside functions do not exist outside of those functions

Program

Input



Output



Local variables

- ▶ Functions only know what you tell them
 - ▶ They are oblivious to the outside world

Program

Input
Commands



Local variables

- ▶ Variables defined inside functions do not exist outside of those functions
- ▶ Functions only know what you tell them
 - ▶ They are oblivious to the outside world

