

# CSC 111:

# Intro to Computer Science through Programming

Spring 2017  
Prof. Sara Mathieson



# Admin

- + Homework 9 and Quiz 9 are due **tomorrow (Thursday)**
- + Final project is due May 3 (Wednesday)
- + Remaining graded labs: Lab 10 (today/tomorrow), Lab 11
- + Labs on last two days of classes: practice final
- + **Office hours tomorrow 10am-12pm** (Ford 015)
- + Please read Piazza, lots of hints and important info!

# Outline: 4/19

- + Biology example: **Gene** class
- + Continue classes: variable usage
- + Introduce Lab 10 and Final Project
- + New today: **assert**, list of lists, **\_\_str\_\_**

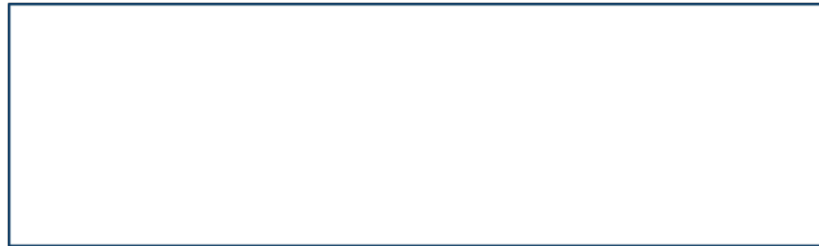
# Continue Classes

## variable usage

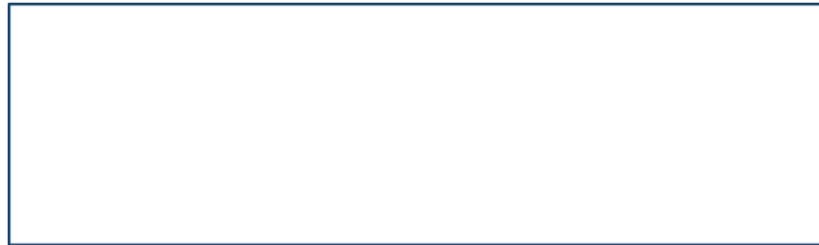
# Global vs. Local



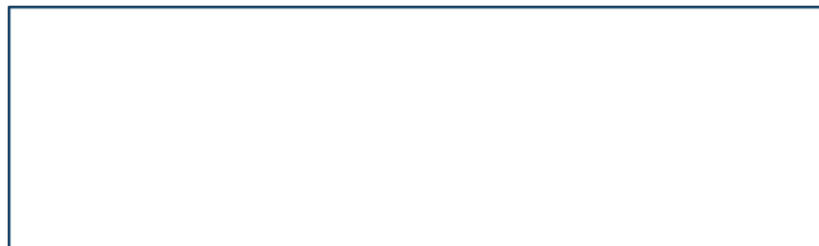
```
def function1(  ,  ,  ):
```



```
def function2(  ,  ):
```



```
def main():
```



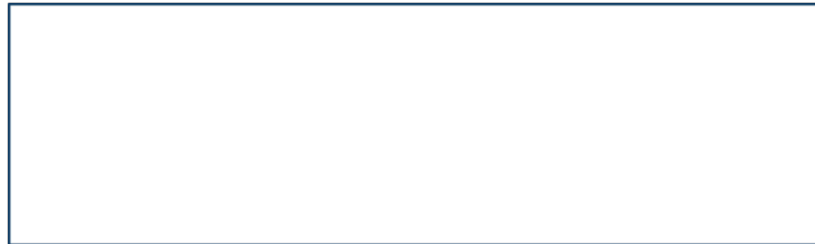
# Global vs. Local



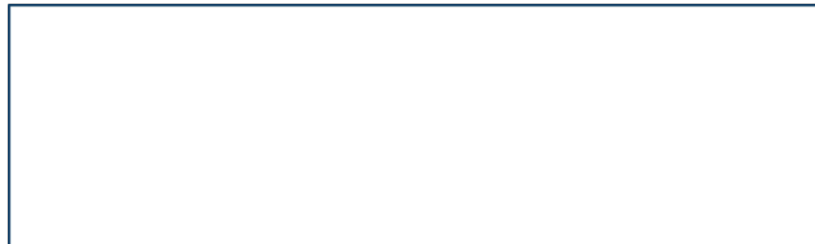
```
def function1(  ,  ,  ):
```



```
def function2(  ,  ):
```



```
def main():
```



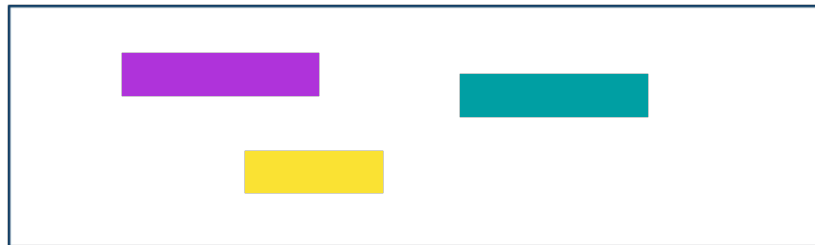
# Global vs. Local



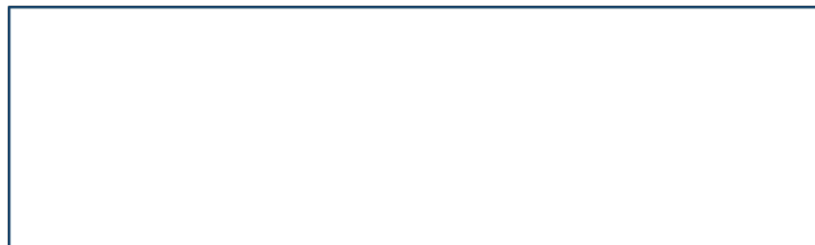
```
def function1( , ,  ):
```



```
def function2( ,  ):
```



```
def main():
```



# Global vs. Local

Global

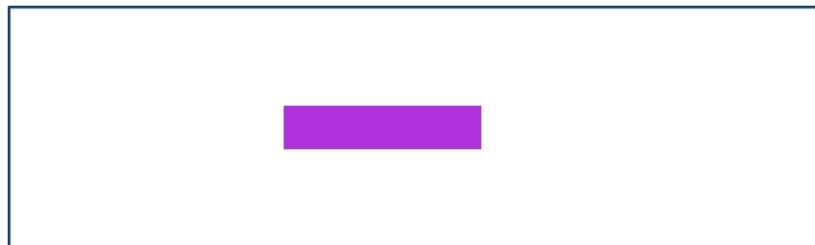
```
def function1(  ,  ,  ):
```



```
def function2(  ,  ):
```



```
def main():
```



Everything else:  
local to their  
functions



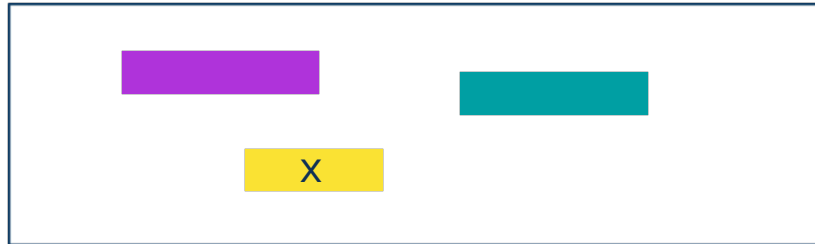
# Global vs. Local

Global

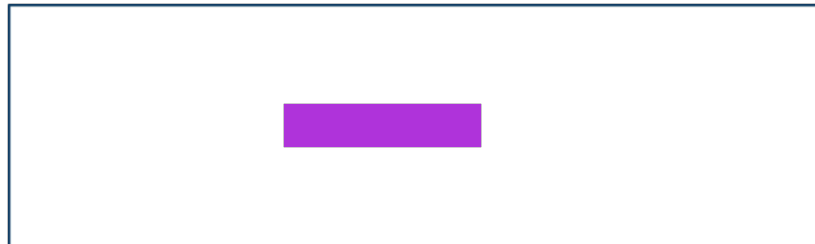
```
def function1(  ,  ,  ):
```



```
def function2(  ,  ):
```



```
def main():
```





Everything else:  
local to their  
functions

Same name?  
still different!


# How the picture changes for classes

```
class MyClass:
```



```
    def __init__(self, , ):
```

```
        self. 
```

```
        self. 
```

```
    def method1(self, ):
```

```
        self.
```

```
    def method2(self, , ):
```

```
        self.
```



```
def function1(, , ):
```

```
def function2(, ):
```


```
def main():
```

# How the picture changes for classes



```
class MyClass:
```

```
    def __init__(self, , ):
```

```
        self.   
        self.
```

```
    def method1(self, ):
```

```
        self.
```

```
    def method2(self, , ):
```

```
        self.
```

```
def function1(, , ):
```

```
    
```

```
def function2(, ):
```



```
    
```

```
def main():
```


```
    
```

# How the picture changes for classes



```
class MyClass:
```

```
    def __init__(self, , ):
```

```
        self.   
        self.
```

```
    def method1(self, ):
```

```
        self.
```

```
    def method2(self, , ):
```

```
        self.
```

```
def function1(, , ):
```

```
       
     
```

```
def function2(, ):
```



```
       
    
```






```
def main():
```


```
    
```

# How the picture changes for classes



```
class MyClass:
```




```
    def __init__(self, , ):
```

```
        self.    
        self.   
```

```
    def method1(self, ):
```

```
        self.   
        
```

```
    def method2(self, , ):
```

```
        self.   
         
```

```
    def function1(, , ):
```

```
           
         
```

```
    def function2(, ):
```



```
           
        
```






```
    def main():
```


```
        
```

# How the picture changes for classes



```
class MyClass:
```

```
    def __init__(self, , ):
```

```
        self.   
        self.   
            
```

```
    def method1(self, ):
```

```
        self.   
         
```

```
    def method2(self, , ):
```

```
        self.   
          
```

```
    def function1(, , ):
```

```
           
         
```

```
    def function2(, ):
```

```
           
        
```

```
    def main():
```

```
        
```

# Biology Example: Gene class

# Biology example: Gene class

- + Goal: test whether a query position is inside a gene or not
- + Motivation: some genomic locations show up on scans for natural selection or disease association
- + If these positions are inside genes, we might understand how they affect physical traits and diseases

```
BASES = ["A", "C", "G", "T"]

# TODO Gene class here

def main():

    # start and end positions for a single gene
    start = 2934
    end   = 5247

    # list of positions - we want to know if they are inside the gene
    query_lst = [2384, 4928, 8374, 238, 3872, 1278, 4374, 12898, 5019]

    # TODO: construct a gene, then check each query position

main()
```



New Today

# Assert (keyword: **assert**)

- + Important debugging strategy
- + Also very useful for checking user input
- + The expression used with assert must evaluate to a boolean
  - **True**, assert holds, nothing happens
  - If this boolean is **False**, assert fails and throws an error

# Assert (keyword: **assert**)

- + Important debugging strategy
- + Also very useful for checking user input
- + The expression used with assert must evaluate to a boolean
  - **True**, assert holds, nothing happens
  - If this boolean is **False**, assert fails and throws an error

```
>>> lst = [4,6,3,2,5,3]
>>> assert 3 in lst
>>> assert 10 in lst
Traceback (most recent call last):
  File "<pyshell#11>", line 1, in <module>
    assert 10 in lst
AssertionError
```

```
>>> user_input = "hello"
>>> assert isinstance(user_input,int)
Traceback (most recent call last):
  File "<pyshell#22>", line 1, in <module>
    assert isinstance(user_input,int)
AssertionError
>>> assert isinstance(user_input,str)
```

# List of Lists (matrix)

- + List of lists, also called a matrix or 2D array
- + Row is always first, then column
- + Example: what is the entry in row 3, column 4?

```
[[4, 8, 9, 0, 7],  
 [4, 9, 0, 2, 8],  
 [7, 6, 7, 2, 0],  
 [5, 7, 5, 5, 6],  
 [3, 8, 2, 0, 0]]
```

# List of Lists (matrix)

- + List of lists, also called a matrix or 2D array
- + Row is always first, then column
- + Example: what is the entry in row 3, column 4?

Column 4

Row 3

```
[[4, 8, 9, 0, 7],  
 [4, 9, 0, 2, 8],  
 [7, 6, 7, 2, 0],  
 [5, 7, 5, 5, 6],  
 [3, 8, 2, 0, 0]]
```

The diagram shows a 5x5 matrix of numbers. An orange box highlights the entry at row 3, column 4, which is the number 5. An orange arrow points from the text 'Row 3' to the start of the third row, and another orange arrow points from the text 'Column 4' to the top of the fourth column.