

CSCI 127: Joy and Beauty of Data

Lecture 8.5: Variable Scope

Reese Pearsall
Summer 2021

<https://reese.github.io/classes/summer2021/127/main.html>

Announcements (Tuesday)

- Lab 4 due **tonight** @ 11:59 PM
- Lab 5 due **tomorrow** @ 11:59 PM
 - > Review Session tomorrow @ 2:00 PM in my Zoom (We will be going over the practice exam)
- Midterm Exam on Thursday
 - > Extra Credit Opportunity Posted
- Program 2 due on Sunday @ 11:59 PM

No class on Monday 5/31

Today

Recursion, Variable Scope

Starting Hack....
Hacking FBI 0%
Hacking FBI 20%
Hacking FBI 40%
Hacking FBI 60%
Hacking FBI 80%
Hacking FBI 100%
FBI Hacked Successfully

In the movies

```
1 print("Starting Hack...");  
2 print("Hacking FBI 0%");  
3 print("Hacking FBI 20%");  
4 print("Hacking FBI 40%");  
5 print("Hacking FBI 60%");  
6 print("Hacking FBI 80%");  
7 print("Hacking FBI 100%");  
8 print("FBI Hacked Successfully");
```

Behind the scenes

Variable Scope

The **scope** of a variable describes the region of code where the variable exists and can be accessed

A **local** variable is a variable that is declared in a function and cannot be accessed from any other functions

A **global** variable is a variable that is declared outside a function and can be accessed from any function

```
g = "CSCI 127"
```

```
def function1(num1, num2):  
    x = num1 + num2  
    return x
```

```
def function2():  
    string = "Hello World"  
    print(string)
```

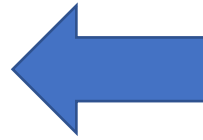
```
def main():  
  
    x = 5  
    y = 2  
  
    answer = function1(x,y)  
    print(answer)  
  
    function2()
```

```
main()
```

```
g = "CSCI 127"
```

```
def function1(num1, num2):  
    x = num1 + num2  
    return x
```

```
def function2():  
    string = "Hello World"  
    print(string)
```



string is a local variable, so trying to reference string variable from outside of the function will result in an error

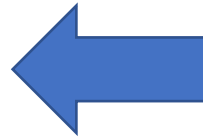
```
def main():  
  
    x = 5  
    y = 2  
  
    answer = function1(x,y)  
    print(answer)  
  
    function2()
```

```
main()
```

```
g = "CSCI 127"
```

```
def function1(num1, num2):  
    x = num1 + num2  
    return x
```


```
def function2():  
    string = "Hello World"  
    print(string)
```



string is a local variable, so trying to reference string variable from outside of the function will result in an error

```
def main():
```

```
    x = 5  
    y = 2
```

```
    answer = function1(x,y)  
    print(answer)  
    print(string)   
    function2()
```

```
main()
```

Traceback (most recent call last):

File "C:/Users/Reese/AppData/Local/Programs/Python/Python39/variable_scope.py", line 24, in <module>
 main()

File "C:/Users/Reese/AppData/Local/Programs/Python/Python39/variable_scope.py", line 21, in main
 print(string)

NameError: name 'string' is not defined

```
g = "CSCI 127"
```

```
def function1(num1, num2):  
    x = num1 + num2  
    return x
```

```
def function2():  
    string = "Hello World"  
    print(string)
```

```
def main():  
  
    x = 5  
    y = 2  
  
    answer = function1(x,y)  
    print(answer)  
  
    function2()
```

```
main()
```

The “Scope” of variable string

```
g = "CSCI 127"
```

```
def function1(num1, num2):  
    x = num1 + num2  
    return x
```

```
def function2():  
    string = "Hello World"  
    print(string)
```

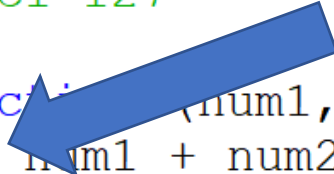
```
def main():  
  
    x = 5  
    y = 2  
  
    answer = function1(x,y)  
    print(answer)  
  
    function2()
```

```
main()
```

The “Scope” of variable num1, num2, and x

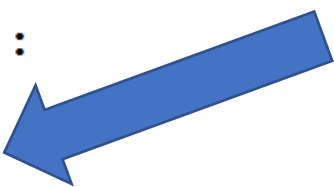

```
g = "CSCI 127"
```

```
def function1(num1, num2):  
    x = num1 + num2  
    return x
```



```
def function2():  
    string = "Hello World"  
    print(string)
```

```
def main():  
    x = 5  
    y = 2  
  
    answer = function1(x, y)  
    print(answer)  
  
    function2()
```



These variables may have the same name, but they are actually two completely different variables!

```
main()
```

```
g = "CSCI 127"
```

```
def function1(num1, num2):  
    x = num1 + num2  
    return x
```

```
def function2():  
    string = "Hello World"  
    print(string)
```

```
def main():
```

```
    x = 5  
    y = 2  
  
    answer = function1(x,y)  
    print(answer)  
  
    function2()
```

```
main()
```

The “scope” of each x variable

```
g = "CSCI 127"
```



The variable g is a **global variable**, which means we can access it from any function

```
def function1(num1, num2):  
    x = num1 + num2  
    return x
```

```
def function2():  
    string = "Hello World"  
    print(string)
```

```
def main():  
  
    x = 5  
    y = 2  
  
    answer = function1(x, y)  
    print(answer)  
  
    function2()
```

```
main()
```

```
g = "CSCI 127"
```

```
def function1(num1, num2):  
    x = num1 + num2  
    return x
```

```
def function2():  
    string = "Hello World"  
    print(string)
```

```
def main():  
  
    x = 5  
    y = 2  
  
    answer = function1(x,y)  
    print(answer)  
  
    function2()
```

```
main()
```

The “scope” of g

```
g = "CSCI 127"
```

```
def function1(num1, num2):  
    x = num1 + num2  
    return x
```

This is the only information that the function1() function can see..

It can only see the local variables, input parameters, and global variables

```
g = "CSCI 127"
```

```
def function1(num1, num2):  
    x = num1 + num2  
    return x
```

This is the only information that the function1() function can see..

It can only see the local variables, input parameters, and global variables



```
g = "CSCI 127"
```

```
def function2():  
    string = "Hello World"  
    print(string)
```

This is the only information that the function2() function can see..

It can only see the local variables, input parameters, and global variables

```
g = "CSCI 127"
```

This is the only information that the main() function can see..

It can only see the local variables, input parameters, and global variables

```
def main():  
  
    x = 5  
    y = 2  
  
    answer = function1(x,y)  
    print(answer)  
  
    function2()  
  
main()
```



```
g = "CSCI 127"
```

```
def function1(num1, num2):  
    x = num1 + num2  
    return x
```

```
def function2():  
    string = "Hello World"  
    print(string)
```

```
def main():
```

```
    x = 5
```

```
    y = 2
```

```
    answer = function1(x, y)  
    print(answer)
```

```
    function2()
```

```
main()
```

Our functions are isolated from each other.

The only ways we can communicate from one function to another is giving the function some input or getting some output from a function