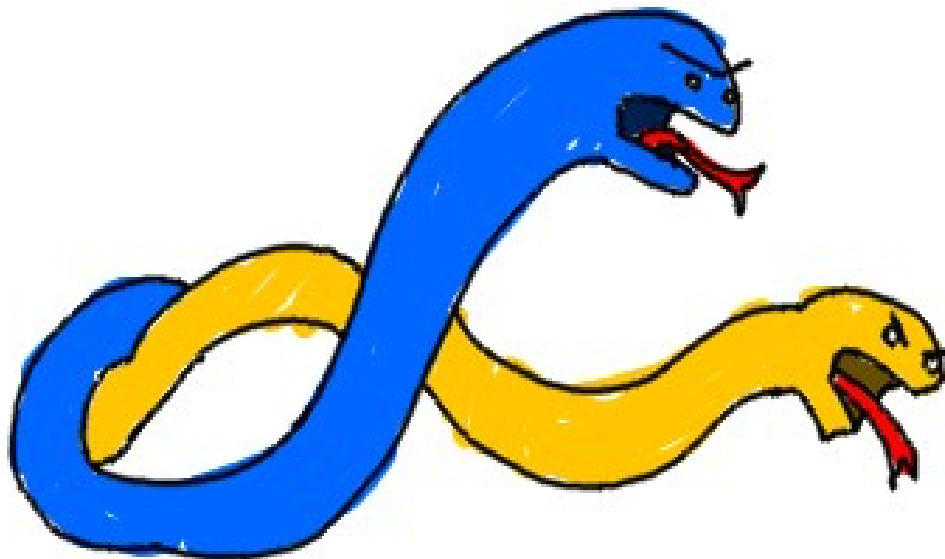


# Python vs. C++



# Loops

- In Python, you probably used several different types of loops:

- While Loops

```
while ( readAllFiles == False ):
```

- For Loops

```
for index in range( 0, len( priceList ) ):
```

- For-In (or For-Each) Loops

```
for student in studentList:
```

- C++ has these loops as well
  - Except C++'s version of the For-Each loop is ugly in C++98, so we will substitute that out with standard For loops.



# While Loops

- C++ has the **while** and the **do while** loops:

```
while ( fees > 0 )  
{  
    balance -= 10.00;  
    fees -= 10.00;  
}
```

The **while** loop checks to see if a conditional statement is **true**. If it is, it will begin iterating through the loop until the conditional statement is **false**.

```
do  
{  
    cin >> choice;  
} while ( choice < 0 )
```

The **do while** loop is similar to the while loop, except that it runs through the inner statement **at least once** before it checks whether the conditional value is true or not.

If the conditional statement is **false**, it only runs through once.

# For Loops

- C++ For Loops look pretty different from Python's Python

```
for [var_name] in range( start_val, end_val ):
    ...
```

## C++

```
for ( int var_name = start_val; var_name < end_val; var_name++ )
{
}
```



# For Loops

```
for ( int var_name = start_val; var_name < end_val; var_name++ )  
{  
}
```

- The statements in the for loop are separated by semicolons ;
- We begin by declaring a variable
  - **int var\_name = start\_val;**
  - The variable can be declared ahead of time. Then it would be  
**var\_name = start\_val;**

# For Loops

```
for ( int var_name = start_val; var_name < end_val; var_name++ )  
{  
}
```

- The second part is to specify the ending condition:
  - Think of  
**if ( var\_name < end\_val )**
  - Once **var\_name** is equal to **end\_val**, the loop will stop running.



# For Loops

```
for ( int var_name = start_val; var_name < end_val; var_name++ )  
{  
}
```

- The third part specifies how much we are going to increment our **var\_name** variable each time through the loop.
  - We could increment it by more than one, or we could do other math to it:
    - **var\_name** += 50    // Add 50
    - **var\_name** \*= 2    // Multiply by 2
    - **var\_name** -= 10    // Subtract 10
    - **var\_name** /= 3    // Divide by 3

# For Loops

## Python

```
for index in range( 0, 10 ):  
    print( index )
```

## C++

```
for ( int index = 0; index < 10; index++ )  
{  
    cout << index << endl;  
}
```

**variable**, **start value**, **end value**



# Loops

## Additional Reading

- <http://www.cplusplus.com/doc/tutorial/control/>
- <http://www.learncpp.com/cpp-tutorial/55-while-statements/>
- <http://www.learncpp.com/cpp-tutorial/56-do-while-statements/>
- <http://www.learncpp.com/cpp-tutorial/57-for-statements/>