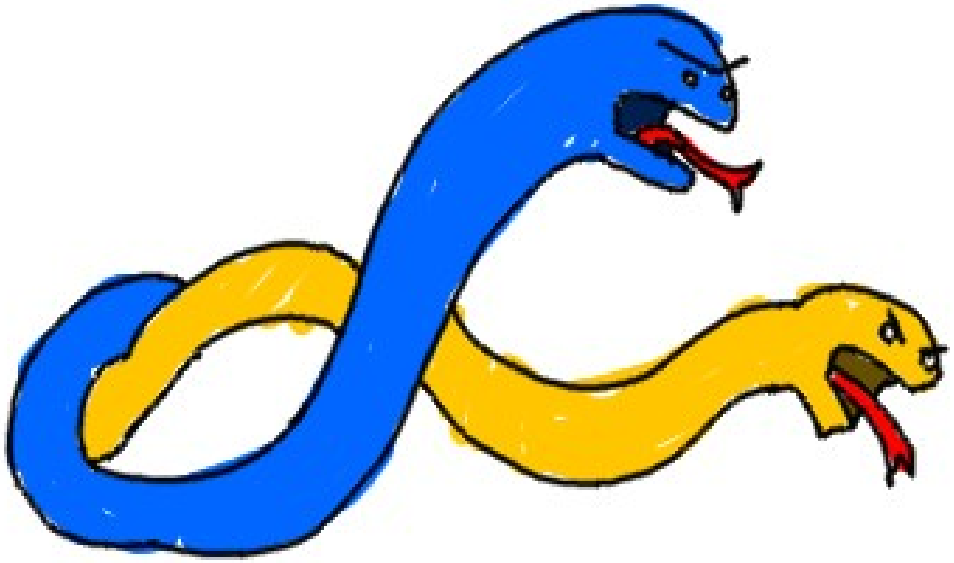
Python vs. C++



Written by Rachel J. Morris, last updated 2013-05-24

Functions

- Functions are a way to *delegate* a task in your program to another section.
- It helps keep your program *modular*, making it easier to maintain and to debug.
  - "Oh, the airline ticket checkout is adding too much tax, I better check the line-item calculation function!"

Functions

You can put functions in a separate file, or above the **main**/program execution.

Python

```
def SumNumbers( number1, number2 ):
    return number1 + number2




num1 = input( "Enter first number: " )
num2 = input( "Enter second number: " )


sum = SumNumbers( num1, num2 )
print( "Sum is " + str( sum )
```

C++

```
#include <iostream>
using namespace std;

int SumNumbers( int number1, int number2 )
{
    return number1 + number2;
}

int main()
{
    int num1, num2;

    cout << "Enter two numbers: ";
    cin >> num1;
    cin >> num2;

    int sum = SumNumbers( num1, num2 );
    cout << "Sum is " << sum << endl;

    return 0;
}
```

Functions

Python

```
def SumNumbers( number1, number2 ):
    return number1 + number2
```

C++

```
int SumNumbers( int number1, int number2 )
{
    return number1 + number2;
}
```

In Python, you create a function by beginning it with the **def** keyword, and then the **function name**.

Inside of parenthesis, you specify the function **parameters** – these are variables that will be passed in when the function is called.

The function may or may not return values, but there is no way specify it in the **function header** (the line with the def & parameters)

Functions

Python

```
def SumNumbers( number1, number2 ):
    return number1 + number2
```

C++

```
int SumNumbers( int number1, int number2 )
{
    return number1 + number2;
}
```

In C++, you begin your function header with a **return-data-type**. Here, it is specified to be an integer.

A C++ function can return any data-type – even a class you wrote yourself.

If you wish to not return any values in the function, you use **void** instead of **int, float, string,** or anything else.

Functions

Python

```
def SumNumbers( number1, number2 ):
    return number1 + number2
```

C++

```
int SumNumbers( int number1, int number2 )
{
    return number1 + number2;
}
```

The **parameters** in C++ must also have a specified data-type. Here, we can only have **int** variables passed into our SumNumbers function.

In the SumNumbers function in C++, you can only pass in integer values (if you pass in a float, it will ***drop*** the numbers to the right of the decimal – no rounding).

This function will only return an int value as well.

Functions

Python

```
def DoubleNum( number1, number2 ):
    return number1 * 2, number2 * 2
```

- Python functions can return more than one value.
  - C++ cannot return more than one value via the **return** statement (but there are other ways to achieve this)
- Python function parameters can be any data-type. The SumNumbers function will work with integers and floats.
  - In C++, the **data-type is concrete; set in stone:** you would have to write another function to handle summing together floats. However, you can use the same “SumNumbers” function name.

Functions

- C++ functions and their parameters must specify a return-type:

```
string FormatName( string firstName, string lastName )
{
   return lastName + ", " + firstName;
}
```

```
float AddTax( float price, float taxRate )
{
   return price + ( price * taxRate );
}
```

```
bool IsPasswordTooShort( string password )
{
   return ( password.size() < 4 );
}
```

Functions

- (Void return type)
- (Parameters vs. Arguments)
  - Names don't need to match
- (Scope)

Functions

Additional Reading

- http://www.cplusplus.com/doc/tutorial/functions/
- http://www.learncpp.com/cpp-tutorial/71-function-parameters-and-arguments/