

Java Functions

Python	Java
<pre>def function_name(var1, var2, ...): statement statement ...more statements...</pre>	<pre>public static returnType functionName(type param1, type param2, ...) { statement statement ...more statements... }</pre>

Comparison:

- Java forces the programmer to declare the *data type* of each parameter to a function, along with the data type of the return value.
 - If the function takes no arguments, the parentheses can be left empty (like Python).
 - If the function does not return anything, you must use the return type of **void**.
- The **return** keyword in Java works just like in Python. When returning a value from a Java function, you must make sure the value being returned has the same data type as the declared return type.

Function overloading

Because Java forces you to declare the types of your parameters to the functions you write, Java allows you to define multiple functions with the same name, but different numbers of parameters or different parameter data types. Java will figure out which one to call based on the arguments that are passed to the function when it is called. These three functions can coexist just fine:

```
public static void g(int x) {  
    System.out.println("In g, x is an integer:" + x);  
}
```

```
public static void g(String x) {  
    System.out.println("In g, x is a string:" + x);  
}
```

```
public static void g() {  
    System.out.println("In g, no parameters!");  
}
```

To be clear, the three functions above are completely separate and independent from each other; they just happen to share the same name (g). You can put whatever code you want inside each function. Java figures out which one to call based on the numbers and types of the parameters you pass to g.