Function Syntax

Here are the syntax rules for defining functions.

```
type name(parameters)
{
   ... body ...
}
```

- **type** is the kind of value returned by the function
- **name** is the function name used when calling it
- **parameters** are a list of variable declarations separated by commas, giving the type and name of each input to the function.

Here is an example function convert, from the *f2c* program which you saw earlier:

```
1  double convert(double temp)
2  {
3    return (temp - 32) * 5.0 / 9.0;
4  }
```

1. The **type** of this function is `double`.
2. The **name** of the function is `convert`.
3. The function has **one parameter** of type `double`.

A **parameter** (*aka* **formal parameter**) is a placeholder for one of the **arguments** (*aka* **actual parameters**), supplied in the function **call**. It acts like a local variable.

Each parameter is initialized at the time the function is called, using a copy of the value of its corresponding argument. Matching is done **by position**, and not by name. If a function has no parameters, the parameter list in the header is empty.