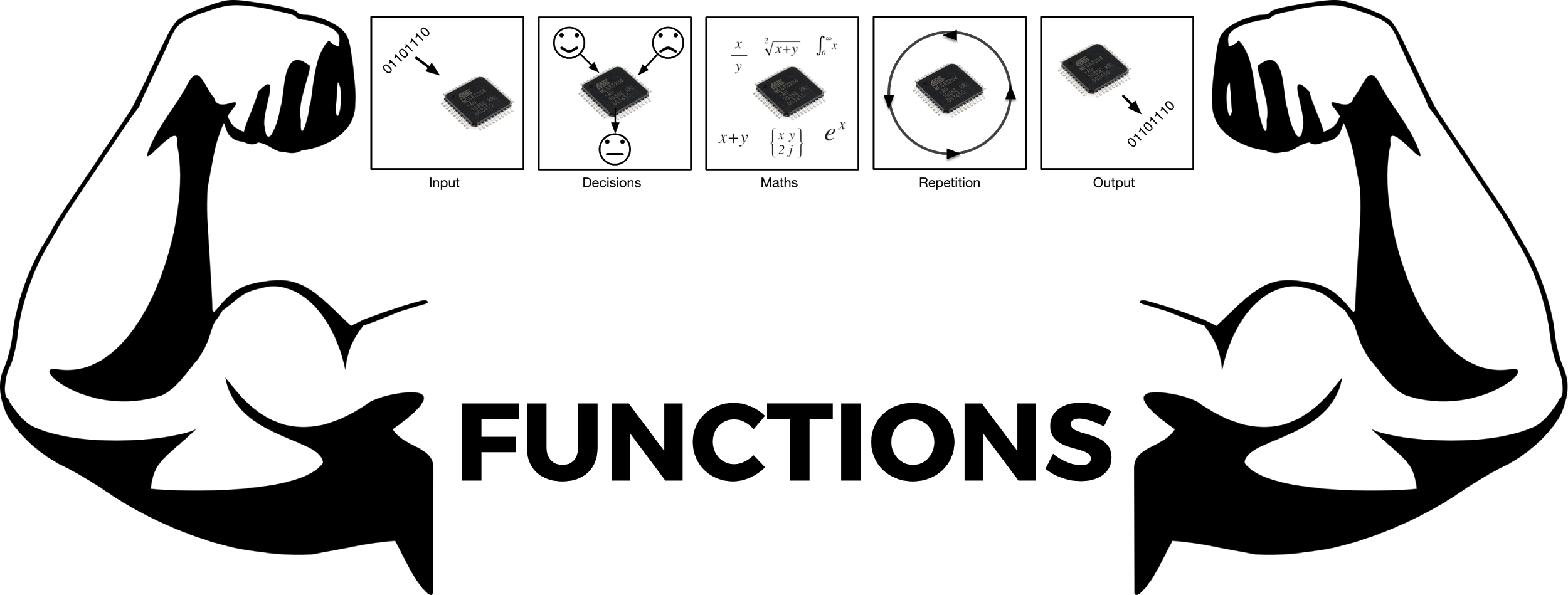
|  |  |
| --- | --- |
| **Reading Time** | 20 mins |
| **Difficulty** | Intermediate |

|  |
| --- |
| **Sections** |
| Inputs |
| Outputs |
| Writing functions |

**Functions (H1)**

**A function is a group of statements that perform a task. They are the building blocks of any program, performing the computations, decisions and IO.**



**Inputs (H1)**

**Functions can take input parameters if they need to work with data. Not all functions do this, but most do. The function below is called digitalWrite, and it takes two inputs, pin number and state.**

digitalWrite(3, LOW);

**This function will write a value of LOW (0 volts) to digital pin number 3.**

**All functions expect you to give them the right type of data. For example if you tried to write the statement below…**

digitalWrite(3, -304923);

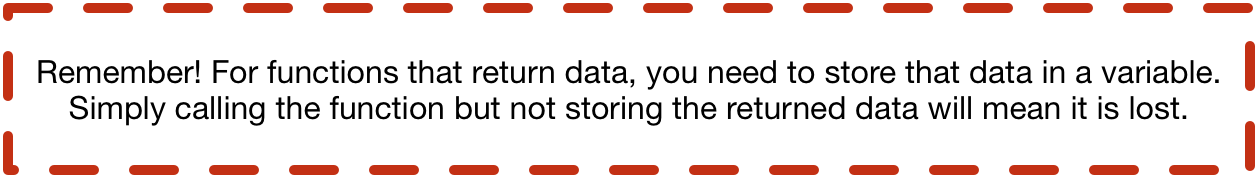
**You’d be giving the function the value of -304923 when it either a HIGH or a LOW value. In this case, the function would just convert -304923 to a HIGH value.**

**Outputs**

**Some functions return data after they have run. Math functions tend to do this because they have to return a result.**

value = sin(1.20);

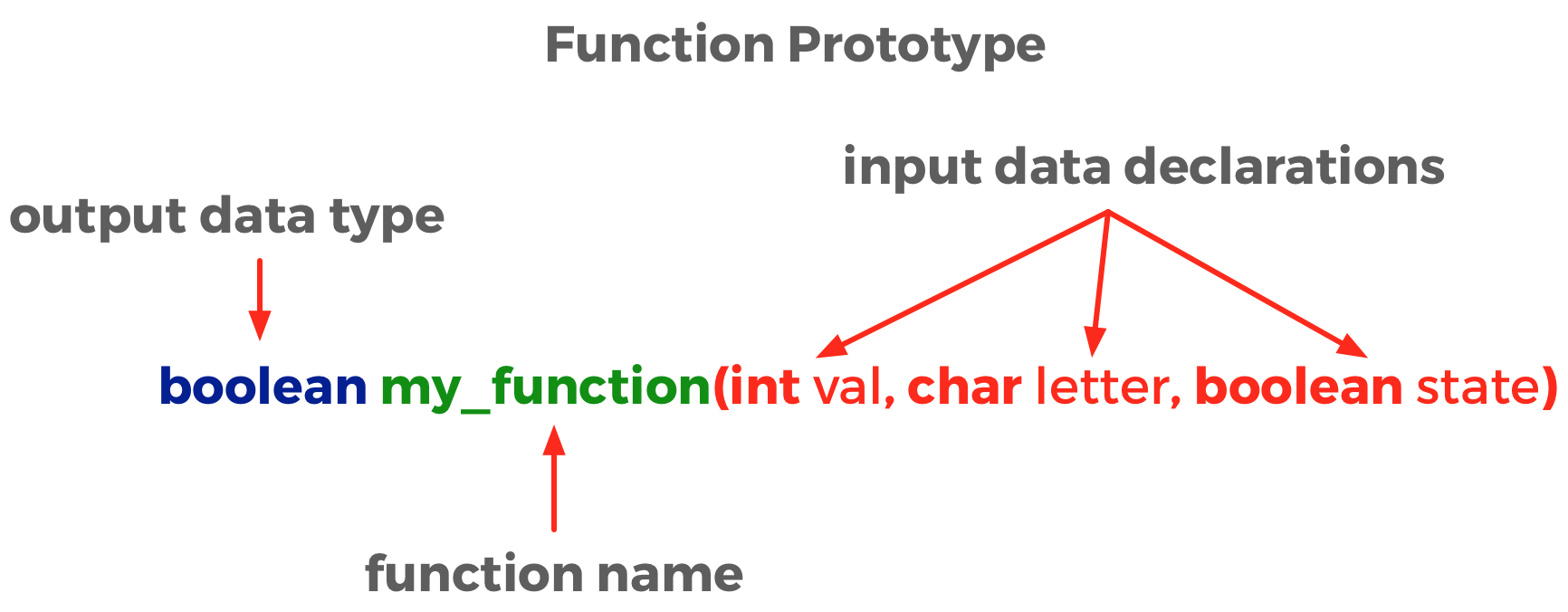
**The sin function returns the calculated value, which is stored in value.**



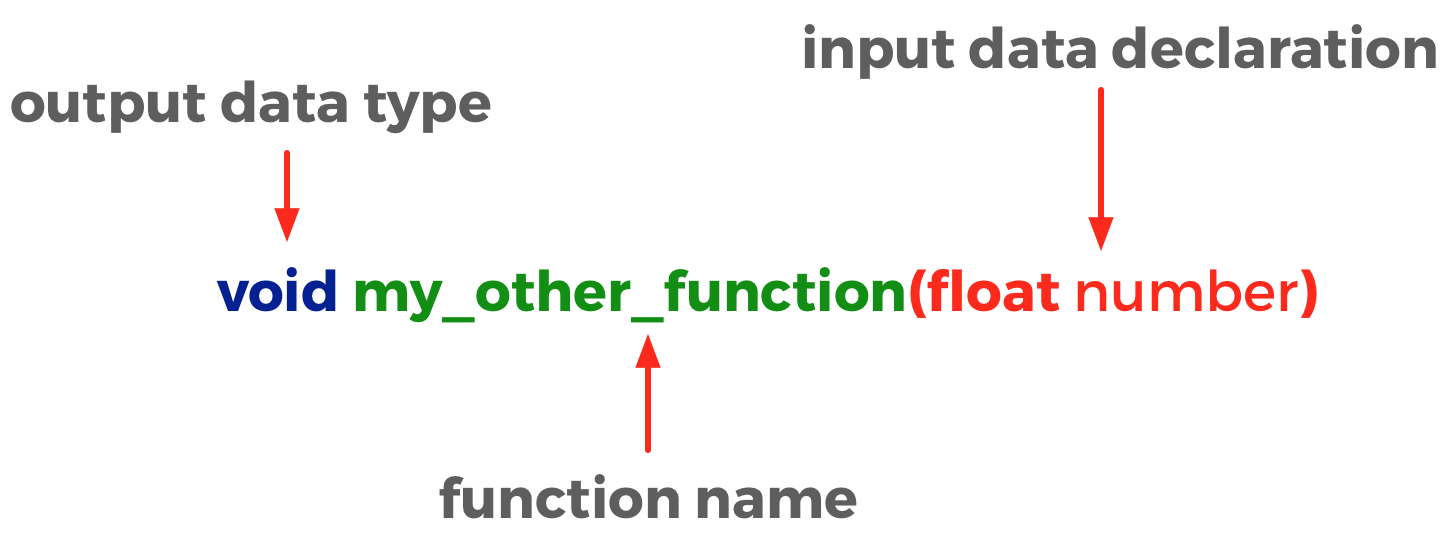
**Writing functions**

**All the functions we’ve used so far come with the Arduino environment, but it’s very useful to be able to write our own functions.**

**To do this we must first define the function prototype, this defines the name of the function, the inputs and the output.**



**Defines a function called “my\_function”, that takes three inputs; an integer called** val**, a character called** letter **and a boolean called** state**. The function outputs a boolean type.**



**This defines a function called “my\_other\_function” that takes a float type input. The “void” return type means the function doesn’t return anything.**

**After the function prototype we must write the operations that make up the function. Let’s take our example of a flashing LED from the previous example and write a function for it.**

**Here is the initial code using only the Arduino functions**

int outputPin = 3;

Int delayTime = 1000;

Void setup(){

pinMode(outputPin, OUTPUT);

}

Void loop(){

digitalWrite(outputPin, LOW);

delay(delayTime);

digitalWrite(outputPin, HIGH);

delay(delayTime);

}

**Let’s write our own function for the code that flashes the LED.**

void flash(int pinNumber){

digitalWrite(outputPin, LOW);

delay(1000);

digitalWrite(outputPin, HIGH);

delay(1000);

}

**Now we can rewrite our code like this…**

int outputPin = 3;

Int delayTime = 1000;

Void setup(){

pinMode(outputPin, OUTPUT);

}

Void loop(){

flash(3);

}

void flash(int pinNumber){

digitalWrite(outputPin, LOW);

delay(1000);

digitalWrite(outputPin, HIGH);

delay(1000);

}

**Look how simple the code inside loop() has become. This is one benefit of functions, it makes our code a lot easier to read and understand. We could even place the function definition in a seperate file and keep our main program even cleaner!**

**You might notice now that loop() and setup() are both functions that take no input data and output no data!**