

## CODE TO NOTE

---

### ANALOG OUTPUT (PWM):

```
analogWrite(RedPin, 100);
```

The `analogWrite()` function outputs a voltage between 0 and 5V on a pin. The function breaks the range between 0 and 5V into 255 little steps. Note that we are not turning the LED on to full brightness (255) in this code so that the night-light is not too bright. Feel free to change these values and see what happens.

---

### NESTED IF STATEMENTS:

```
if(logic statement){  
    if(logic statement){  
    }  
}
```

A nested **if** statement is one or more **if** statements “nested” inside of another **if** statement. If the parent **if** statement is true, then the code looks at each of the nested **if** statements and executes any that are true. If the parent **if** statement is false, then none of the nested statements will execute.

---

### MORE LOGICAL OPERATORS:

```
(potentiometer > 0 &&  
    potentiometer <= 150)
```

These **if** statements are checking for two conditions by using the AND `&&` operator. In this line, the **if** statement will only be true if the value of the variable `potentiometer` is greater than 0 AND if the value is less than or equal to 150. By using `&&`, the program allows the LED to have many color states.

---

### DEFINING A FUNCTION:

```
void function_name(){  
}
```

This is a definition of a simple function. When programmers want to use many lines of code over and over again, they write a function. The code inside the curly brackets “executes” whenever the function is “called” in the main program. Each of the colors for the RGB LED is defined in a function.

---

### CALLING A FUNCTION:

```
function_name();
```

This line “calls” a function that you have created. In a later circuit, you will learn how to make more complicated functions that take data from the main program (these pieces of data are called **parameters**).

---