Control Wink's Eyes

Lesson 2

Eye Control Functions...

Lets play with Wink's eyes. It's super easy. Prepare to be amazed.

```
void loop(){
  EyesPurple(100);  //both eyes purple at 100 brightness
  delay(3000);      //wait 3 seconds
  RightOff();       //turn off right eye (so he “winks”)
  delay(250);      //wait 1/4 second
}
```

Wink_Ch02Eyes_Ex01

This is where the fun happens. Everything inside the loop() function is run over and over. You'll put most of your code inside this function.

The loop function runs each line of code from top to bottom. When the last line inside the loop() function completes, the first line is immediately run again and the loop continues in this way.

The function EyesPurple(); turns on both eyes to purple color. The delay() function causes the processor to wait for a certain number of milliseconds. There are 1000 milliseconds in one second. You can add any number of eyes commands and delays inside the loop to create all kinds of interesting eye color sequences.

Here is a list of all the functions you can call to control the eyes.

Make both eyes turn the same color:

```
EyesRed();     EyesGreen();  EyesBlue();    EyesPurple();  EyesPink();  EyesYellow();
EyesOrange();  EyesCyan();   EyesWhite();
```

Make just the LEFT eye change color:

```
LeftRed();     LeftGreen();  LeftBlue();    LeftPurple();  LeftPink();  LeftYellow();
LeftOrange();  LeftCyan();   LeftWhite();
```

Make just the RIGHT eye change color:

```
RightRed();     RightGreen();  RightBlue();  RightPurple();  RightPink();  RightYellow();
RightOrange();  RightCyan();   RightWhite();
```

Set exact red, green, and blue values. Values can range from 0 to 255:

```
EyesRGB(red,green,blue);     // sets both eyes to the same value
LeftRGB(red,green,blue);     // sets left eye to specific value
RightRGB(red,green,blue);    // sets right eye to specific value
```

Turn eyes off:

```
EyesOff();     // turns off both eyes
LeftOff();     // turn off the left eye
RightOff();    // turn off the right eye
```

Remember Previous Color:

```
EyesOn(); //sets eyes to previous color
LeftOn(); //sets left eye to prev color
RightOn();//sets right eye to prev color
```

1

SKILL LEVEL

Try a working example...

We’ve seen Wink do his “wink”, now lets have fun with a longer example. Study the example below then experiment by making your own changes to it. Try changing the “LeftCyan()” functions to some of the functions on the previous page. Also play with different values in the delay() function. If you get compiler errors, look at the last page of this lesson for suggestions to fix them.

```
void loop(){
  LeftCyan(100);    //make left eye cyan color, 100 brightness
  delay(20);        //brief blink time
  EyesOff();        //turn eyes off
  delay(100);       //delay between blinks
  LeftCyan(100);     //begin second blink
  delay(20);        //brief blink time
  EyesOff();        //turn eyes off again

  delay(2000);      //time before 2nd eye blinks

  RightCyan(100);   //make right eye cyan color, 100 brightness
  delay(20);        //brief blink time
  EyesOff();        //turn eyes off
  delay(100);       //delay between blinks
  RightCyan(100);    //begin second blink
  delay(20);        //brief blink time
  EyesOff();        //turn eyes off again

  delay(2000);      //time before 2nd eye blinks
}
```

Wink_Ch02Eyes_Ex02

Blink the left eye two times. The code is run one line at a time.

Each line is called a “statement”, and each statement needs to end with a semicolon. This is the ; symbol.

Now blink the right eye two times.

The functions are “case sensitive”, which means you must use the correct upper case and lower case letters.

Remembering a previous color...

You can use EyesOn(), LeftOn(), and RightOn() to set the eyes back to a previous value. This is useful if you want to blink the eyes. Have a look at this example.

```
void loop(){
  EyesRed(150);     //make eyes red color, 150 brightness
  delay(20);        //brief blink time
  EyesOff();        //turn eyes off
  delay(100);       //delay between blinks
  EyesOn();        //set eyes back to previous value (Red 150)
  delay(20);        //brief blink time
  EyesOff();        //turn eyes off again
  wait(2);          //wait 2 seconds
}
```

Wink_Ch02Eyes_Ex03

When you run EyesOff(); Wink will remember what color his eyes were before being turned off.

When you run EyesOn(); both eyes will be restored to this previous color.

This works the same way with LeftOn(); and RightOn(); Experement with these on your own.

Advanced Eyes() functions tricks...

Using functions like EyesGreen() and EyesPink() are useful for setting a color. You can also use these to control the brightness of the color. Each eye can be set to a brightness ranging from 0 to 255, where 0 is completely off, and 255 is max bright. You can control the brightness of the color by changing the value inside the parenthese. A starting value of 100 is pretty good. The eyes are super bright, so using larger numbers normally isn't necessary and will use battery power more quickly but sometimes it is useful to make the eyes super bright. Check out this example.

```
void loop(){
  EyesBlue(25);      //set both eyes to blue, at brightness 20
  delay(100);        //wait a short time
  EyesBlue(75);      //set both eyes to blue, at brightness 75
  delay(100);        //wait a short time
  EyesBlue(125);     //set both eyes to blue, at brightness 125
  delay(100);        //wait a short time
  EyesBlue(175);     //set both eyes to blue, at brightness 175
  delay(100);        //wait a short time
  EyesBlue(225);     //set both eyes to blue, at brightness 225
  delay(100);        //wait a short time

  EyesOff();         //turn eyes off
  delay(1000);       //wait 1 second then repeat
}
```

Wink_Ch02Eyes_Ex04

Set both eyes to a low starting value. Show each level for a short time then go to a brighter value.

Eyes become sequentially brighter.

At the end we turn off both lights and wait about 1 second before the sequence begins again.

Using the EyesRGB() function...

Using functions like EyesBlue() can quickly set certain colors, but you can make the eyes any color you can think of. Each eye actually contains three individual color lights. One is red, one is green, and one is blue. By making these three individual lights various brightnesses, you can create any color you can image. You can set the exact level of red, green, and blue by using the EyesRGB() functions. Like this...

```
void loop(){
  //set red to 220, green to 20, blue to 160
  EyesRGB(220, 30, 160);    // Plum purple!!
  delay(100);
  EyesRGB(100, 100, 100);   // set all the same for white!
  delay(100);
}
```

Wink_Ch02Eyes_Ex05

The EyesRGB() function accepts three "arguments". The arguments are the three values you pass into the function. They are separated by commas. The first value is the red level (220 in this case), the second is the green level, and the last is the blue level. These values can each be set to any level between 0 and 255.

Setting R, G, and B to the same amount will result in a white color. Cool!

Errors compiling?

In order to turn your human readable code into instructions the robot's brain can process, the code needs to be written in a certain way. Certain rules must be followed. You'll eventually make a mistake in your code and Arduino will refuse to compile it. (You'll get orange errors in the message window at the bottom). Let's consider some common mistakes you may encounter.

```
EyesBlue(50);     //correct use of semicolon ; symbol
EyesBlue(50)      //forgot semicolon. This won't compile.
EyesBlue(50):     //this is a colon, not a semicolon. incorrect.
EyesBlue(50),     //this is a comma, not a semicolon. incorrect.
```

Don't forget your semicolons at the end of each statement. This is an easy mistake to make.

```
EyesBlue(50);     //correct upper-lowercase letters. This works.
eyesBlue(50);     //doesn't work (lower case e is wrong)
eyesblue(50);     //doesn't work
EYESBLUE(50);     //doesn't work

delay(100);       //this works. "delay" is all lower case.
Delay(100);       //doesn't work
```

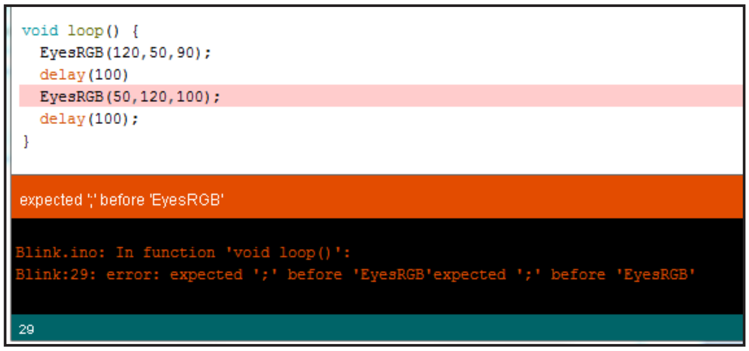
Be sure you're using correct upper case and lower case letters. Some functions need all lower case letters like delay(), and some functions use upper and lower case letters like EyesBlue().

```
EyesRGB(120, 50, 90);      //correct parenthesis. This works.
EyesRGB{120, 50, 90};      //curly braces are incorrect.
EyesRGB[120, 50, 90];      //square brackets are incorrect.
```

Be sure you're using normal parentheses around arguments.

```
EyesRGB(120,50,90);        //This is okay.
EyesRGB( 120, 50, 90);    //This is okay.
EyesRGB (120,  50, 90) ; //This is also okay, but not as common.
```

Extra spaces are called "white space" in programming. In the C language, which is what you're learning, white space is usually okay. You should get used to writing code formatted like either of the top two lines.



```
void loop() {
  EyesRGB(120,50,90);
  delay(100)
  EyesRGB(50,120,100);
  delay(100);
}
```

```
expected ';' before 'EyesRGB'
```

```
Blink.ino: In function 'void loop()':
Blink:29: error: expected ';' before 'EyesRGB'expected ';' before 'EyesRGB'
```

This is the Arduino sketch window when a mistake is found. In this example we forgot a semicolon after the first delay(). Often the error will highlight the line following your mistake. Study the line above the highlighted line if a mistake is found.

By reading the error message in orange, you can often get a clue of what your mistake was.