

# Night Light Solution

Since the night light was left as an open ended problem, your solution probably won't match the one below; but as long as it achieved the design goals of turning the LED off when the lights went on and on when the lights went off, it's still perfectly valid.

```
void setup() { }
```

Here we have a totally empty setup function, it doesn't do anything. We are required by the Arduino compiler, however, to include this empty setup function. The program will not compile without it.

```
int threshold = 150;
```

Here we define a global variable (one defined at the top level of the program), to specify the reading from the light sensor (expressed in ADU) at which we consider the lights to go from on to off.

```
void loop() {  
  int light = readLightSensor();  
  if(light > threshold){  
    ledOff();  
  }  
  else{  
    ledOn();  
  }  
}
```

Here in the main program loop, we check the value of the light sensor and assign it to a local variable called light. We compare that value to the threshold. If the value is greater than the threshold, we consider the room lights to be on, and we turn the led off. If the value is less than the threshold, we consider the room lights to be off, and we turn the led on.

```
int readLightSensor() {  
  return analogRead(5);  
}
```

This is the exact same function we defined in the LightSensor code. We simply encapsulate the reading of the ADC for the light sensor into a function.

```
int ledPin = 13;

void ledOn(){
    digitalWrite(ledPin, HIGH);
}

void ledOff(){
    digitalWrite(ledPin, LOW);
}
```

This is exactly the same function as defined in the Blink sketch. Here we have a block of code where we first instantiate a new variable that defines to which pin the LED is attached. The next two functions turn the LED on and off respectively using the digitalWrite function and setting that pin to either HIGH (on) or LOW(off).