```
// Blink an LED
// ------------

/*-------------

We've heavily commented this code for you. If you're a pro, feel free to ignore it.

Comments start with two slashes or are blocked off by a slash and a star.
You can read them, but your device can't.
It's like a secret message just for you.

Every program based on Wiring (programming language used by Arduino, and Particle devices) has two essential parts:
setup - runs once at the beginning of your program
loop - runs continuously over and over

You'll see how we use these in a second.

This program will blink an led on and off every second.
It blinks the D7 LED on your Particle device. If you have an LED wired to D0, it will blink that LED as well.

-------------*/


// First, we're going to make some variables.
// This is our "shorthand" that we'll use throughout the program:

// Instead of writing D0 over and over again, we'll write led1
// You'll need to wire an LED to this one to see it blink.

int led2 = D7; // Instead of writing D7 over and over again, we'll write led2
// This one is the little blue LED on your board. On the Photon it is next to D7, and on the Core it is next to the USB jack.

// Having declared these variables, let's move on to the setup function.
// The setup function is a standard part of any microcontroller program.
// It runs only once when the device boots up or is reset.

void setup() {

  // We are going to tell our device that D0 and D7 (which we named led1 and led2 respectively) are going to be output
  // (That means that we will be sending voltage to them, rather than monitoring voltage that comes from them)
  // It's important you do this here, inside the setup() function rather than outside it or in the
loop function.
  pinMode(led2, OUTPUT);

}

// Next we have the loop function, the other essential part of a microcontroller program.
// This routine gets repeated over and over, as quickly as possible and as many times as
possible, after the setup function is called.
// Note: Code that blocks for too long (like more than 5 seconds), can make weird things
happen (like dropping the network connection).  The built-in delay function shown below
safely interleaves required background activity, so arbitrarily long delays can safely be done
if you need them.

void loop() {
  // To blink the LED, first we'll turn it on...
  //Turn the LED off
  digitalWrite(led2, LOW);
  delay(5000);


  //Letter J
  digitalWrite(led2, HIGH);
  delay(500);
  digitalWrite(led2, LOW);
  delay(500);

  digitalWrite(led2, HIGH);
  delay(1000);
  digitalWrite(led2, LOW);
  delay(500);

  digitalWrite(led2, HIGH);
  delay(1000);
  digitalWrite(led2, LOW);
  delay(500);

  digitalWrite(led2, HIGH);
  delay(1000);
  digitalWrite(led2, LOW);
  delay(500);

  //Letter O
  digitalWrite(led2, HIGH);
  delay(1000);
  digitalWrite(led2, LOW);
  delay(500);
```

```
digitalWrite(led2, HIGH);
delay(1000);
digitalWrite(led2, LOW);
delay(500);

digitalWrite(led2, HIGH);
delay(1000);
digitalWrite(led2, LOW);
delay(500);

//Letter H:
digitalWrite(led2, HIGH);
delay(500);
digitalWrite(led2, LOW);
delay(500);

digitalWrite(led2, HIGH);
delay(500);
digitalWrite(led2, LOW);
delay(500);

digitalWrite(led2, HIGH);
delay(500);
digitalWrite(led2, LOW);
delay(500);

digitalWrite(led2, HIGH);
delay(500);
digitalWrite(led2, LOW);
delay(500);

//Letter N:
digitalWrite(led2, HIGH);
delay(1000);
digitalWrite(led2, LOW);
delay(500);

digitalWrite(led2, HIGH);
delay(500);
digitalWrite(led2, LOW);
delay(500);
```

```
  // Wait 1 second...
  delay(5000);

  // And repeat!
}
```