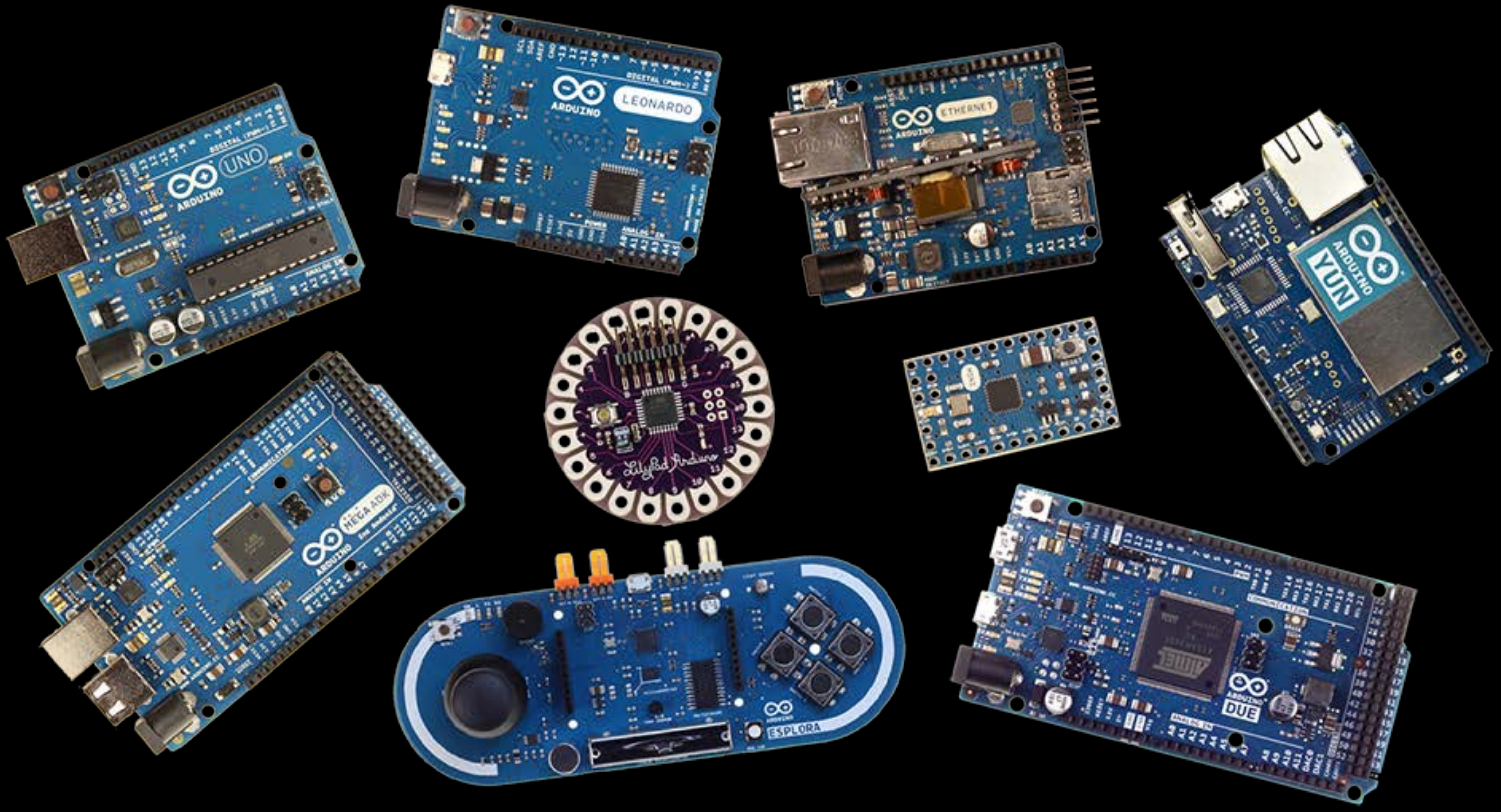
Part of a Balanced Breakfast

SerialPort

Ard…what now?


ARDUINO UNO
ARDUINO LEONARDO
ARDUINO ETHERNET
ARDUINO YUN
LilyPad Arduino
ARDUINO MEGA ADK
ARDUINO ESPLORA
ARDUINO DUE

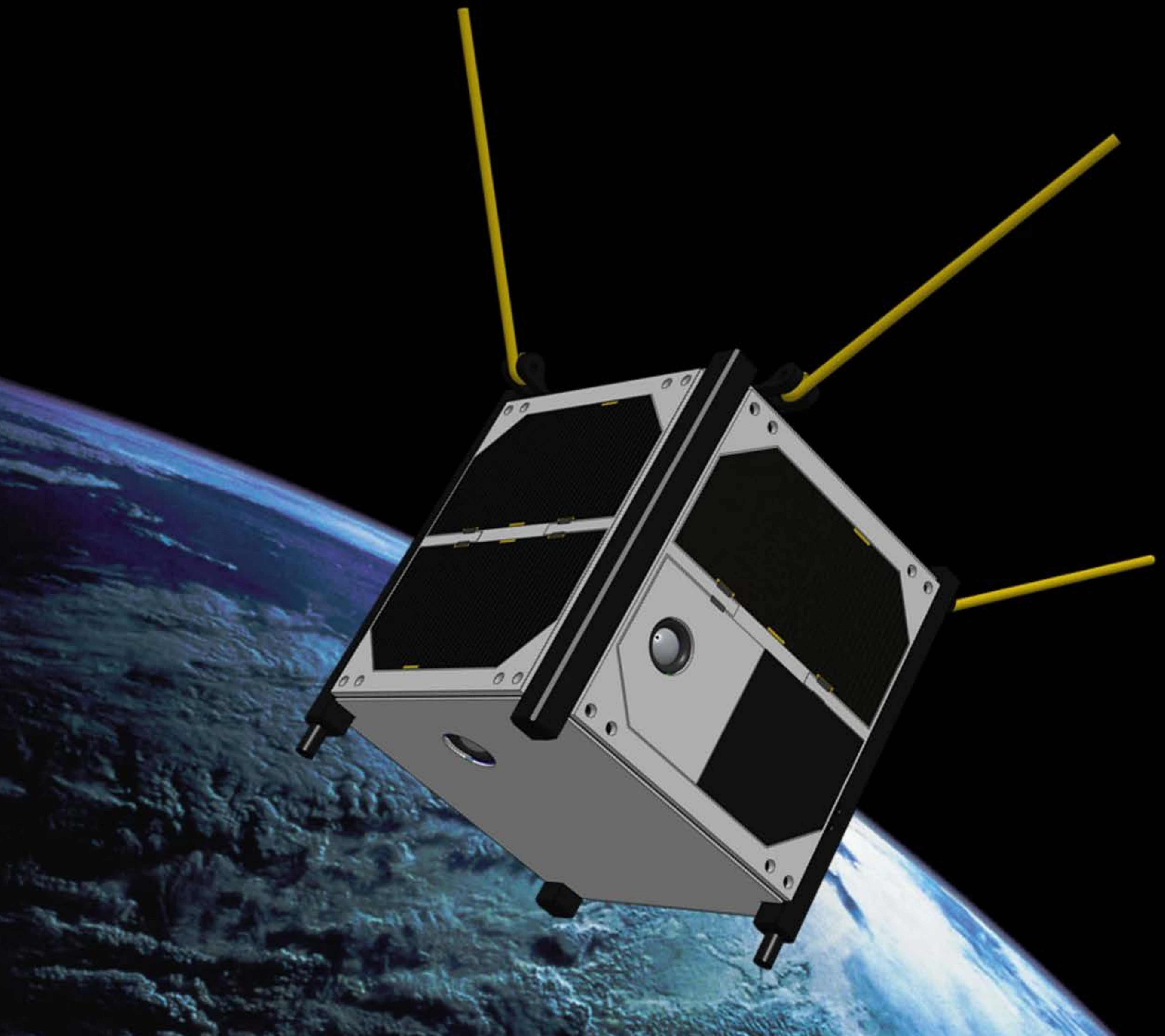
What can I do with it?

- Programatically Control Power Output
- Read Strength of Power Input
- Somewhat Limited Logic Computation*
  (16MHz)

*2.048MHz Computers used for moon landing computers

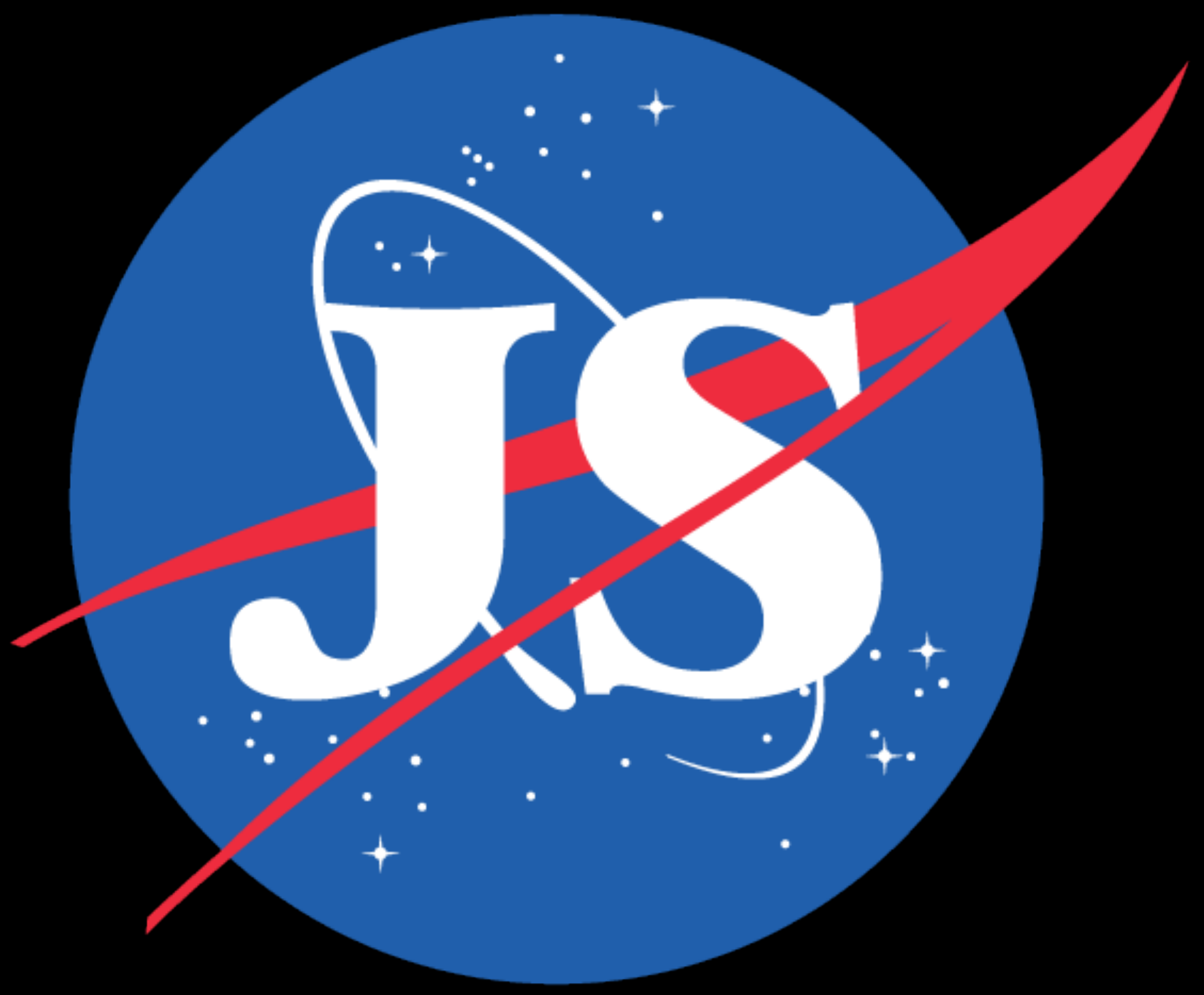
ArduSat

ardusat.com



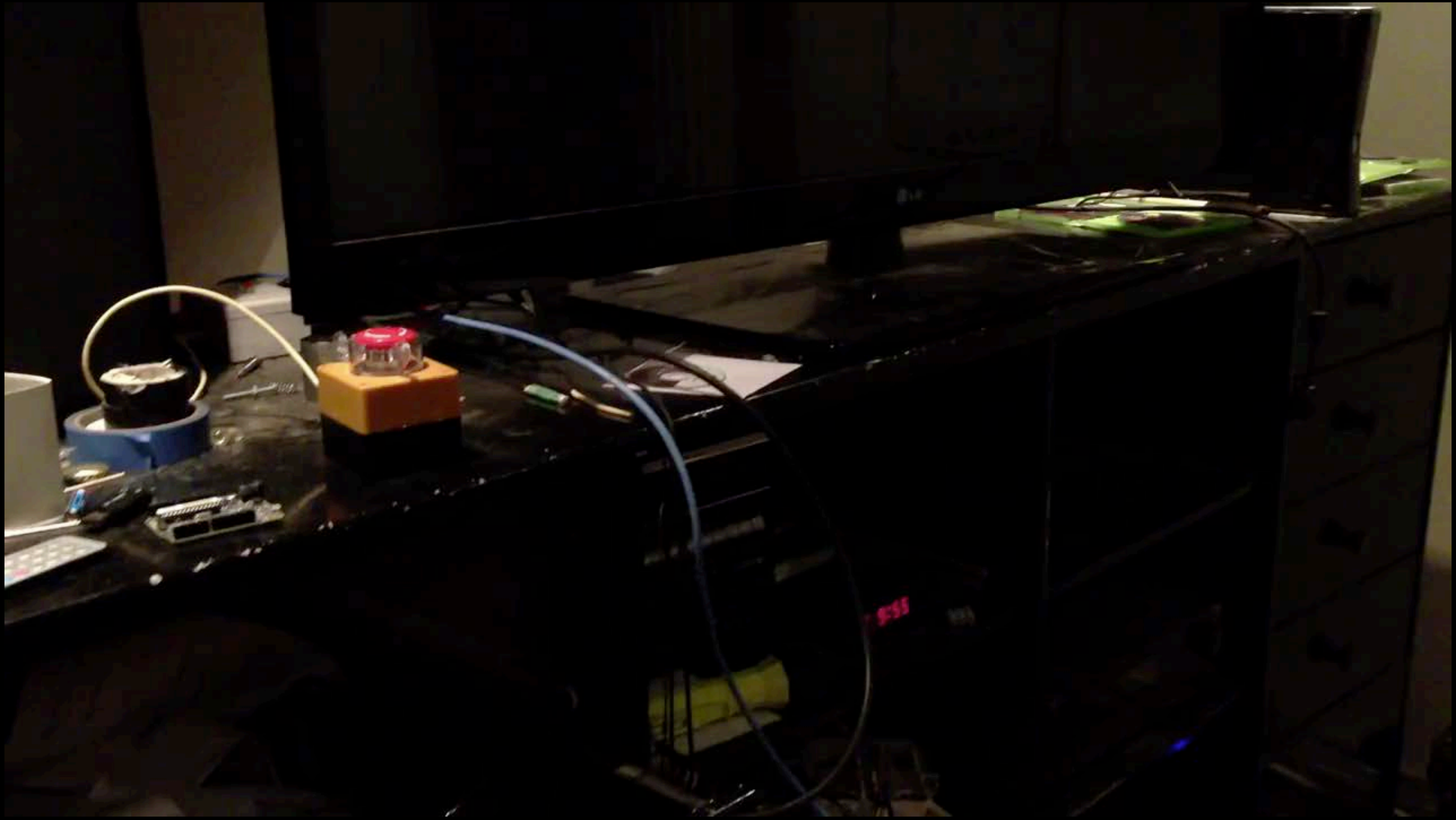
JavaScript in (Near) Space

Pawel Szymczykowski
@makenai



What can I **DO** with it though?

Aside from go to space



Wide Range of Things

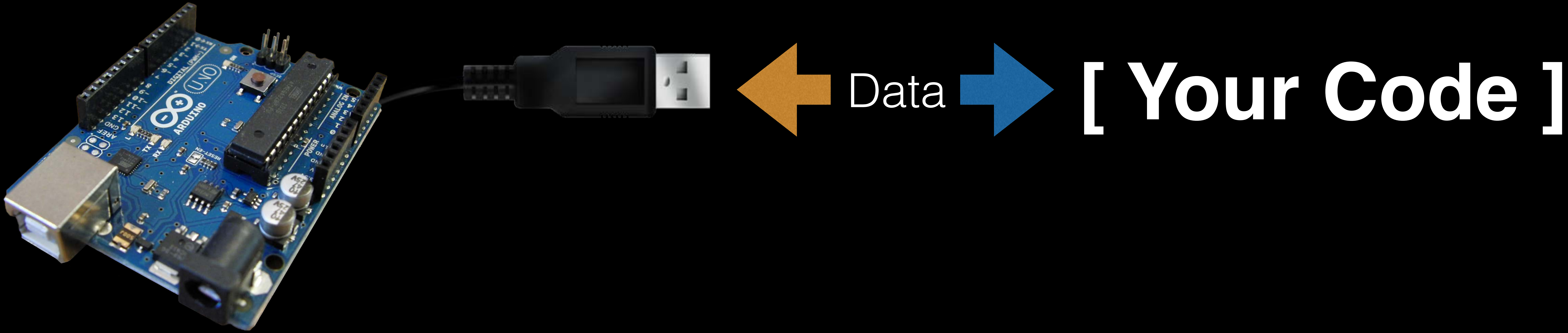
- Home Automation (Lights, etc…)
- Robots
- Safety Equipment
- Pan/Tilt Cameras
- Remote Control Stuff
- Output for Website
- Physically Interactive Website
- Accessibility Enhancement
- Multimedia Controller
- Connected Clothing
- Research & Development Tools
- Physical Notifications & Databoards
- Game Development
- Musical Instrument
- Flaming Stuff*
- Much, much more…

This Seems Complex, What’s ~~Cereal~~ Serial?

- USB (Universal Serial Bus)
- Ethernet
- Fiber Optics
- PCI Express
- Bluetooth

“Serial communication is the process of sending data one bit at a time, sequentially, over a communication channel or computer bus.”

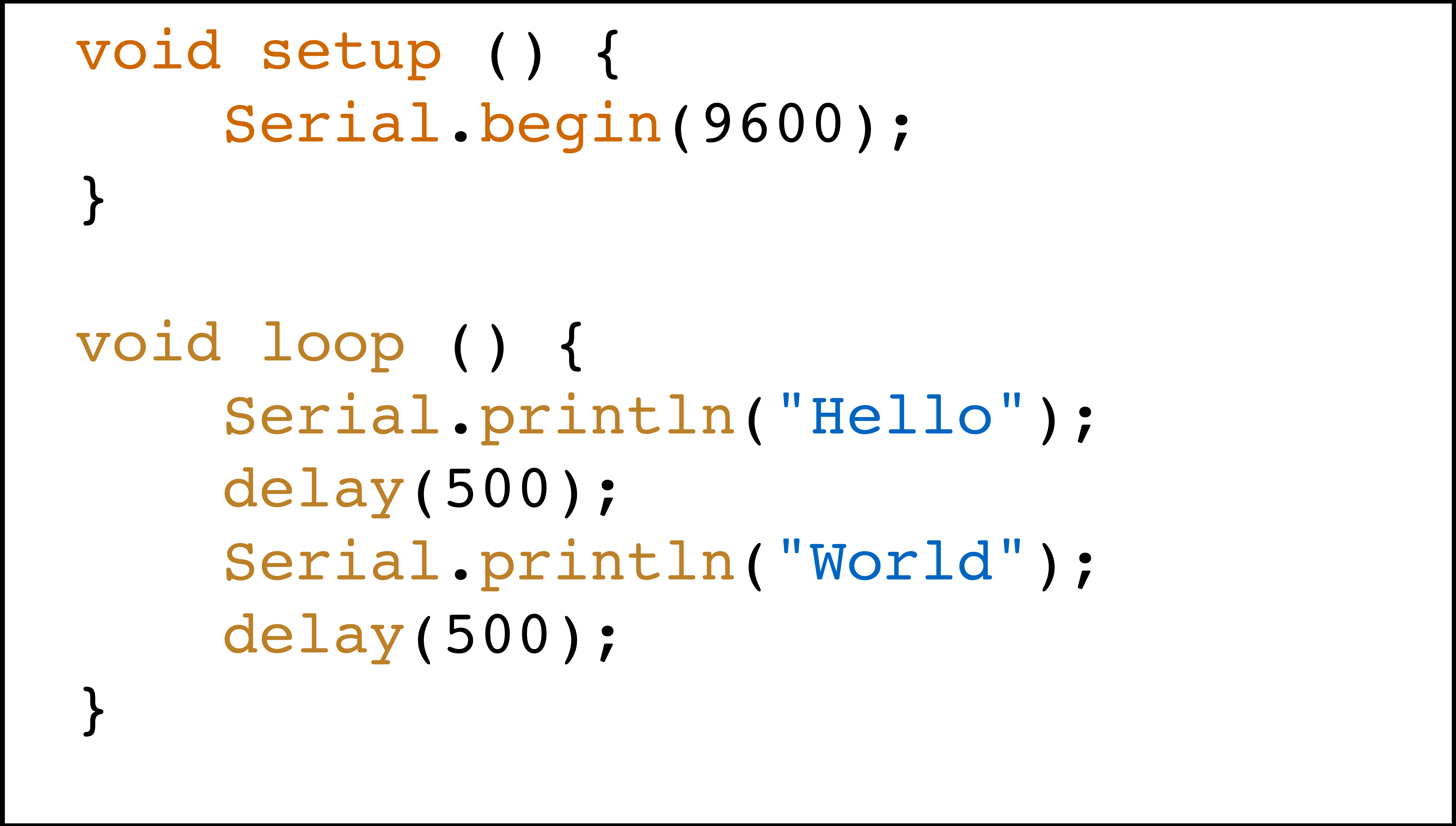
Process



Data

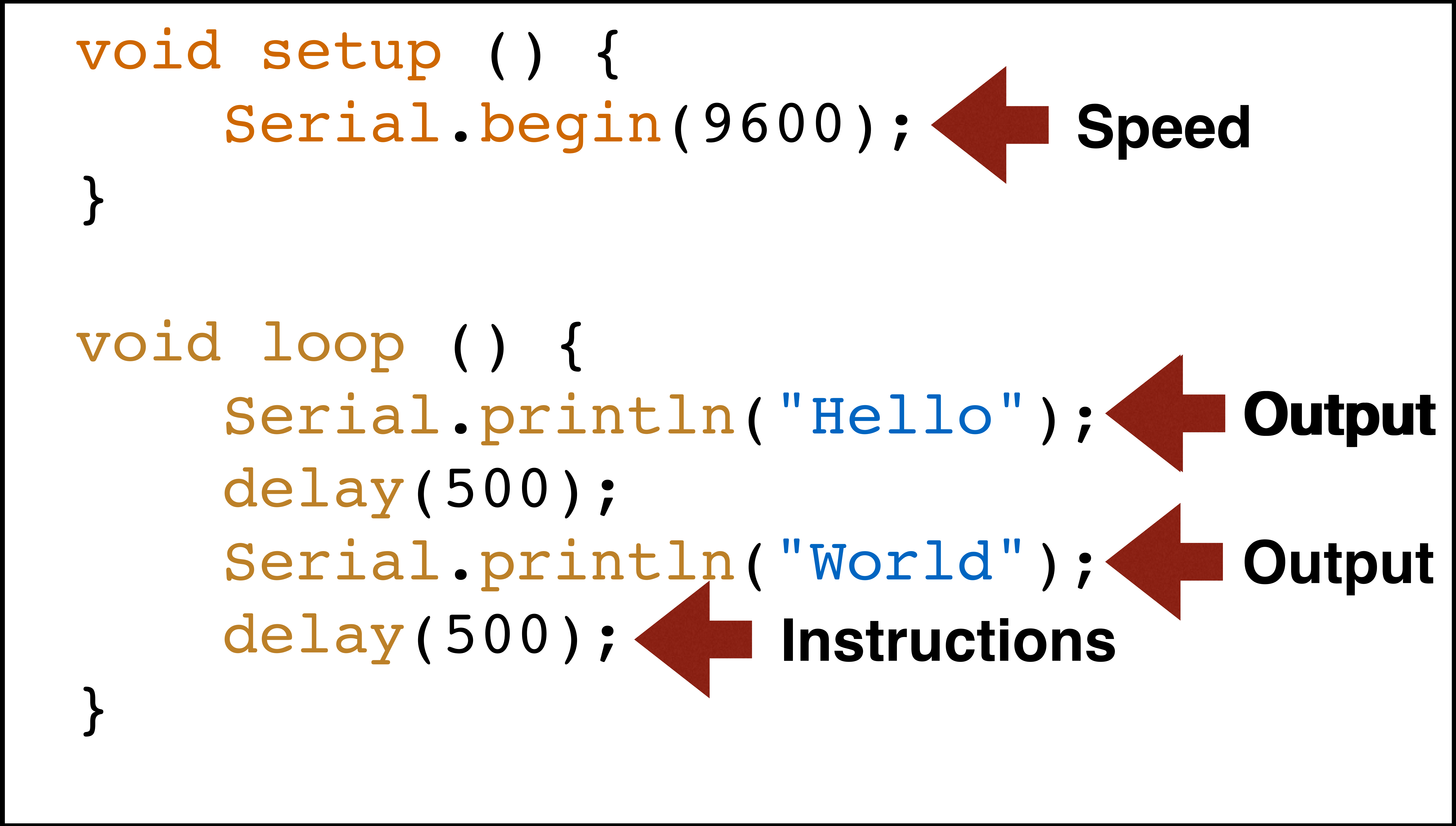
[Your Code]

Simple Usage



```
void setup () {
    Serial.begin(9600);
}

void loop () {
    Serial.println("Hello");
    delay(500);
    Serial.println("World");
    delay(500);
}
```

Simple Output Usage



```
void setup () {
    Serial.begin(9600);
}

void loop () {
    Serial.println("Hello");
    delay(500);
    Serial.println("World");
    delay(500);
}
```

Speed

Output

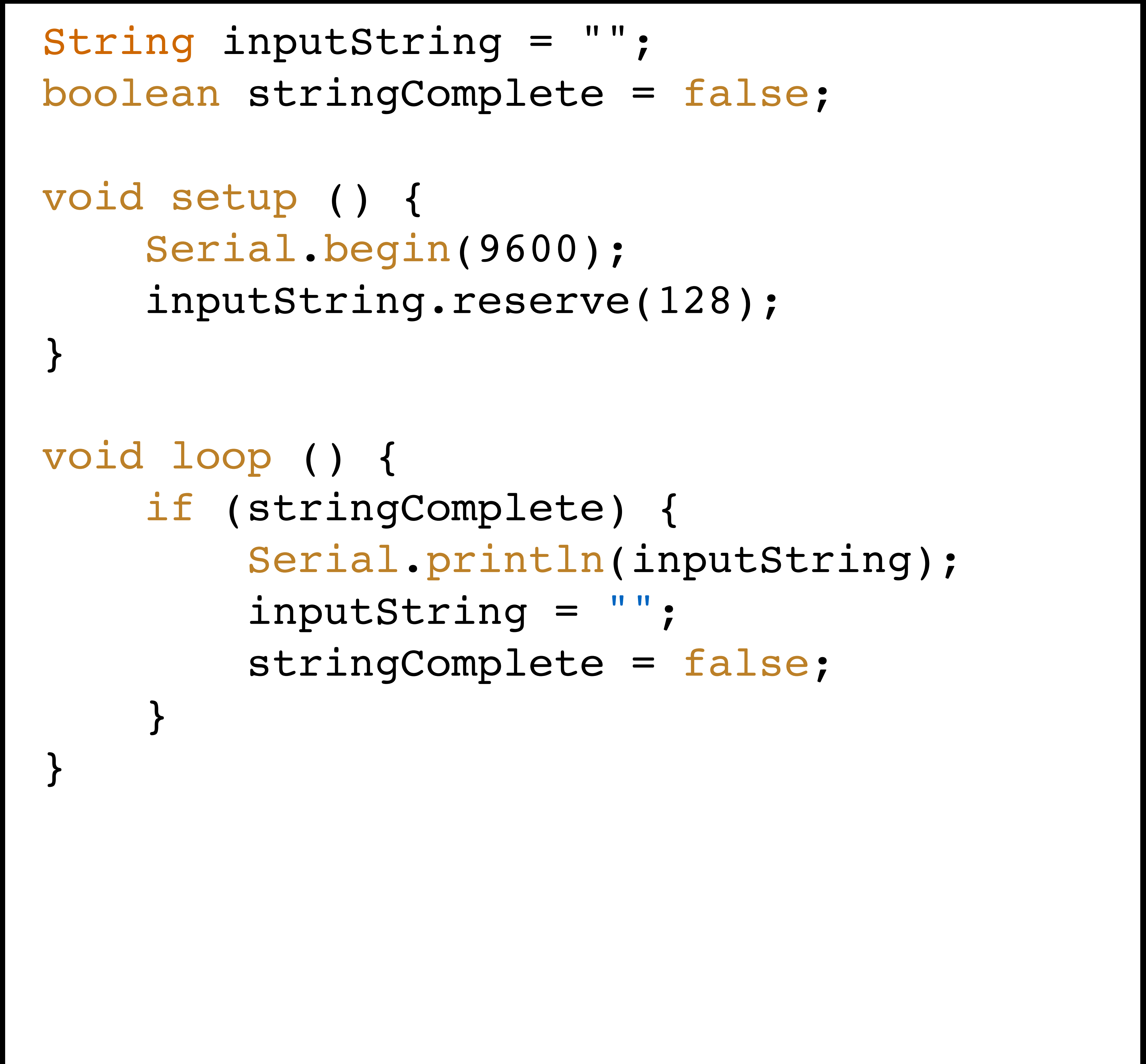
Output

Instructions

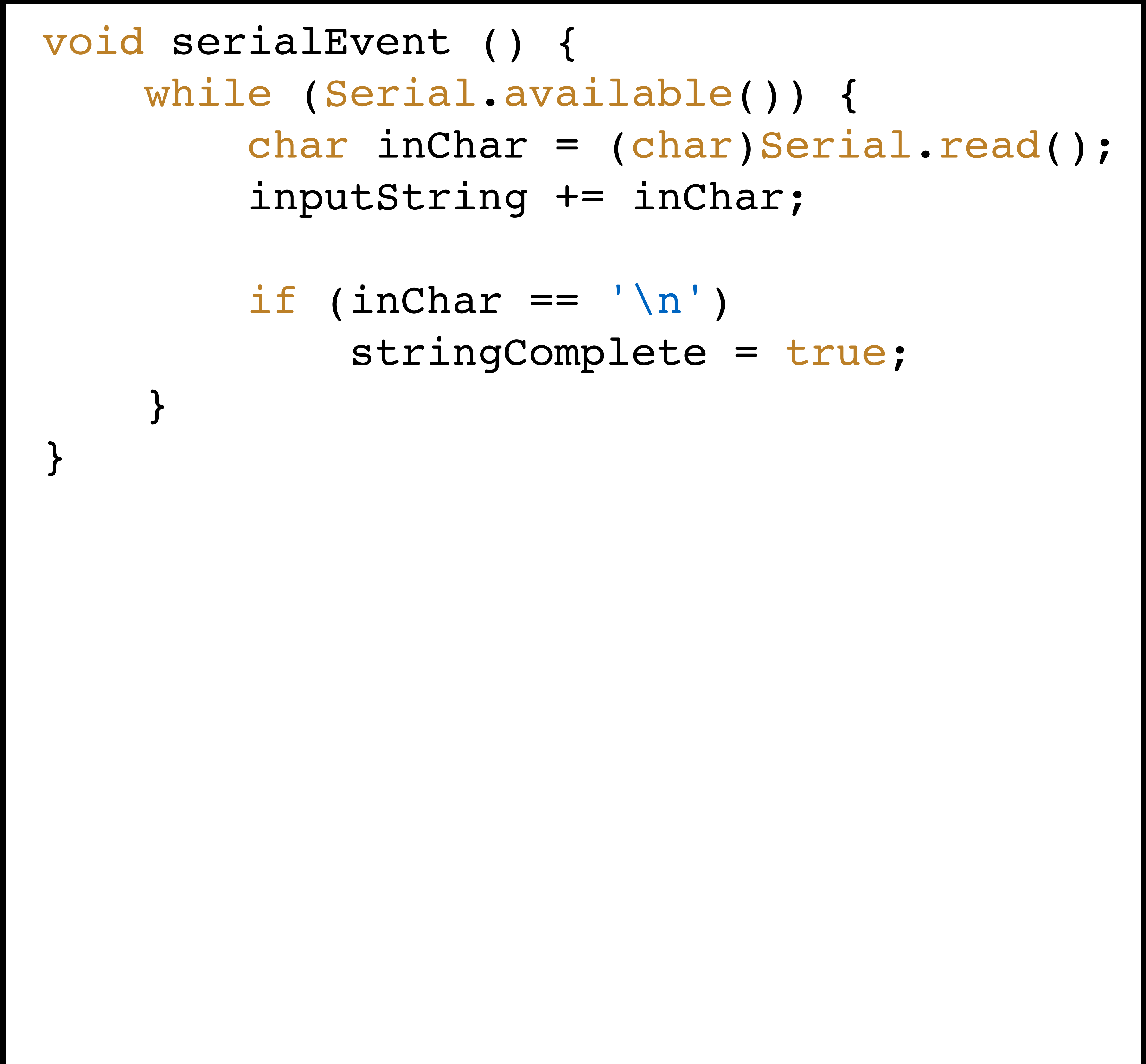
Output? But I Want Control!

- void setup happens when starting
- void loop happens over an over again
- … void serialEvent happens when communication is passed!

How does that work?

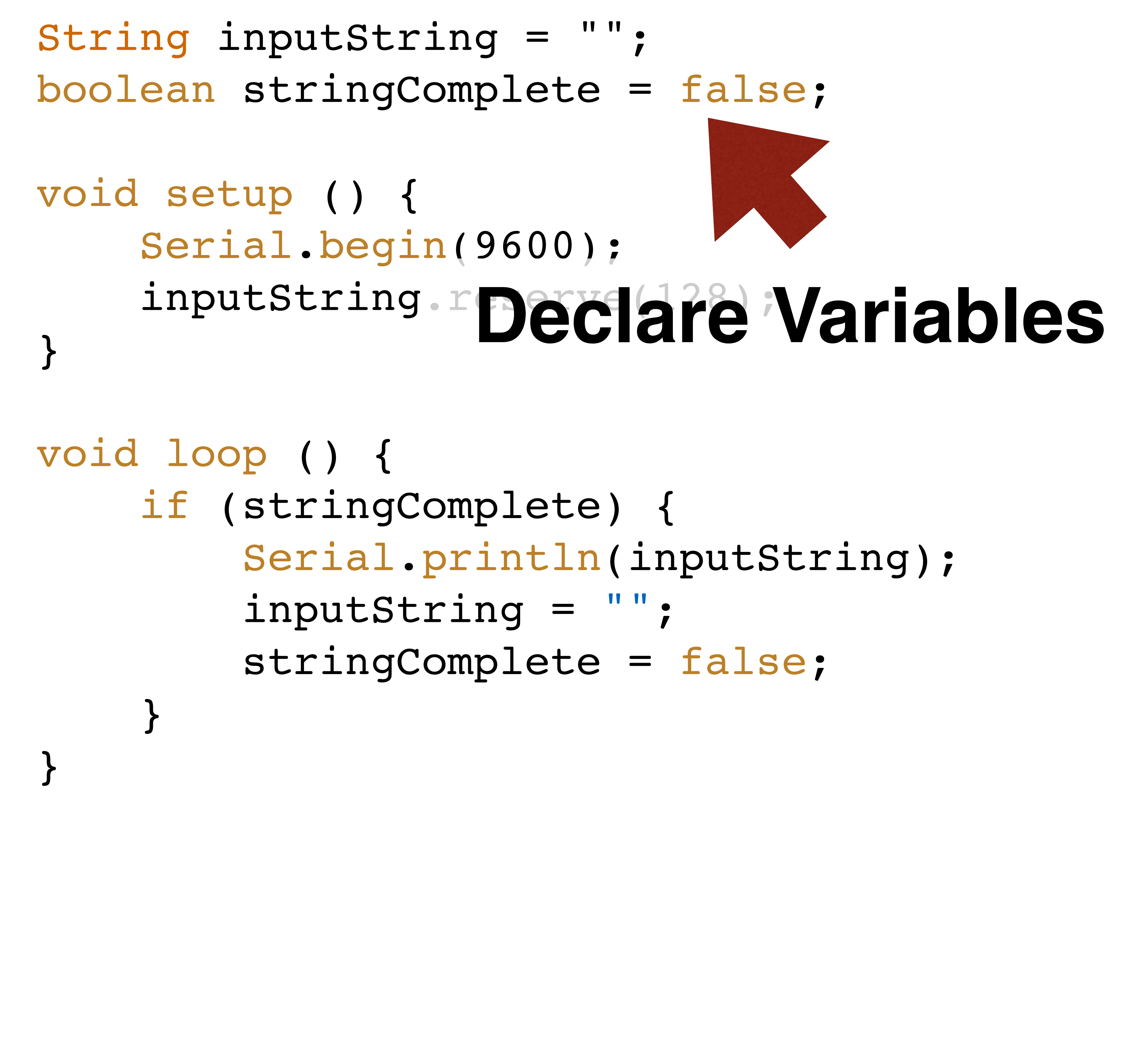


```
String inputString = "";
boolean stringComplete = false;

void setup () {
    Serial.begin(9600);
    inputString.reserve(128);
}

void loop () {
    if (stringComplete) {
        Serial.println(inputString);
        inputString = "";
        stringComplete = false;
    }
}
```



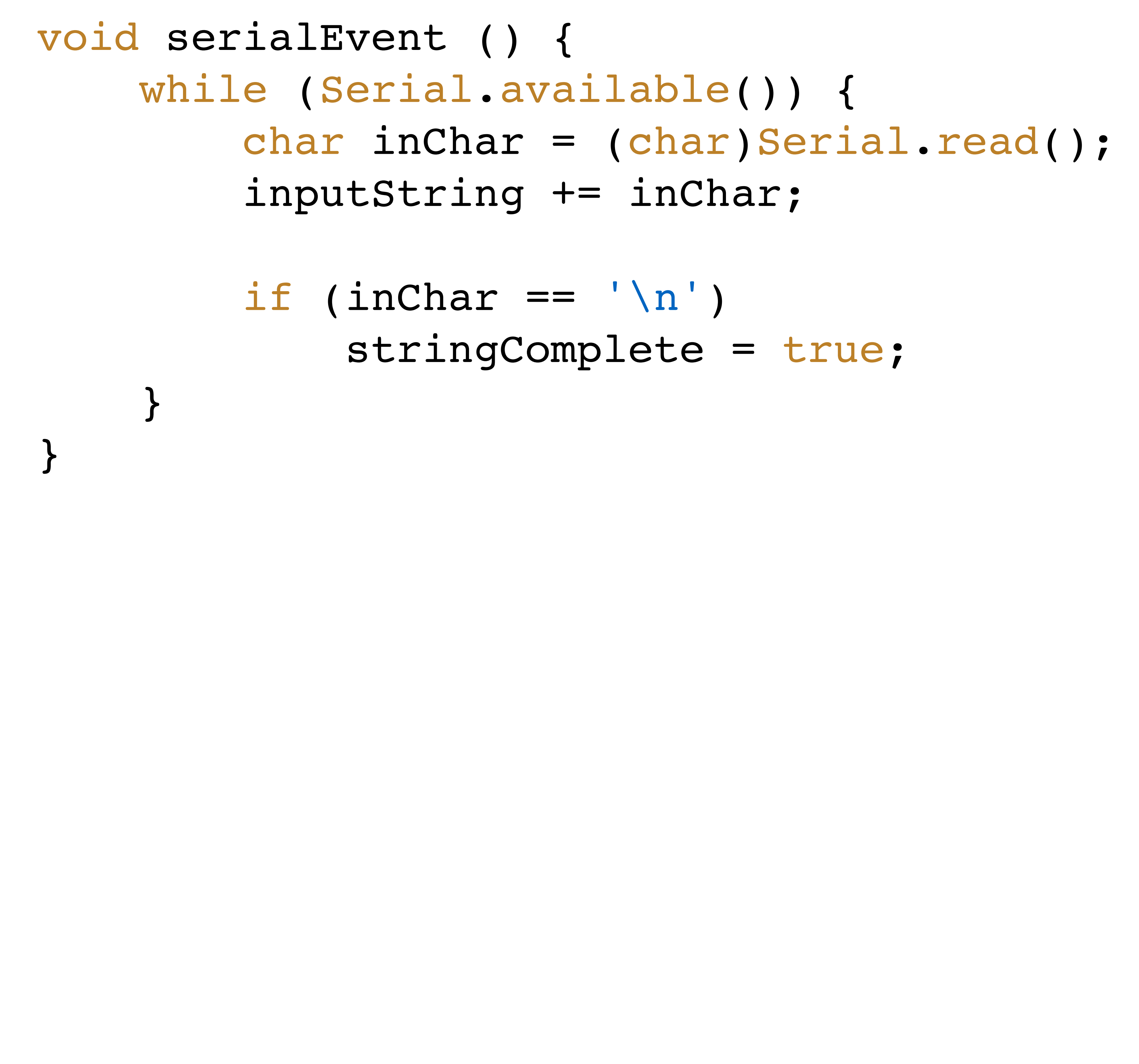
```
void serialEvent () {
    while (Serial.available()) {
        char inChar = (char)Serial.read();
        inputString += inChar;

        if (inChar == '\n')
            stringComplete = true;
    }
}
```

How does that work?



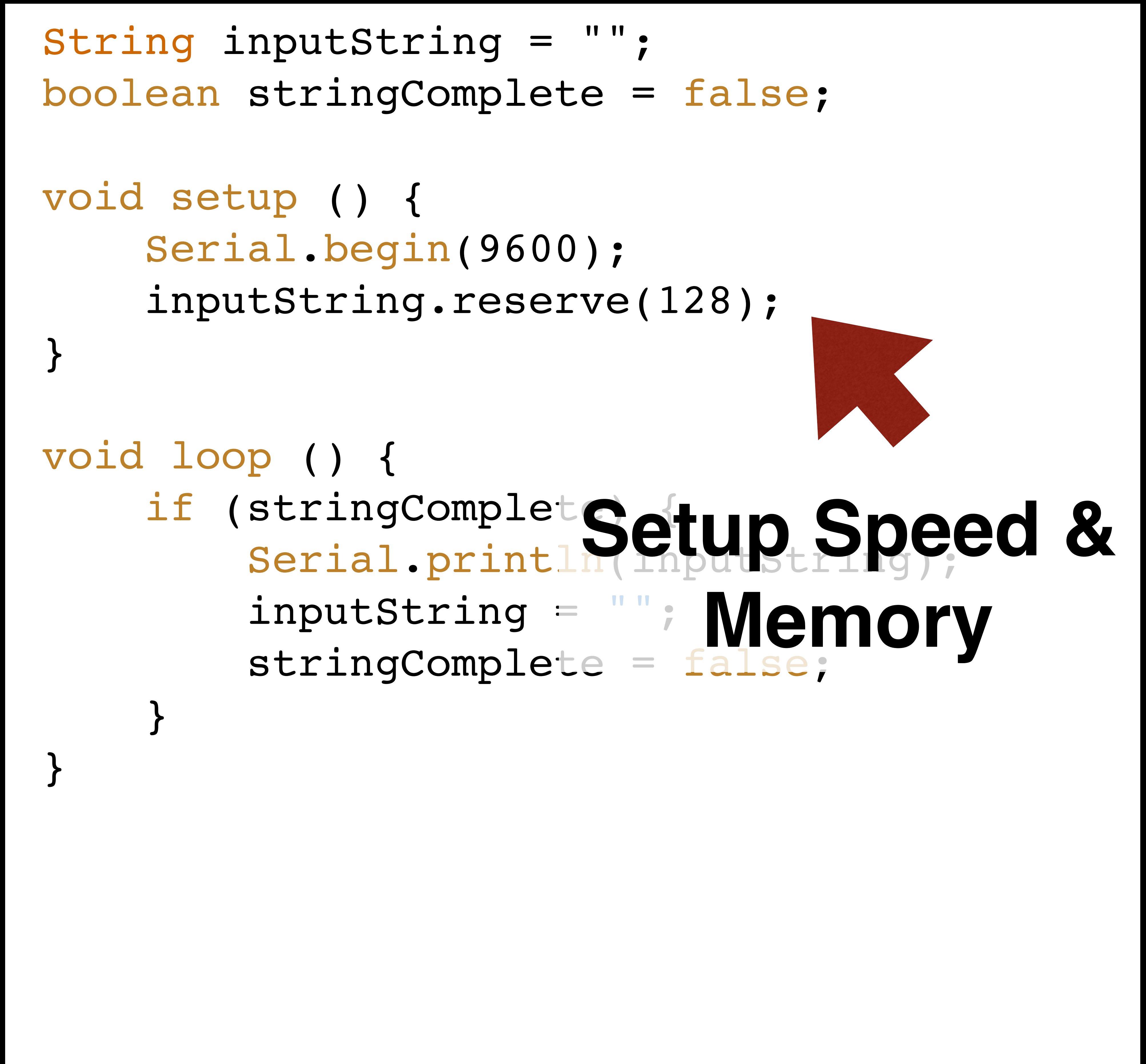
```
String inputString = "";
boolean stringComplete = false;

void setup () {
    Serial.begin(9600);
    inputString.reserve(128);
}

void loop () {
    if (stringComplete) {
        Serial.println(inputString);
        inputString = "";
        stringComplete = false;
    }
}
```

Declare Variables



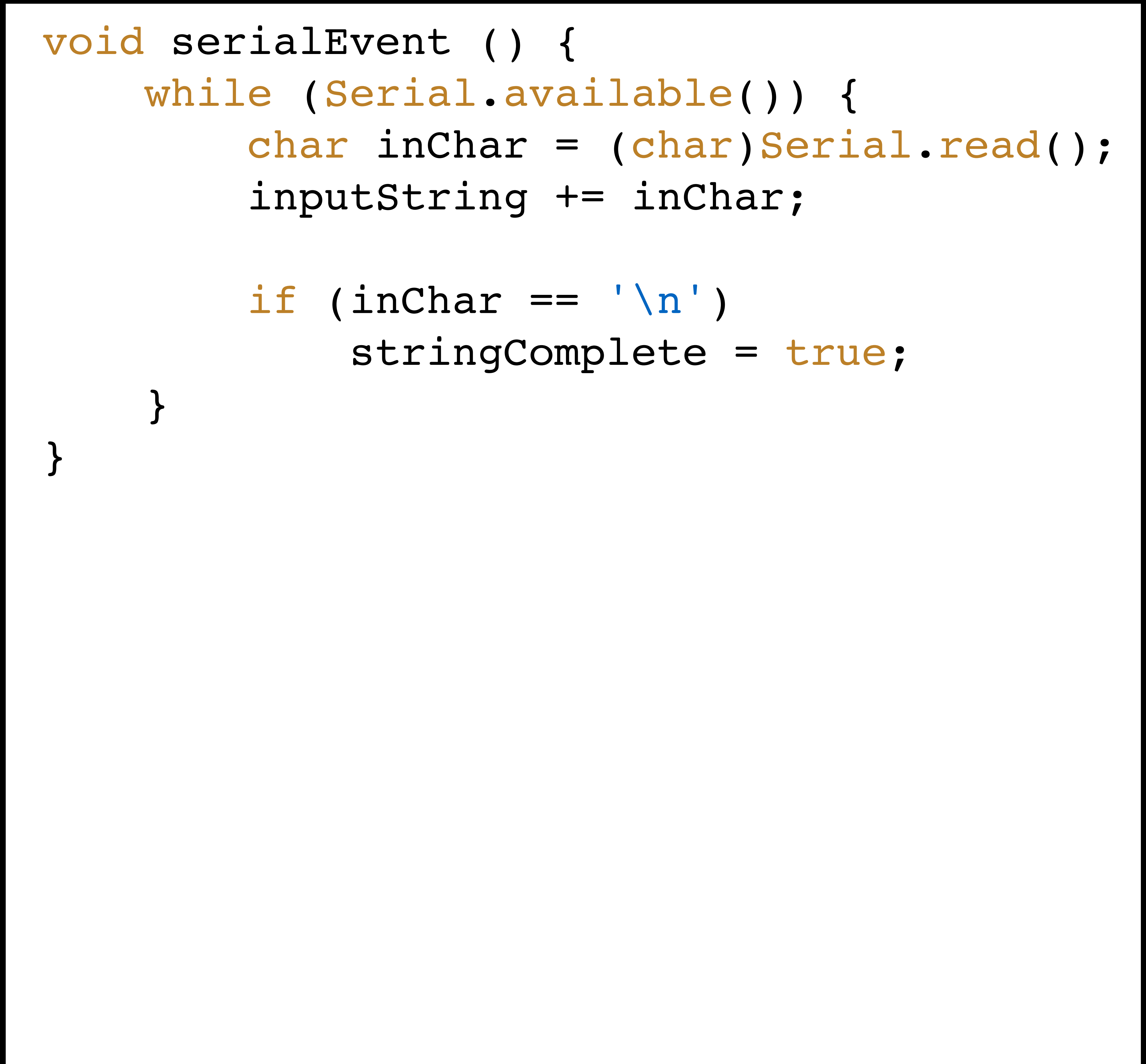
```
void serialEvent () {
    while (Serial.available()) {
        char inChar = (char)Serial.read();
        inputString += inChar;

        if (inChar == '\n')
            stringComplete = true;
    }
}
```

How does that work?



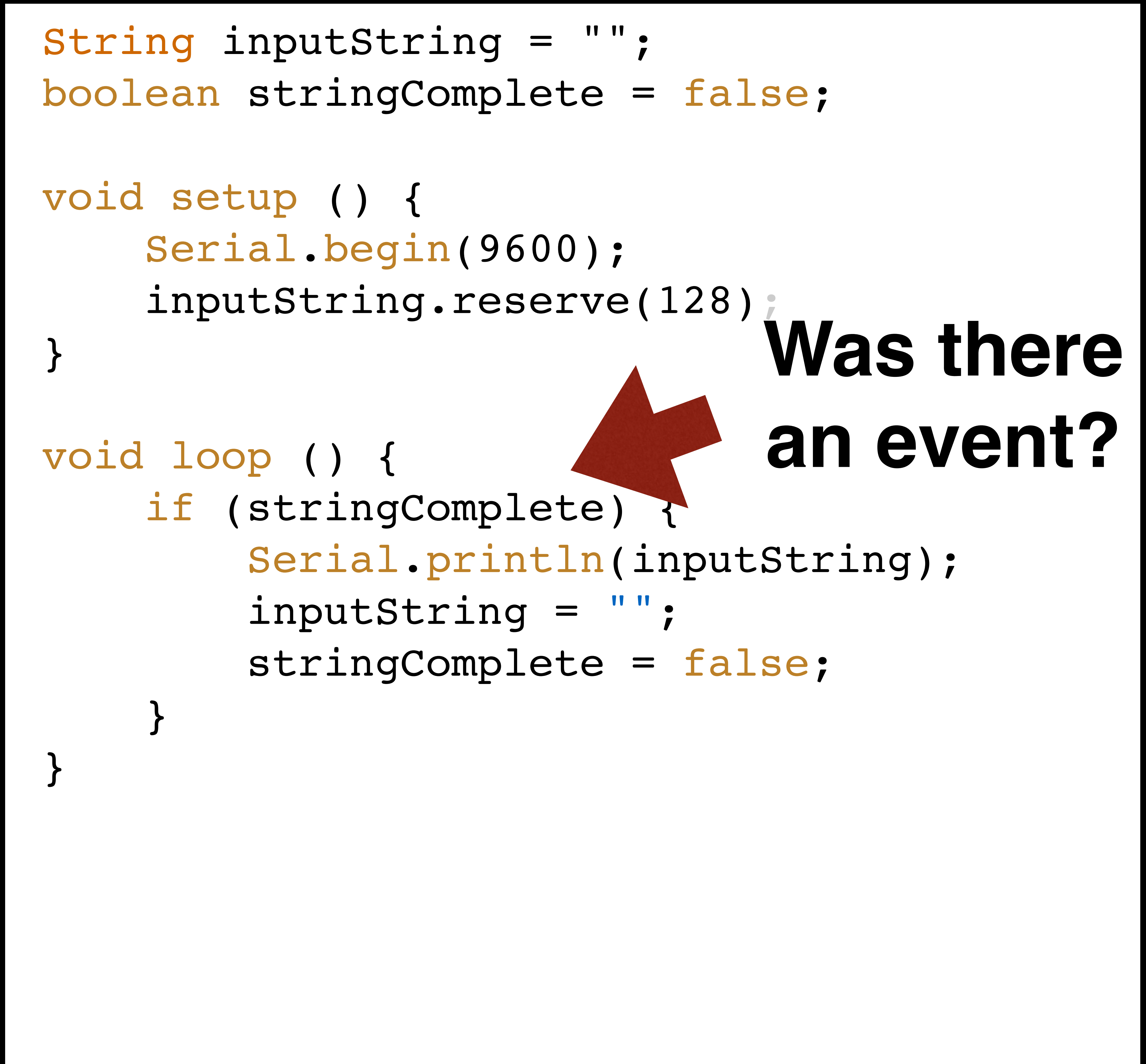
```
String inputString = "";
boolean stringComplete = false;

void setup () {
    Serial.begin(9600);
    inputString.reserve(128);
}

void loop () {
    if (stringComplete) {
        Serial.println(inputString);
        inputString = "";
        stringComplete = false;
    }
}
```

Setup Speed & Memory



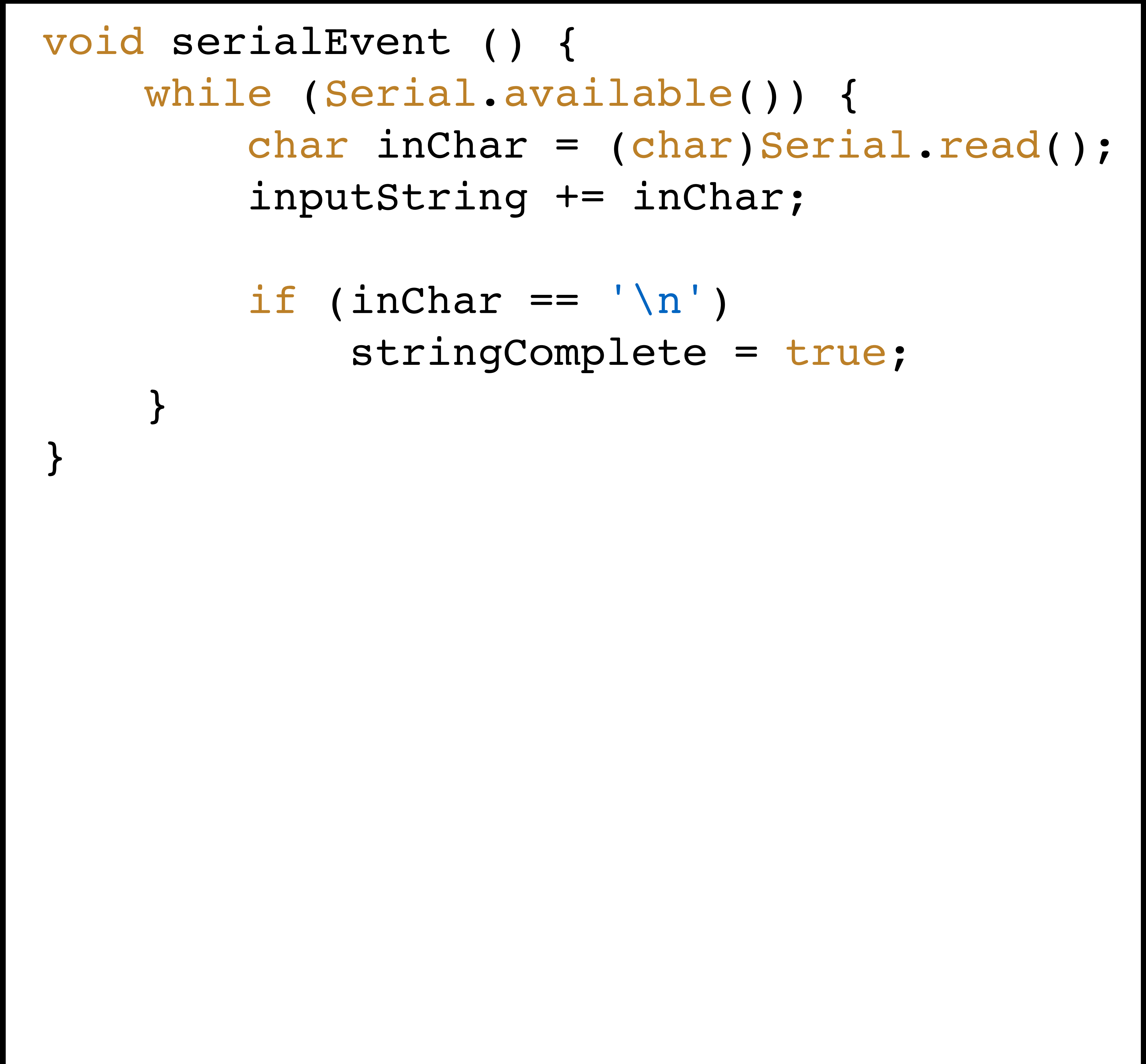
```
void serialEvent () {
    while (Serial.available()) {
        char inChar = (char)Serial.read();
        inputString += inChar;

        if (inChar == '\n')
            stringComplete = true;
    }
}
```

How does that work?



```
String inputString = "";
boolean stringComplete = false;

void setup () {
    Serial.begin(9600);
    inputString.reserve(128);
}

void loop () {
    if (stringComplete) {
        Serial.println(inputString);
        inputString = "";
        stringComplete = false;
    }
}
```

Was there an event?



```
void serialEvent () {
    while (Serial.available()) {
        char inChar = (char)Serial.read();
        inputString += inChar;

        if (inChar == '\n')
            stringComplete = true;
    }
}
```

How does that work?

```
String inputString = "";
boolean stringComplete = false;

void setup () {
    Serial.begin(9600);
    inputString.reserve(128);
}

void loop () {
    if (stringComplete) {
        Serial.println(inputString);
        inputString = "";
        stringComplete = false;
    }
}
```

Perform an action and reset for next

```
void serialEvent () {
    while (Serial.available()) {
        char inChar = (char)Serial.read();
        inputString += inChar;

        if (inChar == '\n')
            stringComplete = true;
    }
}
```

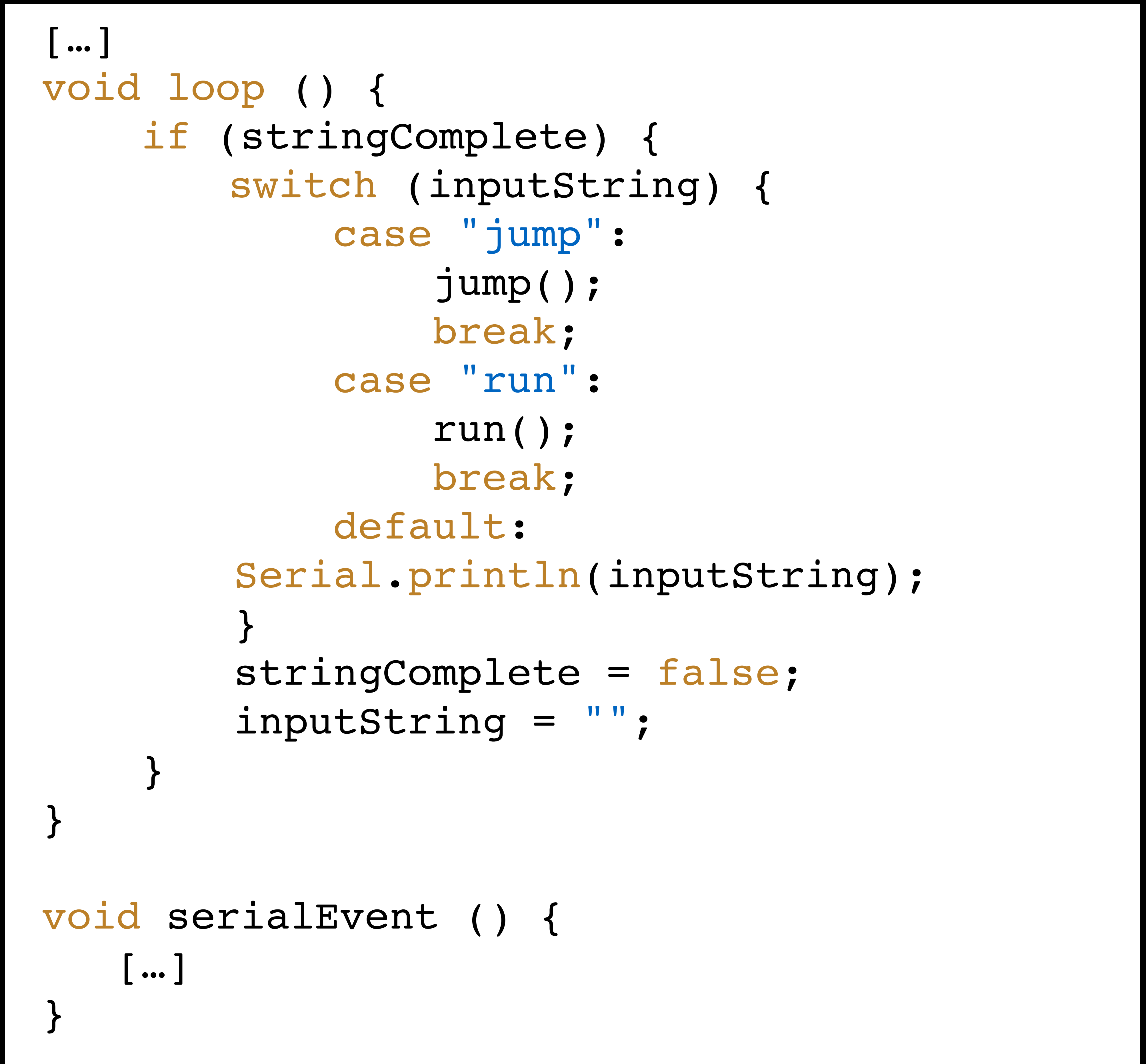
How does that work?

```
String inputString = "";
boolean stringComplete = false;

void setup () {
    Serial.begin(9600);
    inputString.reserve(128);
}

void loop () {
    if (stringComplete) {
        Serial.println(inputString);
        inputString = "";
        stringComplete = false;
    }
}
```

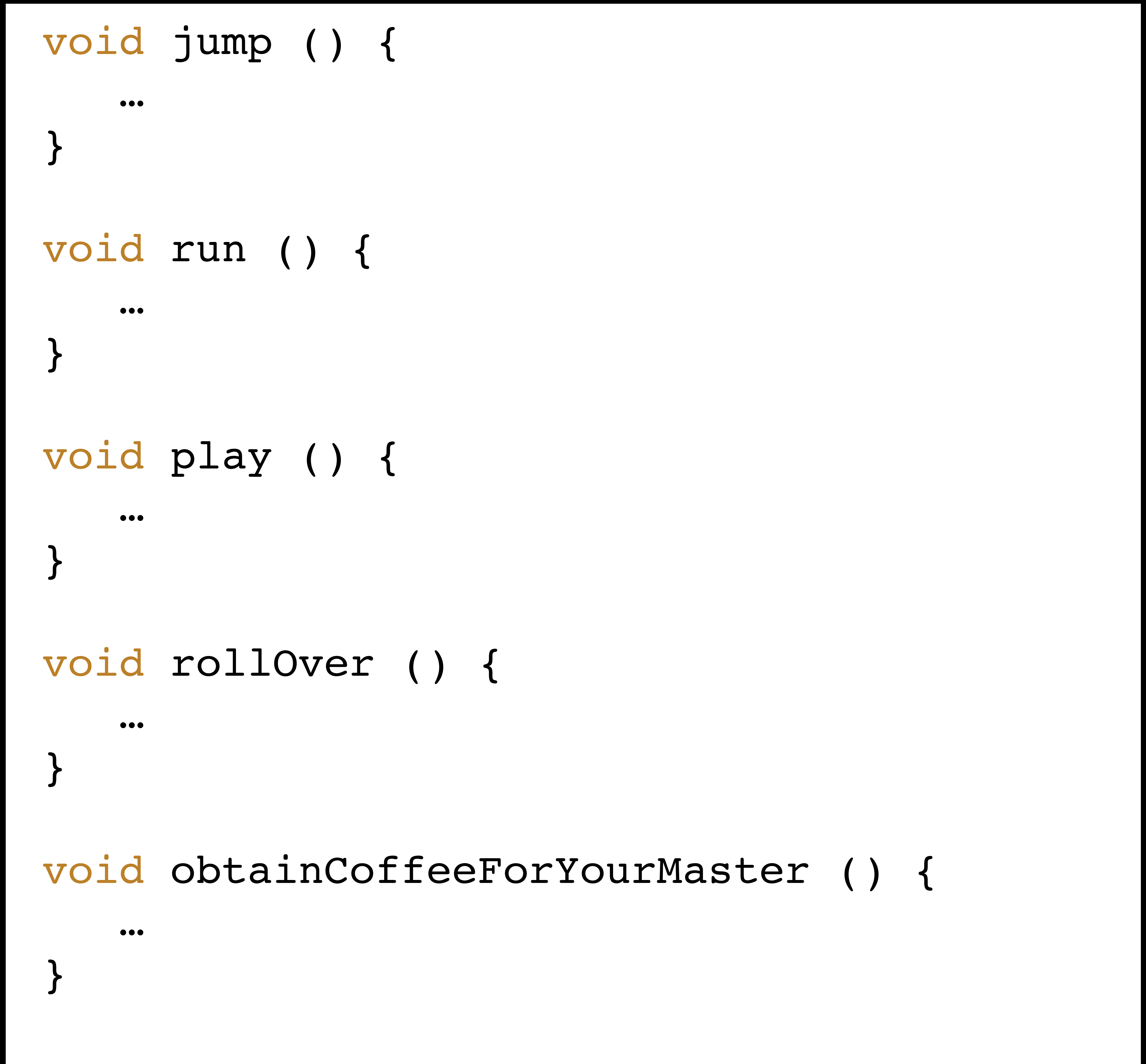
```
void serialEvent () {
    while (Serial.available()) {
        char inChar = (char)Serial.read();
        inputString += inChar;

        if (inChar == '\n')
            stringComplete = true;
    }
}
```

On input, append the string to everything so far

What kind of actions?



```
[…]
void loop () {
    if (stringComplete) {
        switch (inputString) {
            case "jump":
                jump();
                break;
            case "run":
                run();
                break;
            default:
        Serial.println(inputString);
        }
        stringComplete = false;
        inputString = "";
    }
}

void serialEvent () {
   […]
}
```



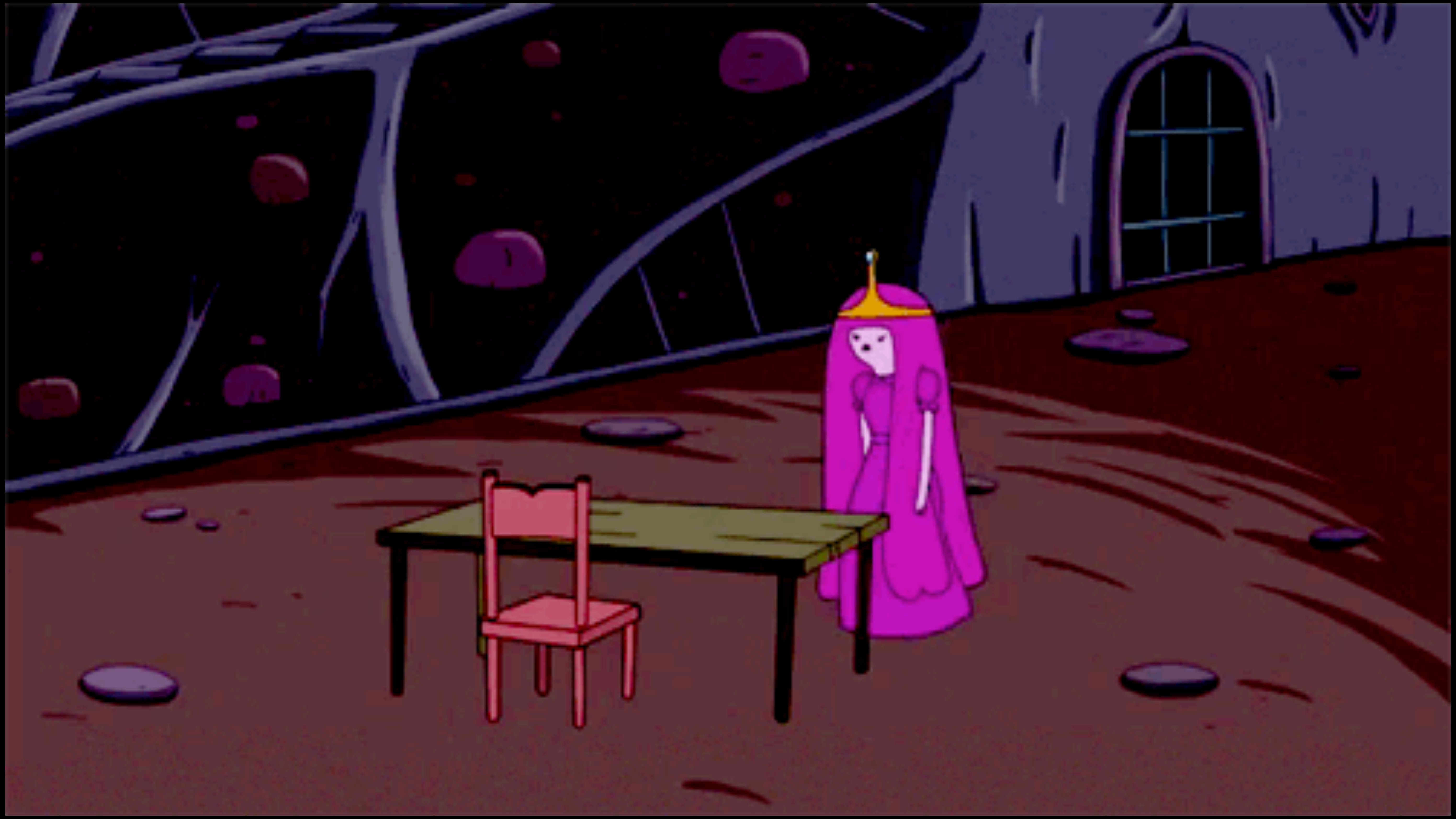
```
void jump () {
   …
}

void run () {
   …
}

void play () {
   …
}

void rollOver () {
   …
}

void obtainCoffeeForYourMaster () {
   …
}
```

Code Walkthrough

It’ll be quick

Outputting a Numeric Value
(with pretty lights)

HEY! I’m Here for JavaScript, not C/C++!



We’re Getting there…

Interacting with the data demo

- Quick ask of the audience
- Terminal used to see data
- Quartz Composer to interact

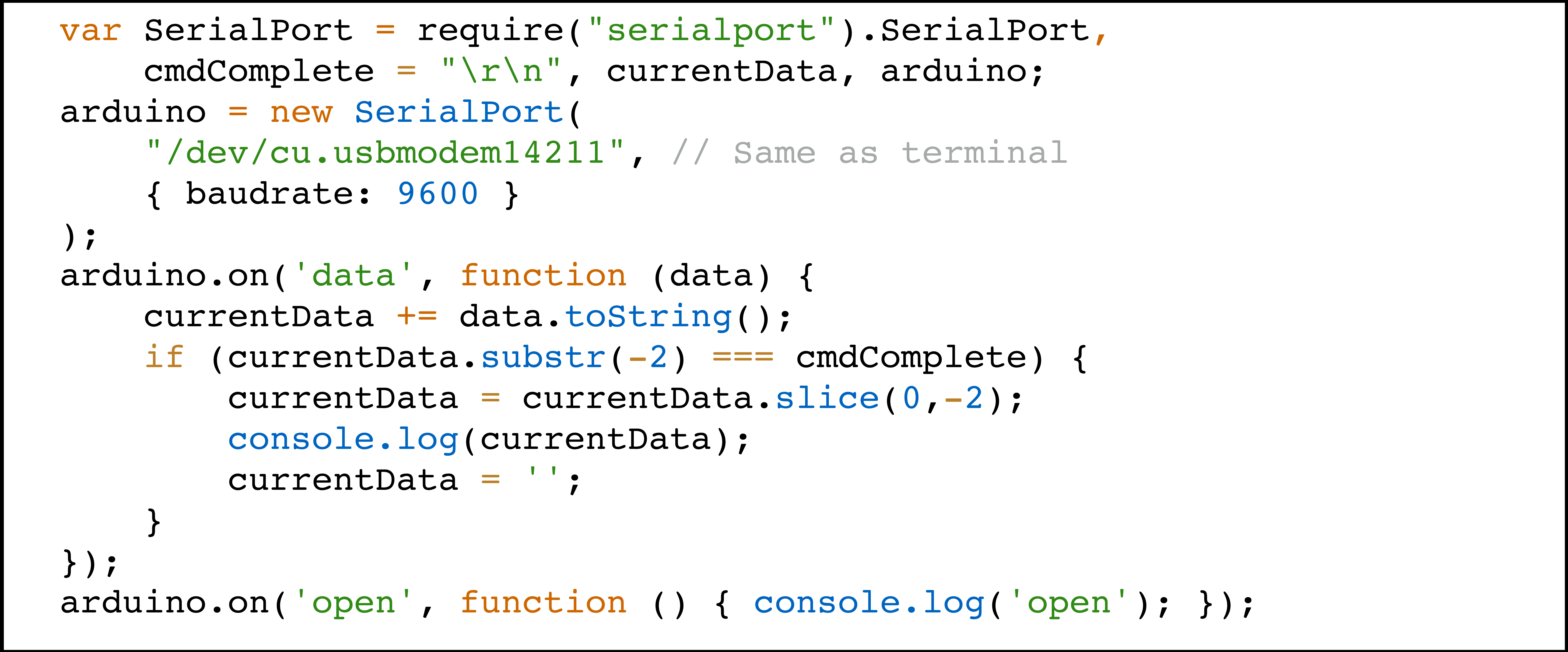
Firmata

Control without code… at least not from you.

Breath It In, We're Starting

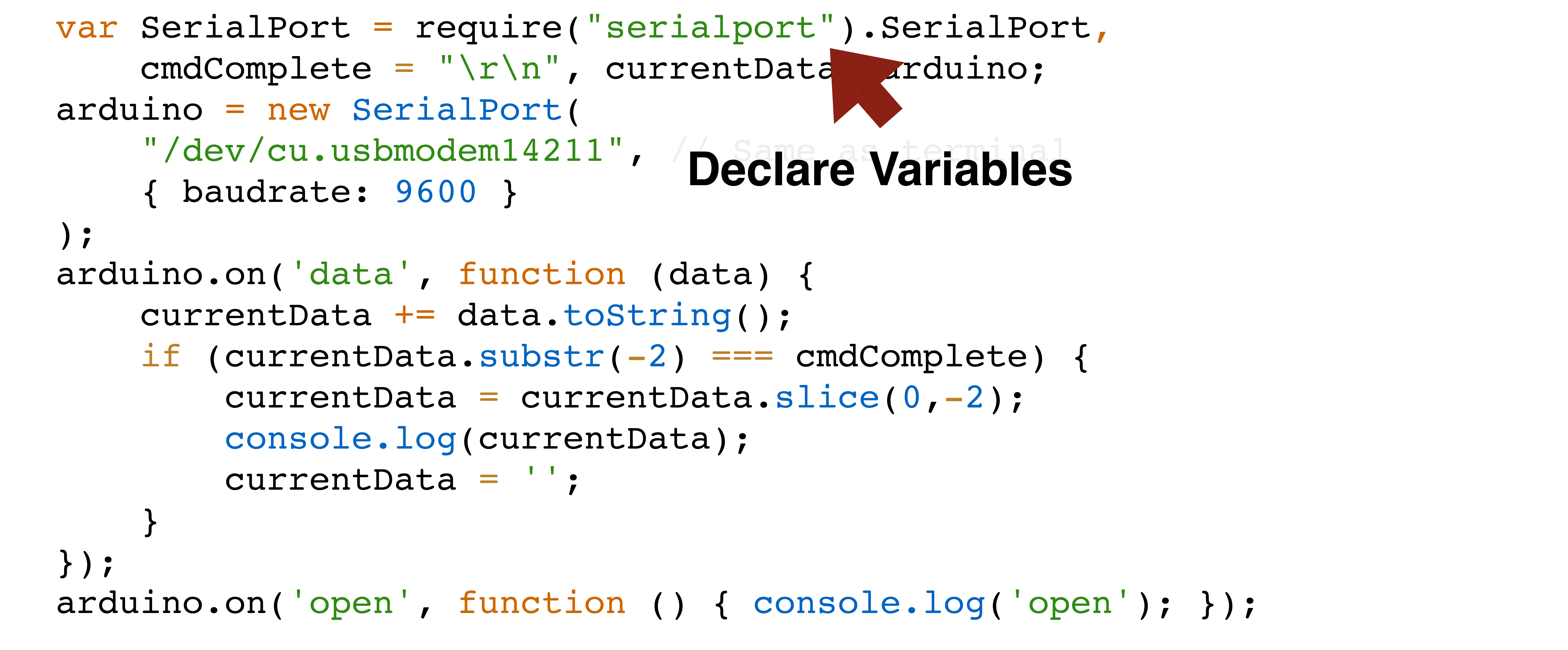
- GitHub @voodootikigod
- /voodootikigod/node-serialport
- npm install serialport

Again…Simple Usage



```
var SerialPort = require("serialport").SerialPort,
    cmdComplete = "\r\n", currentData, arduino;
arduino = new SerialPort(
    "/dev/cu.usbmodem14211", // Same as terminal
    { baudrate: 9600 }
);
arduino.on('data', function (data) {
    currentData += data.toString();
    if (currentData.substr(-2) === cmdComplete) {
        currentData = currentData.slice(0,-2);
        console.log(currentData);
        currentData = '';
    }
});
arduino.on('open', function () { console.log('open'); });
```

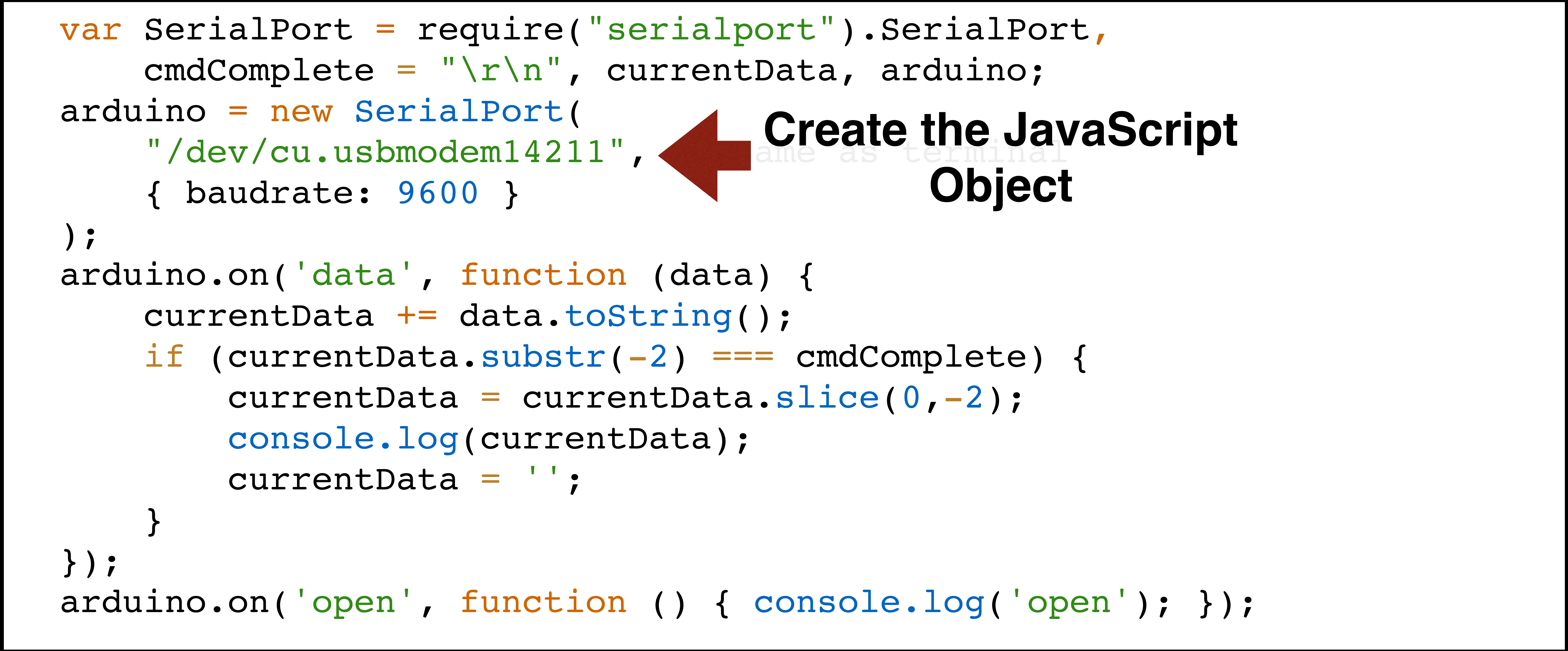
Again…Simple Usage



```
var SerialPort = require("serialport").SerialPort,
    cmdComplete = "\r\n", currentData, arduino;
arduino = new SerialPort(
    "/dev/cu.usbmodem14211", // Same as terminal
    { baudrate: 9600 }
);
arduino.on('data', function (data) {
    currentData += data.toString();
    if (currentData.substr(-2) === cmdComplete) {
        currentData = currentData.slice(0,-2);
        console.log(currentData);
        currentData = '';
    }
});
arduino.on('open', function () { console.log('open'); });
```

Declare Variables

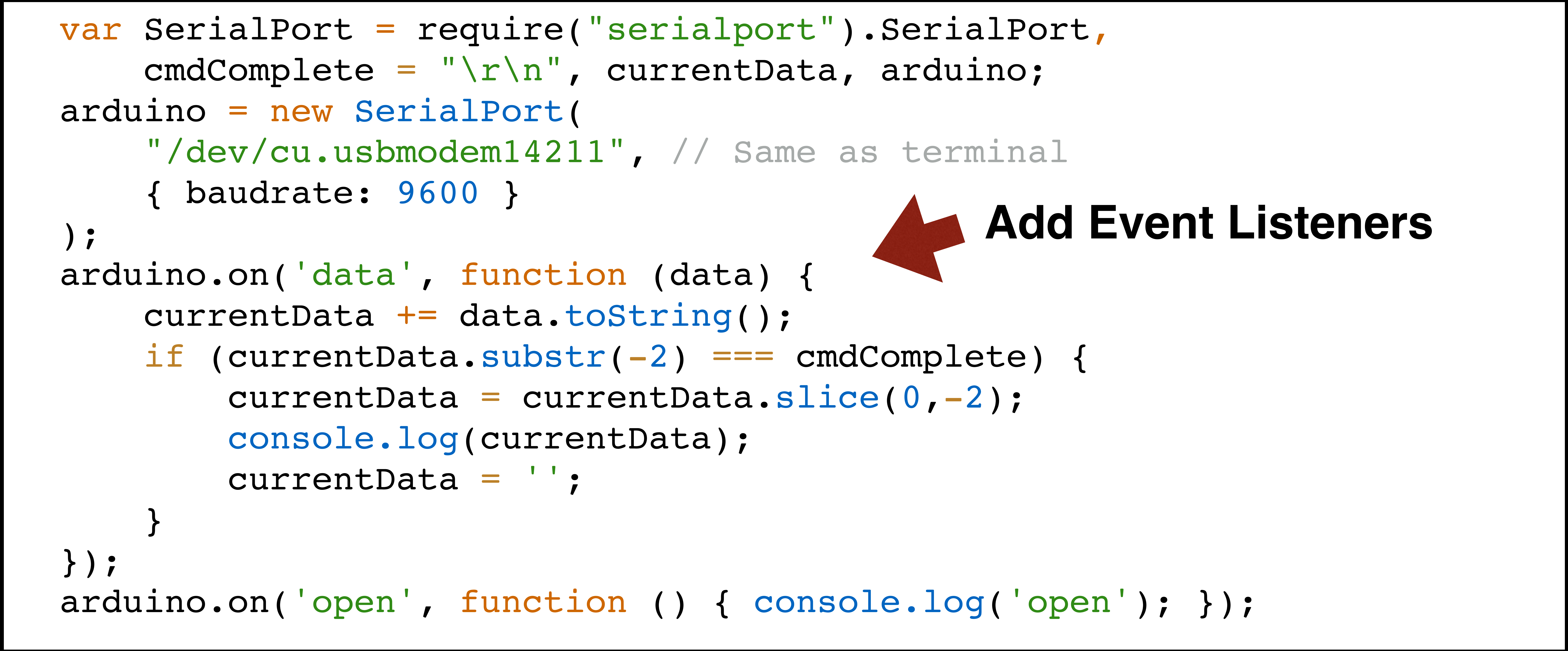
Again…Simple Usage



```
var SerialPort = require("serialport").SerialPort,
    cmdComplete = "\r\n", currentData, arduino;
arduino = new SerialPort(
    "/dev/cu.usbmodem14211",     ame as terminal
    { baudrate: 9600 }
);
arduino.on('data', function (data) {
    currentData += data.toString();
    if (currentData.substr(-2) === cmdComplete) {
        currentData = currentData.slice(0,-2);
        console.log(currentData);
        currentData = '';
    }
});
arduino.on('open', function () { console.log('open'); });
```

Create the JavaScript Object

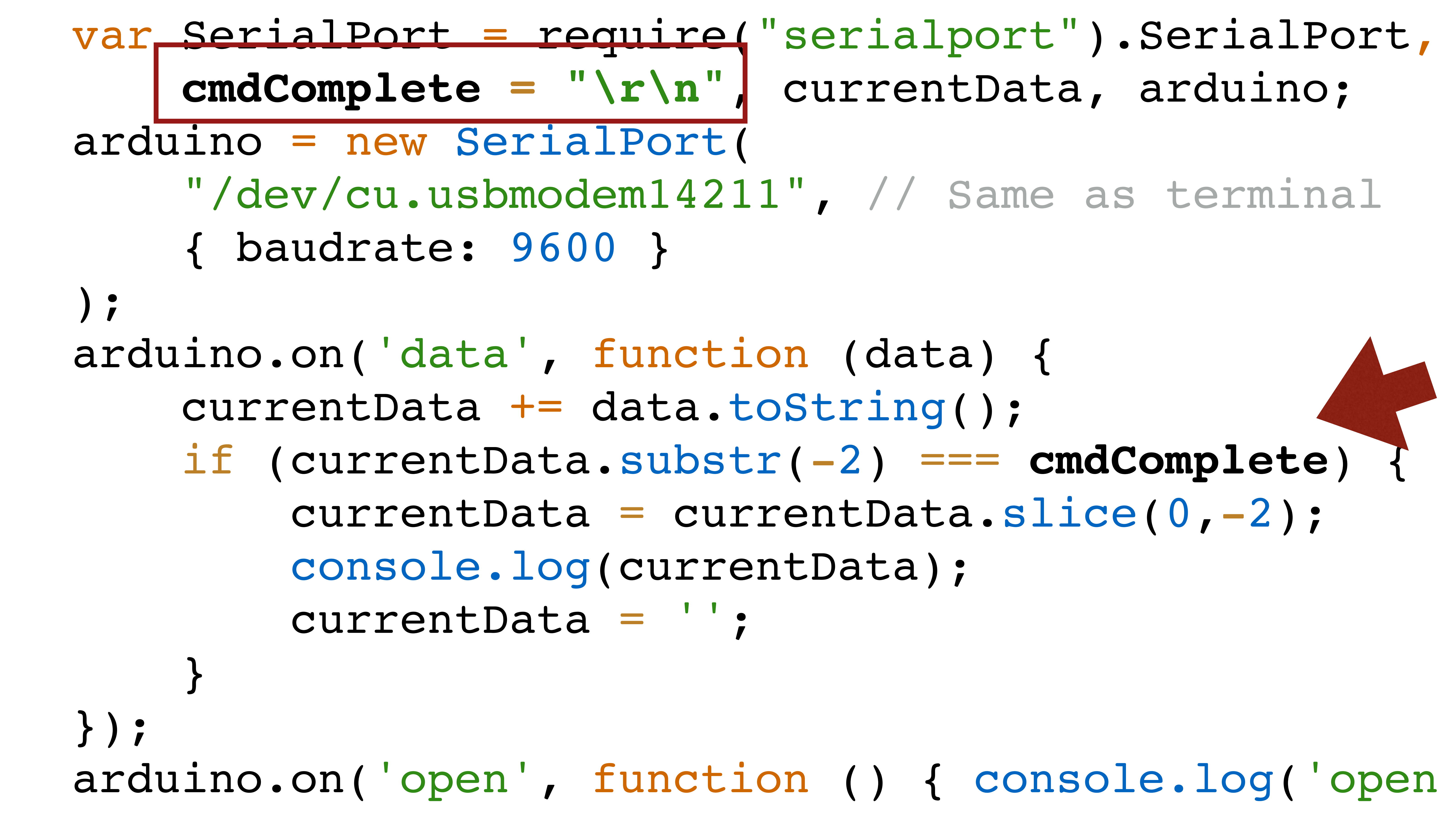
Again…Simple Usage



```
var SerialPort = require("serialport").SerialPort,
    cmdComplete = "\r\n", currentData, arduino;
arduino = new SerialPort(
    "/dev/cu.usbmodem14211", // Same as terminal
    { baudrate: 9600 }
);
arduino.on('data', function (data) {
    currentData += data.toString();
    if (currentData.substr(-2) === cmdComplete) {
        currentData = currentData.slice(0,-2);
        console.log(currentData);
        currentData = '';
    }
});
arduino.on('open', function () { console.log('open'); });
```

Add Event Listeners

Again…Simple Usage



```
var SerialPort = require("serialport").SerialPort,
    cmdComplete = "\r\n", currentData, arduino;
arduino = new SerialPort(
    "/dev/cu.usbmodem14211", // Same as terminal
    { baudrate: 9600 }
);
arduino.on('data', function (data) {
    currentData += data.toString();
    if (currentData.substr(-2) === cmdComplete) {
        currentData = currentData.slice(0,-2);
        console.log(currentData);
        currentData = '';
    }
});
arduino.on('open', function () { console.log('open'); });
```

Note that we're looking again for a new line to know when complete

NodeJS Demo

Simple Output Demo

So Event Based? That’s simple!

SerialPort(path, options [, *openImmediately* [, *callback*]])

.**open**(callback)

Opens the connection if *openImmediately* is set to false.

.**isOpen**() returns boolean

Ability to check whether a connection is open.

.**write**(buffer/string, callback)

Send data back to the device. *See? Told you it would be handy.*

.**on**() events: open, data, close, error

Usual events for NodeJS, expect to use “data” the most.

NodeJS Demo 2

Re-Doing Quartz Demo

Localhost FTW!

localhost…? Really?

- Localhost can be powerful!
- Allows for further control and logic
- Server side is socket.io-client
- Works with Raspberry Pi!

Socket.io-Client

```
…
arduino.on('data', function (data) {
    currentData += data.toString();
    if (currentData.substr(-2) === cmdComplete) {
        currentData = currentData.slice(0,-2);
        ioClient.emit('arduinoData', currentData);
        currentData = '';
    }
});
…
```

1 Line Difference!*

*Aside from the connection, authentication, user management and server side code.

NodeJS Demo 3

Re-Doing Quartz Demo, but with YOU!

(…if my server is up…)

Pfft: Give Me An Example

- **Walk up to Machine**
- **Tweet using hashtag**
- **Raspberry Pi connected to Twitter**
- **Receives tweet from Socket.io**
- **Sends commend to Arduino**
- **Arduino vends**

Code Walkthrough

There Can Be Only One

(1 tweet per hashtag, than reset)

Sorry… couldn’t find
a good Highlander sword pic

Just pretend there’s one here.

What else?

- 1 Arduino per swing,
  5 in total
- 1 PC running NodeJS
- Sensors relaying data back to serialPort
- Arduinos controlling LED lighting colours based on sensors
- PC playing music based on sensor data

Code Walkthrough

SymphonyOfSwings

20' Tall Interactive Light-up Swing Set
(for Everybody!)

One more, but…

TOP SECRET

- **Super Secretive Client**
- **16 units of Raspberry Pi, Arduino and LEDs**
- **About 5x the LEDs of the swings**
- **All communicating together along with video**

Internet of Everything!

What can you build?

Tweet me your IOT
@AskMP

Questions?