

Working with devices in general

```
// Get the device's identifier
var id = device.getIdentifier();

// Get the device's name
var name = device.getName();

// Get the device's unique identifier (uid)
var uid = device.getUID();
```

Working with the RGB LED Light

```
// Set the color of the RGB light
rgb.setColor(255, 0, 0);

// Let the RGB light blink in a color
rgb.blink(0, 255, 0, 500);

// Turn the RGB light off
rgb.off();
```

Working with the RGB LED Button

```
// Set the color of the button light
btn.setColor(255, 0, 0);

// Let the button light blink in a color
btn.blink(0, 255, 0, 500);

// Turn the button light off
btn.off();

// Get informed when the button is pressed
btn.addListener(buttonPressed);

function buttonPressed(val) {
    var state = val.getValue();

    if(state.value == "PRESSED")
        ...

    if(state.value == "RELEASED")
        ...
}
```

Getting a particular device

```
// A global variable to store all devices
var devices;
var rgb;

// Initialize the connections
tf.initDevices(initDone);

// Call this when all connections are established

function initDone(connectedDevices) {

    // Store the devices on the global variable
    devices = connectedDevices;

    // Option 1: Get a device by its unique identifier
    rgb = devices.getDeviceByUid("dmW");

    // Option 2: Get a device by its device identifier
    rgb = devices.getDeviceByIdentifier(271);
}
```

Working with the Humidity 2.0 sensor

```
// Set the interval for new values to 5 seconds
humiditySensor.setCallbackInterval(5000);

// Get informed when the sensor value changes
humiditySensor.addListener(humidityChanged);

// Define the callback function for new values
function humidityChanged(val) {

    var value = val.getValue();

    // Check if it is temperature or humidity
    if(value.type == "temperature") {
        console.log("Temperature is: " + value.value / 100 + "°C");
    }

    if(value.type == "humidity")
        log("Humidity is: " + value.value / 100 + "%");
}
```

Listening to events in general (sensors, buttons)

```
// Add a listener
device.addListener(eventHappened);

// Define the callback function
function eventHappened(eventInfo) {
    // Do something with the new info
}
```