{POWER.CODERS}

APIs

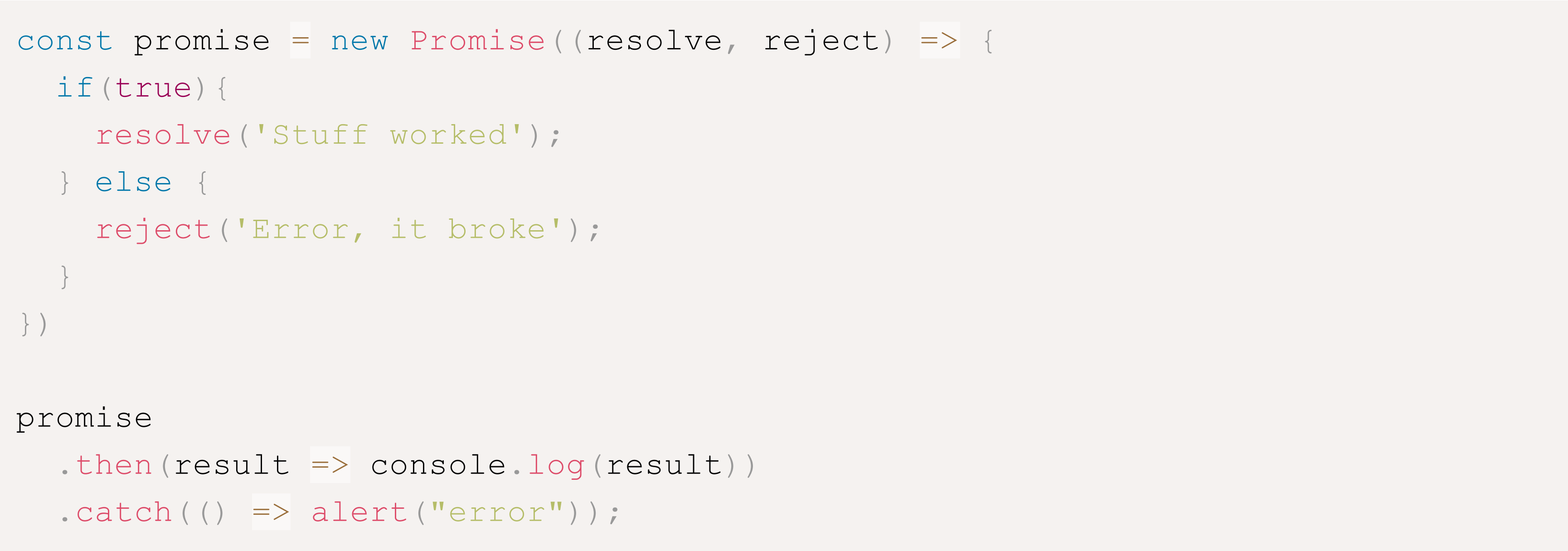
AGENDA

Today we will learn

- Recap Promise
- What is an API
- How to use an API

PROMISE

A **promise** is an **object** that may produce a single value some time **in the future**. Either a **resolve** value, or a reason why it's not resolved (**rejected**).



```
const promise = new Promise((resolve, reject) => {
  if(true){
    resolve('Stuff worked');
  } else {
    reject('Error, it broke');
  }
})

promise
  .then(result => console.log(result))
  .catch(() => alert("error"));
```

EXAMPLE

Website users can upload several images. We need to get the image dimensions for each image. We will use a **promise** to get the dimensions of each image.

RESOURCES

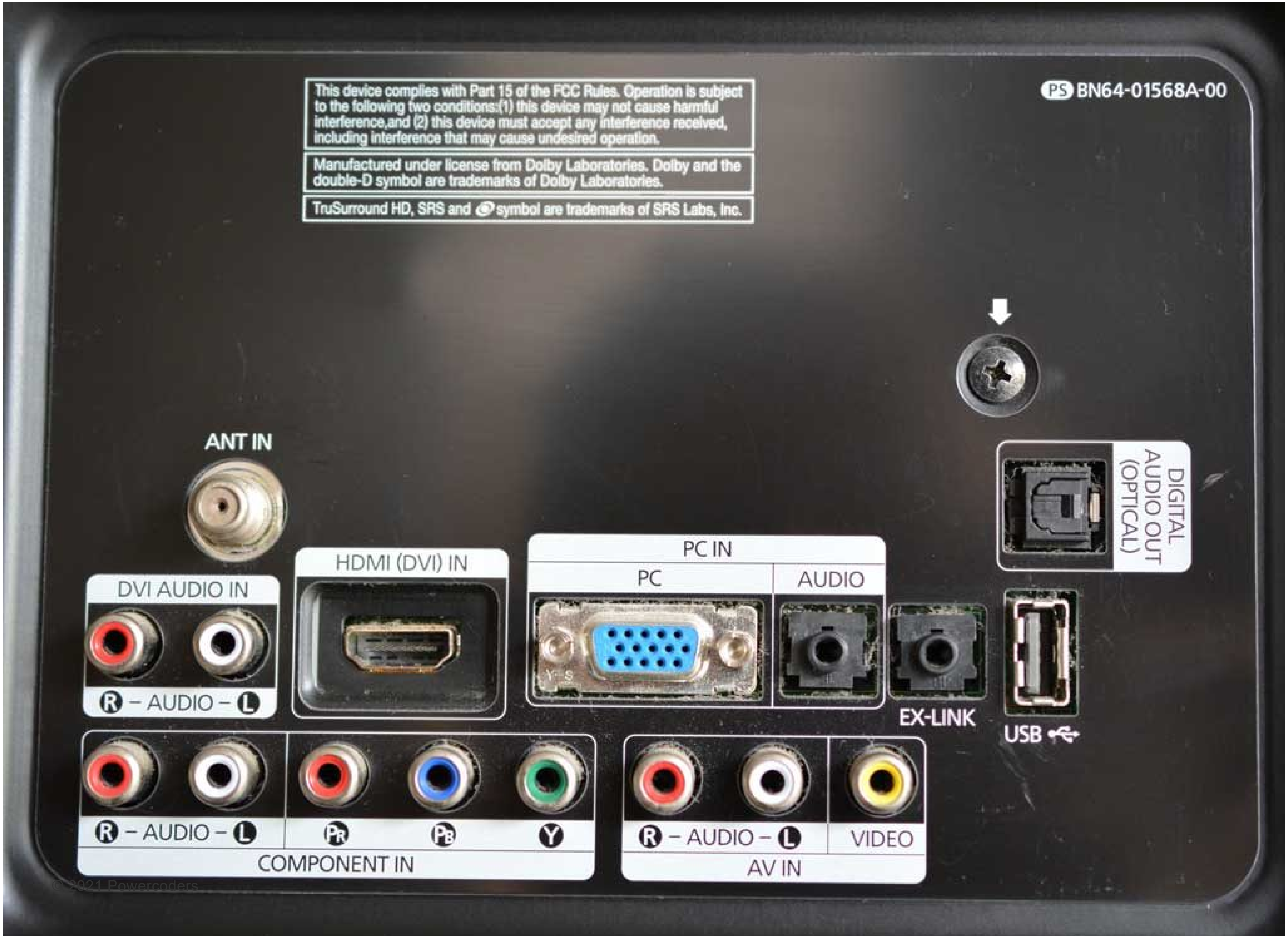
For live coding example

- Web API: File
- Web API: HTML Image
- Web API: URL

LIVE CODING

Code along with me

What is an API?



This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions:(1) this device may not cause harmful interference,and (2) this device must accept any interference received, including interference that may cause undesired operation.

Manufactured under license from Dolby Laboratories. Dolby and the double-D symbol are trademarks of Dolby Laboratories.

TruSurround HD, SRS and symbol are trademarks of SRS Labs, Inc.

What is an API?

Application Programme Interface

- Allows strangers to **communicate** with each other
- Allows to share **data** across machines and systems

Remember fetch API?

```
fetch("https://jsonplaceholder.typicode.com/todos/1")
   .then(response => response.json())
   .then(data => console.log(data));
```

`fetch`

1. returns a **promise**: I promise to let you know when the response of my request is returned

`fetch`

1. returns a **promise**: I promise to let you know when the response of my request is returned
2. then it returns a **response**: with its own method `json()` to parse the response

`fetch`

1. returns a **promise**: I promise to let you know when the response of my request is returned
2. then it returns a **response**: with its own method `json()` to parse the response
3. then it returns a **object**: with the data of your AJAX call

Live coding

Remember our image slider?

Github image slider

RESOURCES

for our live coding example

- Photos API
- Pixabay API (with key)
- unsplash API (with key)
- Swiper Plugin

Free APIs

For fun and testing

- Star wars api
- Robohash API
- Where is the IIS?
- Fake JSON API

Business APIs

Usually charge for use

- Google APIs, e.g. Maps
- Twilio API for messaging
- Mailchimp API for newsletters
- Stripe API for payment

Other APIs

There are so many...

- Speech recognition
- Face recognition
- Aggreagtor to find public APIs

EXAMPLE

Add Google Maps to a website

Code along while going through the slides.

API KEYS

For authentication

To use any Google API you need to authenticate first.

Google Developer Console

Google Developer Console

You need to go through several steps to setup authentication. You need a Google account for that, like your Powercoders-Account.

1. Create a new Google API Project
2. Enable the libraries you want For our example we will select **Maps JavaScript API**.
3. Create the Authentication Credentials For our example we will use the **API key**.
4. Restrict the key's usage

With our API key we can start coding.

Google Maps

We will use the **JavaScript API**, but there are also choices for native apps and many more.

Google Maps

We will use the **JavaScript API**, but there are also choices for native apps and many more.

Google has a very extensive documentation which makes integration its services relatively easy.

Using the API

1. Include the API script to your HTML page

```
<script src="https://maps.googleapis.com/maps/api/js?
key=YOUR_API_KEY&callback=initMap" async defer>
```

Using the API

1. Include the API script to your HTML page

```
<script src="https://maps.googleapis.com/maps/api/js?
key=YOUR_API_KEY&callback=initMap" async defer>
```

2. Include a placeholder for the map

```
<div id="map"></div>
```

Using the API

1. Include the API script to your HTML page

```
<script src="https://maps.googleapis.com/maps/api/js?
key=YOUR_API_KEY&callback=initMap" async defer>
```

2. Include a placeholder for the map

```
<div id="map"></div>
```

3. Include the map into the placeholder with JS

```
var map;
function initMap() {
  map = new google.maps.Map(document.getElementById('map'), {
    center: {lat: -34.397, lng: 150.644},
    zoom: 8
  });
}
```

Finding the right location

```
center: {lat: -34.397, lng: 150.644}
```

Finding the right location

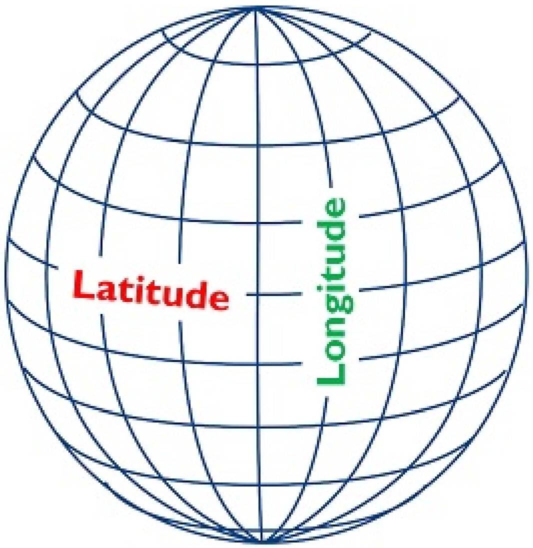
```
center: {lat: -34.397, lng: 150.644}
```

Each location on earth can be found by geo coordinates, the latitude and longitude.

Finding the right location

```
center: {lat: -34.397, lng: 150.644}
```

Each location on earth can be found by geo coordinates, the latitude and longitude.



Geo coordinates

One way to find the correct coordinates is to use Google Maps and copy the coordinates from the URL.

```
...@47.3739089,8.5328035,17z/...
```

Geo coordinates

One way to find the correct coordinates is to use Google Maps and copy the coordinates from the URL.

```
...@47.3739089,8.5328035,17z/...
```

You could also use the Geocoding Service API to dynamically get the coordinates based on the address.

Add a marker

```
var marker = new google.maps.Marker({
  position: {lat: 47.3739089, lng: 8.5328035},
  map: map
});
```

ADD A MARKER

```
var marker = new google.maps.Marker({
  position: {lat: 47.3739089, lng: 8.5328035},
  map: map
});
```

You can also use custom markers.

STYLE YOUR MAP

```
var styledMapType = new google.maps.StyledMapType([JSON styling object],
{name: 'Styled Map'});

// Create a map object, and include the MapTypeId to add
// to the map type control.
var map = new google.maps.Map(document.getElementById('map'), {
  ...
  mapTypeControlOptions: {
    mapTypeIds: ['roadmap', 'satellite', 'hybrid', 'terrain', 'styled_map']
  }
});

//Associate the styled map with the MapTypeId and set it to display.
map.mapTypes.set('styled_map', styledMapType);
map.setMapTypeId('styled_map');
```

STYLE YOUR MAP

```
var styledMapType = new google.maps.StyledMapType([JSON styling object],
{name: 'Styled Map'});

// Create a map object, and include the MapTypeId to add
// to the map type control.
var map = new google.maps.Map(document.getElementById('map'), {
  ...
  mapTypeControlOptions: {
    mapTypeIds: ['roadmap', 'satellite', 'hybrid', 'terrain', 'styled_map']
  }
});

//Associate the styled map with the MapTypeId and set it to display.
map.mapTypes.set('styled_map', styledMapType);
map.setMapTypeId('styled_map');
```

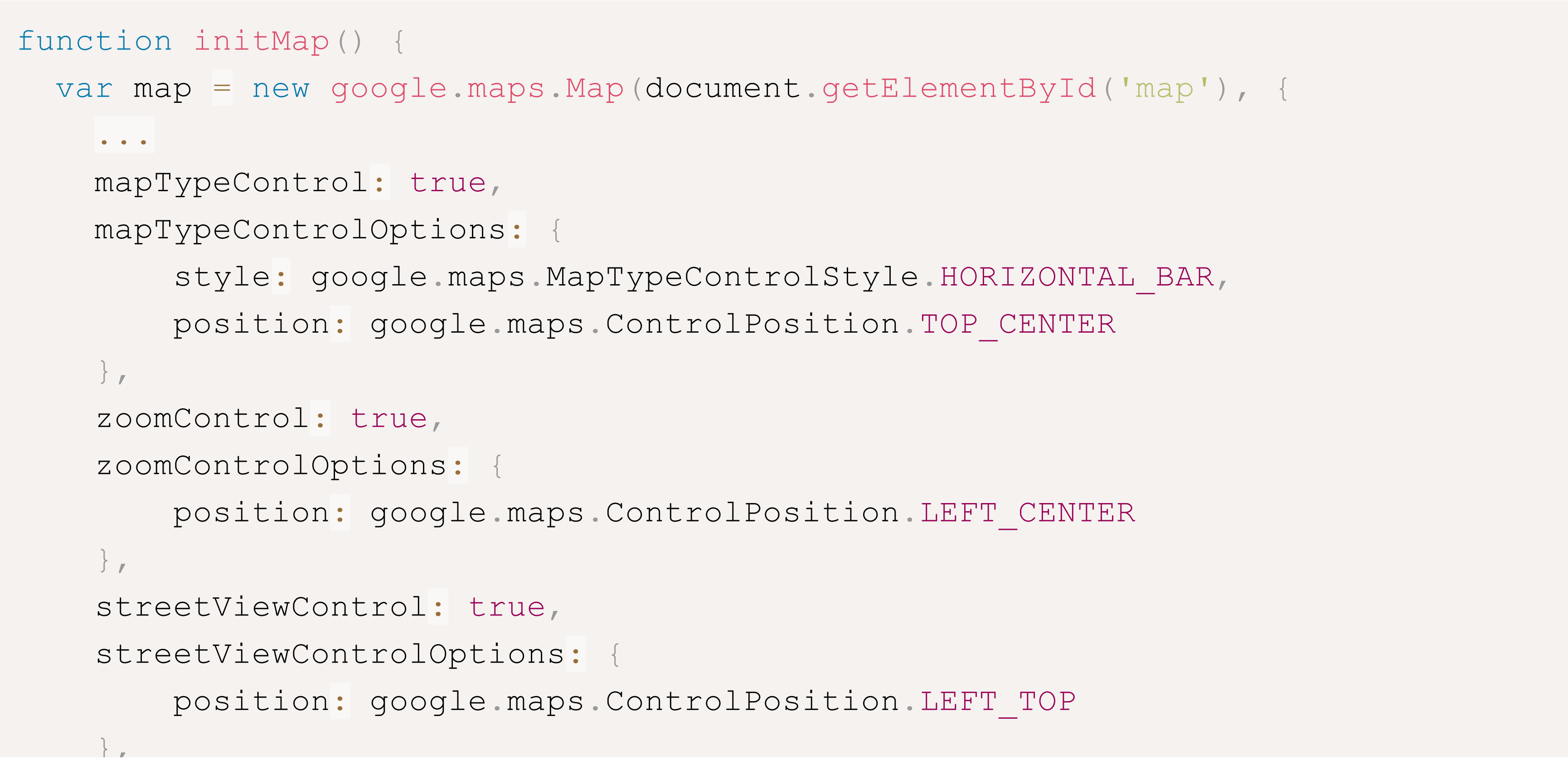
Easiest way to get the JSON styling object:
Google Maps Platform Styling Wizard.

Configure controls

```
...disableDefaultUI: true...
```

```
{
  zoomControl: boolean,
  mapTypeControl: boolean,
  scaleControl: boolean,
  streetViewControl: boolean,
  rotateControl: boolean,
  fullscreenControl: boolean
}
```

Example



```
function initMap() {
  var map = new google.maps.Map(document.getElementById('map'), {
    ...
    mapTypeControl: true,
    mapTypeControlOptions: {
        style: google.maps.MapTypeControlStyle.HORIZONTAL_BAR,
        position: google.maps.ControlPosition.TOP_CENTER
    },
    zoomControl: true,
    zoomControlOptions: {
        position: google.maps.ControlPosition.LEFT_CENTER
    },
    streetViewControl: true,
    streetViewControlOptions: {
        position: google.maps.ControlPosition.LEFT_TOP
    },
```

Alternatives to Google Maps

If you need more than 25'000 transactions per month Google Maps can become quite expensive. There might be other reasons as well like the length of static image map urls.

- Mapbox
- HERE
- Tomtom
- Open Street Map

ONLINE RESSOURCES

- Authentication
- 8 amazing Google APIs
- W3schools Google Maps tutorial
- Official Google Maps documentation
- Cheaper alternatives

Exercises

1. IMPROVE YOUR FACEBOOK APP

On my Github

1. Use an API to get users into the database variable
2. Use an API to get news into the newsfeed variable
3. Add a form to the HTML and remove the prompts
4. Use AJAX on submitting the form
5. If the login is correct, remove the form from the DOM
6. If the login is correct, add found news to DOM
7. If the login is incorrect, return an error message

2. Improve your background generator

On my Github

1. Write code so that the colour inputs match the background generated on the first page load.
2. Display the initial CSS linear gradient property on page load.
3. Add a random button which generates two random numbers for the colour inputs.