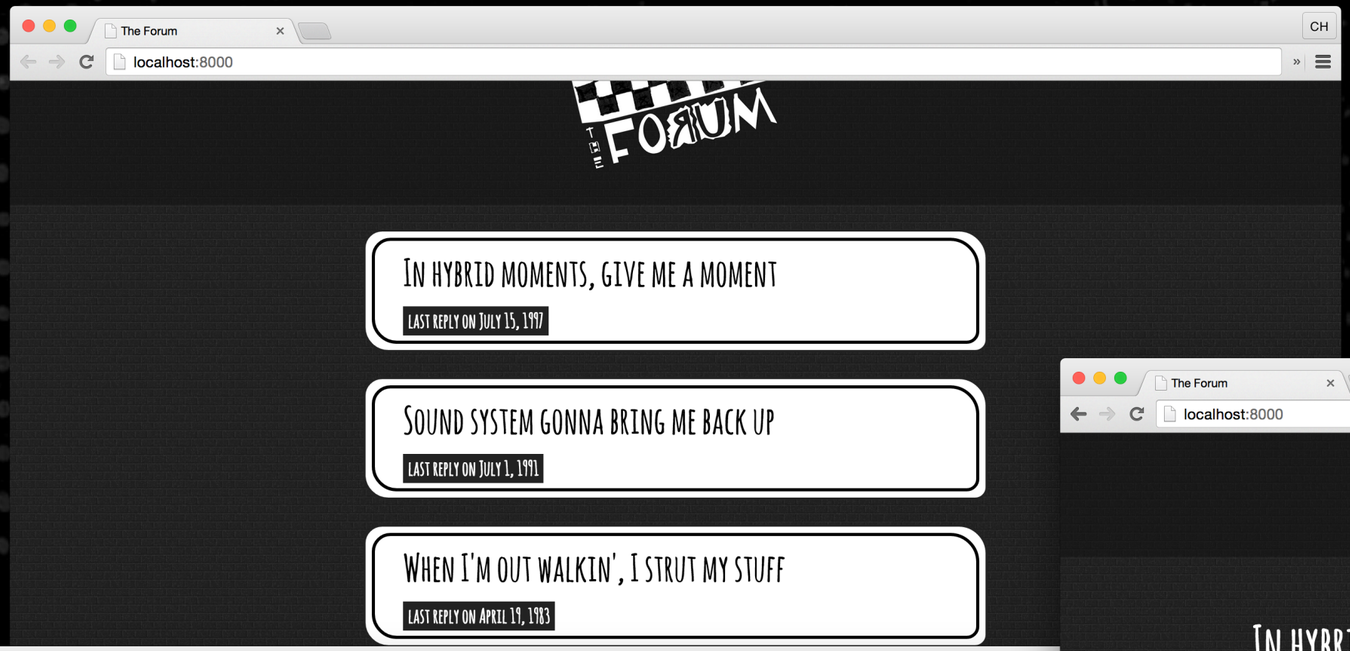
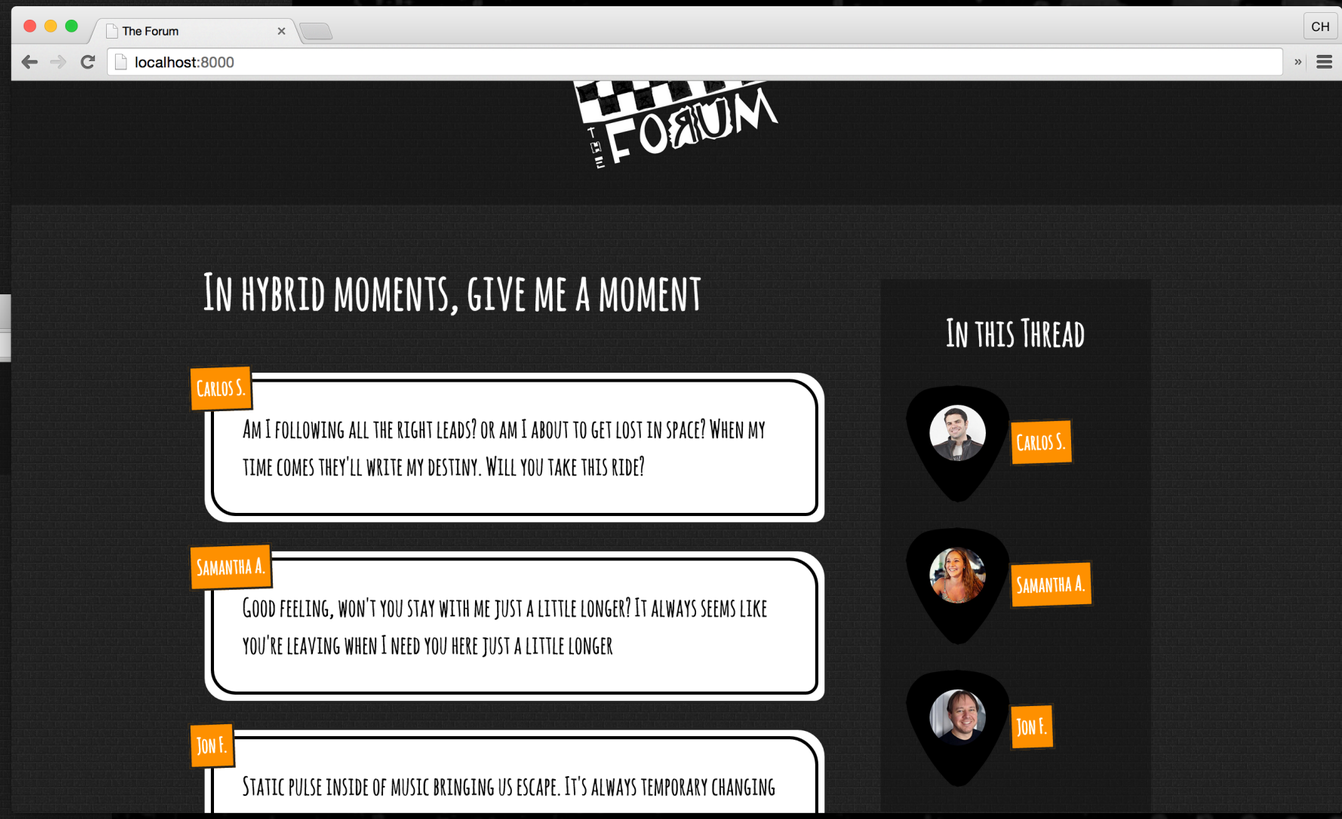
Declarations with *let*

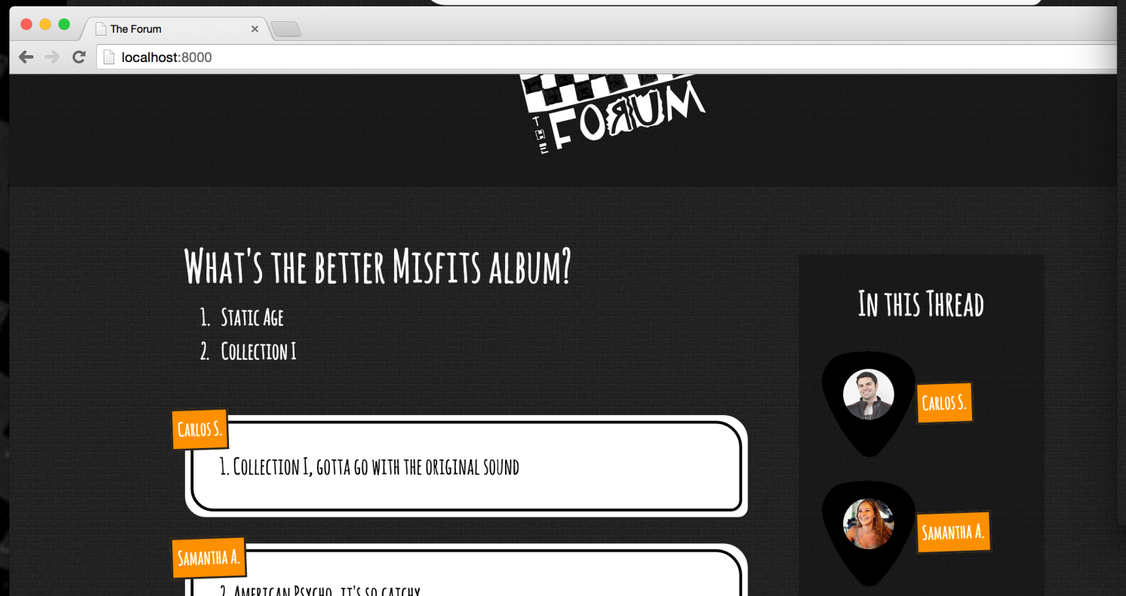
Level 1 – Section 1

Forum Web App



Together, we'll build features that are part of a web forum!



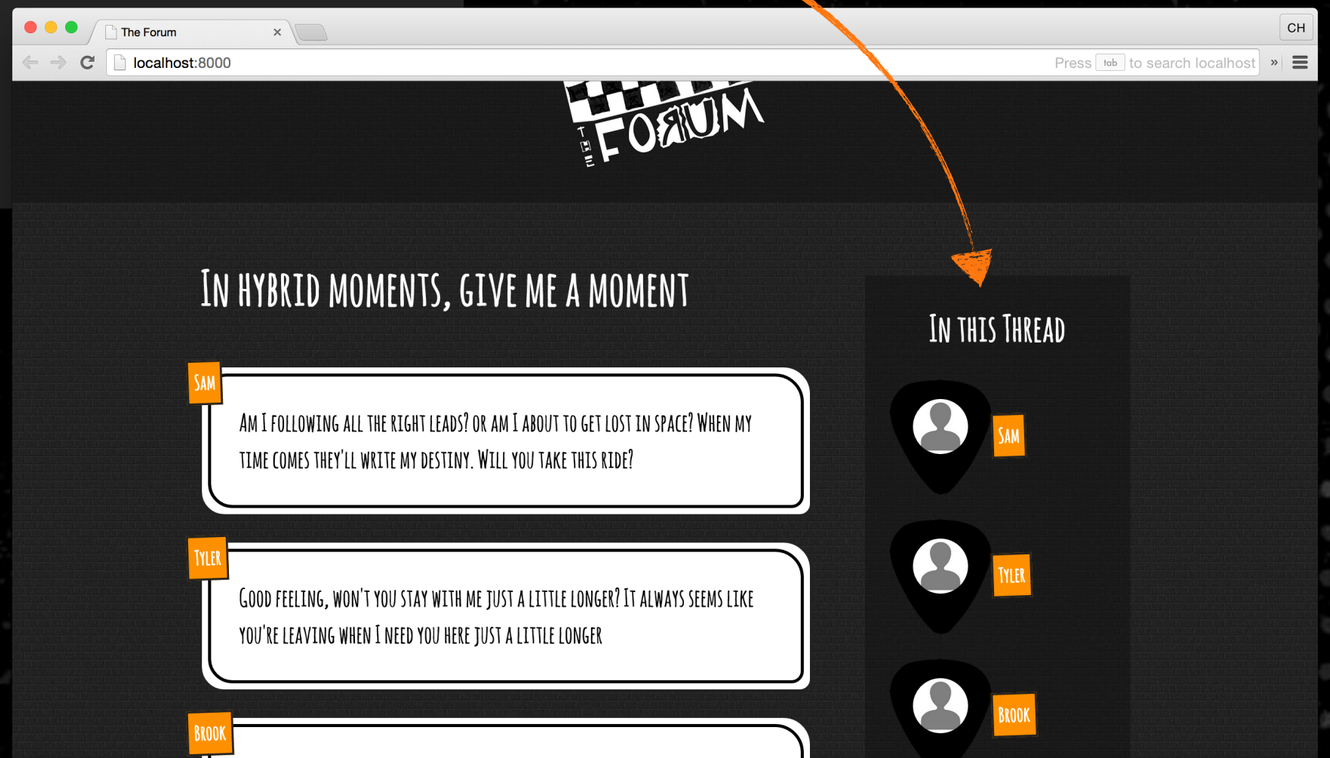


Loading Profiles

The *loadProfiles* function takes an **array of users** and adds their profile to the sidebar.

```
<!DOCTYPE html>
    //...
    <script src="./load-profiles.js">
    <script>
      loadProfiles(["Sam", "Tyler", "Brook"]);
    </script>
  </body>
</html>
```

displays active users for this topic



The loadProfiles Function

This function displays a different message according to the number of arguments.

```
function loadProfiles(userNames){
  if(userNames.length > 3){
    var loadingMessage = "This might take a while...";
    _displaySpinner(loadingMessage);
  }else{
    var flashMessage = "Loading Profiles";
    _displayFlash(flashMessage);
  }

  //... fetch names and build sidebar
}
```

Might take a while to process, so a loading message is displayed

Shouldn't take that long, so we display a different message

How loadProfiles Is Executed

```
loadProfiles(["Sam", "Tyler", "Brook", "Alex"]);
```

1. Executes the if block

2. The loadingMessage variable is declared and assigned a value

```
function loadProfiles(userNames){
  if(userNames.length > 3){
    var loadingMessage = "This might take a while...";
    _displaySpinner(loadingMessage);
  }else{
    var flashMessage = "Loading Profiles";
    _displayFlash(flashMessage);
  }

  //...
}
```

3. The displaySpinner function is called

The else block is not executed

Unexpected Behavior With var

```
function loadProfiles(userNames){
  if(userNames.length > 3){
    var loadingMessage = "This might take a while...";
    _displaySpinner(loadingMessage);
    console.log(flashMessage);
  } else{
    var flashMessage = "Loading Profiles";
    _displayFlash(flashMessage);
  }

  console.log(flashMessage);
}
```

> undefined

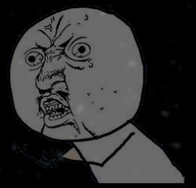
This indicates a console output

> undefined

...but outputs undefined and does not generate error (?!)

flashMessage is never declared...

Why no **ReferenceError** ?



Understanding Hoisting

Prior to executing our code, JavaScript moves *var* declarations all the way up to the top of the scope. This is known as **hoisting.**

```
function loadProfiles(userNames){
  var loadingMessage, flashMessage;

  if(userNames.length > 3){
    var loadingMessage = "This might take a while...";
    _displaySpinner(loadingMessage);
  }else{
    var flashMessage = "Loading Profiles";
    _displayFlash(flashMessage);
  }

  //....
}
```

Automatically moved here by the JavaScript runtime

Declaring Variables With let

ES2015

let variables are scoped to the nearest **block** and are **NOT hoisted.**

(A block is any code section within **curly braces**, like *if*, *else*, *for*, *while*, etc.)

```
function loadProfiles(userNames){
  if(userNames.length > 3){
    let loadingMessage = "This might take a while...";
    _displaySpinner(loadingMessage);
  }else{
    let flashMessage = "Loading Profiles";
    _displayFlash(flashMessage);
  }

  //....
}
```

Not visible outside if block

Not visible outside else block

Expected Behavior With let

Using *let*, variables are “trapped” inside their respective *if* and *else* blocks.

```
loadProfiles(["Sam", "Tyler", "Brook", "Alex"]);
```

```
function loadProfiles(userNames){
  if(userNames.length > 3){
    let loadingMessage = "This might take a while...";
    _displaySpinner(loadingMessage);

    console.log(flashMessage);
  }else{
    let flashMessage = "Loading Profiles";
    _displayFlash(flashMessage);
  }

  console.log(flashMessage);
}
```

> ReferenceError: flashMessage is not defined

Expected behavior, similar to other programming languages!

> ReferenceError: flashMessage is not defined