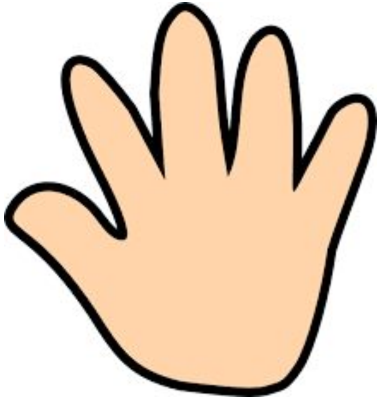
Introduction to JavaScript and React

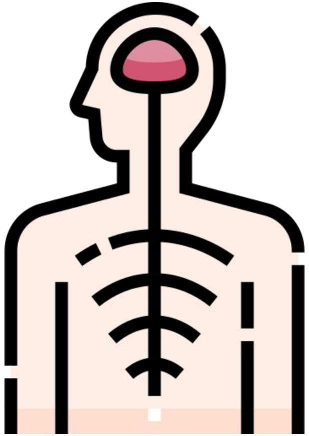
Created by Zane Harrison
Duke Catalyst
October 25th, 2023

Body Analogy

HTML - Skeleton



CSS - Skin



JavaScript - Nervous system, heart, organs, etc.

What is JavaScript?

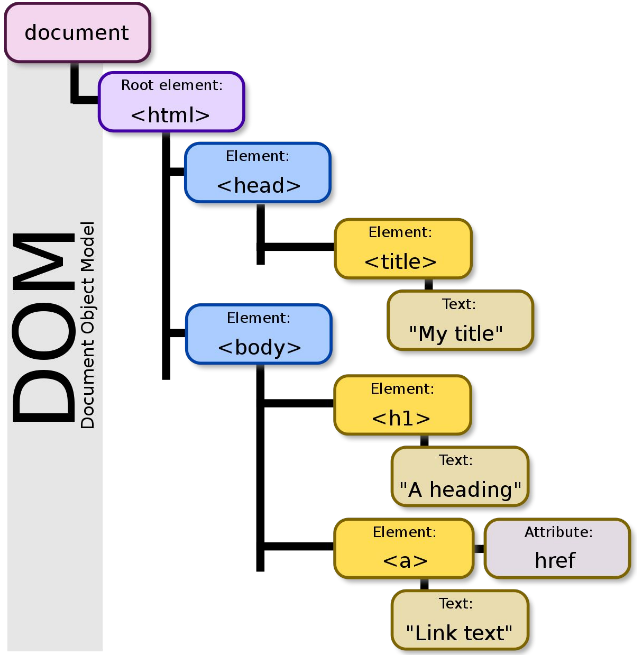
JavaScript (JS) is a programming language that gives web pages interactivity and dynamic content. It can be used to:

- ❑ Modify the structure and content of a page based on user actions
- ❑ Create visual effects on a page
- ❑ Store and manipulate data
- ❑ Give functionality to page elements (such as buttons and forms)
- ❑ Interact with APIs
- ❑ And do much more

JavaScript - DOM Manipulation

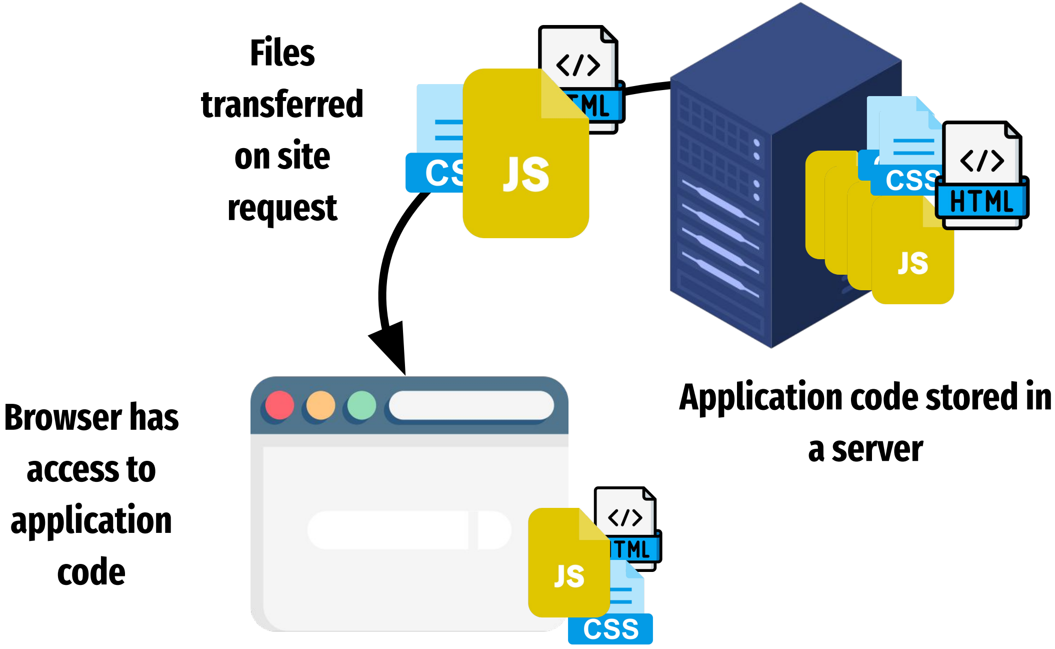
Recall that the DOM (Document Object Model) is a way of representing the structure and content of a web page

JavaScript can be used to manipulate the DOM, allowing the programmer to do things such as creating new elements, removing elements, and altering the contents or attributes of existing ones



JavaScript - Incorporating JS Code Into A Site

When a user requests a website in their browser, the response from the server contains JS files, giving the browser access to JS code



Files transferred on site request

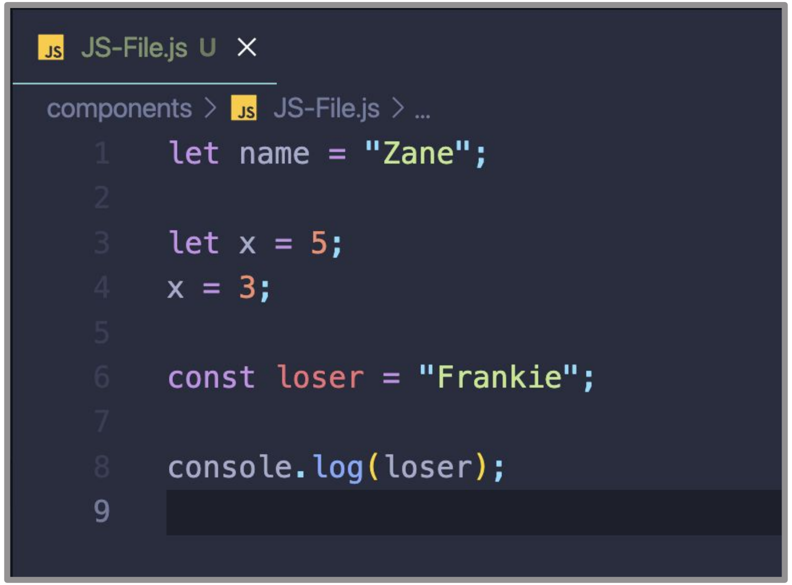
Application code stored in a server

Browser has access to application code

JavaScript Syntax - Variable Declaration

As in other programming languages, variables are used to store data

“**let**” (mutable) and “**const**” (immutable) are used to declare variables in JavaScript

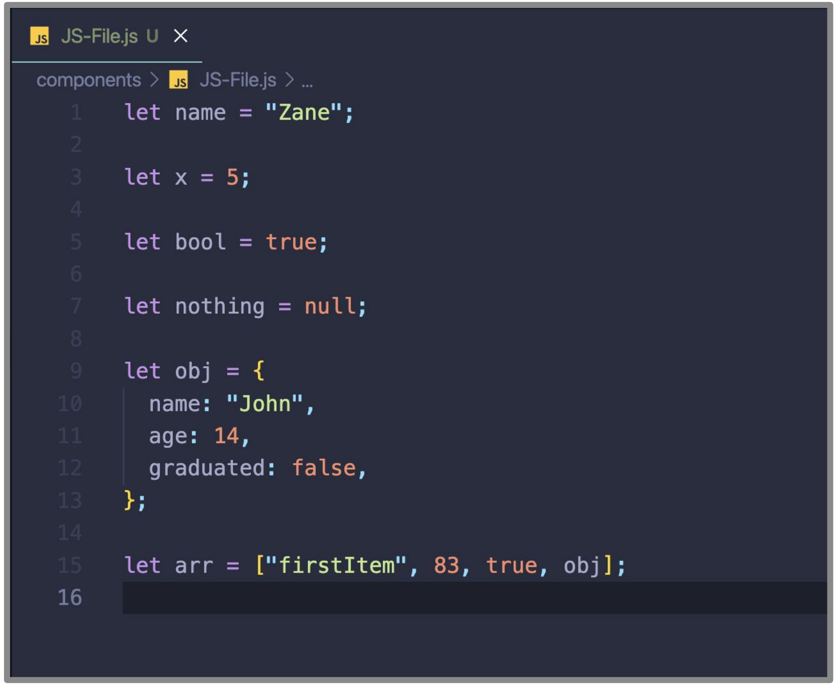


```
let name = "Zane";

let x = 5;
x = 3;

const loser = "Frankie";

console.log(loser);

```

JavaScript Syntax - Data Types

Some basic data types:

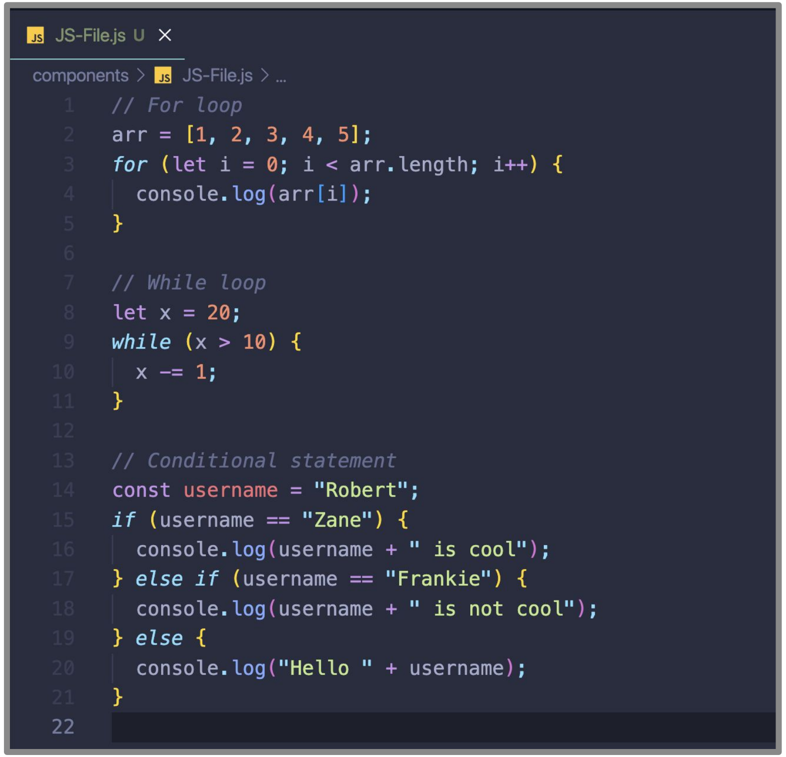
- String
- Number
- Boolean
- Null
- Object
  - Object
  - Array
  - Date



```
let name = "Zane";

let x = 5;

let bool = true;

let nothing = null;

let obj = {
  name: "John",
  age: 14,
  graduated: false,
};

let arr = ["firstItem", 83, true, obj];

```

JavaScript Syntax - Loops and Conditionals

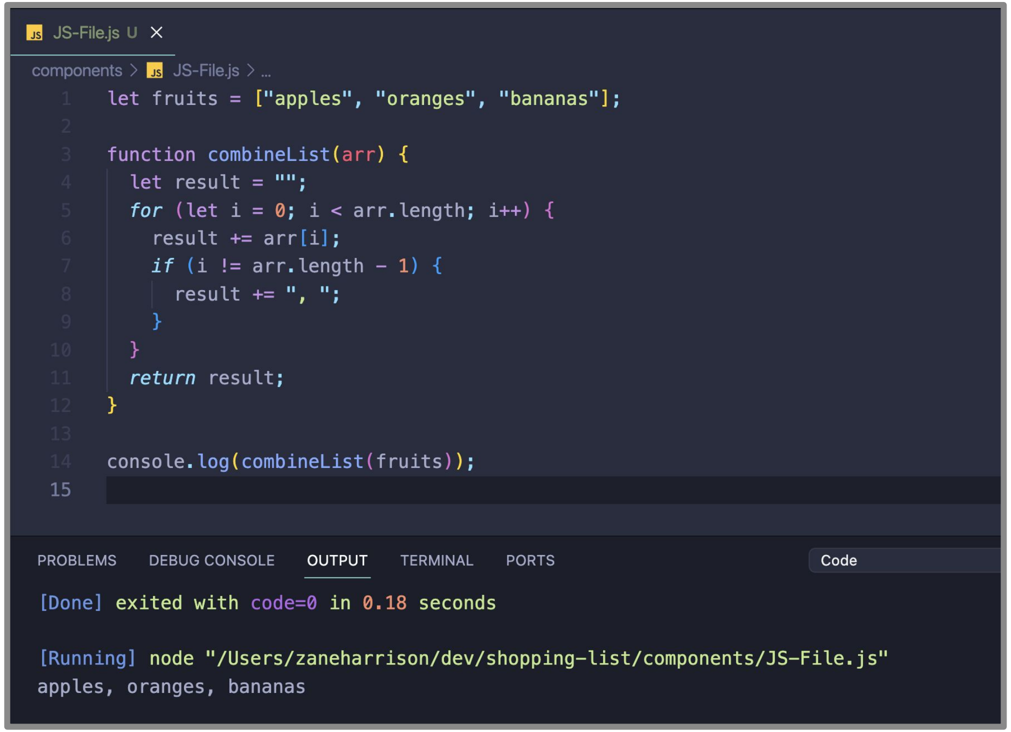
You can use “for” and “while” loops to iterate through lists of data, and conditional statements can be used to control the logical flow of your code



```javascript
// For loop
arr = [1, 2, 3, 4, 5];
for (let i = 0; i < arr.length; i++) {
  console.log(arr[i]);
}

// While loop
let x = 20;
while (x > 10) {
  x -= 1;
}

// Conditional statement
const username = "Robert";
if (username == "Zane") {
  console.log(username + " is cool");
} else if (username == "Frankie") {
  console.log(username + " is not cool");
} else {
  console.log("Hello " + username);
}

```

JavaScript Syntax - Functions

As in other programming languages, functions are blocks of reusable code that can receive arguments and return data



```javascript
let fruits = ["apples", "oranges", "bananas"];

function combineList(arr) {
  let result = "";
  for (let i = 0; i < arr.length; i++) {
    result += arr[i];
    if (i != arr.length - 1) {
      result += ", ";
    }
  }
  return result;
}

console.log(combineList(fruits));
```

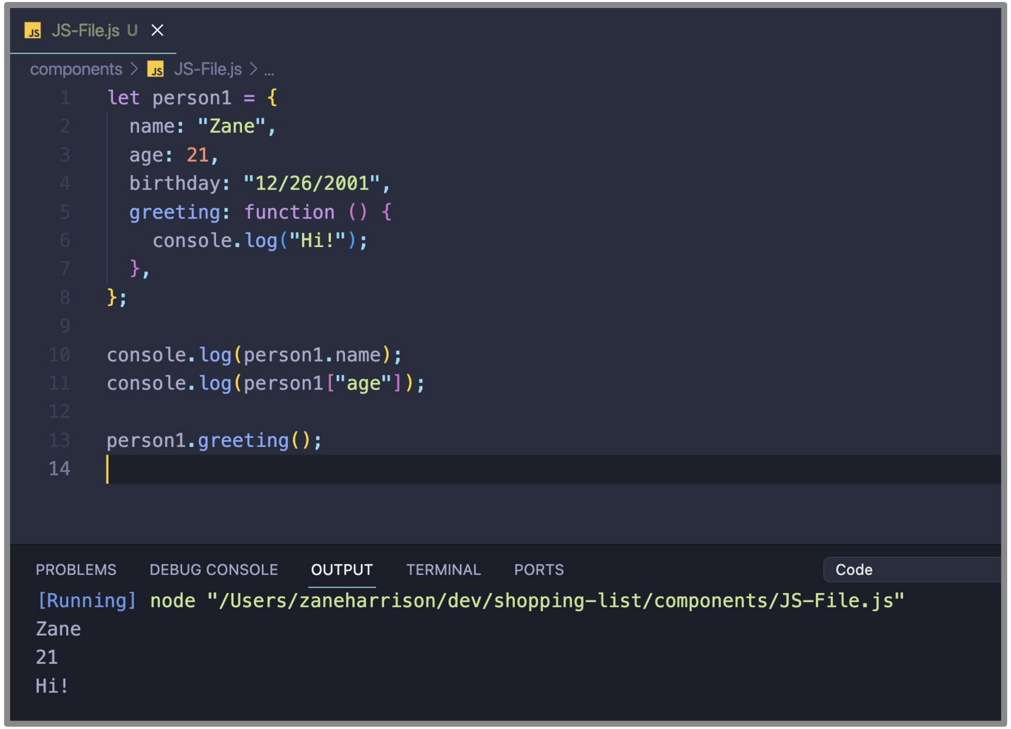
```
[Done] exited with code=0 in 0.18 seconds

[Running] node "/Users/zaneharrison/dev/shopping-list/components/JS-File.js"
apples, oranges, bananas
```

JavaScript Syntax - Objects

Objects are custom data types that contain collections of properties

An object can be used to represent a person, a car, a country, or anything else!

You can access an object's properties with dot or bracket syntax



```
let person1 = {
  name: "Zane",
  age: 21,
  birthday: "12/26/2001",
  greeting: function () {
    console.log("Hi!");
  },
};

console.log(person1.name);
console.log(person1["age"]);

person1.greeting();

```

```
[Running] node "/Users/zaneharrison/dev/shopping-list/components/JS-File.js"
Zane
21
Hi!
```

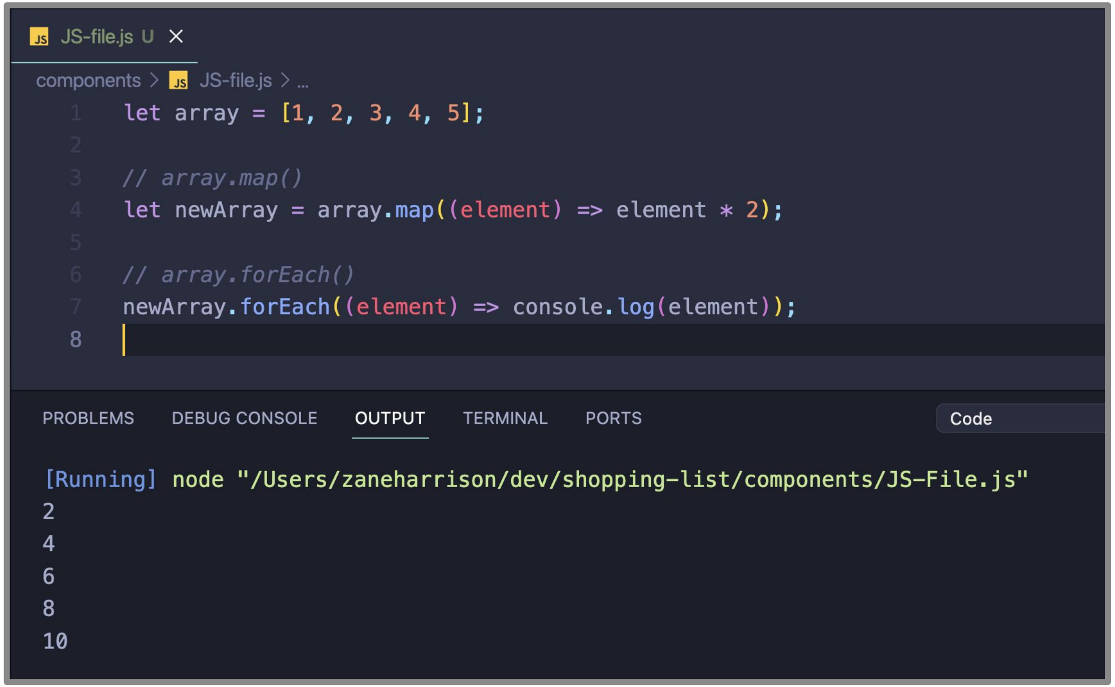
JavaScript Syntax - Array Functions

A few useful built-in array functions:

array.map()

array.forEach()

Both take functions as arguments



```
let array = [1, 2, 3, 4, 5];

// array.map()
let newArray = array.map((element) => element * 2);

// array.forEach()
newArray.forEach((element) => console.log(element));

```

```
[Running] node "/Users/zaneharrison/dev/shopping-list/components/JS-File.js"
2
4
6
8
10
```

What is React?

React is a **JavaScript library** developed by Meta that is used to build user interfaces

React applications are built using reusable **components**

React efficiently updates the user interface by working with a virtual representation of the DOM

JSX, a JavaScript extension, allows your code to interact with the DOM

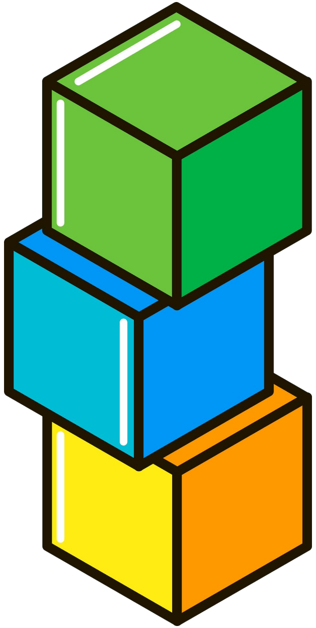
React - Components

Components are at the core of React, and can be thought of as the building blocks of the user interface

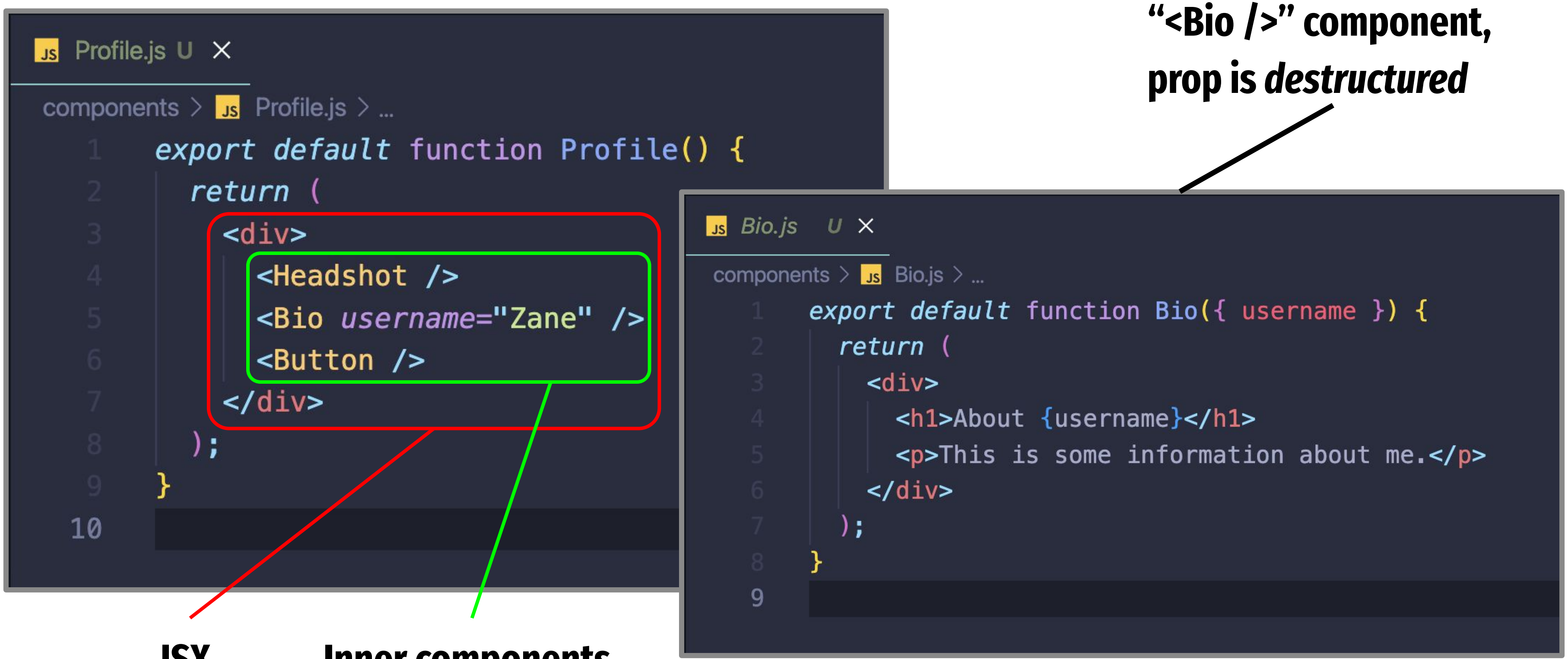
In essence, they are functions that return pieces of UI in the form of **JSX code**

They can be things such as buttons, forms, or more complex pieces of your application

Components can accept arguments, which are often referred to as “**props**”



React - Components



```
export default function Profile() {
  return (
    <div>
      <Headshot />
      <Bio username="Zane" />
      <Button />
    </div>
  );
}
```

```
export default function Bio({ username }) {
  return (
    <div>
      <h1>About {username}</h1>
      <p>This is some information about me.</p>
    </div>
  );
}
```

“<Bio />” component, prop is *destructured*

JSX

Inner components

React - Shopping List Demo

Okay enough theory, let’s put some of this information into practice by building a simple React website

GitHub Project - Shopping List

1. Go to this GitHub repository: https://github.com/zaneHarrison/shopping-list
2. Click “Fork”
   a. Uncheck the checkbox that says “Only fork main branch”
   b. Click “Create fork”
3. In your forked repository in GitHub, click the green button that says “Code”, click “HTTPS”, and copy the link
4. Open a terminal window
5. Navigate to the directory where you want your project to live
6. Run “git clone ___” with the copied HTTPS link in the blank
7. In your terminal, cd into the newly created repository
8. Run this command to install Node version manager:
   a. curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.5/install.sh | bash
9. Run “nvm install node”
10. Run “npm install”
11. Run “npm run dev”
12. Open http://localhost:3000/ in a browser window

GitHub Project - Your Starting Point

- The provided repository gets you started with this project. It contains:
    - A “page.js” file that represents the main (and only) page of the site
    - A “components” directory that contains two components
        - “ListTitle.js” is a component that represents the title of the shopping list
        - “List.js” is a component that represents the body of the shopping list. This component also contains the list data, represented as an array of objects
    - A “styles” directory that contains two CSS files
        - “list-title.module.css” contains CSS classes to style the “ListTitle” component
        - “list.module.css” contains CSS classes to style the “List” component

GitHub Project - Your Goal

- ❑ Your goal is to give the list content. This will require that you:
    - ❑ **Add some data to the “items” array within the “List” component**, which represents shopping items
    - ❑ **Create a component to represent individual list items**
        - ❑ This component needs to interact with the data in the “items” array somehow...
        - ❑ It’s probably a good idea to create some styles for this component, too
    - ❑ **Populate the shopping list with the shopping list items**

GitHub Project - Pushing Your Repo to GitHub

If you want to save the changes you've made to your version of the site, you need to push your local repository to your remote repository, which is the GitHub repository that was created from your fork. To do this, follow these steps:

1. Go to your GitHub settings and click "Developer settings" at the bottom of the left panel
2. Click "Personal access tokens" → "Tokens (classic)" → "Generate new token"
   a. Check all the boxes to grant your token access to all scopes
   b. Click "Generate token"
3. Copy the token link and save it somewhere
4. In your terminal, run "git add .", "git commit -m "Filled shopping list"", and "git push"
5. You'll be prompted to enter your GitHub username and password, and you should use the token you created in place of your password

Thanks for attending!

CATALYST