**Props:** Value that gets passed to the component.

*function* Name(*props*) {

  return <h2>I am { *props*.firstname }!</h2>; //Display name that was passed

}

*const* myelement = <*Name* firstname="Sam"/>; //Passes value into function

**State:** Value that is managed within the component itself.

*const* {useState} = React;

*const* Count = () *=>* {

*const* [num, setNum] = useState(0);

*const* CounterI = () *=>* {

       setNum (num+1); //Each time function is called value is incremented by 1

    }

    }

    return(

        <div>

            <p>Count: {num}</p>

            <button onClick={CounterI}>Counter</button>

        </div>

    );

}

**Functor:** A functor is a data structure you can map functions over and doing so modifying the contents of.

console.log([ 2, 4, 6 ].map(*x* *=>* *x* + 3))

// => [ 5, 7, 9 ]

**Promises:**

* **Pro:** Easy error handling
* **Con:** Only one object can be returned

**Callbacks:**

* **Pro:** Easy to create asynchronous javascript code
* **Con:** Error handling difficult

**Streams:**

* **Pro:** Memory Efficiency
* **Con:** More difficult than other techniques

**CSS Box Model:** The CSS Box Model is used by browser engines as a way to define how elements are organised and displayed on the screen.

Graphical user interface

Description automatically generated

* **Content**: Content of the box, where things like texts and images are displayed
* **Padding**: The padding is an area around the content itself, and is transparent
* **Border**: Border around both the padding and the content
* **Margin**: Area outside the border, like padding is transparent
* <!DOCTYPE html>
* <html>
* <head>
* <style>
* div.test{
* *background-color*: lightgrey;
* *width*: 300px;
* *height*: 100px;
* *border*: 15px solid green;
* *padding*: 50px;
* *margin*: 20px;
* }
* </style>
* </head>
* <body>
* <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse ornare ultricies arcu, a fermentum metus aliquam eu. Duis arcu orci. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse ornare ultricies arcu, a fermentum metus aliquam eu. Duis arcu orci. </p>
* <div class = test>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse ornare ultricies arcu, a fermentum metus aliquam eu. Duis arcu orci. </div>
* </body>
* </html>

**Bootstrapping a WebApp:**

* User navigates to URL and browser goes to Domain Name Servers
* Browser gets TCP/IP connection to server IP address and its associated port to that specific URL
* HTTP proxy accepts connection from browser after listening at that IP address
* Browser Accepts HTTP request over open TCP/IP connection
* Application server parses request and responds to browser over same connection delivering files