

CS390

WEB

APPLICATION

DEVELOPMENT

NISARG KOLHE

Working with the **DOM**

Working with the DOM

Getting DOM Elements

```
let element = document.getElementById( 'myId' );  
  
let elements = document.getElementsByClassName( 'myClass' );
```

Creating DOM Elements

```
let element = document.createElement( 'div' );  
  
document.body.appendChild(element);
```

Working with the **Elements**

Changing HTML

```
element.innerHTML = 'Some text or valid HTML';
```

Adding events

```
element.addEventListener('event', callbackFunc);
```

Adding CSS classes

```
element.classList.add("mystyle");
```

ES2015+

let vs var

let uses proper
scoping

var does not!

```
tests.reset();
function varExample() {
  var example = "HI!";
  if(1) {
    var example = "BYE!";
  }
  return example;
}
expect(varExample()).toBe("HI!"); // Fails
```

const

'nuff said.

```
tests.reset();  
const abc = "abc";  
//abc = "ANOTHER"; //ERROR! Can't reassign a constant variable  
/*
```

Template Strings!

```
console.log('My name is ' + person.firstName + ' ' + person.lastName + ' and my birthday is ' + person.birthday);
```

VS


```
console.log(`My name is ${person.firstName} ${person.lastName} and my birthday is ${person.birthday}`);
```


Array Functions

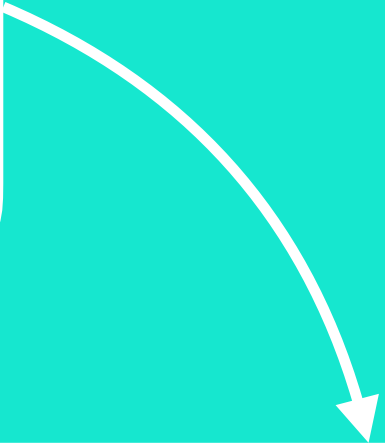
.map	maps every element to a new one
.forEach	iterates through every element...
.filter	returns items that fit a criteria

Fat **=>** Arrows

```
let hello = function () {  
  return "Hello";  
}
```



```
let hello = () => {  
  return "Hello";  
}
```



```
let hello = () => "Hello";
```

Fat => Arrows

- “this keyword scoping makes total sense”
-- **no one** ever.
- Fat arrows are used to:
 - Fix **scoping** by using lexical this
 - Provide a shorthand for small functions
- With no curly braces, is self-returning

Resources

- **Mozilla MDN Web Docs**
 - https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side_web_APIs/Manipulating_documents
- **W3Schools JS HTML DOM Docs**
 - https://www.w3schools.com/js/js_htmlDOM.asp
- **Plain JS**
 - <https://plainjs.com/javascript/manipulation/>