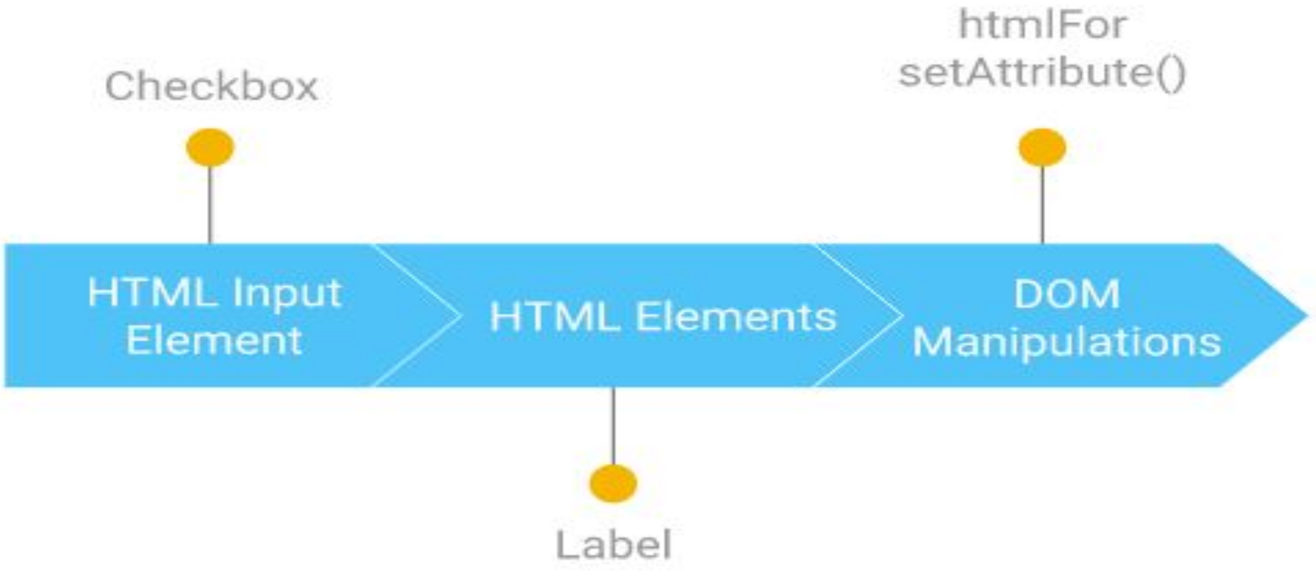
Todoapp

Explanation

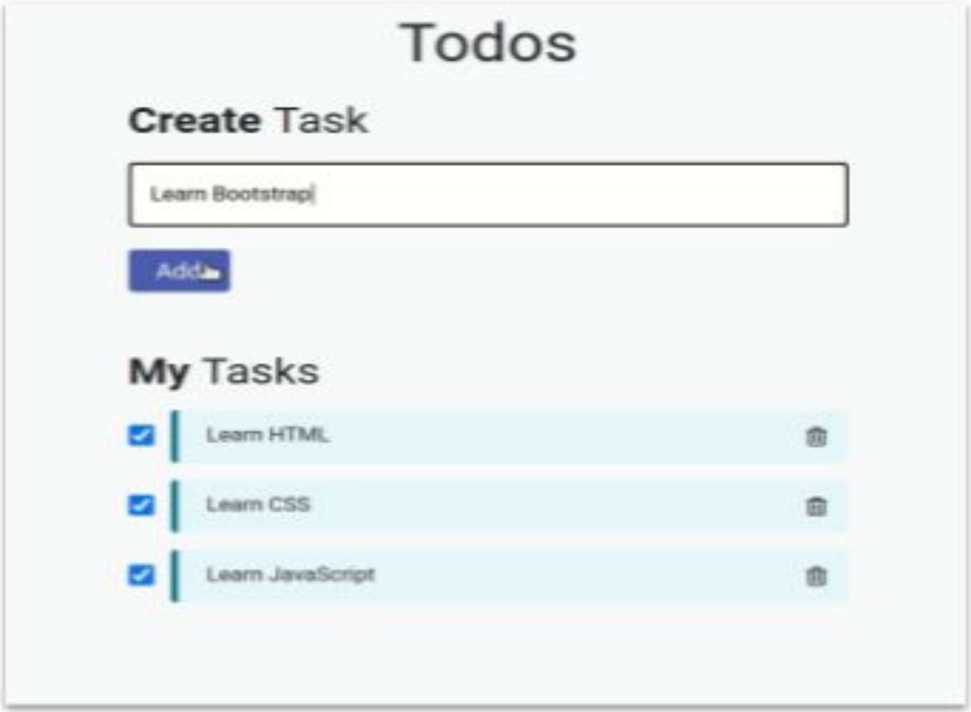
Agenda

Todos Application



Example

Todos Application


Todos

Create Task

Learn Bootstrap

Add

My Tasks

- ✓ Learn HTML
- ✓ Learn CSS
- ✓ Learn JavaScript

How to add

A Checkbox

Statically using HTML?

Checkbox

Label

HTML

```html
<body>

  <input type="checkbox" id="myCheckbox" />

  <label for="myCheckbox">Graduated</label>

</body>
```

Graduated
Graduated

DOM Manipulations

Creating Checkbox Input Dynamically

HTML

```html
<input type="checkbox" id="myCheckbox" />
<label for="myCheckbox">Graduated</label>
```

JS

```js
let inputElement = document.createElement('input');
inputElement.type = "checkbox";
inputElement.id = "myCheckbox";
document.body.appendChild(inputElement);
```

NXT WAVE Powered by iB HUBS

Creating Checkbox Input Dynamically

HTML

```
<input type="checkbox" id="myCheckbox" />
<label for="myCheckbox">Graduated</label>
```

JS

```
let labelElement = document.createElement('label');
labelElement.htmlFor = "myCheckbox";
labelElement.textContent = "Graduated";
document.body.appendChild(labelElement);
```

Graduated
Graduated

DOM Manipulations

setAttribute()

To set a **value of an attribute**

for a **specified element,** we use

setAttribute() method

```
element.setAttribute(attribute, value);
```

If the attribute already exists,

the value of the attribute gets **updated**

DOM Manipulations

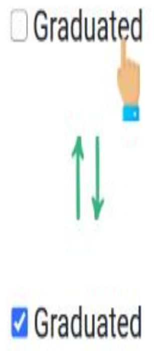
Creating Checkbox Input Dynamically

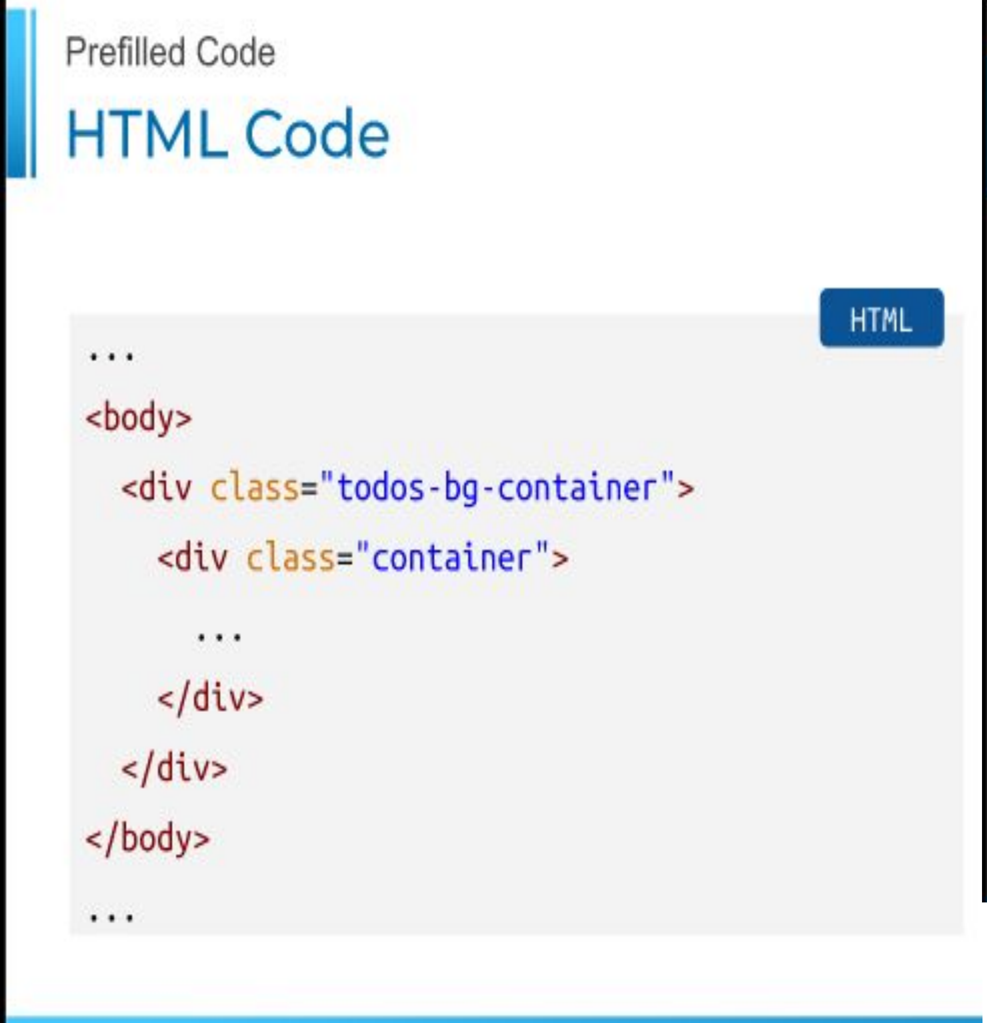
HTML

```html
<input type="checkbox" id="myCheckbox" />
<label for="myCheckbox">Graduated</label>
```

JS

```js
let labelElement = document.createElement('label');
labelElement.setAttribute("for", "myCheckbox");
labelElement.textContent = "Graduated";
document.body.appendChild(labelElement);
```

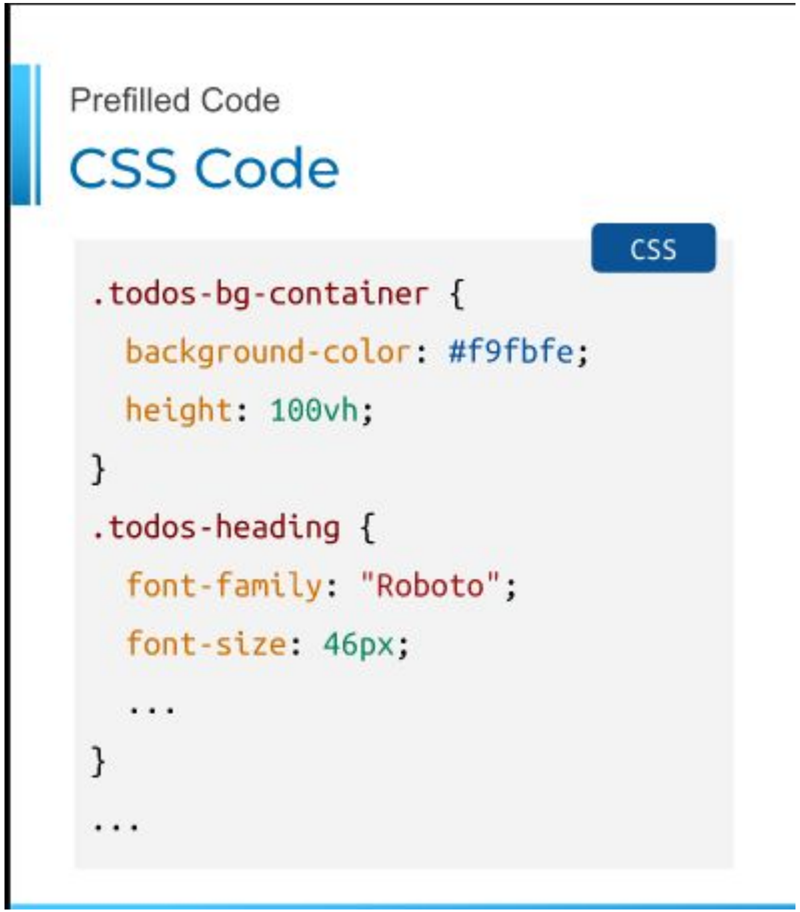



Prefilled Code

HTML Code

HTML

```html
...
<body>
  <div class="todos-bg-container">
    <div class="container">
      ...
    </div>
  </div>
</body>
...
```


Prefilled Code

CSS Code

CSS

```css
.todos-bg-container {
  background-color: #f9fbfe;
  height: 100vh;
}
.todos-heading {
  font-family: "Roboto";
  font-size: 46px;
  ...
}
...
```

Example

Todos Application

Steps:

- Create a Single Todo Item
- Create Multiple Todo Items
- Take User Input and Create Todos Dynamically
- Add Delete Todo Item Functionality

Powered by

Creating Todo Item

Todo List Item

HTML

```
<ul class="todo-items-container" id="todoItemsContainer">
  <li class="todo-item-container d-flex flex-row">
  </li>
</ul>
```

CSS

```
.todo-item-container {
  margin-top: 15px;
}
```

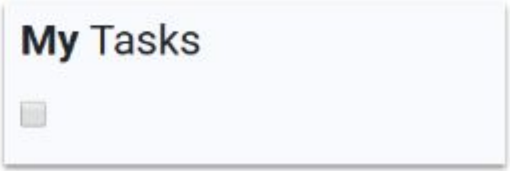
Creating Todo Item

Checkbox Input

HTML

```html
<ul class="todo-items-container" id="todoItemsContainer">
  <li class="todo-item-container d-flex flex-row">
    <input type="checkbox" class="checkbox-input" />
  </li>
</ul>
```

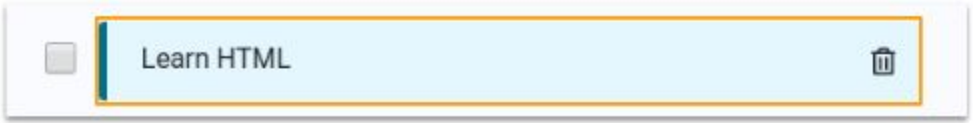
```css
.checkbox-input {
  width: 20px;
  height: 20px;
  margin-top: 12px;
  margin-right: 12px;
}
```



My Tasks

Creating Todo Item

Adding Label Container



HTML

```
...
<input type="checkbox" class="checkbox-input" />
<div class="d-flex flex-row label-container">

</div>
...
```

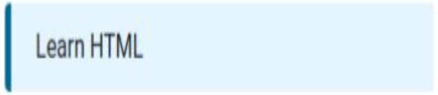
NXT WAVE

Creating Todo Item

Adding Label Container

HTML

```
.label-container {
  background-color: #e6f6ff;
  width: 100%;
  border-radius: 4px;
  border-style: solid;
  border-width: 5px;
  border-color: #096f92;
  border-right: none;
  border-top: none;
  border-bottom: none;
}
```



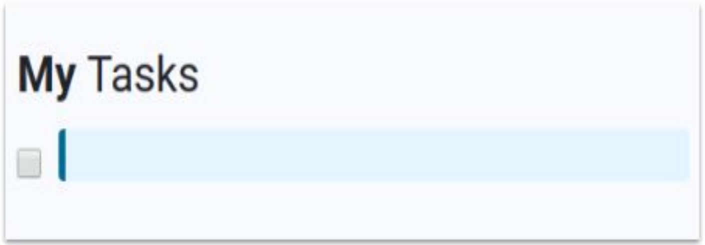
Learn HTML

#e6f6ff

#096f92

Adding Label Container

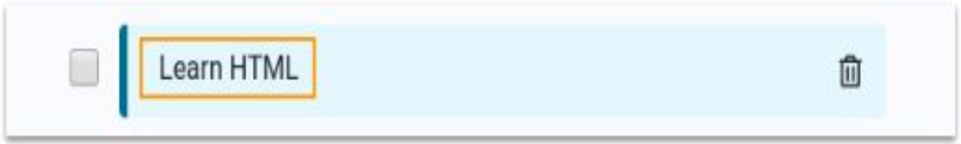
Output



My Tasks

Creating Label

Adding for and id attributes



HTML

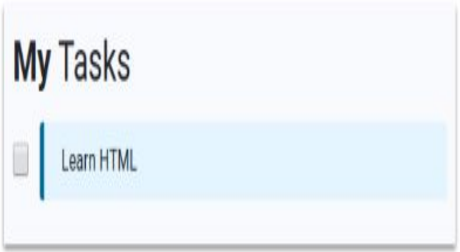
```html
...
<input type="checkbox" id="checkboxInput" class="checkbox-input" />

<div class="d-flex flex-row label-container">
  <label for="checkboxInput" class="checkbox-label">
    Learn HTML
  </label>
</div>
...
```

Creating Todo Item

Adding Label

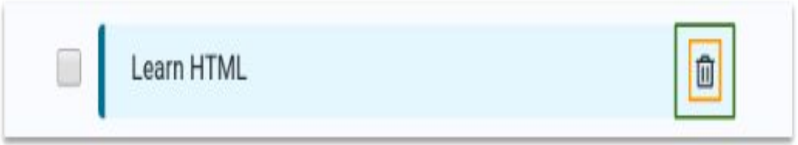
CSS

```css
.checkbox-label {
  font-family: "Roboto";
  font-size: 16px;
  font-weight: 400;
  width: 82%;
  margin: 0px;
  padding-top: 10px;
  padding-bottom: 10px;
  padding-left: 20px;
  padding-right: 20px;
  border-radius: 5px;
}
```



Creating Todo Item

Adding Delete Icon



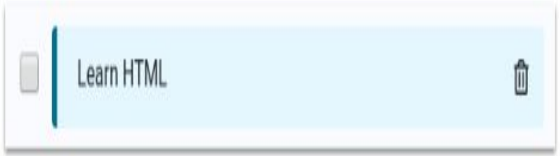
HTML

```html
...
  <div class="d-flex flex-row label-container">
    ...
    <div class="delete-icon-container">
      <i class="far fa-trash-alt delete-icon"></i>
    </div>
  </div>
...
```

Adding Delete Icon

CSS

```
.delete-icon-container {
  text-align: right;
  width: 18%;
}

.delete-icon {
  padding: 15px;
}
```

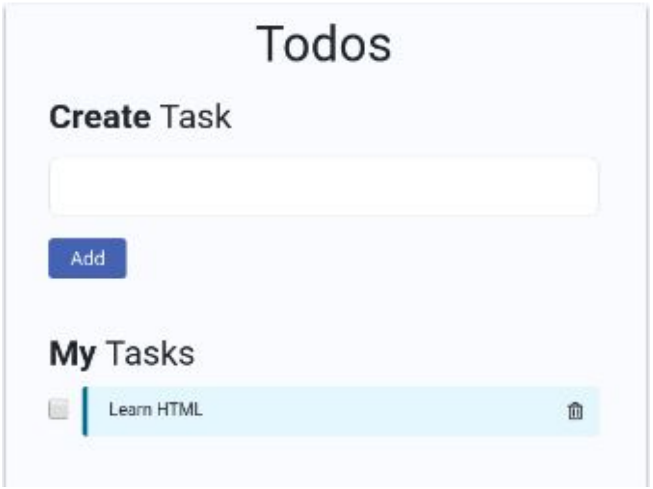


Creating Todo Item

Todo Item

HTML

```html
<ul class="todo-items-container" id="todoItemsContainer">
  <li class="todo-item-container d-flex flex-row">
    <input type="checkbox" id="checkboxInput" class="checkbox-input" />
    <div class="d-flex flex-row label-container">
      <label for="checkboxInput" class="checkbox-label">
        Learn HTML
      </label>
      <div class="delete-icon-container">
        <i class="far fa-trash-alt delete-icon"></i>
      </div>
    </div>
  </li>
</ul>
```

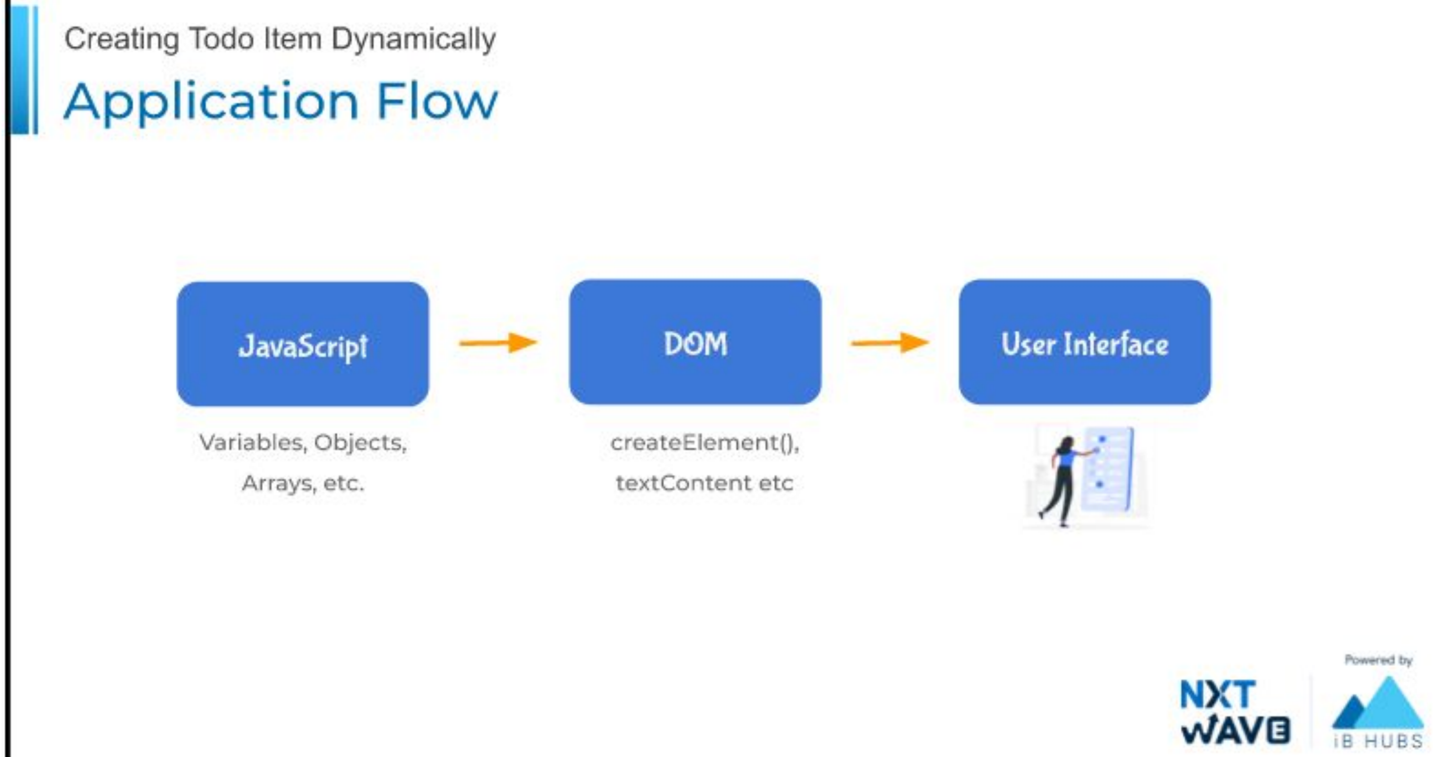

Todos

Create Task

My Tasks

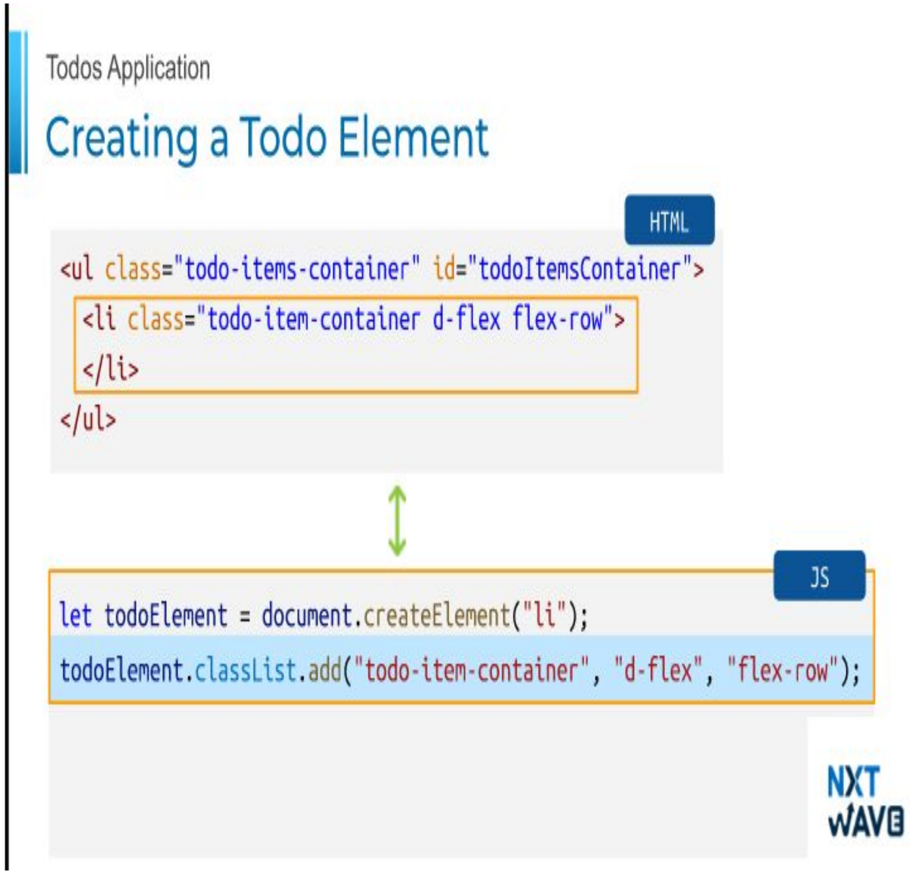
Learn HTML

Lecture:2 Let’s create todo dynamically


Creating Todo Item Dynamically

Application Flow

JavaScript
DOM
User Interface
Variables, Objects, Arrays, etc.
createElement(), textContent etc
Powered by
NXT WAVE
iB HUBS


Todos Application

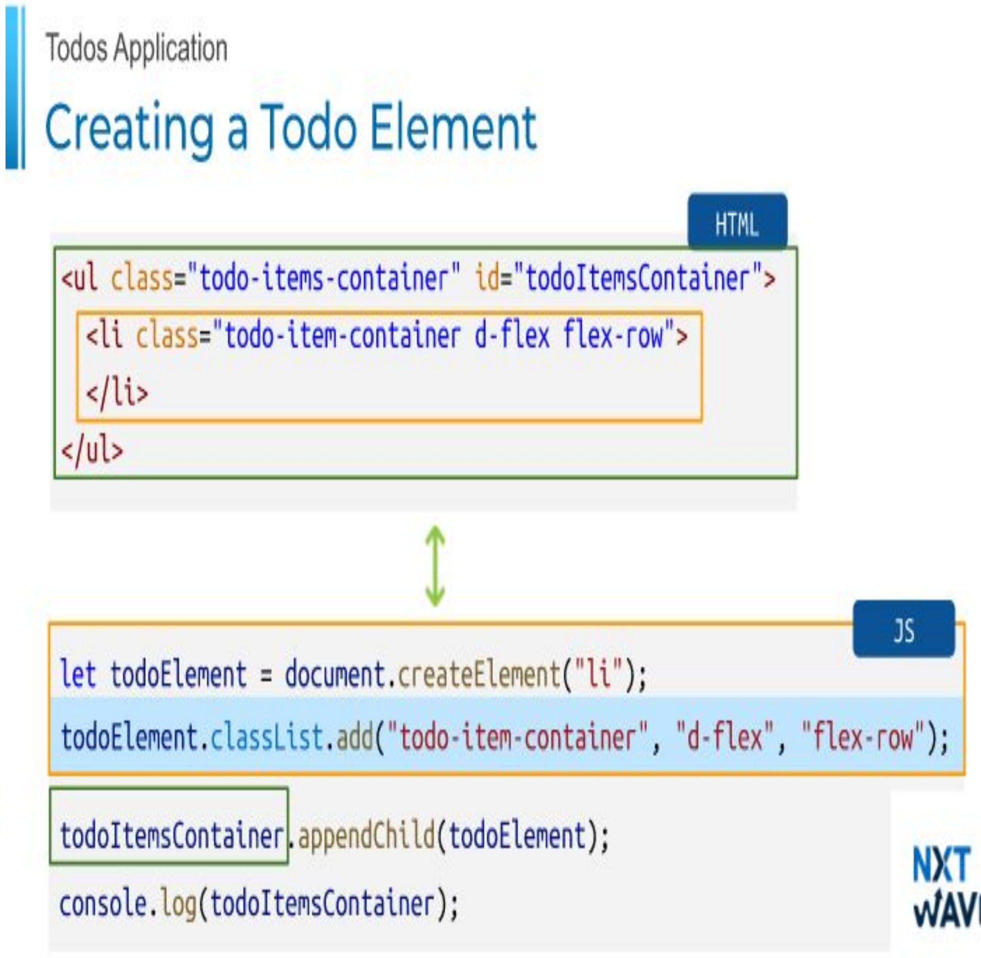
Creating a Todo Element

HTML

```html
<ul class="todo-items-container" id="todoItemsContainer">
  <li class="todo-item-container d-flex flex-row">
  </li>
</ul>
```

JS

```javascript
let todoElement = document.createElement("li");
todoElement.classList.add("todo-item-container", "d-flex", "flex-row");
```


Todos Application

Creating a Todo Element

HTML

```html
<ul class="todo-items-container" id="todoItemsContainer">
  <li class="todo-item-container d-flex flex-row">
  </li>
</ul>
```

JS

```javascript
let todoElement = document.createElement("li");
todoElement.classList.add("todo-item-container", "d-flex", "flex-row");
todoItemsContainer.appendChild(todoElement);
console.log(todoItemsContainer);
```

Todos Application

Creating a Checkbox

HTML

```html
<li class="todo-item-container d-flex flex-row">
  <input type="checkbox" id="checkboxInput" class="checkbox-input" />
</li>
```

JS

```js
let inputElement = document.createElement("input");
inputElement.type = "checkbox";
inputElement.id = "checkboxInput";
inputElement.classList.add("checkbox-input");

todoElement.appendChild(inputElement);
```

Todos Application

Creating a Label Container

HTML

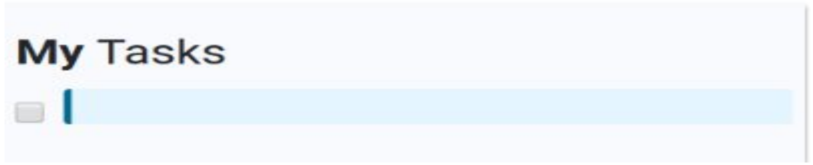
```html
...
<input type="checkbox" id="checkboxInput" class="checkbox-input" />
<div class="d-flex flex-row label-container">
</div>
...
```

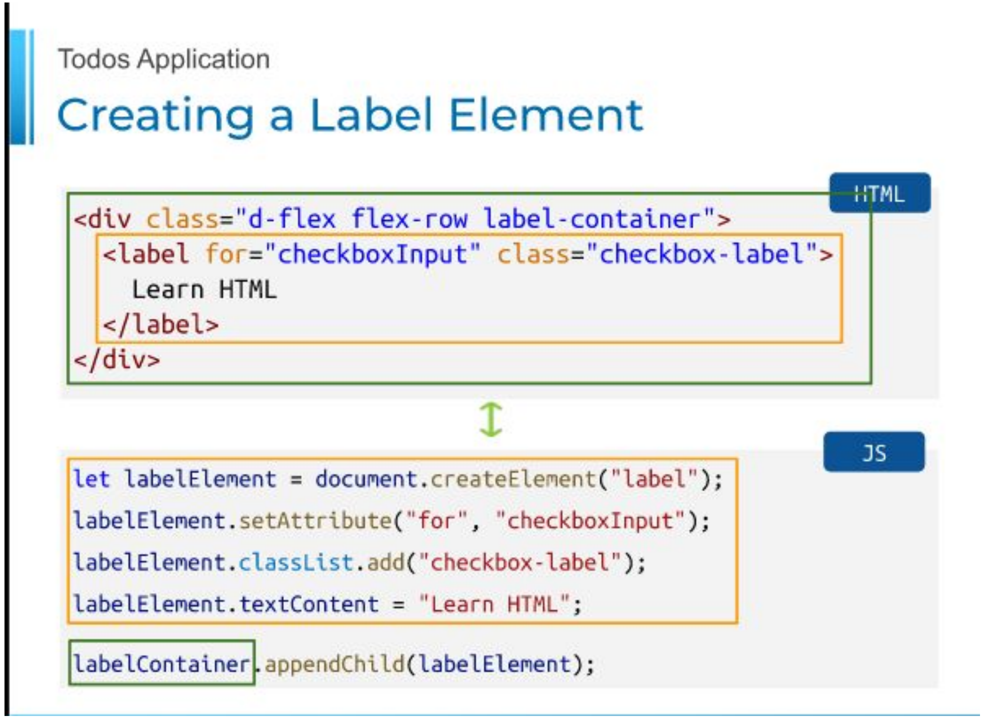
↕

JS

```js
let labelContainer = document.createElement("div");
labelContainer.classList.add("label-container", "d-flex", "flex-row");
todoElement.appendChild(labelContainer);
```

Creating label container



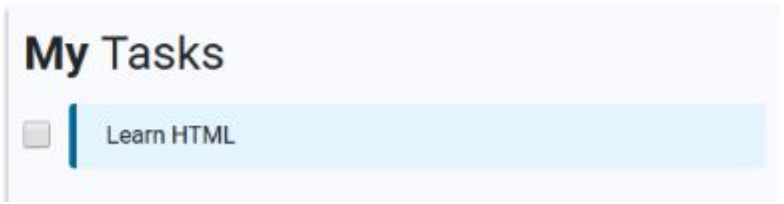

Todos Application

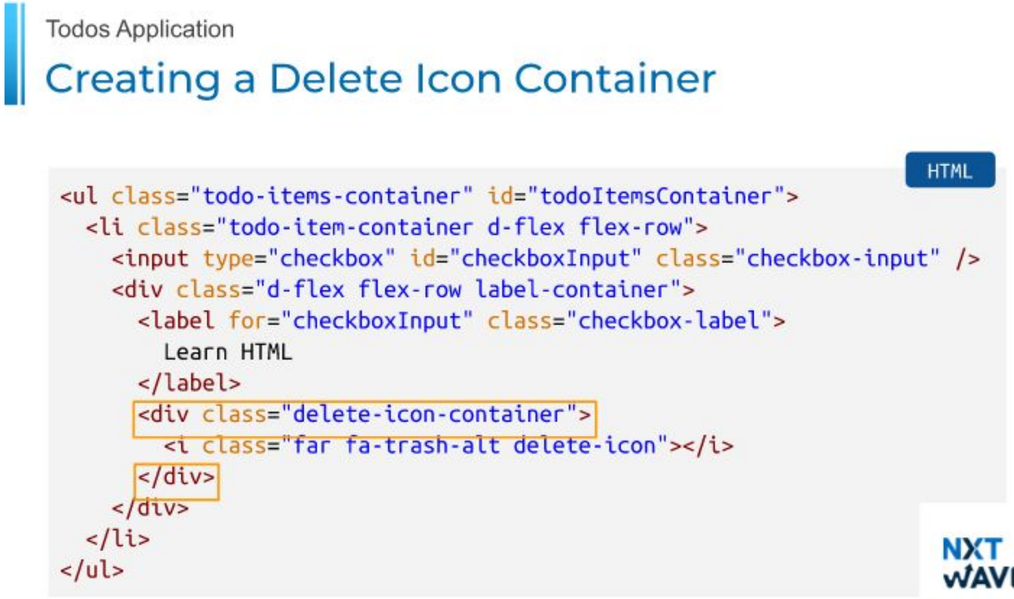
Creating a Label Element

HTML
```
<div class="d-flex flex-row label-container">
  <label for="checkboxInput" class="checkbox-label">
    Learn HTML
  </label>
</div>
```

JS
```
let labelElement = document.createElement("label");
labelElement.setAttribute("for", "checkboxInput");
labelElement.classList.add("checkbox-label");
labelElement.textContent = "Learn HTML";

labelContainer.appendChild(labelElement);
```

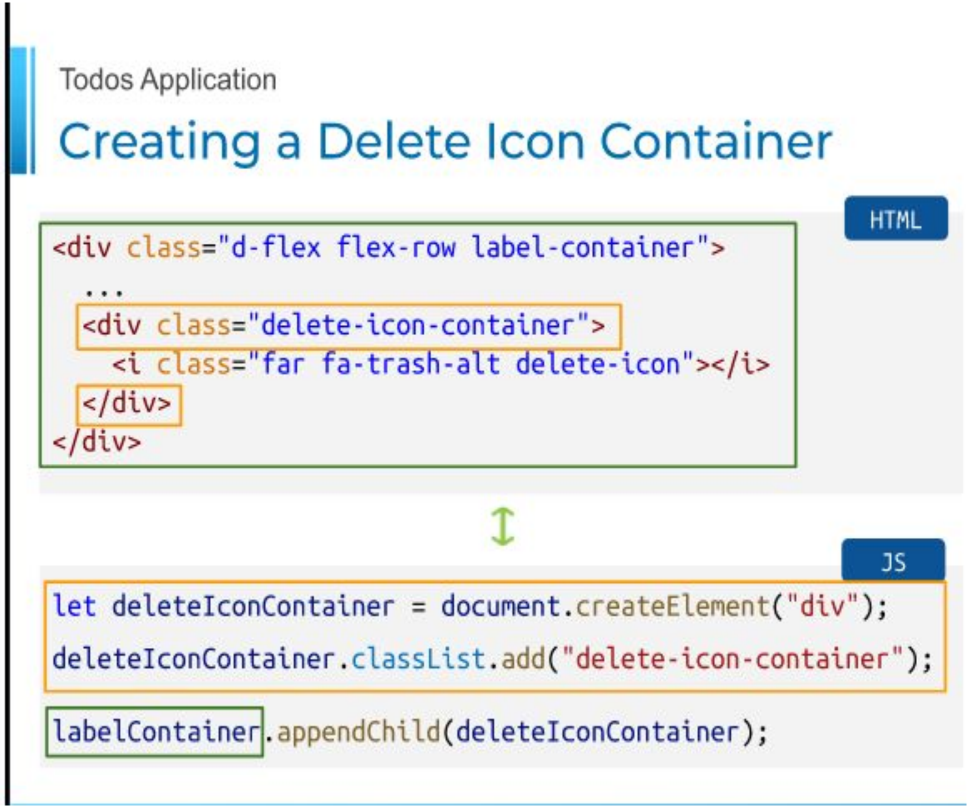
output:




Todos Application

Creating a Delete Icon Container

```html
<ul class="todo-items-container" id="todoItemsContainer">
  <li class="todo-item-container d-flex flex-row">
    <input type="checkbox" id="checkboxInput" class="checkbox-input" />
    <div class="d-flex flex-row label-container">
      <label for="checkboxInput" class="checkbox-label">
        Learn HTML
      </label>
      <div class="delete-icon-container">
        <i class="far fa-trash-alt delete-icon"></i>
      </div>
    </div>
  </li>
</ul>
```



Todos Application

Creating a Delete Icon Container

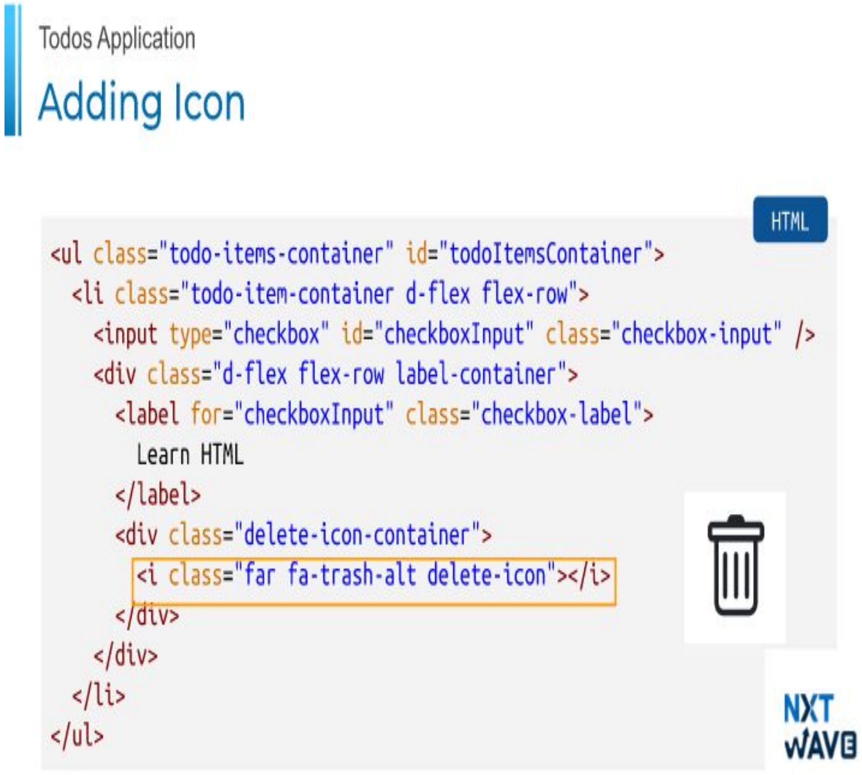
HTML

```html
<div class="d-flex flex-row label-container">
  ...
  <div class="delete-icon-container">
    <i class="far fa-trash-alt delete-icon"></i>
  </div>
</div>
```

↕

JS

```js
let deleteIconContainer = document.createElement("div");
deleteIconContainer.classList.add("delete-icon-container");
labelContainer.appendChild(deleteIconContainer);
```



Todos Application

Adding Icon

HTML

```html
<ul class="todo-items-container" id="todoItemsContainer">
  <li class="todo-item-container d-flex flex-row">
    <input type="checkbox" id="checkboxInput" class="checkbox-input" />
    <div class="d-flex flex-row label-container">
      <label for="checkboxInput" class="checkbox-label">
        Learn HTML
      </label>
      <div class="delete-icon-container">
        <i class="far fa-trash-alt delete-icon"></i>
      </div>
    </div>
  </li>
</ul>
```

NXT WAVE

Todos Application

Adding Icon

HTML

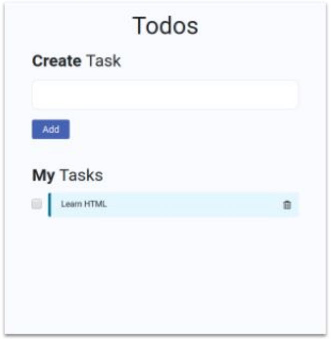
```html
<div class="d-flex flex-row label-container">
  ...
  <div class="delete-icon-container">
    <i class="far fa-trash-alt delete-icon"></i>
  </div>
</div>
```

JS

```js
let deleteIcon = document.createElement("i");
deleteIcon.classList.add("far", "fa-trash-alt", "delete-icon");

deleteIconContainer.appendChild(deleteIcon);
```

Adding Icon

Output



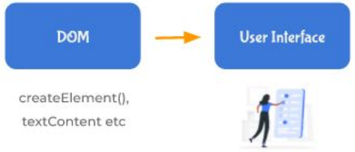
Todos

Create Task

Add

My Tasks

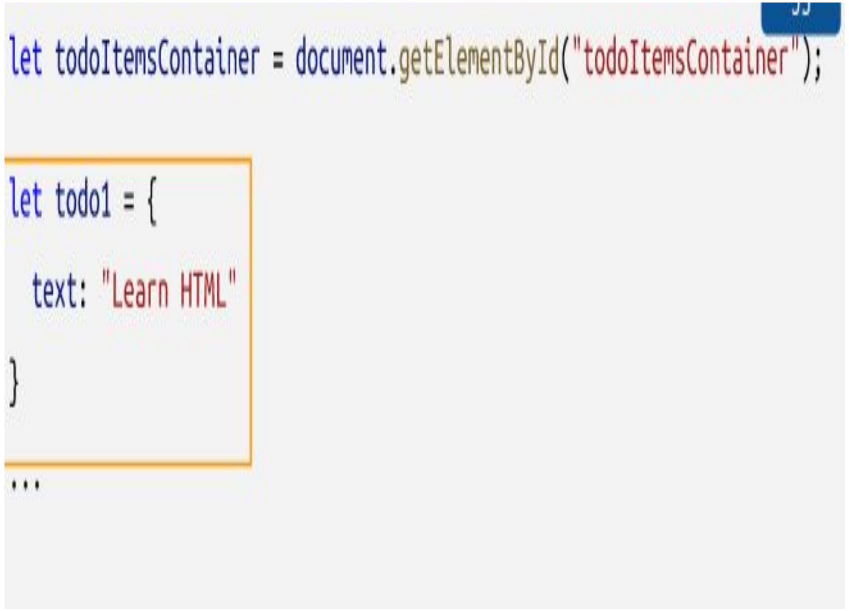
Learn HTML



Powered by NXT WAVE iB HUBS

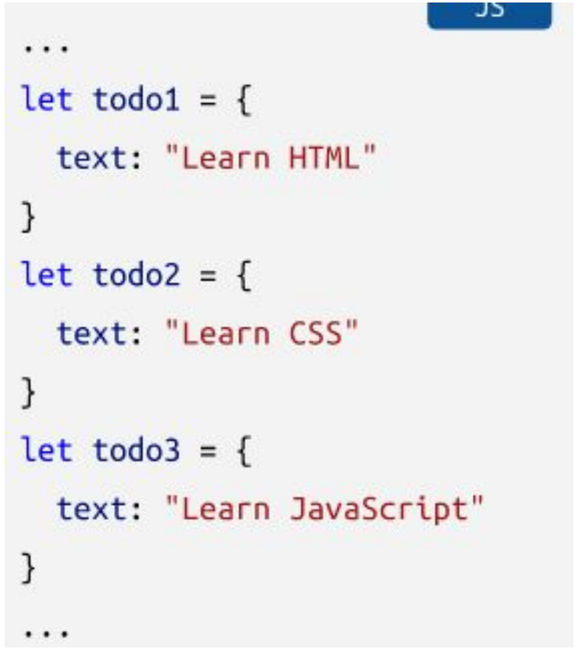
Creating single todo and multiple todo

Single todo



```
let todoItemsContainer = document.getElementById("todoItemsContainer");

let todo1 = {
  text: "Learn HTML"
}
...
```

Multiple todo



```
...
let todo1 = {
  text: "Learn HTML"
}
let todo2 = {
  text: "Learn CSS"
}
let todo3 = {
  text: "Learn JavaScript"
}
...
```

Creating List of Todo Objects

JS

```
...
let todoList = [
  {
    text: "Learn HTML"
  },
  {
    text: "Learn CSS"
  },
  {
    text: "Learn JavaScript"
  }
];
...
```

Creating Reusable Function

JS

```
...
function createAndAppendTodo() {
  let todoElement = document.createElement("li");
  ...
  ...
  let deleteIcon = document.createElement("i");
  deleteIcon.classList.add("far", "fa-trash-alt", "delete-icon");
  deleteIconContainer.appendChild(deleteIcon);
}
```

Making Corresponding Changes

JS

```
...
function createAndAppendTodo(todo) {
  ...
  let labelElement = document.createElement("label");
  labelElement.setAttribute("for", "checkboxInput");
  labelElement.classList.add("checkbox-label");
  labelElement.textContent = todo.text;
  labelContainer.appendChild(labelElement);
  ...
}
```

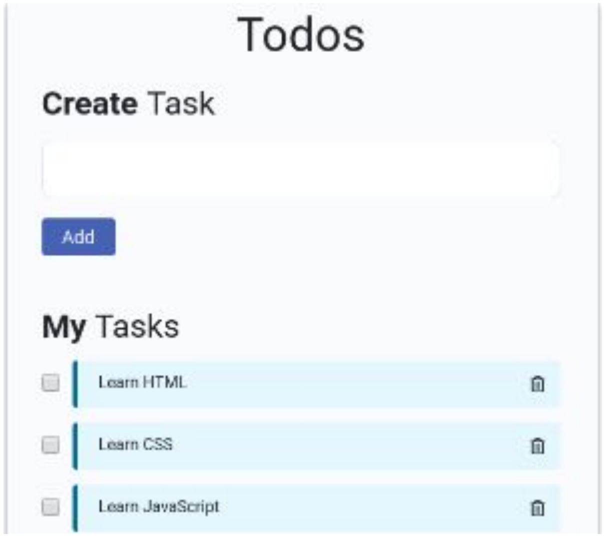
Creating List of Todo Objects

JS

```
...
let todoList = [
  {
    text: "Learn HTML"
  },
  {
    text: "Learn CSS"
  },
  {
    text: "Learn JavaScript"
  }
];
...
```

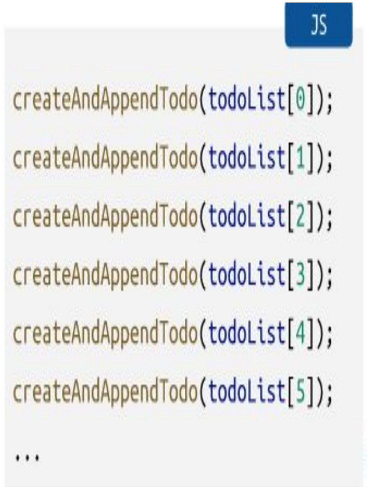
```
...
createAndAppendTodo(todoList[0]);
createAndAppendTodo(todoList[1]);
createAndAppendTodo(todoList[2]);
```

output:



Can we minimize

Duplication of code?



```
createAndAppendTodo(todoList[0]);
createAndAppendTodo(todoList[1]);
createAndAppendTodo(todoList[2]);
createAndAppendTodo(todoList[3]);
createAndAppendTodo(todoList[4]);
createAndAppendTodo(todoList[5]);
...
```

Introduction to Loops

Loops

Loops allow us to **execute** a block of code **several times**

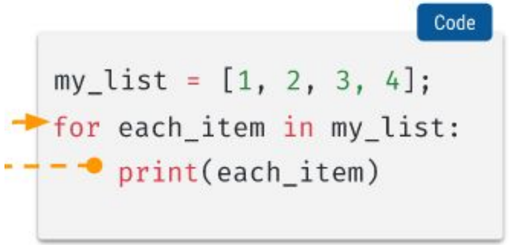
- **for...of Loop**
- for...in Loop
- for Loop
- while Loop

many more...

Loops

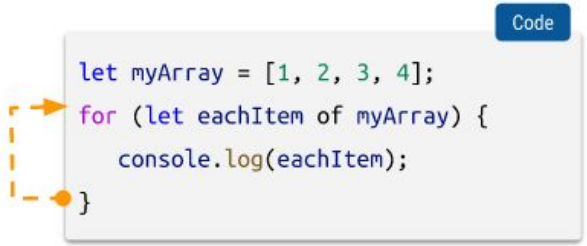
The for...of Loop

Python



```
my_list = [1, 2, 3, 4];
for each_item in my_list:
    print(each_item)
```

JavaScript



```
let myArray = [1, 2, 3, 4];
for (let eachItem of myArray) {
   console.log(eachItem);
}
```

Creating Multiple Todo Items

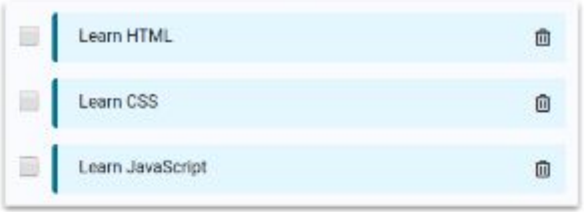
Adding Todo Items to Todos List

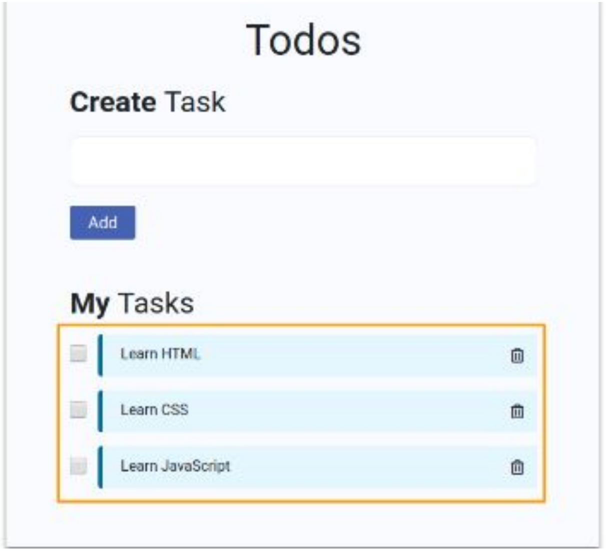
JS

```
[
  {
    text: "Learn HTML"
  },
  {
    text: "Learn CSS"
  },
  {
    text: "Learn JavaScript"
  }
];
```

JS

```
for (let todo of todoList) {
  createAndAppendTodo(todo);
}
```





Todos Application

Features

- ✓ Creating **Single** Todo Item
- ✓ Creating **Multiple** Todo Items
- ☐ Taking User Input and creating Todos **Dynamically**
- ☐ **Checking** a Todo
- ☐ **Deleting** a Todo
- ☐ **Persisting** Todos On Reload using Local Storage

On-Demand Session | Cheat Sheet

1. Most Commonly Made Mistakes

1.1 Most of the JS properties and methods should be in the Camel case.

Most of the JS properties and methods are in the Camel case (the starting letter of each word should be in uppercase except for the first word).

Code	Mistake	Correct Syntax
document.CreateElement()	C in Uppercase	document.createElement()
document.getElementbyId()	b in Lowercase	document.getElementById()
element.textcontent	c in Lowercase	element.textContent
element.classlist.add()	l in Lowercase	element.classList.add()

1.2 The ID should be the same in both the HTML and JS.

1.2.1 Mistake:

HTML

```html
<h1 id="heading">Shopping List</h1>
```

JAVASCRIPT

```javascript
let headingEl = document.getElementById("listHeading");
headingEl.textContent = "Items Needed";
```

In the above Code Snippets, the HTML element's text content doesn't change because the ID used in HTML and JS are different.

So, While accessing an HTML element with the ID using JS, **the ID used in the HTML element and the `document.getElementById` method must be the same**.

HTML

```html
<h1 id="heading">Shopping List</h1>
```

JAVASCRIPT

```javascript
let headingEl = document.getElementById("heading");
headingEl.textContent = "Items Needed";
```

1.2.2 Mistake:

HTML

```html
<h1 id="listHeading ">Shopping List</h1>
```

JAVASCRIPT

```javascript
let headingEl = document.getElementById("listHeading");
headingEl.textContent = "Items Needed";
```

The HTML element's text content doesn't change because there is an extra space at the end of the ID in the HTML code.

So, there shouldn't be any extra spaces in the IDs used in both the HTML and JS.

HTML

```html
<h1 id="listHeading">Shopping List</h1>
```

JAVASCRIPT

```javascript
let headingEl = document.getElementById("listHeading");
headingEl.textContent = "Items Needed";
```

1.3. The Function name must be the same in both the Function declaration and the Function call.

1.3.1 Mistake:

JAVASCRIPT

```javascript
function greeting() {
  let message = "Hello Rahul";
  console.log(message);
}

greet();
```

As there is no function called `greet` , we will get an error in the above Code Snippet.

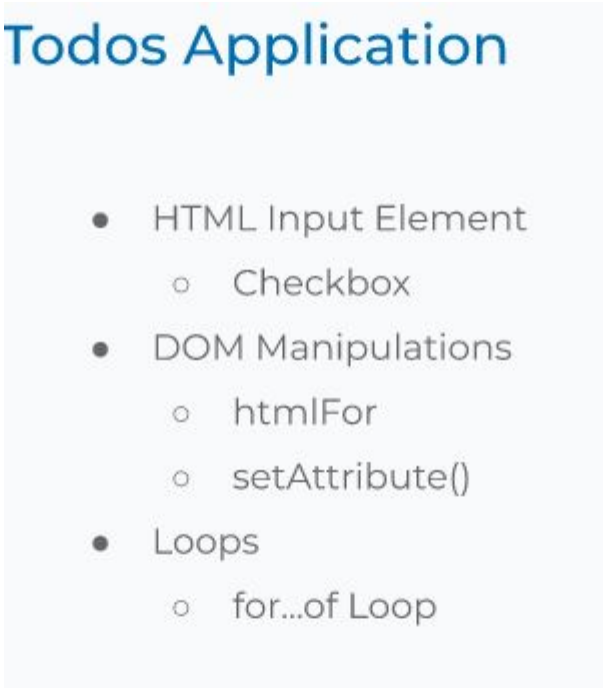
So, while calling a function, you must use the same function name used in the function declaration.

JAVASCRIPT

```javascript
function greeting() {
  let message = "Hello Rahul";
  console.log(message);
}

greeting();
```

Lecture :3 todo

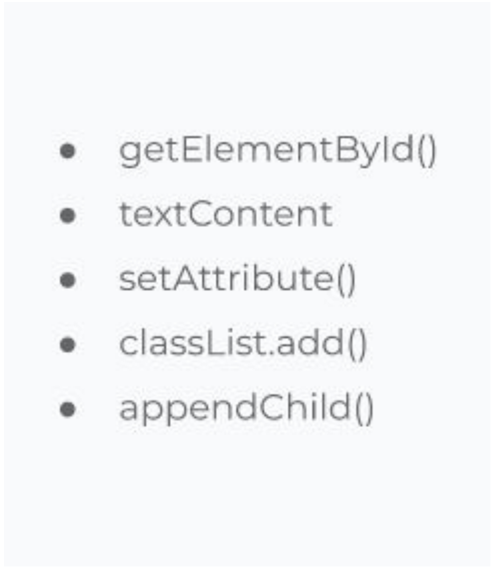
Recap lectures



Todos Application

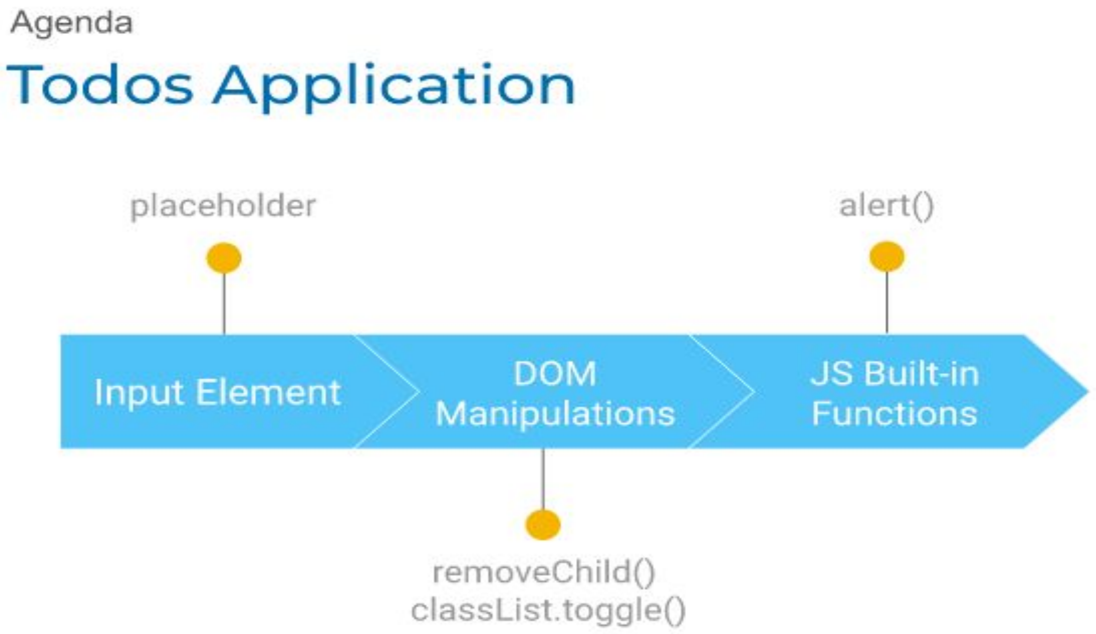
- HTML Input Element
  - Checkbox
- DOM Manipulations
  - htmlFor
  - setAttribute()
- Loops
  - for...of Loop

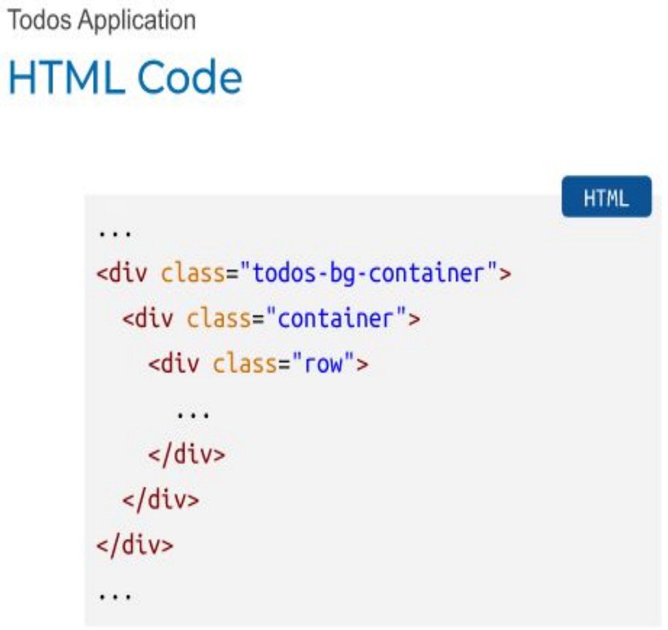
Recap of lectures



- getElementById()
- textContent
- setAttribute()
- classList.add()
- appendChild()

Lecture3 :agenda




Todos Application

HTML Code

HTML

```html
...
<div class="todos-bg-container">
  <div class="container">
    <div class="row">
      ...
    </div>
  </div>
</div>
...
```

Todos Application

CSS Code

CSS

```
.todos-bg-container {
  background-color: #f9fbfe;
  height: 100vh;
}
.todos-heading {
  text-align: center;
  font-family: "Roboto";
  font-size: 46px;
  ...
}
...
```

JS Code

JS

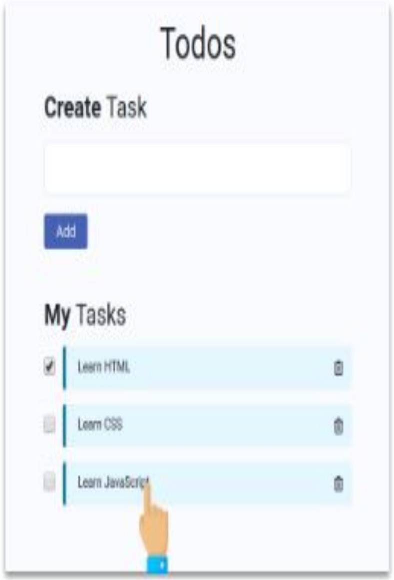
```
let todoItemsContainer = document.getElementById("todoItemsContainer");
let todoList = [
  {
    text: "Learn HTML",
  },
  ...
];
function createAndAppendTodo(todo) {
  let todoElement = document.createElement("li");
  ...
  deleteIconContainer.appendChild(deleteIcon);
}
...
```

Todos Application

Fixing Checkbox Issue

We have to specify a **Unique ID** to each Checkbox.

Provide the same ID to the labels **for** attribute.



NXT WAVE

Fixing Checkbox Issue

Specifying Unique ID

JS

```
let todoList = [
  {
    text: "Learn HTML",
    uniqueNo: 1
  },
  {
    text: "Learn CSS",
    uniqueNo: 2
  },
  {
    text: "Learn JavaScript",
    uniqueNo: 3
  }
];
```

Fixing Checkbox Issue

Adding ID to each Checkbox

JS

```js
...
function createAndAppendTodo(todo) {
  let checkboxId = "checkbox" + todo.uniqueNo;
  ...
  inputElement.type = "checkbox";
  inputElement.id = checkboxId;
  ...
  labelElement.setAttribute("for", checkboxId);
  labelElement.classList.add("checkbox-label");
  ...
}
...
```

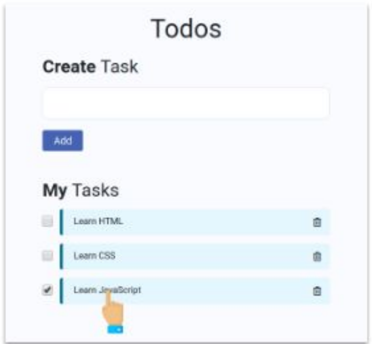
"checkbox" + 1 ⟶ "checkbox1"

"checkbox" + 2 ⟶ "checkbox2"

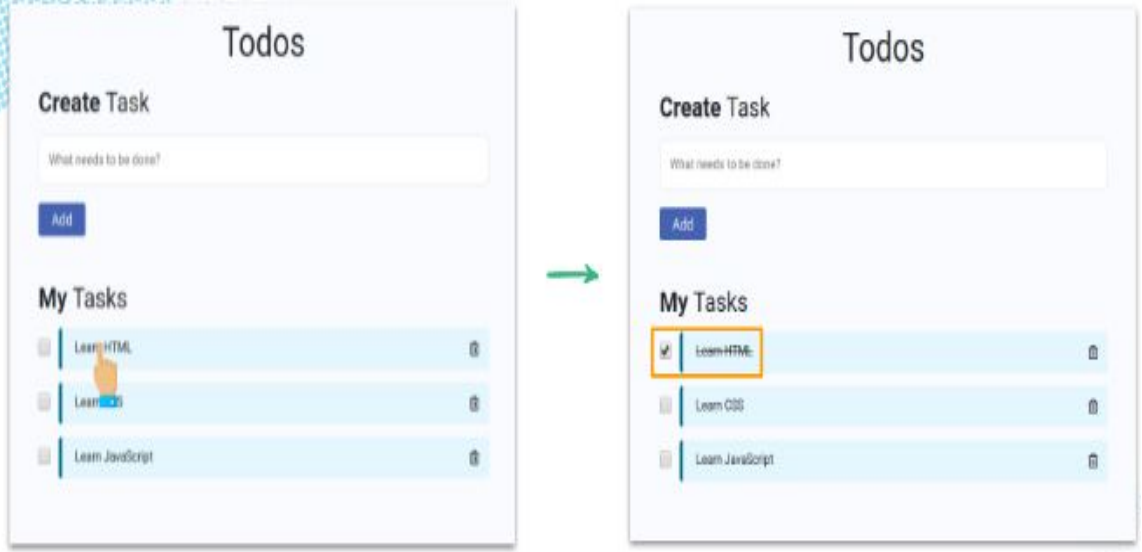
"checkbox" + 3 ⟶ "checkbox3"

Fixing Checkbox Issue

Output



How to Strikethrough the Label Text on Checking Checkbox?



Todos Application

Strikethrough the Label Text

Steps:

- Add Required CSS to Strike a Text
- Specify Unique Id to each Label Element
- Add Event Listeners to the Checkboxes
- Change Styles of Label Element based on Checkbox Check

Step 1

Adding Required CSS to Strike a Text

CSS

```css
.checked {
  text-decoration: line-through;
}
```

Step 3

Adding Event Listeners to Checkboxes

JS

```js
...
function createAndAppendTodo(todo) {
  let checkboxId = "checkbox" + todo.uniqueNo;
  let labelId = "label" + todo.uniqueNo;
  ...
  let inputElement = document.createElement("input");
  inputElement.type = "checkbox";
  inputElement.id = checkboxId;
  inputElement.onclick = function () {
    onTodoStatusChange(checkboxId, labelId);
  };
  ...
}
```

Step 2

Specifying ID to each Label Element

JS

```js
...
function createAndAppendTodo(todo) {
  let checkboxId = "checkbox" + todo.uniqueNo;
  let labelId = "label" + todo.uniqueNo;
  ...
  let labelElement = document.createElement("label");
  labelElement.setAttribute("for", checkboxId);
  labelElement.id = labelId;
}
...
```

"label" + 1 ⟶ "label1"

"label" + 2 ⟶ "label2"

"label" + 3 ⟶ "label3"

NXT WAVE Powered by iB HUBS

Step 4 - Change Styles of Label Element based on Checkbox Check

Accessing Label Elements

JS

```
...
function onTodoStatusChange(checkboxId, labelId) {
  let checkboxElement = document.getElementById(checkboxId);
  console.log(checkboxElement.checked);

  let labelElement = document.getElementById(labelId);
}
...
```

Step 4

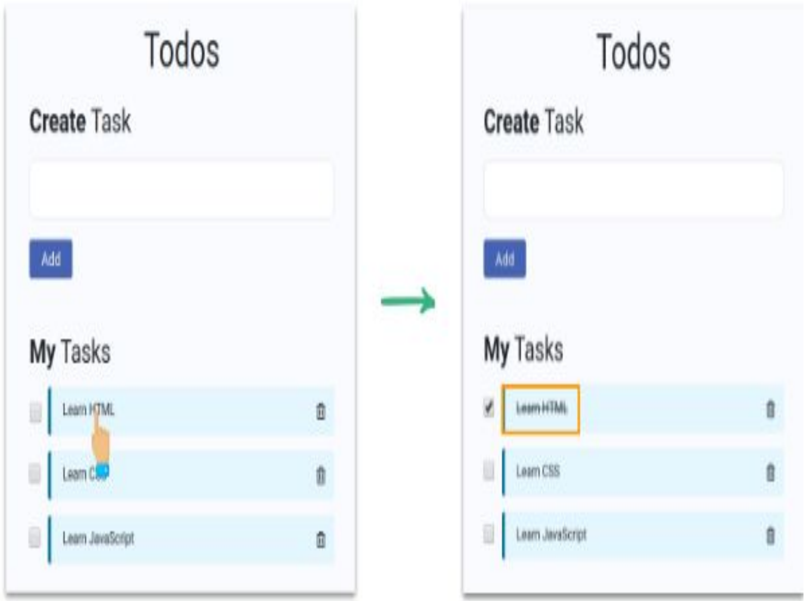
Changing Label Element Styles

JS

```
...
function onTodoStatusChange(checkboxId, labelId) {
  ...
  if (checkboxElement.checked === true) {
    labelElement.classList.add("checked");
  }
  else {
    labelElement.classList.remove("checked");
  }
}
```

Change styles of Label Element based on Checkbox Check

Output



DOM Manipulations

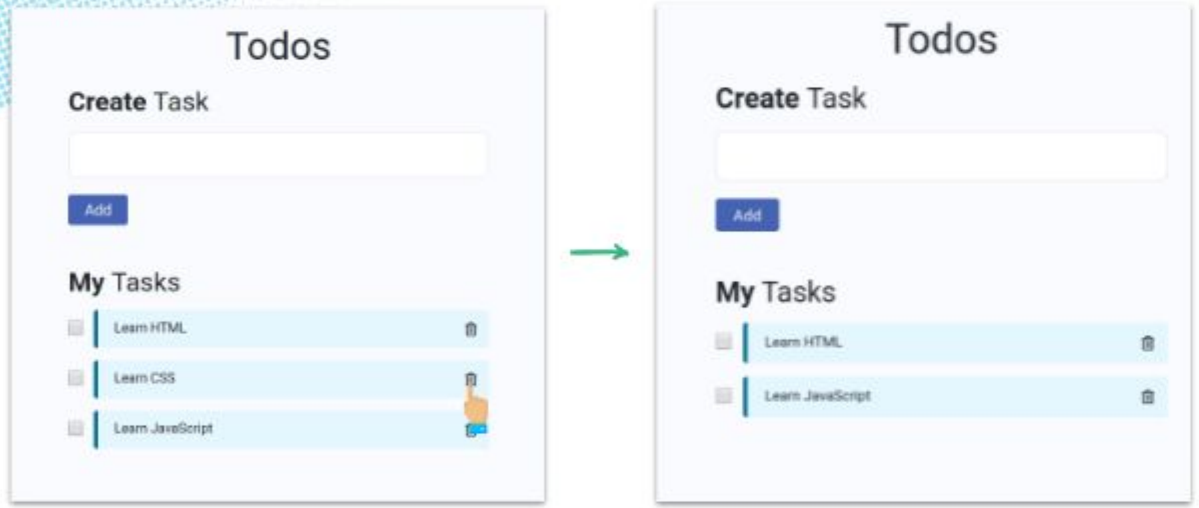
classList.toggle()

Toggle Button

JS

```
...
function onTodoStatusChange(checkboxId, labelId) {
  ...
  labelElement.classList.toggle("checked");
}
```

How to Delete a Todo Item?



Todos Application

Deleting a Todo Item

Steps:

- Specify ID to each Todo Item
- Add Event Listeners to Delete Icon
- Delete Todo Item from the Todo Items Container

Step 1

Specifying ID to each Todo Item

JS

```js
...
function createAndAppendTodo(todo) {
  let todoId = "todo" + todo.uniqueNo;
  ...
  let todoElement = document.createElement("li");
  todoElement.classList.add("todo-item-container", "d-flex", "flex-row");
  todoElement.id = todoId;
  todoItemsContainer.appendChild(todoElement);
  ...
}
```

Step 2

Adding Event Listeners to Delete Icon

JS

```js
...
let deleteIcon = document.createElement("i");
deleteIcon.classList.add("far", "fa-trash-alt", "delete-icon");
deleteIcon.onclick = function () {
  onDeleteTodo(todoId);
};
deleteIconContainer.appendChild(deleteIcon);
...
```

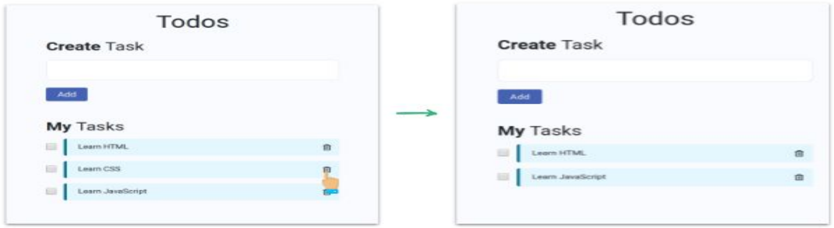
Step 3

Deleting a Todo Item

```js
...
function onDeleteTodo(todoId) {
  let todoElement = document.getElementById(todoId);
  todoItemsContainer.removeChild(todoElement);
}
...
```

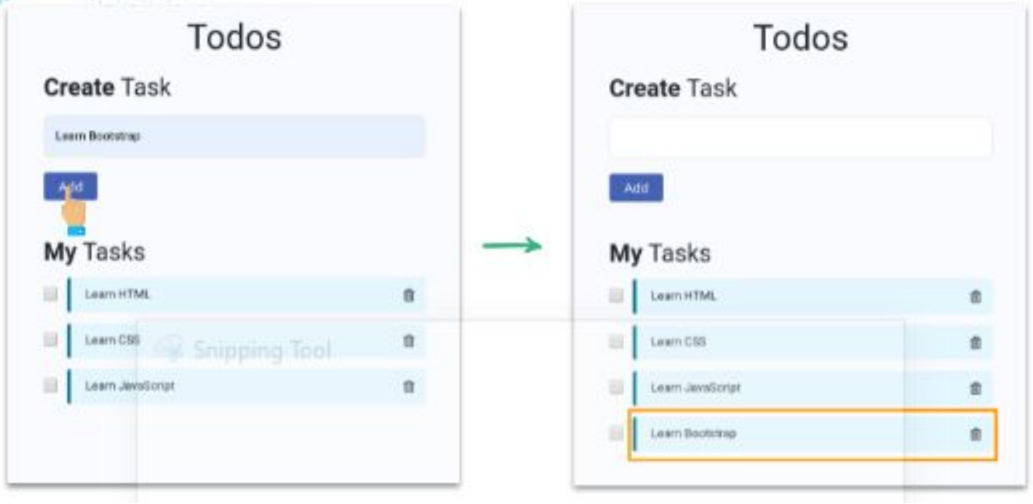
Deleting a Todo Item

Output



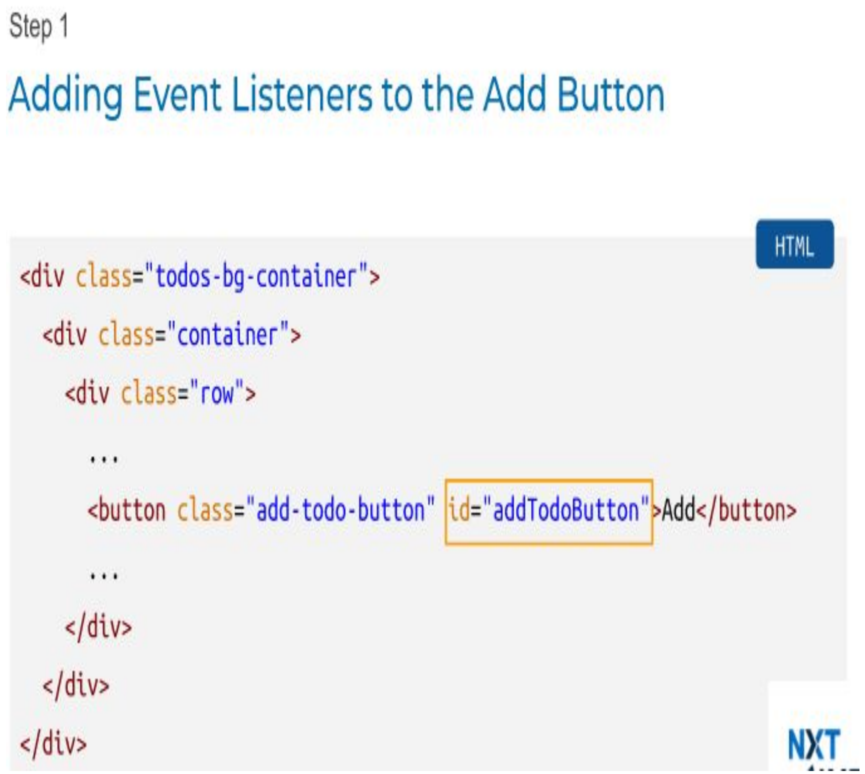
Adding a Todo Item

How to Add a new Todo Item on User Input?



Steps:

- Add Event Listener to the Add Button
- Access User Input Value
- Create New Todo Item

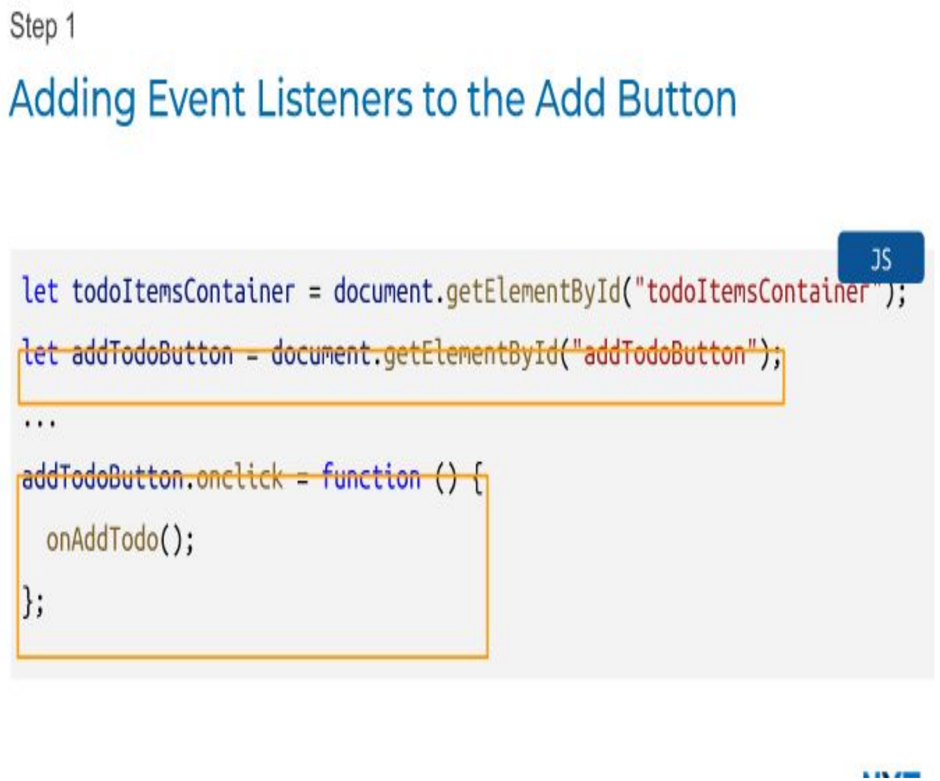


Step 1

Adding Event Listeners to the Add Button

HTML

```html
<div class="todos-bg-container">
  <div class="container">
    <div class="row">
      ...
      <button class="add-todo-button" id="addTodoButton">Add</button>
      ...
    </div>
  </div>
</div>
```



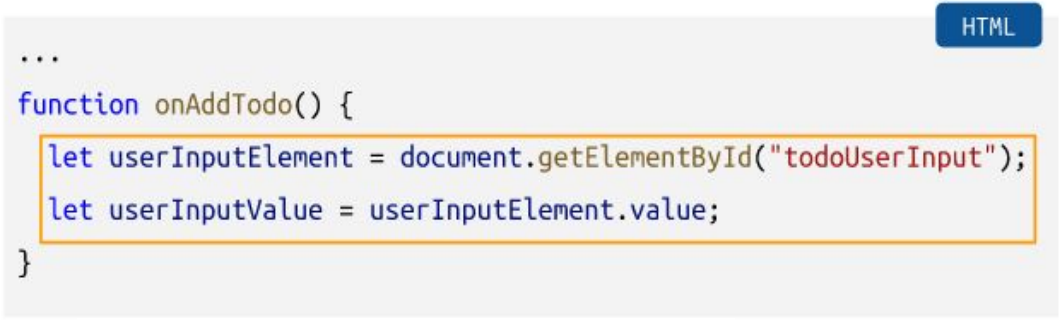
Step 1

Adding Event Listeners to the Add Button

JS

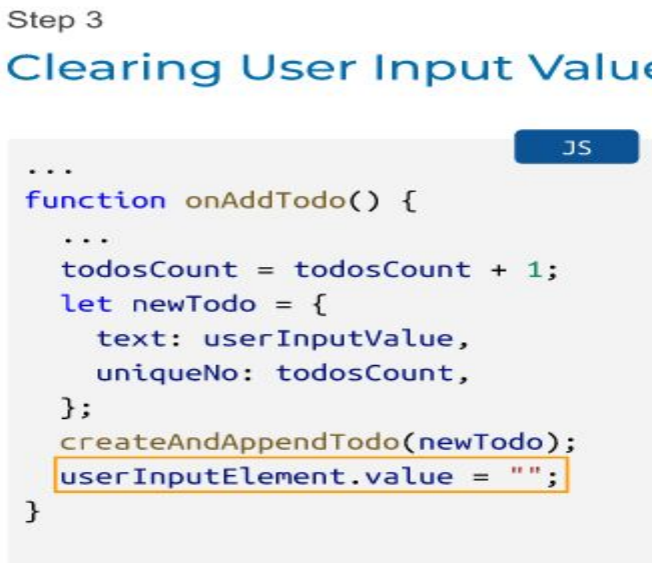
```javascript
let todoItemsContainer = document.getElementById("todoItemsContainer");
let addTodoButton = document.getElementById("addTodoButton");
...
addTodoButton.onclick = function () {
  onAddTodo();
};
```

Accessing user input:



HTML

```
...
function onAddTodo() {
  let userInputElement = document.getElementById("todoUserInput");
  let userInputValue = userInputElement.value;
}
```

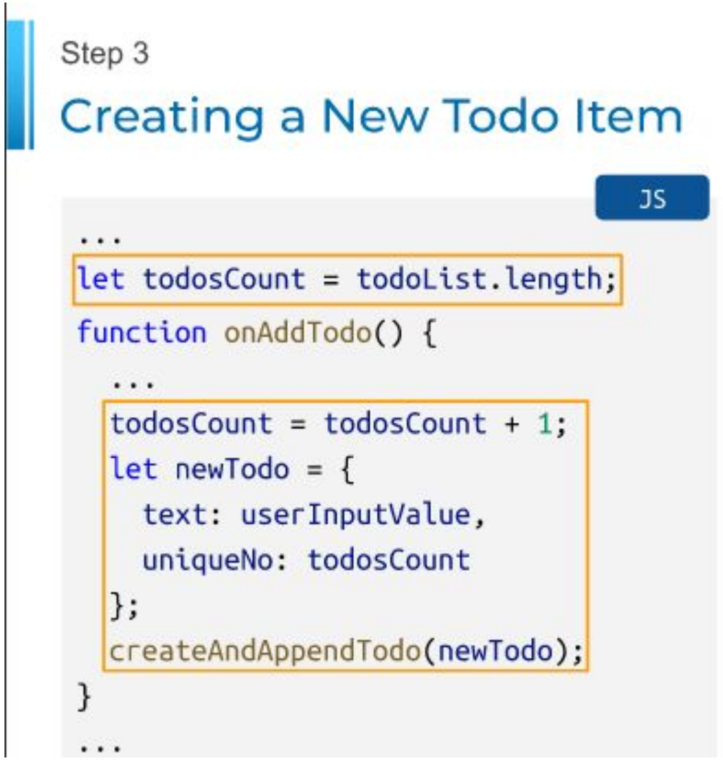


Step 3

Clearing User Input Valu

JS

```
...
function onAddTodo() {
  ...
  todosCount = todosCount + 1;
  let newTodo = {
    text: userInputValue,
    uniqueNo: todosCount,
  };
  createAndAppendTodo(newTodo);
  userInputElement.value = "";
}
```



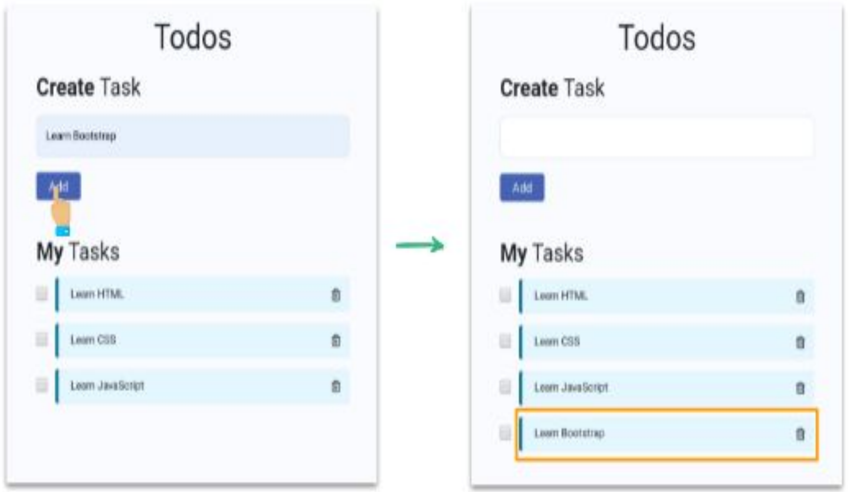
Step 3

Creating a New Todo Item

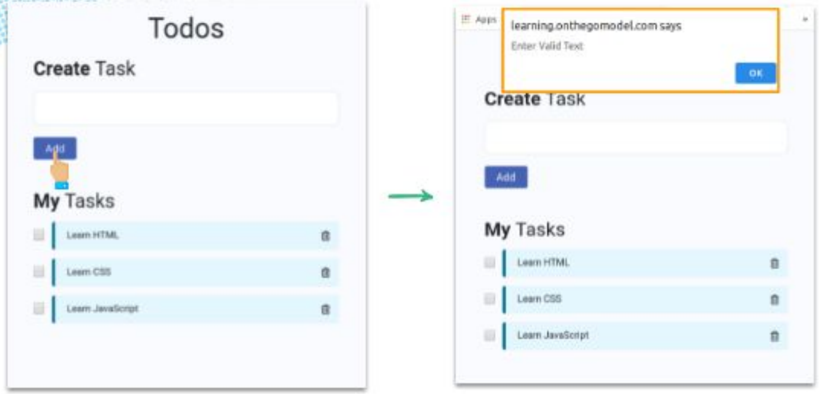
JS

```
...
let todosCount = todoList.length;
function onAddTodo() {
  ...
  todosCount = todosCount + 1;
  let newTodo = {
    text: userInputValue,
    uniqueNo: todosCount
  };
  createAndAppendTodo(newTodo);
}
...
```

Output



How to show a Warning Message on Invalid User Input?



learning.onthegomodel.com says

Enter Valid Text

Todos Application

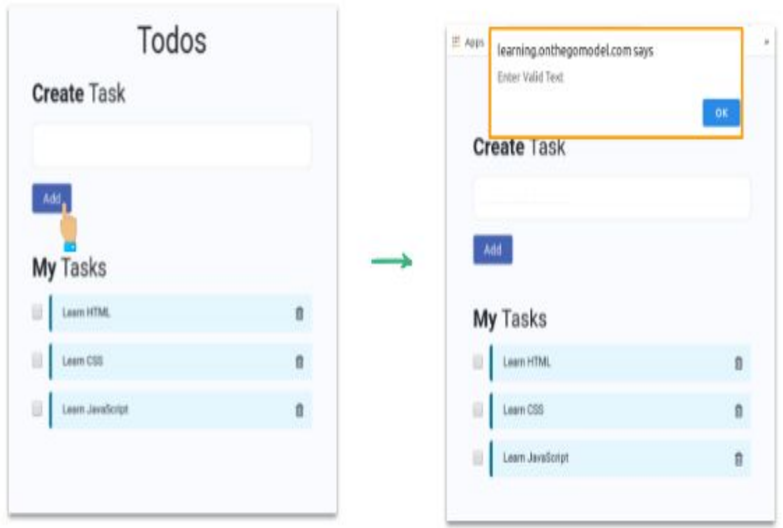
Showing Warning Message

JS

```
...
function onAddTodo() {
  let userInputElement = document.getElementById("todoUserInput");
  let userInputValue = userInputElement.value;
  if (userInputValue === "") {
    alert("Enter Valid Text");
    return;
  }
  ...
}
```

Showing Warning Message

Output



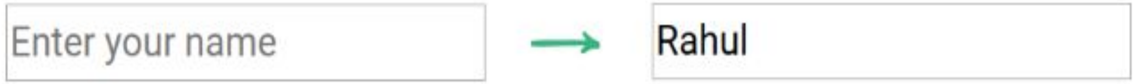
learning.onthegomodel.com says

Enter Valid Text

Adding Placeholder Text

HTML

```
<body>
  <input type="text" placeholder="Enter your name" />
</body>
```



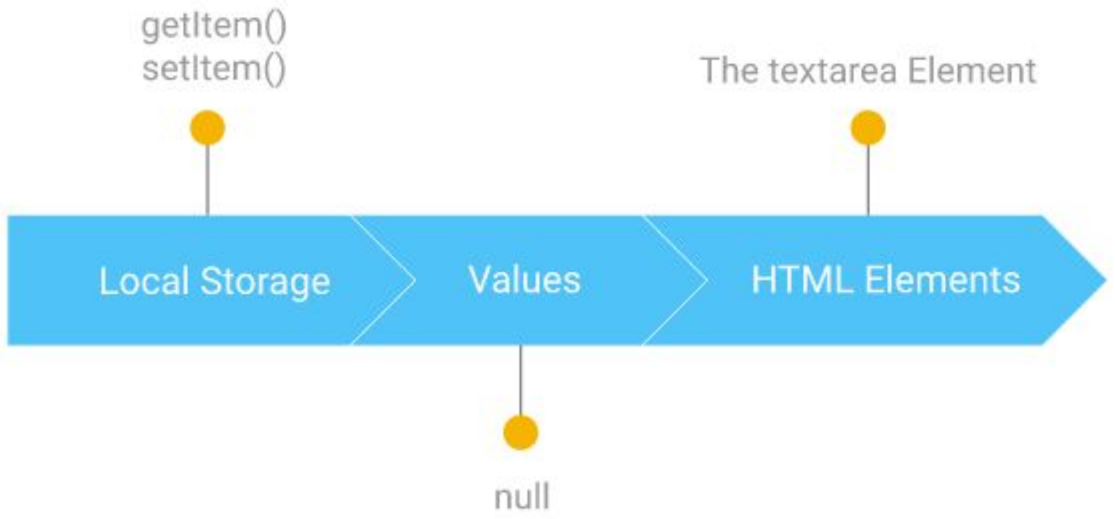
Key Takeaways

- Input Element
  - placeholder
- DOM Manipulations
  - classList.toggle()
  - removeChild()
- JS Built-in Functions
  - alert()

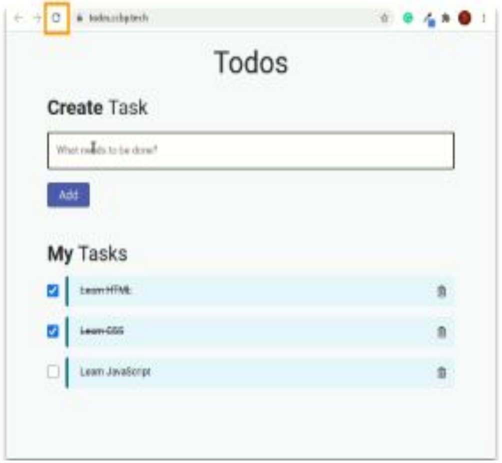
lecture:4

Agenda

Todos Application



What happens to Todo List when we reload the application?


todos.ccbp.tech

Todos

Create Task

What needs to be done?

Add

My Tasks

- [x] ~~Learn HTML~~
- [x] ~~Learn CSS~~
- [ ] Learn JavaScript

Todos Application

Execution Context

The environment in which JS Code runs is called Execution Context

Execution context contains all the variables, objects, and functions

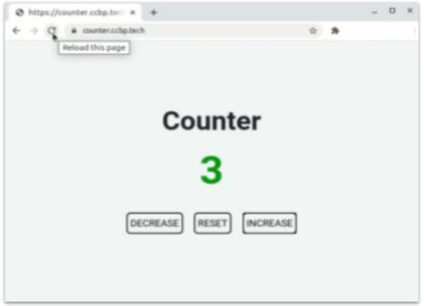
Variables

Objects

Functions

Example

Counter Application


https://counter.ccbp.tech

counter.ccbp.tech

Reload this page

Counter

3

DECREASE RESET INCREASE

Execution Context

On Reloading

Execution Context is destroyed and recreated whenever we reload an Application.

Striked aswell as deleted an
stored element will be delete

How to persist todo items on reload?

Persisting Data even on Reload

Storage Mechanisms

Client-Side Data Storage

- Storing Data on the Client (user's machine)

Server-Side Data Storage

- Storing Data on the Server using some kind of Database

Storage Mechanisms

Client-Side Data Storage Mechanisms

- Local Storage
- Session Storage
- Cookies
- IndexedDB

many more...

Local Storage

It allows web applications
to store **data locally**
within the **User's Browser**

It is a Storage **Object**

Data can be stored in the form of
key-value pairs

Value provided should always be a **string**

Key	Value
name	Rahul
gender	Male
city	Delhi

Client-Side Data Storage Mechanisms

Local Storage

To access and work with Local Storage, we have below methods:

- setItem()
- getItem()
- clear()
- removeItem()

Storing Data in Local Storage

setItem()

Syntax:

```
localStorage.setItem("Key", "Value");
```

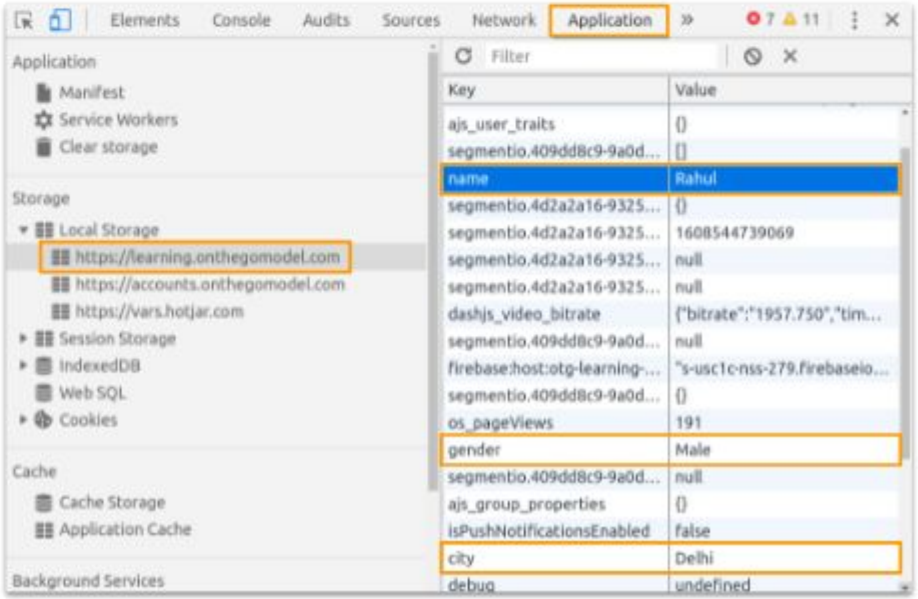
JS

```
localStorage.setItem("name", "Rahul");
localStorage.setItem("gender", "Male");
localStorage.setItem("city", "Delhi");
```

Key	Value
name	Rahul
gender	Male
city	Delhi

Storing Data in Local Storage

setItem()



Key	Value
ajs_user_traits	{}
segmentio.409dd8c9-9a0d...	[]
name	Rahul
segmentio.4d2a2a16-9325...	{}
segmentio.4d2a2a16-9325...	1608544739069
segmentio.4d2a2a16-9325...	null
segmentio.4d2a2a16-9325...	null
dashjs_video_bitrate	{"bitrate":"1957.750","tim...
segmentio.409dd8c9-9a0d...	null
firebase:host:otg-learning-...	"s-usc1c-nss-279.firebaseio...
segmentio.409dd8c9-9a0d...	{}
os_pageViews	191
gender	Male
segmentio.409dd8c9-9a0d...	null
ajs_group_properties	{}
isPushNotificationsEnabled	false
city	Delhi
debug	undefined

Getting Data from Local Storage

getItem()

Syntax:

```
localStorage.getItem("Key");
```

JS

```js
let name = localStorage.getItem("name");
let gender = localStorage.getItem("gender");
let city = localStorage.getItem("city");

console.log(name);
console.log(gender);
console.log(city);
```

Output

```
Rahul
Male
Delhi
```

Getting Data from Local Storage

getItem()

JS

```
let occupation = localStorage.getItem("occupation");
console.log(occupation);
```

Output

```
null
```

Values

null

We use null in situations where we intentionally want a variable to not have a value yet

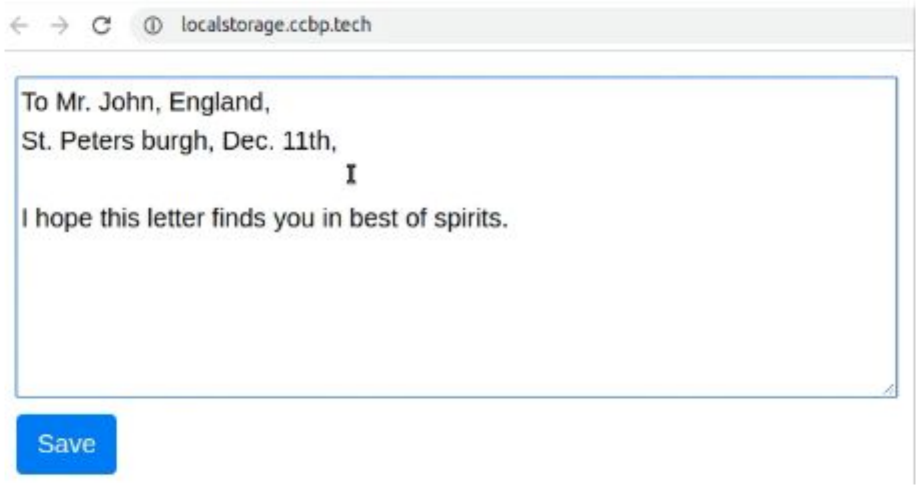
JS

```
let selectedColor = null;
console.log(selectedColor);
console.log(typeof(selectedColor));
```

Output

```
null
object
```

Local storage
example:



localstorage.ccbp.tech

To Mr. John, England,
St. Peters burgh, Dec. 11th,

I hope this letter finds you in best of spirits.

How can we provide
Multiline Text as input?

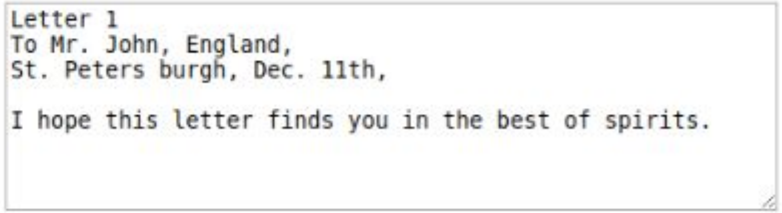
How to provide multiple line as a input?
Below is the way

HTML Elements

The textarea Element

HTML

```html
<textarea rows="8" cols="55">
</textarea>
```



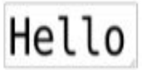
```
Letter 1
To Mr. John, England,
St. Peters burgh, Dec. 11th,

I hope this letter finds you in the best of spirits.
```

- The **rows** attribute specifies the number of **lines**
- The **cols** attribute specifies the number of **characters** per line

The textarea Element

HTML

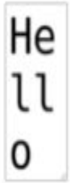
```html
<textarea rows="1" cols="5"></textarea>
```



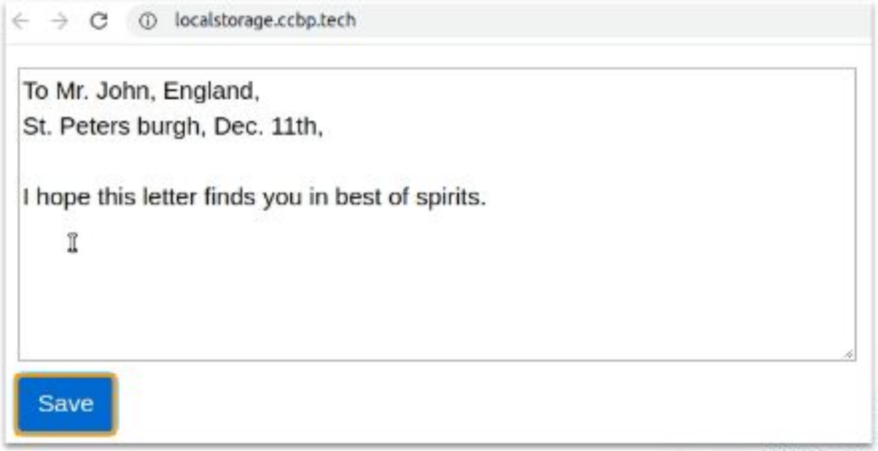
Hello

HTML

```html
<textarea rows="3" cols="2"></textarea>
```



He
ll
o



To Mr. John, England,
St. Peters burgh, Dec. 11th,

I hope this letter finds you in best of spirits.

How to store data

In Local Storage

on Button Click?

Storing Data in Local Storage

Adding Button with Event Listener

HTML

```html
<textarea rows="8" cols="55"></textarea>
<br />
<button class="btn btn-primary mt-1" id="saveButton">Save</button>
```

JS

```js
let saveButton = document.getElementById("saveButton");
saveButton.onclick = function() {

};
```

Storing Data in Local Storage

Accessing textarea Element value

HTML

```html
<textarea rows="8" cols="55" id="message"></textarea>
```

JS

```js
let saveButton = document.getElementById("saveButton");
let textAreaElement = document.getElementById("message");
saveButton.onclick = function() {
  let userEnteredText = textAreaElement.value;
};
```

Storing Data in Local Storage

Storing value in Local Storage

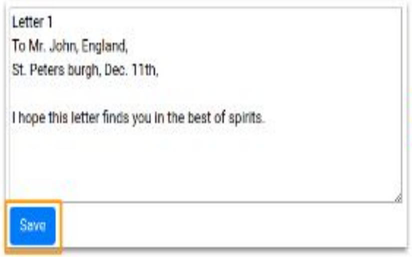
JS

```js
...
saveButton.onclick = function() {
  let userEnteredText = textAreaElement.value;
  localStorage.setItem("userEnteredText", userEnteredText);
};
...
```

Storing Data in Local Storage

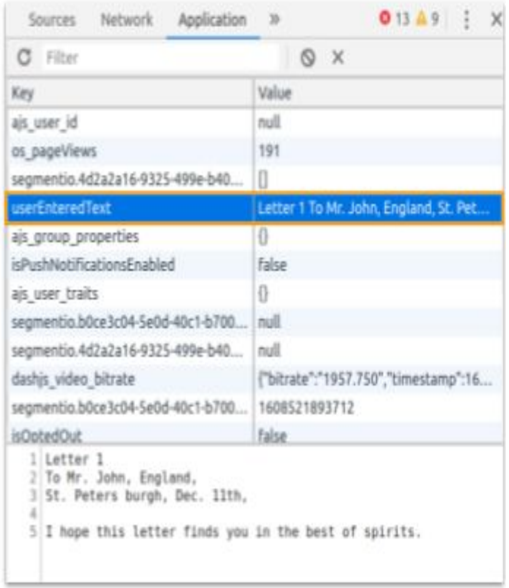
Output

How to load Text Message automatically on Reload?



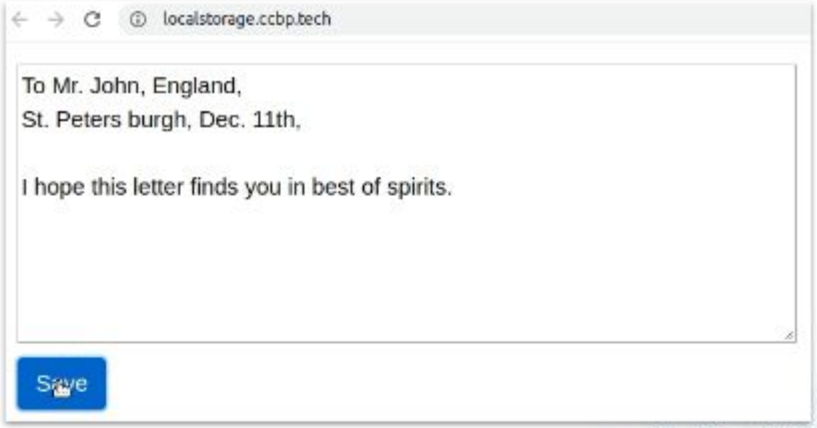
Letter 1
To Mr. John, England,
St. Peters burgh, Dec. 11th,

I hope this letter finds you in the best of spirits.



Key	Value
ajs_user_id	null
os_pageViews	191
segmentio.4d2a2a16-9325-499e-b40...	[]
userEnteredText	Letter 1 To Mr. John, England, St. Pet...
ajs_group_properties	{}
isPushNotificationsEnabled	false
ajs_user_traits	{}
segmentio.b0ce3c04-5e0d-40c1-b700...	null
segmentio.4d2a2a16-9325-499e-b40...	null
dashjs_video_bitrate	{"bitrate":"1957.750","timestamp":16...
segmentio.b0ce3c04-5e0d-40c1-b700...	1608521893712
isOptedOut	false

```
Letter 1
To Mr. John, England,
St. Peters burgh, Dec. 11th,

I hope this letter finds you in the best of spirits.
```



To Mr. John, England,
St. Peters burgh, Dec. 11th,

I hope this letter finds you in best of spirits.

Getting Data from Local Storage

Automatically on Reload

JS

```
let storedUserInputValue = localStorage.getItem("userEnteredText");
if (storedUserInputValue === null) {
  textAreaElement.value = "";
}
else {
  textAreaElement.value = storedUserInputValue;
}
```

Key Takeaways

- Local Storage
  - setItem()
  - getItem()
- Values
  - null
- HTML Elements
  - textArea Element

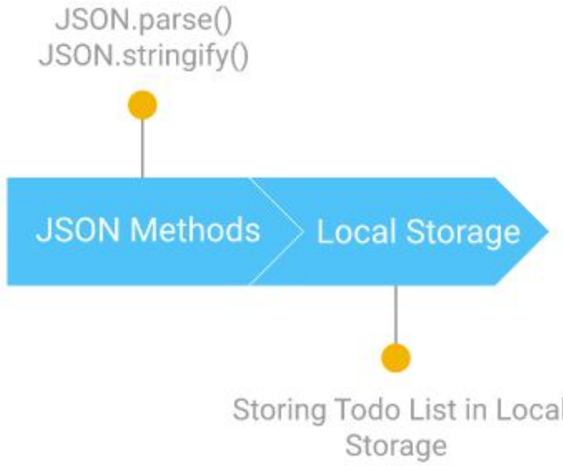
lecture4:todo

Recap

Values

Local Storage can only use **strings** for its keys and value

Trying to store any other type of data (objects) can lead to **unexpected behaviour**



Step:1 question

How can I Store Other Types of Values (Objects, Arrays, etc.) in Local Storage?

step2:anwer

JSON Strings

JSON

Supported Types

Storing Data in Local Storage

JavaScript Object Notation (JSON)

JSON is a data **representation format** used for:

- Storing Data (Client/Server)
- Exchanging Data between Client and Server

- Number
- String
- Boolean
- Array
- Objects
- Null

JSON

Supported Types

JavaScript	JSON
10	10
"hello"	"hello"
true	true
[1, 2, 3]	[1, 2, 3]
null	null

JS Object vs JSON Object

JS Object

```
{
  name: "Rahul",
  age: 29,
  designation: "Web Developer"
}
```

JSON Object

```
{
  "name": "Rahul",
  "age": 29,
  "designation": "Web Developer"
}
```

JavaScript

JSON Methods

JavaScript provides **JSON methods** to convert Data into **JSON format.**

- JSON.stringify()
- JSON.parse()

JSON Methods

JSON.stringify()

It converts the given value into **JSON String**

$$\texttt{JSON.stringify(}\underbrace{\texttt{ value }}_{\text{Value to be converted}}\texttt{)}$$

JSON Methods

JSON.parse()

It parses a **JSON String** and returns a **JS Object**

$$\texttt{JSON.parse(}\underbrace{\texttt{ string }}_{\text{String in JSON format}}\texttt{)}$$

Example

Storing JS Objects

```
let profile = {
  name: "Rahul",
  age: 29,
  designation: "Web Developer"
};
```

JSON Methods

stringify()

```js
let profile = {
  name: "Rahul",
  age: 29,
  designation: "Web Developer"
};
```

↓

```js
JSON.stringify(profile);
```

↓

```
'{"name":"Rahul","age":29,"designation":"Web Developer"}'
```

JSON Object Methods

stringify()

JS

```js
let stringifiedProfile = JSON.stringify(profile);
console.log(stringifiedProfile);
console.log(typeof(stringifiedProfile));
```

Output

```
{"name":"Rahul","age":29,"designation":"Web Developer"}
string
```

JSON Object Methods

parse()

```
let stringifiedProfile = '{"name":"Rahul","age":29,"designation":"Web Developer"}'
```

$\downarrow$

```
JSON.parse(stringifiedProfile);
```

$\downarrow$

```
{
  name: "Rahul",
  age: 29,
  designation: "Web Developer"
}
```

JSON Object Methods

parse()

JS

```
let parsedProfile = JSON.parse(stringifiedProfile);
console.log(parsedProfile);
console.log(typeof(parsedProfile));
```

Output

```
Object {name: "Rahul", age: 29, designation: "Web Developer"}
object
```

Storing and Getting Data

JS

```
localStorage.setItem("profileDetails", JSON.stringify(profile));
let stringifiedProfileDetails = localStorage.getItem("profileDetails");
let parsedProfileDetails = JSON.parse(stringifiedProfileDetails);
console.log(parsedProfileDetails);
```

Output

```
Object {name: "Rahul", age: 29, designation: "Web Developer"}
```

Storing Todo List in Local Storage

Adding Save Button Statically

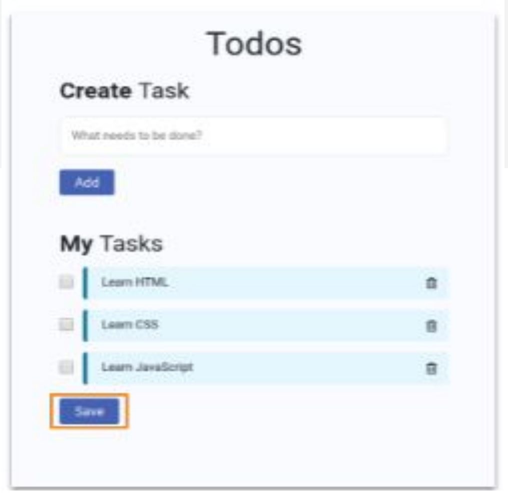
Todo List

JS

```js
let todoList = [
 {
   text: "Learn HTML",
   uniqueNo: 1
 },
 {
   text: "Learn CSS",
   uniqueNo: 2
 },
  ...
];
```

HTML

```html
<div class="todos-bg-container">
 <div class="container">
   <div class="row">
       ...
       <ul class="todo-items-container" id="todoItemsContainer"></ul>
       <button class="button" id="saveTodoButton">Save</button>
       ...
   </div>
 </div>
</div>
```



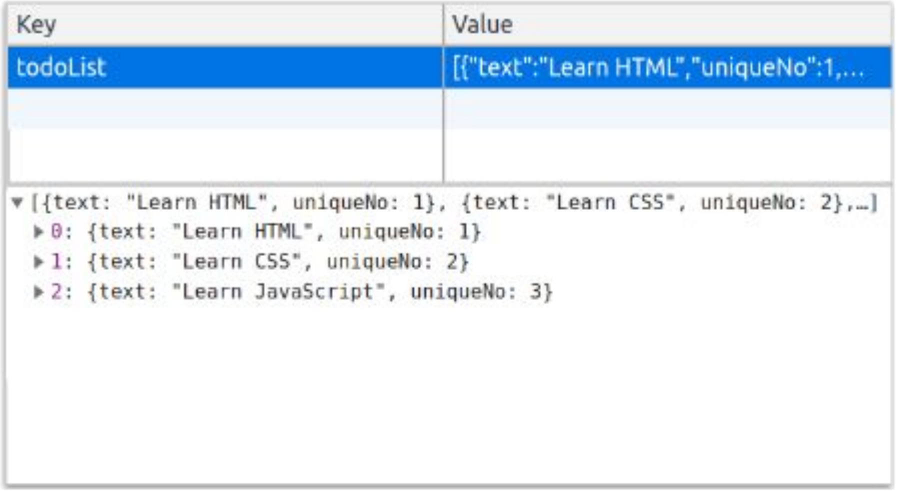
Adding Event Listener Dynamically

JS

```js
...
let todoItemsContainer = document.getElementById("todoItemsContainer");
let addTodoButton = document.getElementById("addTodoButton");
let saveTodoButton = document.getElementById("saveTodoButton");
...
saveTodoButton.onclick = function () {
};
...
```

Storing TodoList

```
...
saveButton.onclick = function () {
  localStorage.setItem("todoList", JSON.stringify(todoList));
};
...
```



Key	Value
todoList	[{"text":"Learn HTML","uniqueNo":1,...

```
▼ [{text: "Learn HTML", uniqueNo: 1}, {text: "Learn CSS", uniqueNo: 2},…]
  ▶ 0: {text: "Learn HTML", uniqueNo: 1}
  ▶ 1: {text: "Learn CSS", uniqueNo: 2}
  ▶ 2: {text: "Learn JavaScript", uniqueNo: 3}
```

How to get Todo List from Local Storage?

Getting Todo List from Local Storage

getItem()

JS

```js
...
function getTodoListFromLocalStorage() {
  let stringifiedTodoList = localStorage.getItem("todoList");
}
...
```

Getting Todo List from Local Storage

Parsing Stringified TodoList

JS

```js
...
function getTodoListFromLocalStorage() {
  let stringifiedTodoList = localStorage.getItem("todoList");
  let parsedTodoList = JSON.parse(stringifiedTodoList);
}
...
```

Getting Todo List from Local Storage

Parsed Todo List

JS

```
...
function getTodoListFromLocalStorage() {
  ...
  if (parsedTodoList === null) {
    return [];
  }
  else {
    return parsedTodoList;
  }
}
...
```

Getting Todo List from Local Storage

Assigning value to Todo List

JS

```
...
function getTodoListFromLocalStorage() {
  ...
  if (parsedTodoList === null) {
    return [];
  }
  else {
    return parsedTodoList;
  }
}
...
let todoList = getTodoListFromLocalStorage();
```

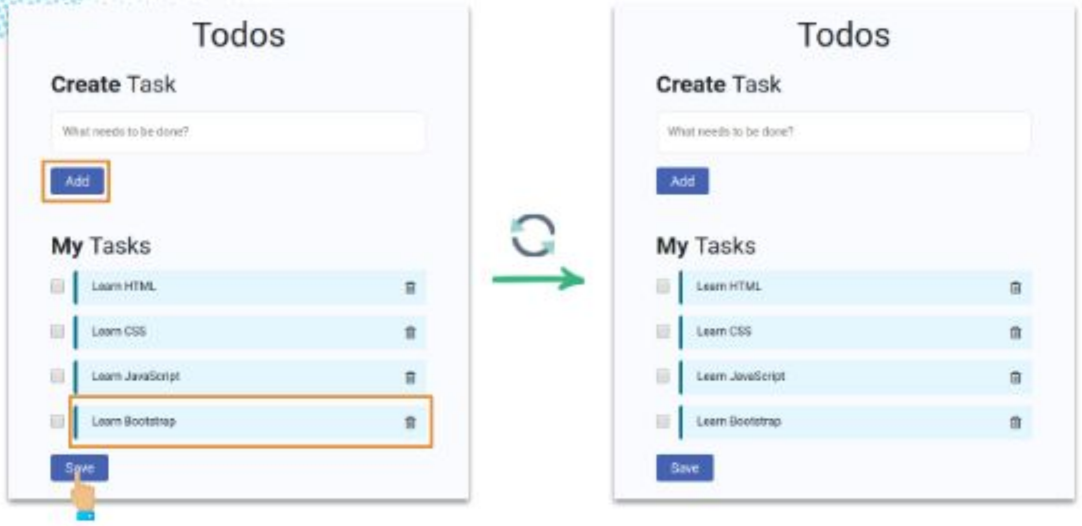
Getting Todo List from Local Storage

Delete existing Todo List

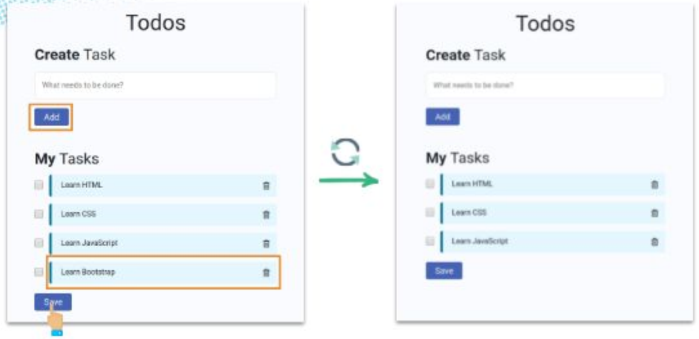
JS

```
let todoList = [
 {
   text: "Learn HTML",
   uniqueNo: 1
 },
 {
   text: "Learn CSS",
   uniqueNo: 2
 },
 ...
];
```

Adding a New Todo Item

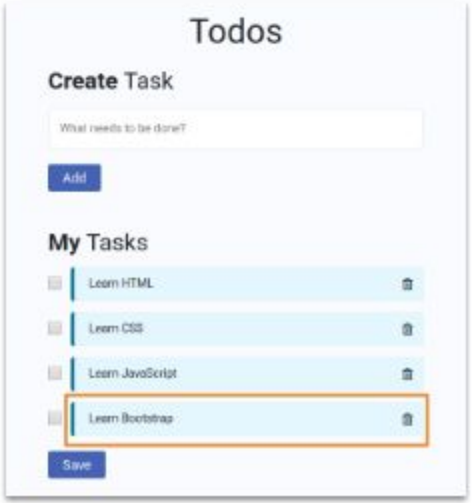


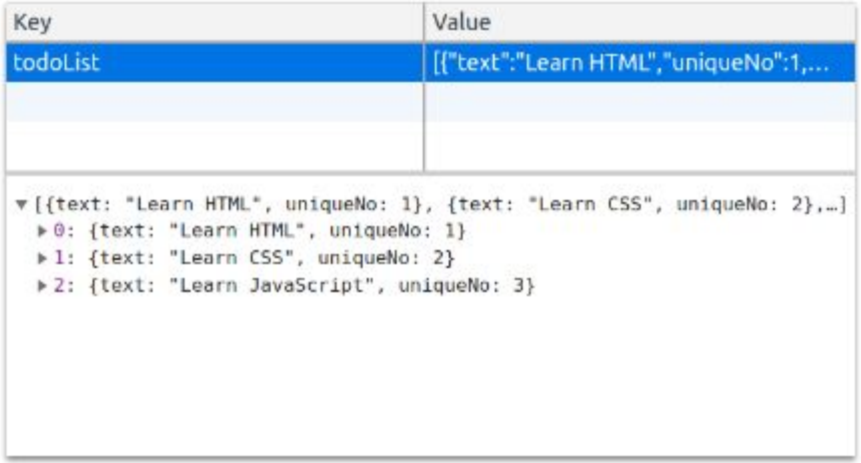
Why is the new Todo Item not persisting on Reload?



On Reload

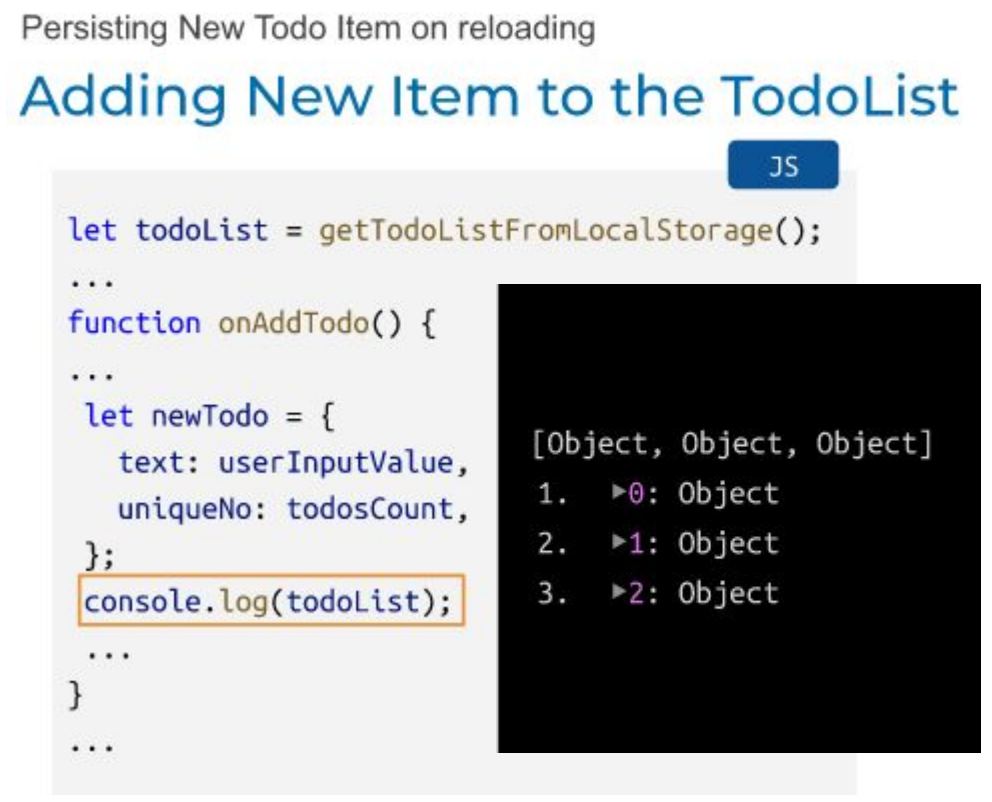
Local Storage





Key	Value
todoList	[{"text":"Learn HTML","uniqueNo":1,...

```
▼[{text: "Learn HTML", uniqueNo: 1}, {text: "Learn CSS", uniqueNo: 2},…]
  ▶0: {text: "Learn HTML", uniqueNo: 1}
  ▶1: {text: "Learn CSS", uniqueNo: 2}
  ▶2: {text: "Learn JavaScript", uniqueNo: 3}
```

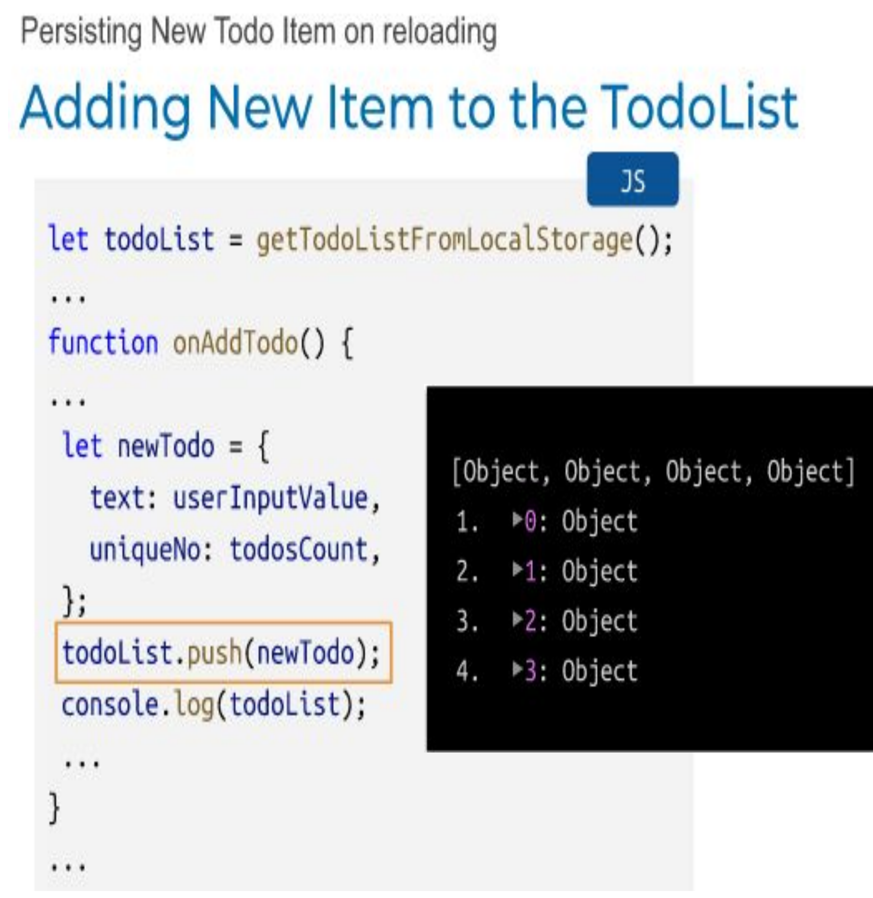

Persisting New Todo Item on reloading

Adding New Item to the TodoList

JS

```
let todoList = getTodoListFromLocalStorage();
...
function onAddTodo() {
...
 let newTodo = {
   text: userInputValue,
   uniqueNo: todosCount,
 };
 console.log(todoList);
  ...
}
...
```

```
[Object, Object, Object]
1.   ▸0: Object
2.   ▸1: Object
3.   ▸2: Object
```


Persisting New Todo Item on reloading

Adding New Item to the TodoList

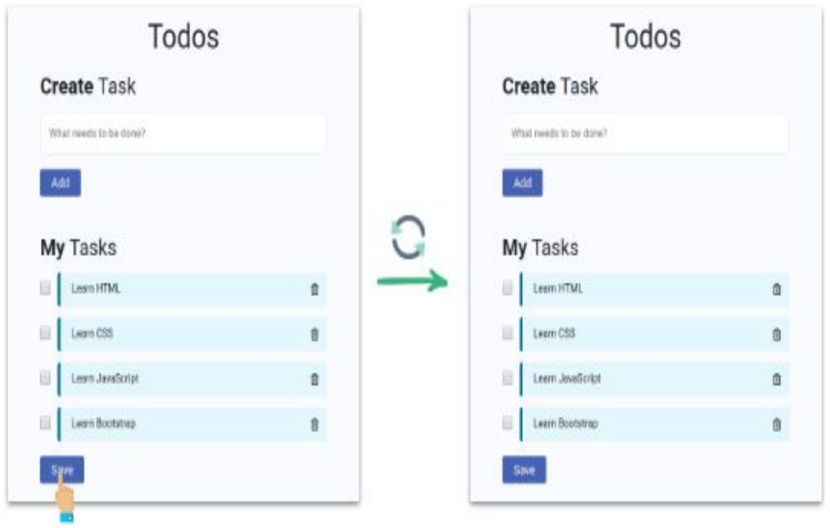
JS

```
let todoList = getTodoListFromLocalStorage();
...
function onAddTodo() {
...
 let newTodo = {
   text: userInputValue,
   uniqueNo: todosCount,
 };
 todoList.push(newTodo);
 console.log(todoList);
  ...
}
...
```

```
[Object, Object, Object, Object]
1.  ▸0: Object
2.  ▸1: Object
3.  ▸2: Object
4.  ▸3: Object
```

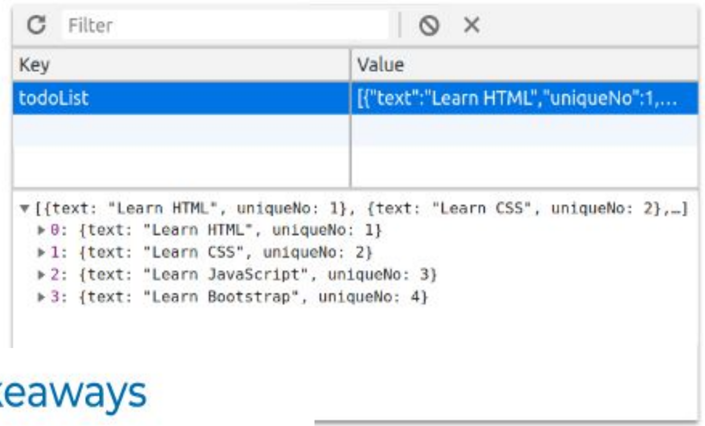
Persisting New Todo Item on reloading

Output



Local Storage

TodoList



Key	Value
todoList	[{"text":"Learn HTML","uniqueNo":1,...

```
▼[{text: "Learn HTML", uniqueNo: 1}, {text: "Learn CSS", uniqueNo: 2},…]
  ▶0: {text: "Learn HTML", uniqueNo: 1}
  ▶1: {text: "Learn CSS", uniqueNo: 2}
  ▶2: {text: "Learn JavaScript", uniqueNo: 3}
  ▶3: {text: "Learn Bootstrap", uniqueNo: 4}
```

Key Takeaways

- JSON Methods
  - JSON.stringify()
  - JSON.parse()
- Local Storage
  - Storing Todo List in Local Storage