

JS and DOM Interactions

Pull from upstream!

Commit any changes first!

Basics of Writing To Document from JavaScript

For now, we will only know one way to dynamically write html from our JavaScript:

```
document.writeln("html tags and text here");
```

For example: `document.writeln("<p>Paragraph Text</p>");`

Basics of Writing To Document from JavaScript

You may also embed variables into your html now!

For example:

```
let x = "Station Wagons";
```

```
document.writeln("<p>My favorite cars are " + x + "</p>");
```

Most of the examples posted use this, so test it out!

JavaScript DOM Manipulation

Accessing Information:

- `document.getElementById('myID');`
- `document.getElementsByTagName('p');`
- `document.getElementsByClassName('mainMenu');`

Advanced DOM Manipulation

- `element.innerHTML = new html`
- ~~`element.attribute = new values`~~
- `element.style.property = new style`
- `element.setAttribute(attribute, value);`

Advanced DOM Manipulation

- `document.createElement(element)`
- `document.removeChild(element)`
- `document.appendChild(element)`
- `document.replaceChild(new, old)`
- `document.write(text)`

Forms in HTML

- Means by which information passes from the user to a server
- Used to read values to be processed by JavaScript
- **<form> tag**
 - Defines the form
 - It has two attributes : **action** and **method**

Form

- **action** : indicates where the form contents will be sent when the form is submitted
- **method** : defines how the contents will be sent
 - **post** : contents sent using the HTTP POST method. Content does NOT appear in the request URL. Content is attached to the request body.
 - **get** : contents sent using the HTTP GET method. Contents become part of the request URL. These are known as “request params”.

input Tag

```
<input type=" ... " id=" ... " name=" ... " />
```

- Appears inside of the <form> tag
- Defines several input data alternatives
- The general format is à <input type="*choices*" />
- Can also use name property to select element in the dom for forms and inputs

Type Choices

- text
- password
- checkbox
- radio
- file
- submit
- button
- reset
- hidden
- date
- email

CheckBoxes

- Allow us to make a selection
- Defined by using type=**“checkbox”**
- We can tell whether an entry is selected by using the **“checked”** property
 - True- an entry has been selected
- Default selection by using checked=**“checked”**

Form Data Access

- We can access data in forms by using getElementById(string)
- It returns a reference to an element whose id property matches with the given string and we can:
- Retrieve the value of the element (e.g., text field in a form)
 - `var login = document.getElementById("loginId").value;`
- Set a function to call when an element is clicked on (e.g., button)
 - `document.getElementById("processButton").onclick = callback;`
- Another way to associate a function is to use `addEventListener` (allows several events to be added)
 - `document.getElementById("displayValueButton").addEventListener("click", callback);`

Form Data Access

- Get/Set Attributes of the element
 - `var imageElement = document.getElementById("myImage");`
 - `var imageName = imageElement.getAttribute("src");`
 - `imageElement.setAttribute("src", "imageFile.jpg");`
- Let's complete ex-updateValueInTextField.html
- Animation
- setInterval
 - `window.setInterval (callback, millisecond, param1, param2, ...);`
 - The return value from setInterval can be used to stop repeating

Other Event Handler Attributes

onclick- mouse button is pressed and released

ondblclick- mouse button is double-click over element

onmousedown- mouse is pressed down while cursor is over the element

onmouseup- mouse is released while the cursor is over the element

onmouseenter- mouse moves onto element

onmouseover- mouse pointer enters into an element and its child elements

onmouseout- mouse moves off element

onmousemove- mouse pointer is moved over an element