Topic/Title: Adding JavaScript to Websites

Keywords/Questions:

Inline Javascript

<body onload="">

<style type="text/javascript">

Internal Javascript

<style src="index.js" charset="utf-8">

External Javascript

styles.css

document.querySelector("h1").innerHTML="Hi";

Notes:

Inline JavaScript:-

-you put your JavaScript code inside the quotes in <body ***onload=" "***>

```
<body onload="alert('Hello');">
```

-Don't make both quotes double quotes the inside one should be a single quote. Otherwise, it won't work.

-When the code is loaded up, we see an alert that says Hello.

-Inline JavaScript has a lot of the downsides of inline CSS. It's not very modular, it's difficult to debug, and it's also not good practice as well, so try to avoid that if you can.

Internal JavaScript:- we can include it by using a <script type="text/plain"> </script> tag inside the body.

```
<body>
  <h1>Hello</h1>
  <script type="text/javascript">
    alert("Hello");
  </script>
</body>
```

HTML code is executed line by line. We place our ***sylesheet code inside the title*** while we place our ***<script> code at the end of the body***. If we put our stylesheet at the end of the body, the styling will be applied after all the code inside the body is executed. For example, the color of an h1 code might change at the end.

Summary:

Inline JavaScript:- added in the body tag.

```
<body onload="alert('Hello');">
```

-The inside quotes should be ' '.

Internal JavaScript:- added in the body part and we enclose our code within the <script type="text/javascript"> </script> tag.

```
<script type="text/javascript">
  alert("Hello");
</script>
```

External JavaScript:- add <script src="index.js" charset="utf-8"></script> to the end of the <body> part.

```
<body>
  <h1>Hello</h1>
  <script src="index.js" charset="utf-8"></script>
</body>
```

document.querySelector("h1").innerHTML ="Good Bye";:-write it in index.js.It changes the h1 Hello to Good Bye

More Notes

External JavaScript :-

-Add *<script src="index.js" charset="utf-8"></script>* to the **end of the <body> part.**

-We create a file outside called "index.js"

```
<body>
  <h1>Hello</h1>
  <script src="index.js" charset="utf-8"></script>
</body>
```

```
document.querySelector("h1").innerHTML = "Good Bye";
```

-This code right above changes the "Hello" h1 text to a Good Bye. It is written in the "index.js" file.

Topic/Title: Introduction to the Document Object Model (DOM)

Keywords/Questions:

DOM model

document;

HTML tree generator

document.firstElementChild;

document.firstElementChild.
firstElementChild;

document.firstElementChild.
lastElementChild;

var x=.....

x.style.color="Red";

x.innerHTML="HI";

document.querySelector("h1").
innerHTML="Hi";

document.querySelector("input").
click();

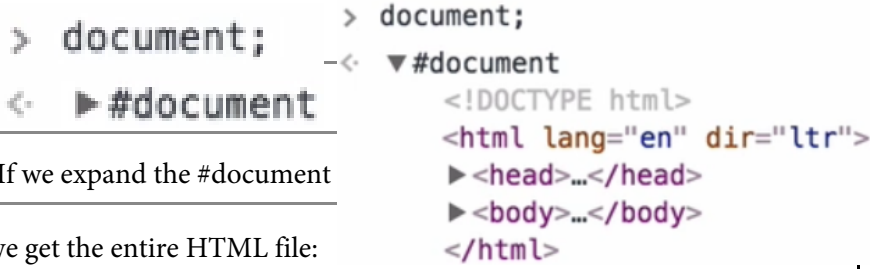
DOM objects have:
-property
-method

Notes:

The DOM model:- the document object model. every element treated as an object.

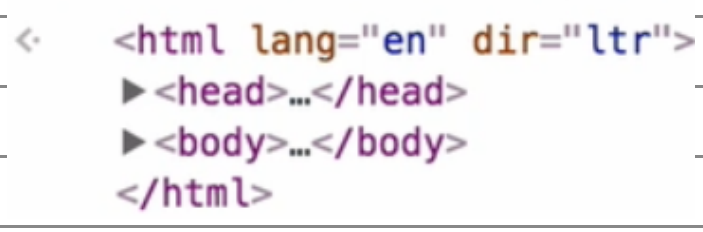
document;

-writing this in the console will print out the whole #document.



-If we expand the #document we get the entire HTML file:

document.firstElementChild;



-It's giving me everything that's inside the HTML.

Summary:

The DOM model :-treating every elements in our page as objects. document; :- prints the whole page code/document

document.firstElementChild; :-accesses the html tag and everthing inside it

document.firstElementChild.firstElementChild:-the head tag -document.firstElementChild.lastElementChild:-the body tag

var x=document.firstElementChild.lastElementChild.firstElementChild; :-means x=the h1 tag in the example in page 2

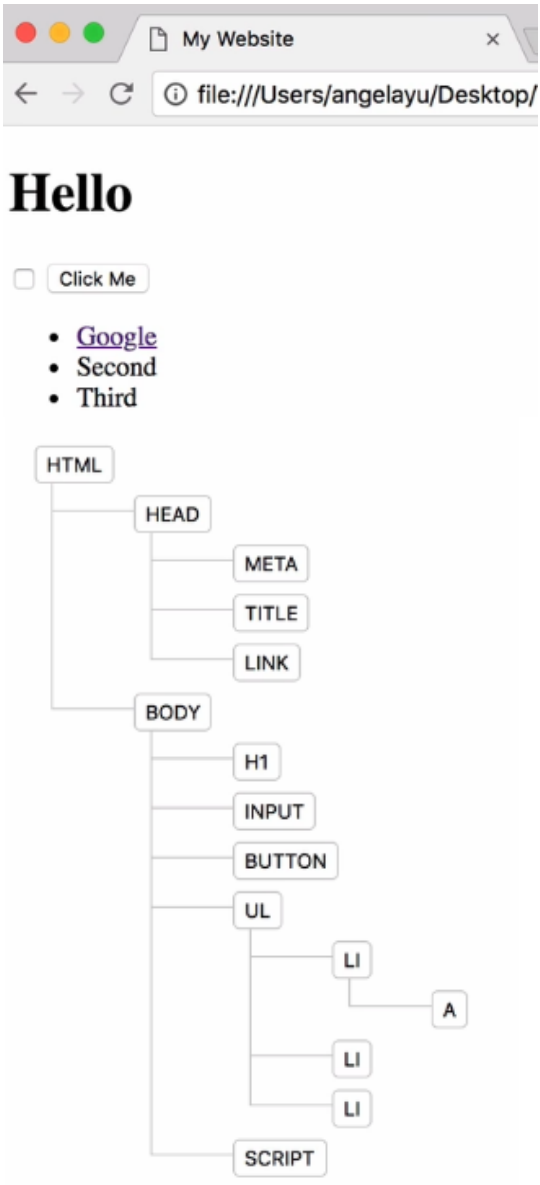
x.style.color="Red":-changes the color of the h1 tag to red -HTML tree generator:-shows us the DOM outline for our page.

x.innerHTML="Hi"; & document.querySelector("h1").innerHTML="Hi"; :-both change the content of the h1 tag to Hi

document.querySelector("input").click(); :-clicks and ticks the empty box of our button

An object inside a DOM has a -property: like color and innerHTML -method(used only for objects): like click()

More Notes



document.firstElementChild.firstElementChild;

```
> document.firstElementChild.firstElementChild;
▼<head>
    <meta charset="utf-8">
    <title>My Website</title>
    <link rel="stylesheet" href="styles.css">
  </head>
```

-It gives us the head. Because the head is the first child of the<html> tag and <html> is the first child of the document

document.firstElementChild.lastElementChild;

```
> document.firstElementChild.lastElementChild;
▼<body>
    <h1>Hello</h1>
    <input type="checkbox">
    <button style=":active: color:red;">Click Me</button>
  ▶<ul>…</ul>
    <script src="index.js" charset="utf-8"></script>
  </body>
```

-It gives us the body. Because the body is the last child of the<html> tag and <html> is the first child of the document

document.firstElementChild.lastElementChild.firstElementChild;

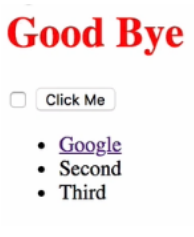
```
> document.firstElementChild.lastElementChild.firstElementChild;
  <h1>Hello</h1>
```

-It gives us the <h1> tag.

```
> var heading = document.firstElementChild.lastElementChild.firstElementChild;
> heading
  <h1>Hello</h1>
> heading.innerHTML = "Good Bye";
"Good Bye"
```

-the heading.*innerHTML* ="Good Bye"; changes "Hello" to "Good Bye".

```
> heading.style.color = "red";
"red"
```



-heading.*style.color*="red" makes the h1 red in color.

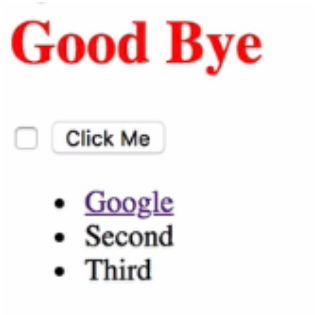
.innerHTML:- used to change the content of an element when used with the DOM model

```
> var heading = document.firstElementChild.lastElementChild.firstElementChild;
```

```
> heading
<-   <h1>Hello</h1>
> heading.innerHTML = "Good Bye";
<- "Good Bye"
```

.style.color :- used to change the color of an element when used with the DOM model.

```
> heading.style.color = "red";
<- "red"
```



The DOM model :-allows us to change our website on the fly using the DOM.

-the above 2(.innerHTML and style.color) are examples of how we can do this.

-we write our code inside our console.

document.querySelector("input").click();

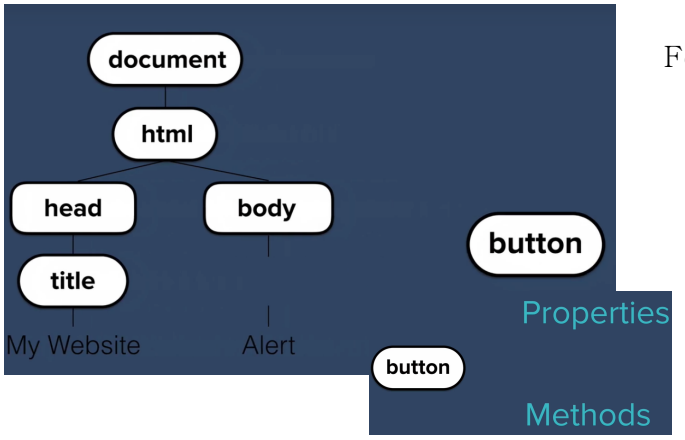
-when we run this code our checkbox gets clicked.

```
> document.querySelector("input").click();
<- undefined
```

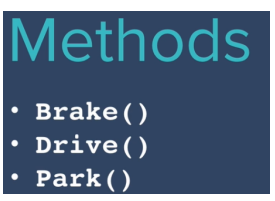
-we can see that our object inside the DOM can have both

-properties:- describe something about the object.

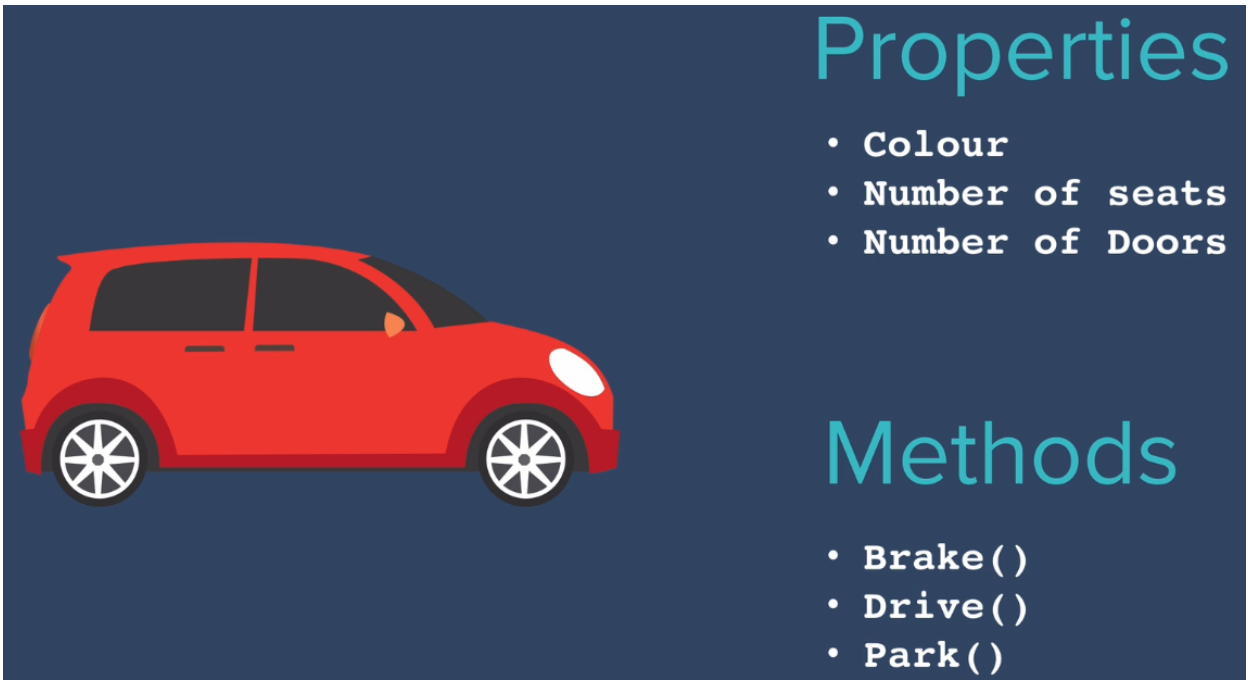
-methods:- are things the object can do.



For example a car



Let us demonstrate properties and methods of an object inside the DOM using this car. Let the objects name be car:



Manipulating Properties

Get Property (getting a property)

-we use it to get a property or in other words the value of a property.

```
car.colour; //red
```

Set Property (setting a property)

-we assign a value using an equal sign.

```
car.numberOfDoors = 0;
```

Manipulating Methods

Call Method (calling a method)

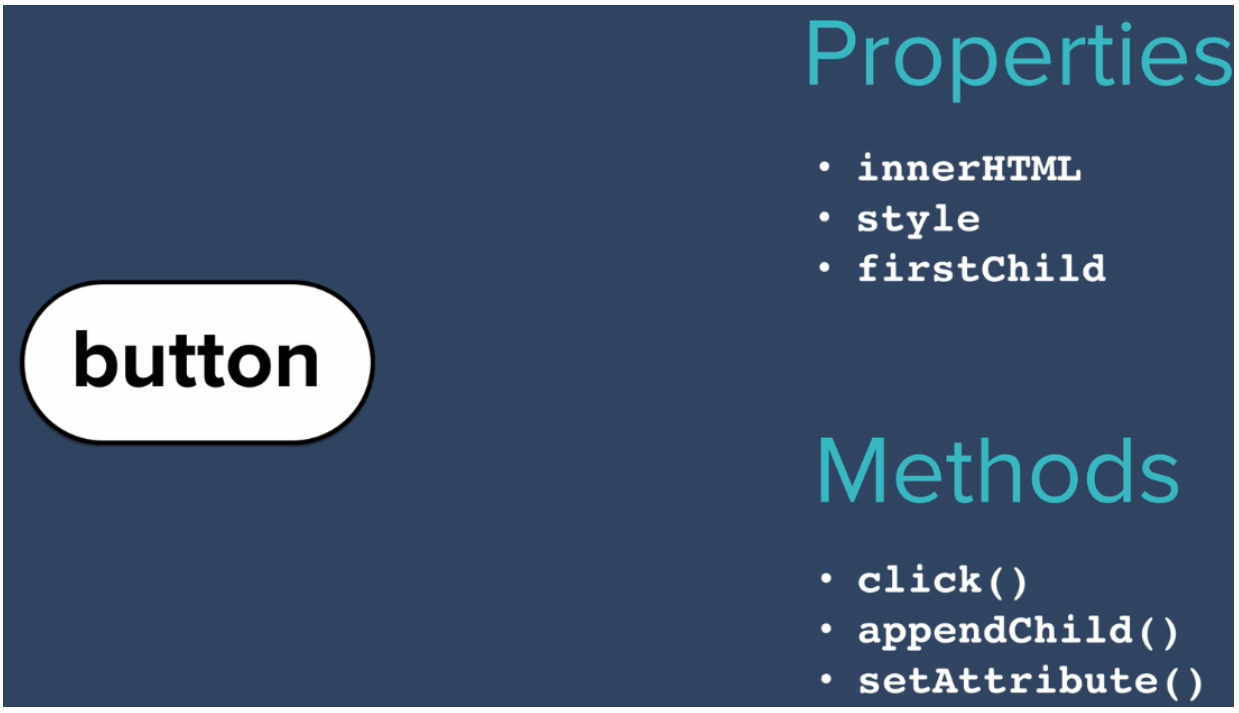
-it is calling a method on our object.

-For example `car.drive();`

-will make the car drive.

-The difference between a method and an object is that a method is something that is only associated with an object. So, when we say method it should be about something our object can do. For example, *car.drive();* from above is a method. Function is more general and doesn't necessarily need to be about sth an object can do. So, a method is also a function.

-Let us go back to our button object.

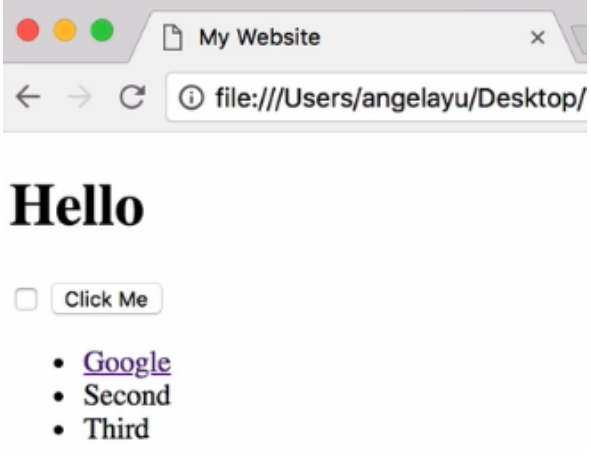


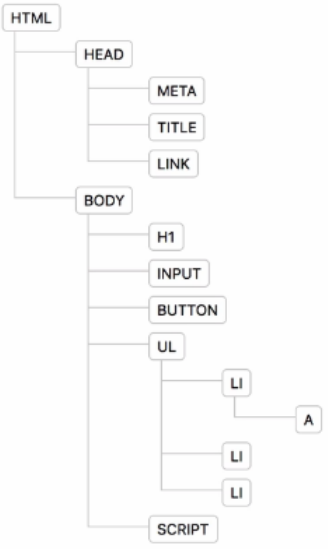
-All methods should have a set of parentheses ( ) at the end. This is how you can tell the difference between a method and a property.

-.appendChild() :-used to add another child.

HTML Tree Generator: generates a DOM outline for our page. If it doesn't work

Just right click on HTML Tree generator and select manage extension n then allow access to URLs...





Topic/Title: Selecting HTML Elements with JavaScript

Hello

Click Me

- Google
- Second
- Third

Keywords/Questions:

Notes:

document.getElementsByTagName("li");

-gets all of the elements with a particular tag name. For example here it will fetch all the li elements

```
> document.getElementsByTagName("li");
◂ ▶HTMLCollection(3) [li.item, li.item, li.item]
```

-It gives us the 3 li items in the above page. It gives them as an array.

```
> document.getElementsByTagName("li").style.color = "purple";
✖ ▶Uncaught TypeError: Cannot set property 'color' of undefined
    at <anonymous>:1:49
```

-we can't change the style of all our li items at once.

document.getElementsByTagName("li")[2].style.color="purple";

-we are targeting the 2nd li tag and changed its color to purple.

-this will also give us the 2nd element of the array.

document.getElementsByClassName("btn")[0].style.color="red";

-

```
<button class="btn">Click Me</button>
```

- this will target the 1st "btn" class and change its color to red.

document.getElementsByClassName("btn").style.color="red";

-

```
✖ ▶Uncaught TypeError: Cannot set property 'color' of undefined
    at <anonymous>:1:52
```

-this is will result in an error even if we have only one btn.

Summary:

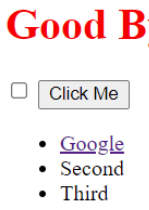
More Notes

document.getElementById("title");

- selects the element with the id="title"

```
<h1 id="title">Hello</h1>
```

- There is only one particular instance of an id so if I select on the title id, then I will only get back a single item as opposed to an array of items.
- document.getElementById("title").innerHTML="Good Bye"



document.querySelector("h1");

- targets the h1 tag

```
> document.querySelector("h1");
<- <h1 id="title">Good Bye</h1>
```

document.querySelector("#title");

- targets the h1 tag

```
> document.querySelector("#title");
<- <h1 id="title">Good Bye</h1>
```

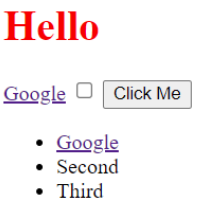
document.querySelector(".btn");

- targets the element with the btn class.

```
<button class="btn">Click Me</button>
> document.querySelector(".btn");
<- <button class="btn" style="color: red;">Click Me</button>
```

```
> document.querySelector("li a");
<- <a href="https://www.google.com">Google</a>
```

- targets the <a> tag inside the
- a hierarchical selector




```
<body>
  <h1 id="title">Hello</h1>
  <a href="https://www.google.com">Google</a>
  <input type="checkbox">
  <button class="btn">Click Me</button>
  <ul id="list">
    <li class="item"><a href="https://www.google.com">Google</a></li>
    <li class="item">Second</li>
    <li class="item">Third</li>
  </ul>
</body>
```

```
> document.querySelector("a");
<- <a href="https://www.google.com">Google</a>
```

- targets the first <a> tag.

```
> document.querySelector("li.item");
<- ▶<li class="item">…</li>
```

- we get our first list item
- we are using a combined selector.

```
> document.querySelector("#list a");
<- <a href="https://www.google.com">Google</a>
```

- targets the anchor tag inside the list. We are using a hierarchical selector.

When using document.querySelector("");, you get the first item that satisfies the particular selector.

```
> document.querySelector("#list .item");
<- ▶<li class="item">…</li>
```

-this gives us the first list item that satisfies the property.

document.querySelectorAll{"#list .item");

-returns a list/array of all items that satisfy the particular selector.

```
> document.querySelectorAll("#list .item");
<- ▶NodeList(3) [li.item, li.item, li.item]
```

-to manipulate specific items you need to use indexes.

```
> document.querySelectorAll("#list .item")[2].style.color = "blue";
<- "blue"
```

```
> document.querySelector("li a").style.color="green";
<- 'green'
```

- Hello

Google Click Me

- Google
- Second
- Third

```
<body>
  <h1 id="title">Hello</h1>
  <a href="https://www.google.com">Google</a>
  <input type="checkbox">
  <button class="btn">Click Me</button>
  <ul id="list">
    <li class="item"><a href="https://www.google.com">Google</a></li>
    <li class="item">Second</li>
    <li class="item">Third</li>
  </ul>
</body>
```

```
> document.querySelector("li").style.color="green";
<- 'green'
```

- Hello

Google Click Me

- Google
- Second
- Third

-Because <li> and <a> are separate elements selecting on <li> will only change the bullet point. The <a> tag inside it will not change its color.

Topic/Title: Manipulating and Changing Styles of HTML Elements with JavaScript

Keywords/Questions:

Notes:

document.querySelector("h1").style.fontSize="10rem";

-Changes the font size to 10 rem.

-***When we use css properties at the end of document.querySelector("h1").style***

-if the property is a:

-**one** letter word: like color. We just write the word color at the end

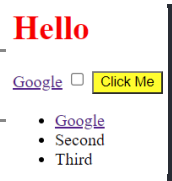
-**document.querySelector("h1").style.color**

->=**two** letter word: like font-size. We remove the hyphen & capitalize the 2 word or any word after the first word.

-**document.querySelector("h1").style.fontSize**

document.querySelector(".btn").style.backgroundColor="yellow";

-changes the background color of the button to yellow




```
<body>
  <h1 id="title">Hello</h1>
  <a href="https://www.google.com">Google</a>
  <input type="checkbox">
  <button class="btn">Click Me</button>
  <ul id="list">
    <li class="item"><a href="https://www.google.com">Google</a></li>
    <li class="item">Second</li>
    <li class="item">Third</li>
  </ul>
</body>
```

Summary:

Topic/Title: The Separation of Concerns: Structure vs Style vs Behavior

Html CSS JS

Keywords/Questions:

document.querySelector("h1").classList

document.querySelector("h1")
.classList.add("huge");

document.querySelector("h1").
classList.remove("huge");

document.querySelector("h1").
classList.toggle("huge");

Notes:


```
<body>
  <h1 id="title">Hello</h1>
  <a href="https://www.google.com">Google</a>
  <input type="checkbox">
  <button class="btn">Click Me</button>
  <ul id="list">
    <li class="item"><a href="https://www.google.com">Google</a></li>
    <li class="item">Second</li>
    <li class="item">Third</li>
  </ul>
</body>
```

```
.huge {
  font-size: 10rem;
}
```

```
.yellowish{
  background-color:yellow;
}
```

-Let the .huge & .yellowish class styling be written in the the styles.css

document.querySelector("h1").**classList**;

-gives us a list of all the classes attached to the element h1

```
> document.querySelector("button").classList;
< ▶ DOMTokenList ["btn", value: "btn"]
```

-gives us a list of all the elements attached to the button element

document.querySelector("h1").classList.**add**("huge");

-adds a class called "huge" to the h1 tag

```
> document.querySelector("button").classList.add("yellowish");
<button class="btn yellowish">Click Me</button>
```
Click Me

document.querySelector("h1").classList.**remove**("huge"); :- removes the "huge" class from the h1

document.querySelector("h1").classList.**toggle**("huge");

-If the class "huge" is included in the h1 it will remove it. If it is no present it will add it

Summary:

document.querySelector("h1").**classList**;

-gives us a list of all the classes of the h1 tag.

document.querySelector("h1").classList.**add**("huge"); :-we don't need to use a dot while writing the class to be added

-The class "huge" will be added to the "h1". All the properties of the class "huge" will be added to the h1 tag

document.querySelector("h1").classList.**remove**("huge"); :-we don't need a . here too

- the class "huge" will be removed from the "h1"

document.querySelector("h1").classList.**toggle**("huge")

-it serves as an on/off switch for the class "huge". it adds/removes it.

Topic/Title: Text Manipulation and the Text Content Property

Keywords/Questions:

document.querySelector("h1").innerHTML;

document.querySelector("h1").
innerHTML="Hi;

document.querySelector("h1").textContent;

document.querySelector("h1").
textContent="Hi";

Notes:

document.querySelector("h1").**innerHTML**;

-targets everything in between the h1 tag even other tags like &

-For example, if the h1 text is in between tags.

```
<h1 id="title"><strong>Hello</strong></h1>
> document.querySelector("h1").innerHTML;
'<strong>Hello</strong>'
```

document.querySelector("h1").innerHTML="Good Bye"

-italicizes and changes the h1 text to Good Bye

```
> document.querySelector("h1").innerHTML="<em>Good Bye</em>";
'<em>Good Bye</em>'
```

Good Bye

document.querySelector("h1").textContent;

-targets only the text content unlike innerHTML

```
<h1 id="title"><em>Good Bye</em></h1>
> document.querySelector("h1").textContent;
'Good Bye'
```

document.querySelector("h1").textContent="Hale";

-can modify only the text inside the h1. It can't affect its style.

```
> document.querySelector("h1").textContent="Hale";
'Hale'
```

Hale

Summary:

document.querySelector("h1").innerHTML;

-targets everything inside the h1 tag like & tag besides the text inside.

document.querySelector("h1").innerHTML="Hello";

-modifies the text within the h1 tag and also italicizes it

document.querySelector("h1").textContent;

-targets only the text enclosed within the h1 tag.

document.querySelector("h1").textContent="Hi";

-modifies the text enclosed within the h1 tag.

Topic/Title: Manipulating HTML Element Attributes

Keywords/Questions:

HTML Attributes

document.querySelector("a").attributes;

document.querySelector("a").getAttribute;

document.querySelector("a").
setAttribute("href", "google.com");

Notes:

HTML Attributes

-everything that goes inside the html tag, other than the tag name itself, are attributes.

-the attributes are all the things that get colored in orange in Atom.

```
<h1 id="title"><strong>Hello</strong></h1>
<a href="https://www.google.com">Google</a>
<input type="checkbox">
<button class="btn">Click Me</button>
```

-class, type and href are all attributes.

document.querySelector("a").**attributes**;

-lists all the attributes of the <a> tag.

```
> document.querySelector("a");
<-   <a href="https://www.google.com">Google</a>
> document.querySelector("a").attributes;
<- ▶NamedNodeMap {0: href, href: href, length: 1}
```

document.querySelector(""a").**getAttribute**("href");

-shows us the content of the attribute href or any other attribute we input.

```
> document.querySelector("a").getAttribute("href");
<- "https://www.google.com"
```

document.querySelector("a").**setAttribute**("href", "https://www.bing.com");

-we can use this to modify our attribute content. e.g. we modified the href content.

Summary: HTML attributes:- every keyword inside an HTML tag except the tag name.

-have an orange color in atom i.e. href, type, class

document.querySelector("a").attributes; :-lists all the attributes of the <a> tag.

document.querySelector("a").getAttribute("href");

-shows us the content of an attribute

document.querySelector("a").setAttribute("href" , "https:www.bing.com");

-setting the content of our attribut