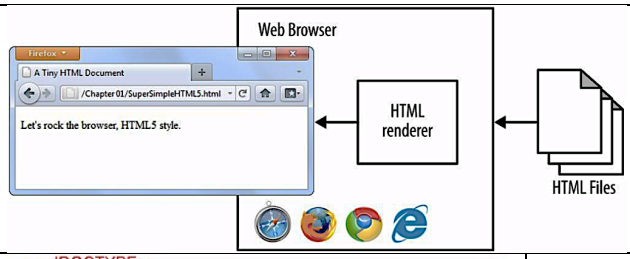
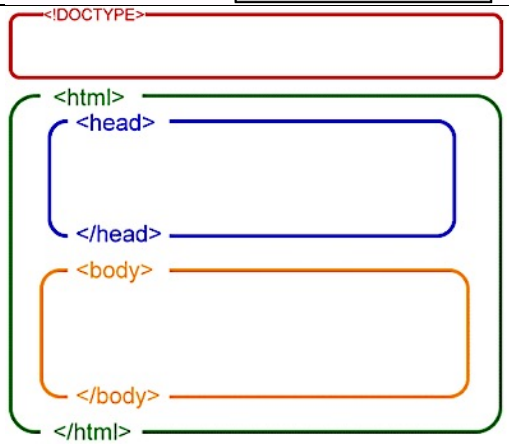
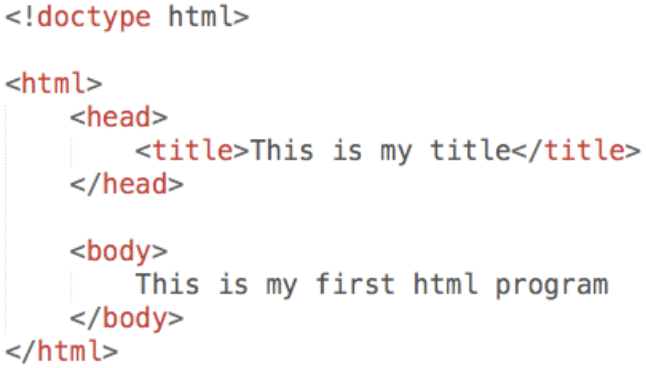
HTML

It is a language for describing Web pages. (created in 1991 by Tim Berners Lee)

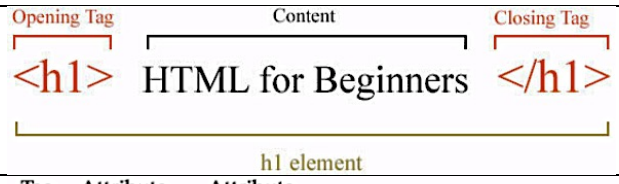
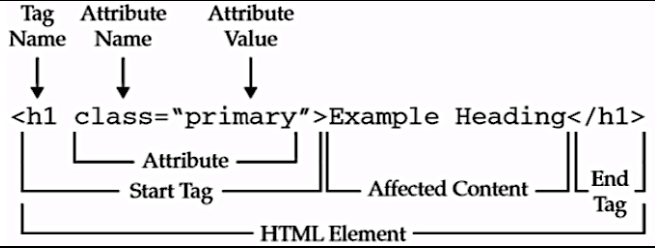
- It stands for Hyper Text Markup Language.
- It is not a programming language, it is markup language.
- A markup language is a collection of markup tags
- It uses markup tags to describe Web pages.

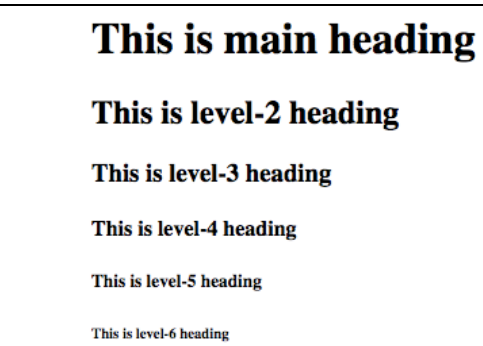
HTML's do;

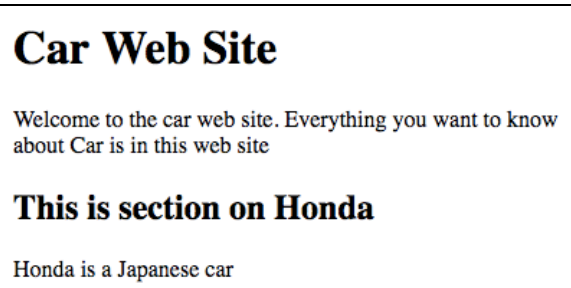
- Storage, Type, Motion, Games, Audio, 3D, Video

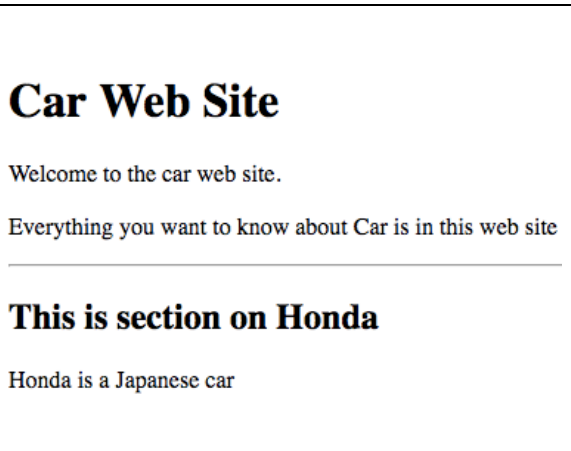
How to write HTML files	Index.html	Just like most programming languages, we type a bunch of HTML into file, so we can send it around.
How to display files		Web browsers are basically "HTML Readers"
Structure of an HTML Page		

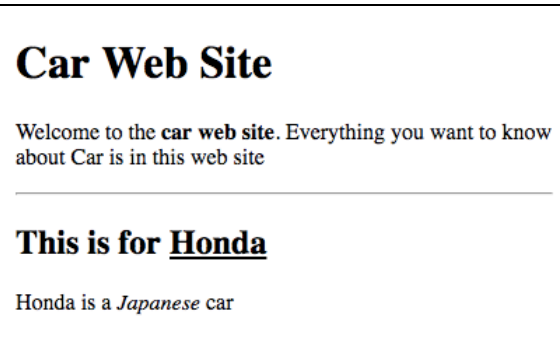
DOCTYPE	It is an instruction to the web browser about what version of HTML the page is written in • Must be at the top of the document • Declaration is not case-sensitive	

Tags	They act like containers. They tell you something about the information that lies between their opening and closing tags.	
Attributes	-provide additional information about the contents of an element. They appear on the opening tag of the element and are made up of two parts: a **name** and a **value,** separated by an equal sign.	

Headings	<body> This is main heading This is level-2 heading This is level-3 heading This is level-4 heading This is level-5 heading This is level-6 heading </body>	

**Paragraphs **	<body>
	   Car Web Site
	  Welcome to the car web site.
	  Everything you want to know about
	  Car is in this web site
	
	   This is section on Honda
	  Honda is a Japanese car
	</body>
		

**LineBreaks
**
 	
**Horizontal Rules   **
	<body>
	   Car Web Site
	  <!--To add a line break inside the
	  middle of a paragraph, use
	-->
	  Welcome to the car web site.
	
	  Everything you want to know about
	  Car is in this web site
	
	  <!--to create break between themes-->
	
	  <!-- new break space
	  test test-->
	   This is section on Honda
	  Honda is a Japanese car
	</body>
		

**Bold
Italic <i>
Underlined <u>**
	<body>
	   Car Web Site
	  <!--to bold we use -->
	   Welcome to the car web site
	  . Everything you want to know about
	  Car is in this web site
	
	  <!--to underlined we use <u> </u>-->
	   This is for <u>Honda</u>
	  <!--to italic we use <i> </i>-->
	   Honda is a <i>Japanese</i> car
	</body>
		

Comment
 	
<!-- -->
	<body>
	  <h1>Car Web Site</h1>
	  Welcome to the car web site
	  . Everything you want to know about
	  Car is in this web site
	  <!--This is the comment part-->
	  <h2>This is section on <u>Honda</u></
	  h2>
	  Honda is a Japanese car
	</body>
		<!-- -->

Align & Style

```
<body>
    <h1>Car Web Site</h1>
    <p>Welcome to the <b>car web site</b>
    . Everything you want to know about
    Car is in this web site</p>
    <h2 align="left" style="color: blue">
    This is section on <u>Honda</u></h2>
    <p>Honda is a <i>Japanese</i> car</p>
</body>
```

<h1>Car Web Site</h1>

Welcome to the **car web site**. Everything you want to know about Car is in this web site

<h2 align="left" style="color: blue">This is section on <u>Honda</u></h2>

Honda is a *Japanese* car

IMAGES

<img src="">

```
<body>
    <h1>Car Web Site</h1>
    <p>Welcome to the <b>car web site</b>
    . Everything you want to know about
    Car is in this web site</p>
    <!--img; element|src; to find image
    location|alt;text description if you
    can not seen it-->
    <img src="Mustang_2015.png" width="
    100" height="100" alt="car image">
    <h2>This is section on <u>Mustang</u>
    </h2>
    <p>Mustang is an <i>American</i> car
    </p>
</body>
```

<h1>Car Web Site</h1>

Welcome to the **car web site**. Everything you want to know about Car is in this web site

<h2>This is section on <u>Mustang</u></h2>

Mustang is an *American* car

<u>LIST</u>
- **Ordered L.**
<ol>
- **Unordered L.**
<ul>
- **Definition L.**
<dl>
<dt>
<dd>
- **Nested L.**

```
<body>
    <ol><!--Ordered List-->
        <li>Honda</li>
            <ul><!--Nested List-->
                <li>Civic</li>
                <li>Accord</li>
            </ul>
        <li>Toyota</li>
        <li>Nissan</li>
    </ol>
    <ul><!--Unordered List-->
        <li>Mercedes</li>
        <li>BMW</li>
        <li>Jaguar</li>
    </ul>
    <dl><!--Definition List-->
        <dt>HTML</dt><!--def. Term-->
        <dd>HTML stands for <b>H</b>yper<
        b>T</b>ext<b>M</b>arkup<b>L</b>
        anguage</dd>
        <dt>JavaScript</dt>
        <dd>JavaScript is oop language</
        dd><!--definition-->
    </dl>
</body>
```

1. Honda
    - Civic
    - Accord
2. Toyota
3. Nissan

- Mercedes
- BMW
- Jaguar

HTML
: HTML stands for **H**yper**T**ext**M**arkup**L**ang

JavaScript
: JavaScript is oop language

<table>
<tr>
<td rowspan="2"><u>LINKS</u>

<a href=””></td>
<td>

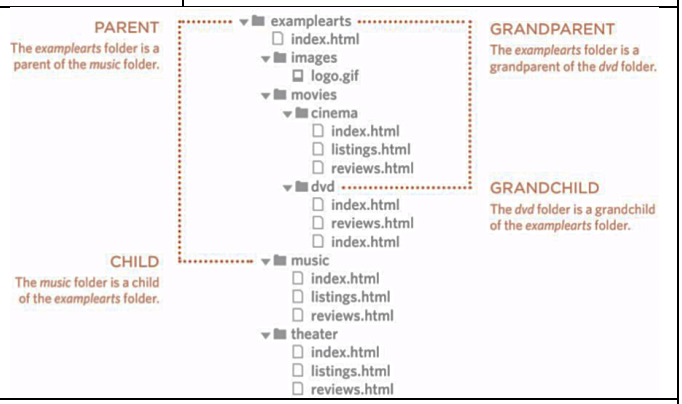
```
<body>
    <!--Linking to outside of the page-->
    <p>please click on this link
        <a href="https://www.cars.com/">Car</a>
    </p>

    <!--Opening Links in a New Window-->
    <p>please click on this link
        <a href="https://www.carmax.com/" target="
        _blank">CarMax</a>
    </p>
</body>
```

</td>
<td>please click on this link Car

← OPEN LINKs in a NEW WINDOW
target attribute and its value is
_blank</td>
</tr>
<tr>
<td>

```
<body>
    <!--Linking to other pages on the same sites
    -->
    <p>Car review:
    <!--if all pages of the site are in the same
    folder, href attribute is just the name of
    the file-->
        <li><a href="index.html/">Home</a></li>
        <li><a href="about.html/">Home</a></li>
        <li><a href="contact.html/">Home</a></li>
    </p>
</body>
```

</td>
<td>Car Reviews:

• Home
• About
• Contact</td>
</tr>
<tr>
<td>Directory Structure</td>
<td>On larger websites it is a good idea to organize the code by placing the pages for each different section of the site into a new folder. Folders on a website are sometimes referred to as directories.</td>
<td colspan="2"> 
</td>
</tr>
<tr>
<td rowspan="5">Relative URLs</td>
<td>SAME FOLDER
To link to a file in the same folder, just use the file name</td>
<td colspan="2">To link to music reviews from music homepage
Reviews</td>
</tr>
<tr>
<td>CHILD FOLDER
name of the child folder/ the file name.</td>
<td colspan="2">To link to music listings from music homepage
Listings</td>
</tr>
<tr>
<td>GRANDCHILD FOLDER
Use the name of the child folder / the name of the grandchild folder/file name</td>
<td colspan="2">To link to DVD reviews from homepage
Reviews</td>
</tr>
<tr>
<td>PARENT FOLDER
Use ../ to indicate the folder above the current one / file name</td>
<td colspan="2">To link to homepage from music reviews
Home</td>
</tr>
<tr>
<td>GRANDPARENT FOLDER
Repeat the ../ to indicate that you want to go up two folders / file name</td>
<td colspan="2">To link to homepage from DVD reviews
Home</td>
</tr>
</table>

TABLE

<table>
<tr>
<td>
<th>

```
<body>
    <table><!--Element-->
        <tr><!--Tag to table ROW-->
            <th>The Planets</th><!--Heading-->
            <th>The Oceans</th>
            <th>The Deserts</th>
        </tr>
        <tr>
            <td>Earth</td><!--Data-->
            <td>Pacific</td>
            <td>Sahara</td>
        </tr>
    </table>
    <br>
    <br>
    <table>
        <tr>
            <th></th>
            <th>Saturday</th>
            <th>Sunday</th>
        </tr>
        <tr>
            <th>Tickets Sold</th>
            <td>120</td>
            <td>135</td>
        </tr>
        <tr>
            <th>Total Sales</th>
            <td>$600</td>
            <td>$1000</td>
        </tr>
    </table>
</body>
```

➢The **<table>** element → to add tables
➢ Each row is created with the **<tr>** element.
➢Inside each row →cells represented by the **<td>** element (or **<th>** if it is a header).
➢ make cells of a table span more than one row or column → **rawspan** and **colspan** attributes.

The Planets	The Oceans	The Deserts
Earth	Pacific	Sahara
Batuhan	Atlantic	Gobi

	Saturday	Sunday
Tickets Sold	120	135
Total Sales	$600	$1000

```
<body>
    <table>
        <tr>
            <th></th>
            <td>9am</td>
            <td>10am</td>
            <td>11am</td>
            <td>12am</td>
        </tr>
        <tr>
            <th>Monday</td><!--Data-->
            <td align="center" colspan="3">HTML</td>
            <td rowspan="2">JS</td>
        </tr>
        <tr>
            <th>Tuesday</td>
            <td>SQL</td>
            <td>JS</td>
            <td>SQL</td>
        </tr>
    </table>
</body>
```

SPANNING COLUMNS & ROWS

	9am	10am	11am	12am
Monday		HTML		JS
Tuesday	SQL	JS	SQL	

GROUPING TEXT & ELEMENT in a BLOCK

** **

```
<body>
    <div>
        <h3>This is first group</h3>
        <ul>
            <li>Java</li>
            <li>JS</li>
            <li>Ruby</li>
        </ul>
    </div>
    <div>
        <h3>This is second group</h3>
        <ul>
            <li>C++</li>
            <li>Python</li>
            <li>Go</li>
        </ul>
    </div>
</body>
```

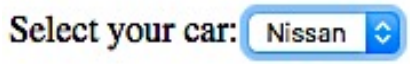
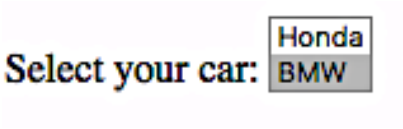
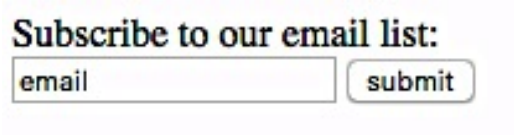
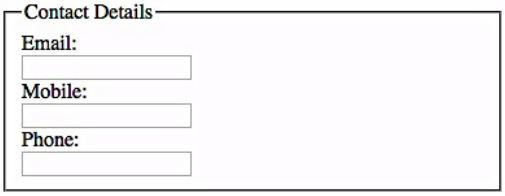
This is first group

- Java
- JS
- Ruby

This is second group

- C++
- Python
- Go

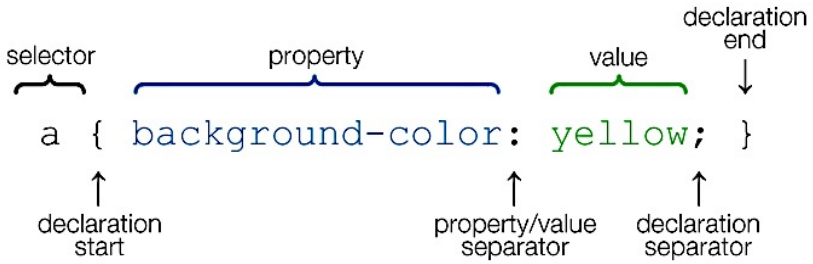
FORMS **<form>**	has referred to a printed document that contains spaces for you to fill in info. 1- Adding Text: Text Input (single Line), Text Area (Multi-Line), Password Input 2- Making Choices: Radio Buttons, Checkboxes, Drop-down boxes 3- Submitting Forms: Submit and Image Buttons 4- Uploading Files	
Text Input **<input>** **1-type="text"** **2-name**	<form> Username: <!--Add a Label--> <input type="text" name="username" size="15" maxlength="30"/> </form>	Username:
Password Input **<input>** **1-type="text"** **2-name**	<form> Password: <input type="text" name="username" value="password" /> <!--Label for username (it has to match)--> <label for="username">Username:</label> <input type="text" id="username" name="username" value="username" /> </form>	value="" Password: password Username: username
Text Area **<textarea>** **1-name="text"**	<form> Comments: <textarea name="comments" cols="20" rows="3">Enter Your Comments</textarea> </form>	Comments: Enter Your Comments
Radio Button **<input>** **1-type="radio"** **2-name=""** **3-value=""**	<form> Choose your programming language: <input type="radio" name="program" value="Java"> Java</> <!--Default check--> <input type="radio" name="program" value="JS" checked="checked">JS</> <input type="radio" name="program" value="HTML"> HTML</> </form>	Choose your programming language: Java JS HTML
Check Button **<input>** **1-type="checkbox"** **2-name=""** **3-value=""**	<form> What is your subject <input type="checkbox" name="sport" value=""> Football <!--Default check--> <input type="checkbox" name="sport" value="" checked="checked">Basketball <input type="checkbox" name="sport" value=""> Baseball </form>	What is your subject Football Basketball Baseball

Drop-Down List Box **<select>** **1-name=””** **<option value="">**	<form> Select your car: <select name="cars"> <option value="">Honda</option> <option value="" selected="">Nissan</option> <option value="">Acura</option> <option value="">Toyota</option> </select> </form>	
Multiple Select Box **<select>** **1-name=””** **2-size=””** **<option value="">**	<form> Select your car: <select name="cars" size="2" multiple="multiple"> <option value="">Honda</option> <option value="" selected="">BMW</option> <option value="">Acura</option> <option value="">Toyota</option> </select> </form>	
Submit Button **<input>** **1-name=”text”** **<input>** **1-name=”submit”**	<form> Subscribe to our email list: <input type="text" name="subsr" value="email"> <input type="submit" name="submit" value="submit"> </form>	
Grouping Form Elements **<fieldset>** **<legend>** **<input>** **1-type=”text”** **2-name=**	<form> <fieldset> <legend>Contact Details</legend> <label>Email: <input type="text" name="email"></label> <label>Mobile: <input type="text" name="mobile"></label> <label>Phone: <input type="text" name="mobile"></label> </fieldset> </form>	
Search Button **<input>** **1-type=”search”** **1-name=””**	<form> Search: <input type="search" name="search" value="Keyword"> <input type="submit" name="submit" value="Search"> </form>	

CSS (Cascading Style Sheets)

It allows you to create rules that specify how the content of an element should appear.

- It describes how HTML elements are to be displayed on screen, paper, or in media.
- It saves a lot of work. It can control the layout of multiple web pages all at once.
- External style sheets are stored in its files.



selector: indicate which element the rule applies to.
declarations: indicate how the elements referred to in the selector should be styled
properties: indicate the aspects of the element you want to change
values: specify the settings you want to use for the chosen properties.

Using Internal CSS

<head>
<style>
1-type="text/css"

```
<head>
    <style type="text/css">
        body{
            background-color: yellow;
        }
        h1, h2{
            background-color: blue;
            color: white;
        }
        h2{
            color: red;
        }
        p{
            color: green;
            font-size: 25px;
            font-family:Tahoma;
            font-style: italic;
        }
    </style>
</head>
```

Using External CSS

<link>
href="" (path to the CSS file)
type="" (type of doc being linked to)
rel="" (relationship)

!!!ExternalCSS must be in the same folder!!!

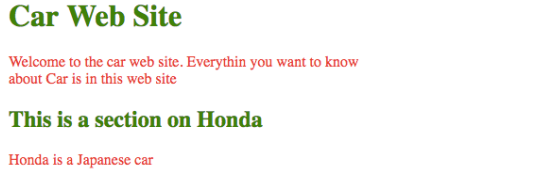
```
<head>
    <link rel="stylesheet" type="text/css" href="
    externalCSS.css">
</head>
```

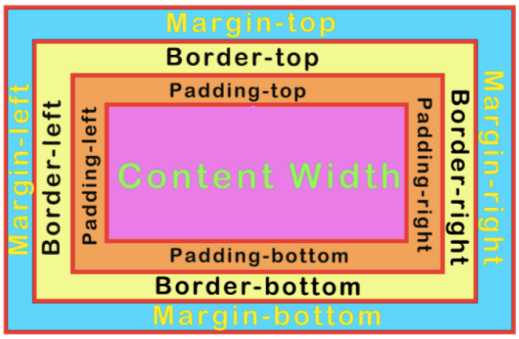
externalCSS.css

```
body{
    background-color: yellow;
}
h1, h2{
    background-color: blue;
    color: white;
}
h2{
    color: red;
}
p{
    color: green;
    font-size: 25px;
    font-family:Tahoma;
    font-style: italic;
}
```

Inheritance

```
body{
    color:green;
}
p{ /*overriding*/
    color:red;
}
```



CSS Properties	1– **Font** Properties 2– **Padding** Properties 3– **Border** Properties 4– **Margin** Properties	

1- Font Properties	`body{`
	`  /*This is the long way*/`
	`  font-style: italic;`
	`  font-size: 24px;`
	`  font-family: Tahoma;`
	`  font-weight: 500;`
	`  font-variant: small-caps;`
	`}`
	`p{`
	`  /*This is the short way*/`
	`  font:italic small-caps 500 24px Arial;`
	`}`
		CAR WEB SITE
		WELCOME TO THE CAR WEB SITE. EVERYTHIN YOU WANT TO KNOW ABOUT CAR IS IN THIS WEB SITE
		THIS IS A SECTION ON HONDA
		HONDA IS A JAPANESE CAR

2- Padding Properties	`h1{`
	`  background-color:red;`
	`  padding-left: 10px;`
	`  padding-top: 10px;`
	`  padding-bottom: 10px;`
	`  padding-right: 10px;`
	`}`
	`h2{`
	`  background-color: blue;`
	`  padding: 20px 40px 30px 20px;`
	`/*padding: top right bottom left;`
	`  if padding:25px 50px 75px;`
	`  top right&left bottom`
	`  if padding:25px 50px;`
	`  top&bottom right&left`
	`  if padding:25px;`
	`  All 25px */`
	`}`
		Car Web Site
		Welcome to the car web site. Everythin you want to know about Car is in this web site
		This is a section on Honda
		Honda is a Japanese car

3- Border Properties	`<style type="text/css">`
	`  h1{`
	`  background-color:red;`
	`  padding:20px 30px;`
	`  /*border: size style color;*/`
	`  border:10px ridge blue;`
	`  border-left-color: yellow;`
	`  border-top-width: 2px;`
	`  }`
	`  h2{`
	`  border-top-style: dotted;`
	`  border-bottom-style: dashed;`
	`  background-color: blue;`
	`  padding: 20px 40px 30px 20px;`
	`  }`
	`</style>`
		Car Web Site
		Welcome to the car web site. Everythin you want to know about Car is in this web site
		This is a section on Honda
		Honda is a Japanese car

4- Margin Properties

```
<style type="text/css">
    h1{
        background-color:red;
      /*margin:top right bottom left;*/
        margin:0px 150px 0px 20px;
        padding:20px 30px;
        border:10px ridge blue;
        border-left-color: yellow;
        border-top-width: 2px;
    }
    h2{
        background-color: blue;
        padding: 20px 40px 30px 20px;
    }
</style>
```

SELECTOR	MEANING	EXAMPLE
UNIVERSAL SELECTOR	Applies to all elements	*** { }**
TYPE SELECTOR	Matches element names	**h1, h2, h3 { }**
CLASS SELECTOR	Matches an element whose class att. has a value that matches the one specified after the period (or full stop) symbol	**.note { }** –target any element whole class attribute has a value of note **p.note { }** –target only <p> elements…
ID SELECTOR	Matches an element whose id att has a value that matches the one specified after the pound or hash symbol	**# introduction { }**
CHILD SELECTOR	Matches an element that is a direct child of another	**li>a { } –** targets any <a> elements that are children of an element (but not other <a> element in the page)
DESCENDANT SELECTOR	Matches an element that is a descendent of another specified element (not just a direct child of that element)	**p a { } –** targets any <a> elements that sit inside a <p> element, even if there are other elements nested between them
ADJACENT SIBLING SELECTOR	Matches an element that is the next sibling of another	**h1+p { } –** targets the first <p> element after any <h1> element (but not other <p>)
GENERAL SIBLING SELECTOR	Matches an element that is a sibling of another, although it does not have to be the directly preceding element	**h1-p{ } –** if you had two <p> elements that are siblings of an <h1> element, this rule would apply to both.

CLASS & ID SELECTOR

```
<body>
    <h1>Car Web Site</h1>
    <!--for CLASS selector-->
    <p class="p1">Welcome to the car web
    site.</p>

    <h2>This is a section on Honda</h2>
    <!--for ID selector-->
    <p id="p2">Honda is a Japanese car</p>
</body>
```

```
<head>
    <style type="text/css">
        /*CLASS selector*/
        .p1{
            color: blue;
        }
        /*ID selector*/
        #p2{
            font-style: italic;
        }
    </style>
</head>
```

DOM (Document Object Model)

getElementById( ) - Select elements by their id

index.html	script.js

`<body>`
 	
`<p id="para1">Some text for this class</p>`
`<input type="button" onclick="changeStyle()" value="Submit">`
 	
`</body>`
	`function changeStyle(){`
	`document.getElementById("para1").style.color="red";`
	`document.getElementById("para1").style.backgroundColor="blue";`
	`document.getElementById("para1").style.fontStyle="italic";`
	`}`

getElementsByTagName( ) - returns a collection of all elements in the document with the specified tag name.

```
<body>
    <p id="para1">Some text for this class-1</p>
    <p id="para2">Some text for this class-2</p>
    <p id="para3">Some text for this class-3</p>
    <p id="para4">Some text for this class-4</p>

    <input type="button" onclick="changeStyle()"
    value="Submit">
    <!--onclick is an EVENT-->
</body>
```

```
function changeStyle(){
                    //I access to my DOM
    var paragraph = document.getElementsByTagName("p")

    for (var i=0;i<paragraph.length;i++){
        paragraph[i].style.color="red";
    }
}
```

getElementsByClassName( ) - Select elements by their class name

```
<body>
    <p class="para">Some text for this class-1</p>
    <p class="para">Some text for this class-2</p>
    <p class="para">Some text for this class-3</p>
    <p class="para">Some text for this class-4</p>

    <input type="button" onclick="changeStyle()" value="Submit">
</body>
```

```
function changeStyle() {
    var paragraph = document.getElementsByClassName("para");
    //paragraph[0].style.color="blue";
    for (var i = 0; i < paragraph.length; i++) {
        paragraph[i].style.color = "green";
    }
}
```

innerHTML - innerHTML property sets or returns the HTML content of an element

```
<body>
    <p id="myP">I am a paragraph</p>
    <p>Click the button get the HTML content of the p element</p>
    <input type="button"  onclick="changeStyle()" value="Submit">

    <p id="demo"></p>
</body>
```

```
function changeStyle() {
    var x = document.getElementById("myP").innerHTML;
    document.getElementById("demo").innerHTML = x;

}
```

TASK: Hello There. -->class="para"
This is Mike Smith -->class="para"
Hello There.This is Mike Smith

```
<body>
    <p class="para">Hello There.</p>
    <p class="para">This is Mike Smith</p>
    <input type="button"  onclick="changeStyle()" value="Submit">

    <p id="para"></p>
</body>
```

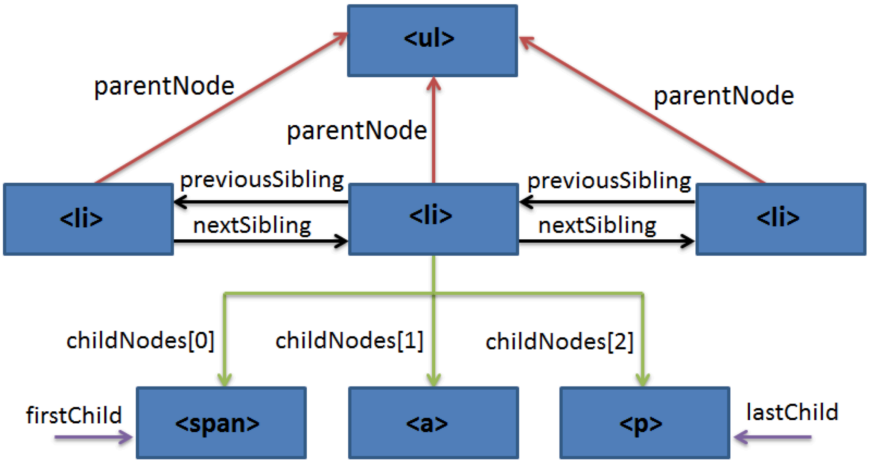
```
function changeStyle() {
    var arr = document.getElementsByClassName("para");
    var text1 = arr[0].innerHTML;
    var text2 = arr[1].innerHTML;

    document.getElementById("para").innerHTML=`${text1} ${text2}`;
}
```

querySelector( ) - Get the first element in the document

```
<body>
    <p>Hello There.</p>
    <p>This is Mike Smith</p>
    <p>I love JavaScript</p>

    <input type="button" onclick="changeStyle()" value="Submit">

    <p id="para"></p>
</body>
```

```
function changeStyle(){
    document.querySelector("p").style.color="red";
}
```

How to TRAVERSE the DOM ? (We can travel between elements)



index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <script src="script.js"></script>
</head>

<body>
    <header id="main header" class="bg-success">
        <div class="container">
            <h1 id="header-title">Item Lister<span style="
                display:none">123</span></h1>
        </div>
    </header>
        <div class="container">
            <div id="main" class="card-body">
                <h2 class="title">Add Items</h2>
                <form class="form-inline">
                    <input type="text" class="form-control">
                    <input type="button" onclick="myFunction
                        ()" value="Try it">
                </form>
                <h2 class="title">Items</h2>
                <ul id="items" class="list-group">
                    <li class="list-group-item">Item1</li>
                    <li class="list-group-item">Item2</li>
                    <li class="list-group-item">Item3</li>
                    <li class="list-group-item">Item4</li>
                </ul>
            <p></p>
            </div>
        </div>
</body>
</html>
```

script.js

```
function myFunction() {
    var itemlist = document.querySelector("#items");
    //console.log(itemlist);

    //parentNode & parentElement
    //console.log(itemlist.parentNode);
    itemlist.parentNode.style.color="green";
    itemlist.parentElement.style.color="green"; //1

    //childNodes - Not good
    //console.log(itemlist.childNodes)

    //firstElementChild
    //console.log(itemlist.firstElementChild)
    itemlist.firstElementChild.innerHTML = "Hello"; //2

    //lastElementChild
    //console.log(itemlist.lastElementChild);
    itemlist.lastElementChild.style.color="red"; //3

    //nextElementChild
    //console.log(itemlist.nextElementChild);
    //itemlist.nextElementChild.style.color="red";

    //previousSibling
    //console.log(itemlist.previousSibling);
}
```

//1

Item Lister

Add Items

Try it

Items

- Item1
- Item2
- Item3
- Item4

//2

Items

- Hello
- Item2
- Item3
- Item4

//3

Items

- Item1
- Item2
- Item3
- Item4

TASK

MINI CALCULATOR PROGRAM

Enter first number:

Enter second number:

Result is:

Addition | Subtraction | Multiplication | Division

index.html

```
<!DOCTYPE html>
<html>

<head>

    <title>Arithmetic Operations</title>
    <script src="cal.js" type="text/javascript"></script>

</head>

<body>
    <h2>MINI CALCULATOR PROGRAM</h2>
    <b>Enter first number: </b><input type="text" size="10"
    id="t1"><br><br>
    <b>Enter second number: </b><input type="text" size="10"
     id="t2"><br><br>
    <b>Result is: </b><input type="text" size="10" id="res">
    <br><br>
    <input type="button" value="Addition" onclick="add()">
    <input type="button" value="Subtraction" onclick="sub()"
    >
    <input type="button" value="Multiplication" onclick="mul
        ()">
    <input type="button" value="Division" onclick="div()">

</body>

</html>
```

cal.js

```
var res;

function add(){
    var num1 =parseInt(document.getElementById("t1").value);
    var num2 =parseInt(document.getElementById("t2").value);
    res=num1+num2;
    document.getElementById("res").value = res;
}

function sub(){
    var num1 =parseInt(document.getElementById("t1").value);
    var num2 =parseInt(document.getElementById("t2").value);
    res=num1-num2;
    document.getElementById("res").value = res;
}

function mul(){
    var num1 =parseInt(document.getElementById("t1").value);
    var num2 =parseInt(document.getElementById("t2").value);
    res=num1*num2;
    document.getElementById("res").value = res;
}

function div(){
    var num1 =parseInt(document.getElementById("t1").value);
    var num2 =parseInt(document.getElementById("t2").value);
    res=num1/num2;
    document.getElementById("res").value = res;
}
```