WEB TECHNOLOGY DEFINITION AND TERMINOLOGIES

TEKNOTURF

Web technology refers to the way computers/devices communicate with each other using mark up languages. It establishes communication between different types of devices, across the internet. It also helps create, deliver and manage web content using hypertext markup language (HTML). *

Internet

A large network of networks connecting thousands of computers that can communicate with one another.

World Wide Web

A system of 'interlinked hypertext documents' that can be accessed by users, via the internet.

Web Browser

An application software on your device to access web.

Web Server

Websites are hosted, here. The request for information is received, processed, and the response is sent back.

*https://www.w3.org/html/

WEB TECHNOLOGY - TERMINOLOGIES

TEKNOTURF

WEB PAGES

A Web Page is a HTML document displayed as a single page in a browser. It can be connected to other pages

Ex:

https://en.wikipedia.org/wiki/Wikipedia:About

WEBSITES

A **Web site** is a collection of several web pages, all connected together. It may contain text, images, audio and video and usually present at the same internet address. The first page of a website is called the home page

Ex: https://en.wikipedia.org/

WEB APPLICATION

Web application is a piece of software that can be accessed by the browser. It ensures authentication before access and requires a highly sophisticated server to manage users' requests.

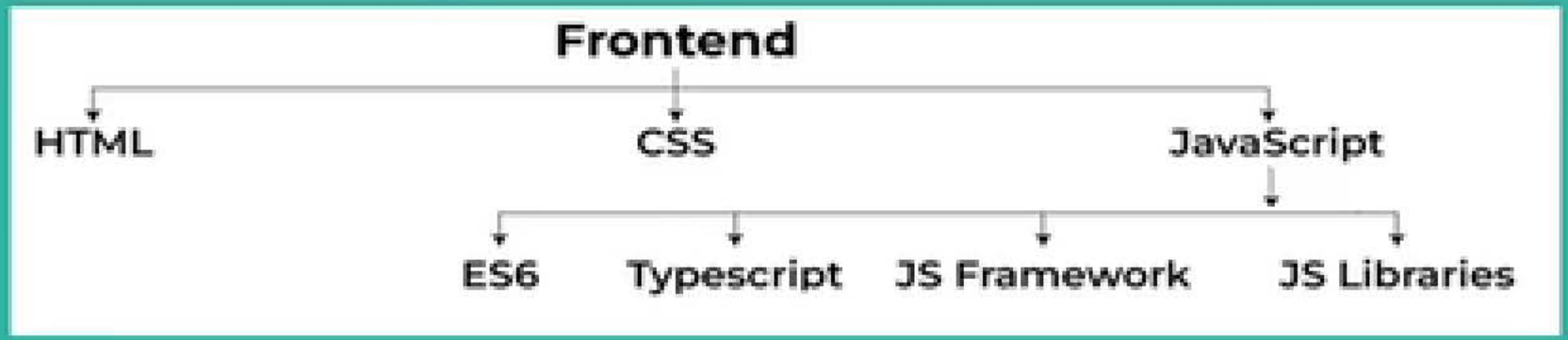
Ex: Google Apps, Amazon, YouTube *

WEB DEVELOPMENT

TEKNOTURF

- Web Development refers to the building, creating, and maintaining of websites. It includes aspects such as web design, web publishing, web programming, and database management. *
- Frontend Web Development refers to building the frontend - ie., the 'client side' of the application; the part of website that a user directly interacts with. *

The figure represents the frontend programming languages.



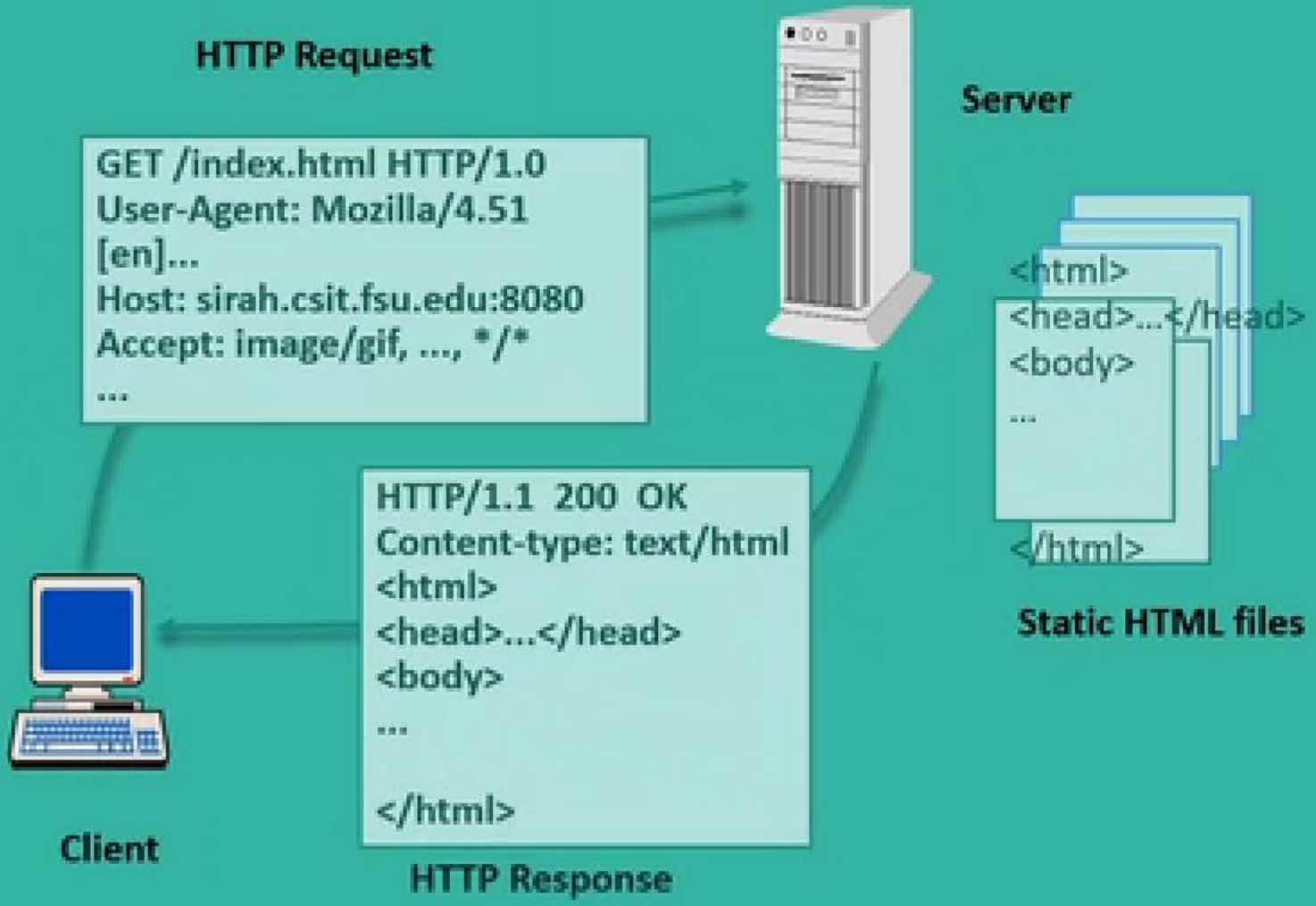
- Backend Web Development refers to the server side of a website. Java, PHP, Python, Node.js, etc., are the languages used for backend web development. .

STATIC WEB PAGE

TEKNOTURF

- A static web page or static website is the basic type of website whose content will be static
- It is usually written in plain HTML
- Pre-built content is the same every time the page is loaded because it sends exactly the same response for every request
- The content only changes when someone publishes and updates the file (and sends it to the web server)
- Flexibility is the main advantage of static website

REQUESTING A STATIC HTML DOCUMENT



DYNAMIC WEB PAGE

TEKNOTURF

- **Client side scripting** generates content at the client computer on the basis of user input. The web browser receives the web page from the server and processes the code within the page to render information to the user.

- In **server side scripting**, the software runs on the server, processes the code and once completed, the pages are sent to the client.

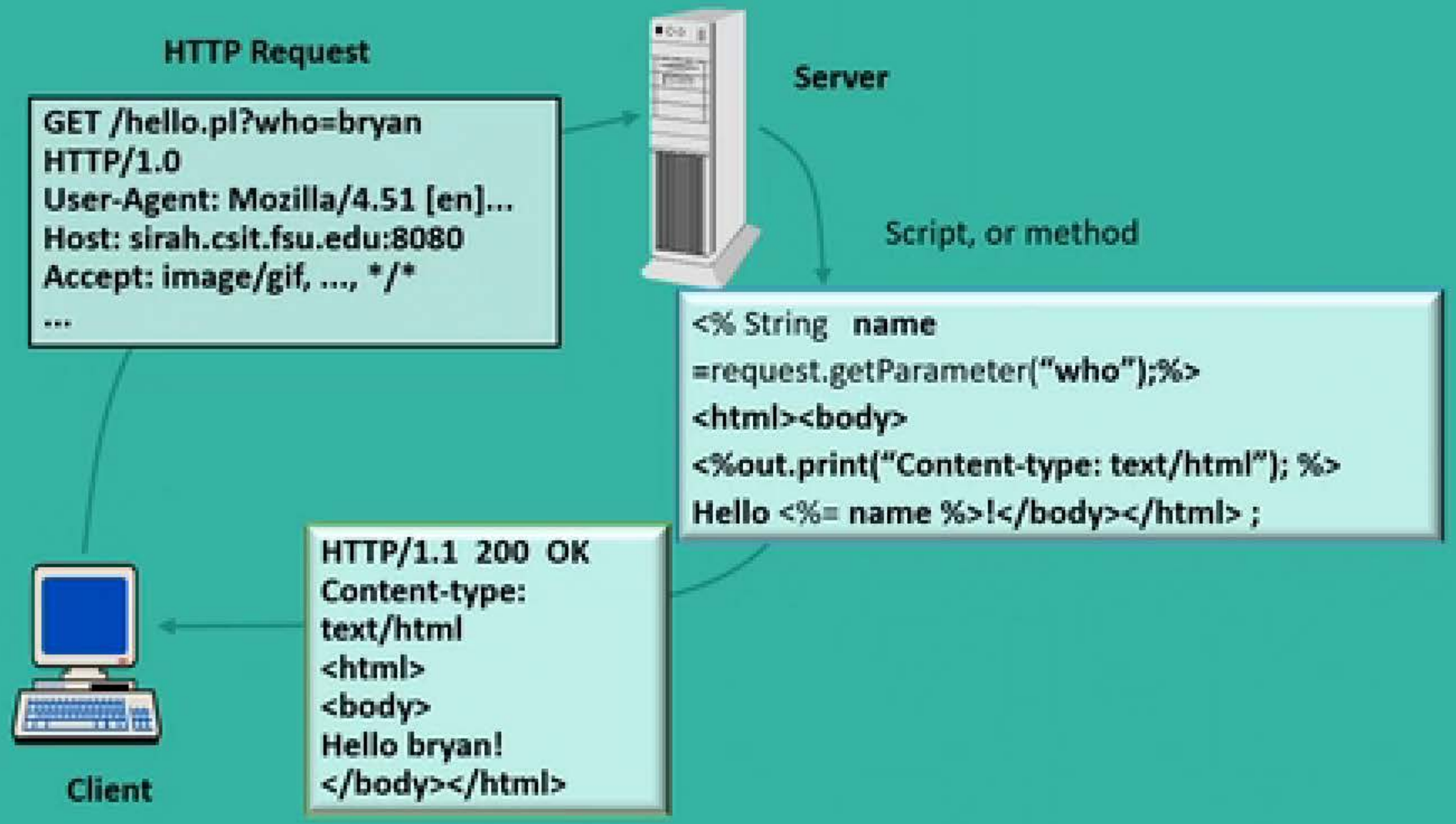
Dynamic website uses client-side scripting or server-side scripting, or both to generate dynamic content.

Dynamic web page is a page whose content changes dynamically. It shows different information at different points of time.

It may generate different HTML for each of the requests.

It accesses content from a database or Content Management System (CMS). When you alter or update the content of the database, the content of the website is also altered or updated.

DYNAMIC GENERATION OF HTML



INTRODUCTION TO HTML

TEKNOTURF

- HTML stands for **Hyper Text Markup Language.**
- **Hyper Text** - Refers to the way web pages are linked to each other
- **Markup** - Usage of tags to structure the web page
- HTML is used for developing web pages for applications. Web pages are text files containing HTML.

HTML is a normal text surrounded by bracketed *tags* that tell browsers how to display web pages

Pages end with ".htm" or ".html"

HTML Editor – A word processor that has been specialized to make the writing of HTML documents more effortless

HTML STRUCTURE

TEKNOTURF

01 HTML is comprised of "**elements**".

02 The structure begins with **<html>** and ends with **</html>**

03 **Elements (tags)** are nested one inside another.

04 HTML describes structure using two main sections: **<head>** and **<body>**

BASIC HTML TAGS

TEKNOTURF

HTML Tags :

<html>

<head>

<title>

<body>

<h1> -- <h6>

<p>

<p>

<hr>

<!-- -->

BASIC HTML TAGS

TEKNOTURF

Heading Tag

Used to provide headings in HTML file

Heading tags start from **<h1>** to **<h6>**

```
<html>
<body>
        <h1>HTML</h1>
        <h2>Basic Tags</h2>
        <h3>Heading</h3>
        <h4>Provides heading section in the web page</h4>
        <h5>Starts from h1 to h6</h5>
        <h6>Can be given in any order</h6>
</body>
```

BASIC HTML TAGS CONT.

TEKNOTURF

Paragraph tag

The <p> tag defines a paragraph.

Browsers automatically add **space** before and after each <p> element.

Break tag

```
<body>
    <p><h2>Learn HTML</h2> </p>
    <p>The paragraph tags are used to define a  <br>block of text as a paragraph</p>
</body>
```

Exit full screen

BASIC HTML TAGS CONT.

TEKNOTURF

Horizontal Rule

The **<HR>** element causes the browser to display a horizontal line (rule) in your document.

```
<html>
<body>
        <p><h2>Learn HTML</h2> </p>
        <p>The paragraph tags are used to define a
        <br>block of text as a paragraph</p>
        <hr>
        <p>The tag HR gives a horizontal line </p>
</body>
</html>
```

BASIC HTML TAGS CONT.

TEKNOTURF

<!DOCTYPE>

- Gives instruction to the web browser about the version of HTML used
- Doctype must be the first line of the web page

```
<!DOCTYPE html>
<html>
<body>
    Contents of Web Page
        </body>
</html>
```

Represents HTML5

BASIC HTML TAGS CONT.

TEKNOTURF

Comments

- Tag used to add information to the HTML page, which will not be displayed in the browser
- Tag : **<!-- -->**

```
<!DOCTYPE html>
<html>
<body>
<!-- body tag contains the details that is to be displayed in the web page -->
    Contents of Web Page
        </body>
</html>
```

Will not be displayed in browser

HTML5

TEKNOTURF

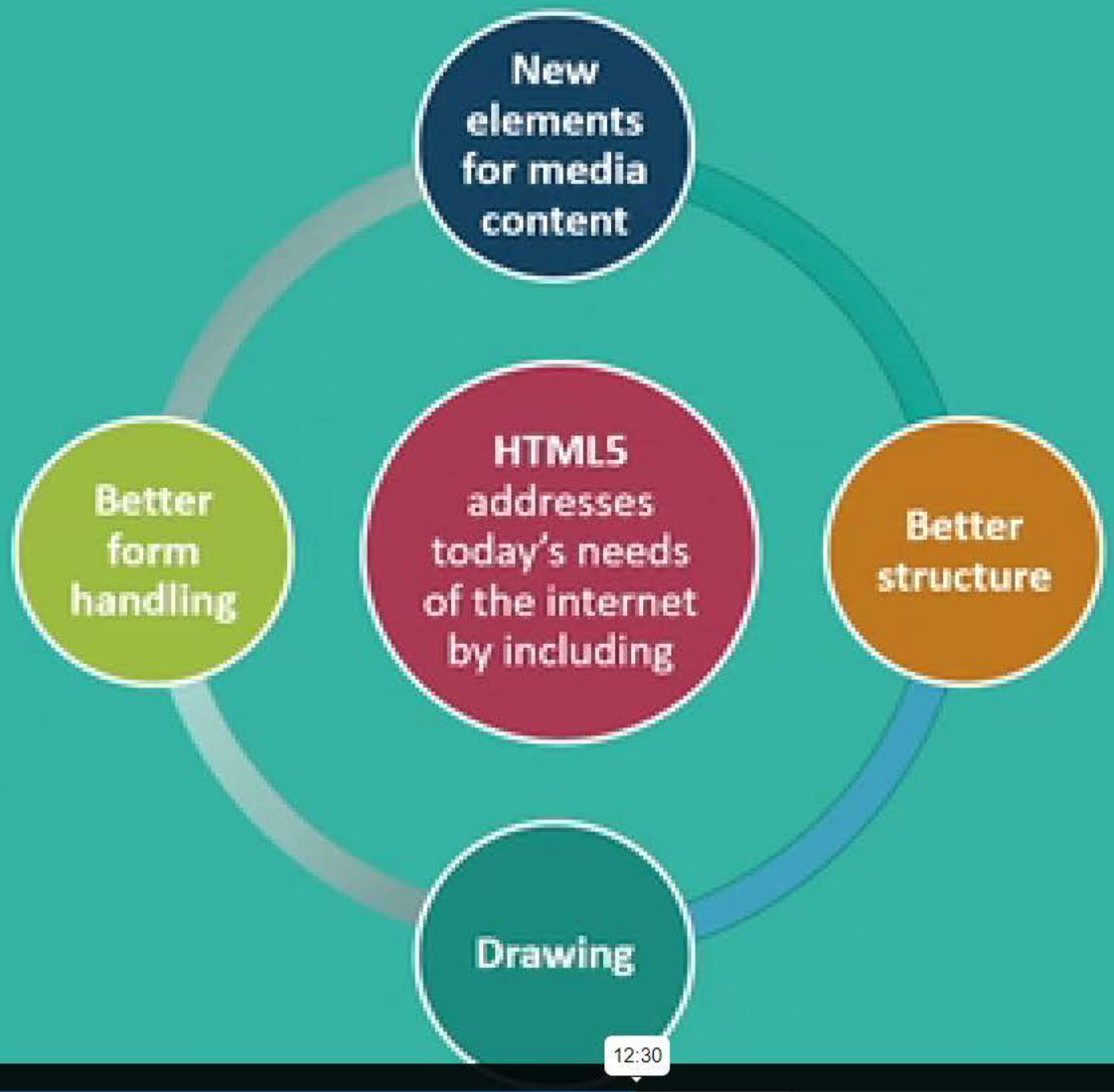
A bit of HTML

A whole sprinkling of JavaScript

A dash of CSS

HTML5 CONT.

TEKNOTURF



12:30

HTML5 CONT.

TEKNOTURF

Browser Support

The latest versions of
- –Apple Safari
- –Google Chrome
- –Mozilla Firefox
- –Opera

All support many HTML5 features

The mobile web browsers that come pre-installed on iPhones, iPads, and Android phones all have excellent support for HTML5.

NEW FEATURES

TEKNOTURF

- Better support for local offline storage
- The canvas element for sketching
- New content specific elements
  - article, footer, header, navigation, section, etc.
- New form controls like: calendar, date, time, email, url, search, etc.
- The video and audio elements for media playback
  - Allows video and audio to be tagged easier as in : <video src=...> and <audio src=...>
- Forms 2.0 and client-side validation
- Native browser support for audio and video

TABLES

```
<table border="1">
    <colgroup>
        <col span="2" style="background-color:#81D5D9">
        <col style="background-color: #DDDB5F">
    </colgroup>
<tr>  <caption>Employee Details</caption> </tr>
<thead>
 <tr>
 <th>ID</th>
 <th>NAME</th> <th>AGE</th> <th>Address</th></tr>
</thead>
<tbody>
<tr> <td>101</td> <td>Johan</td> <td>20</td> <td rowspan="2">Ganapathy</td>
    </tr>
    <tr>    <td>102</td> <td>Mini</td> <td>19</td>  </tr>
    <tr>    <td>103</td> <td>Ivan</td> <td>23</td> <td>Coimbatore</td>   </tr>
    </tbody>
    <tfoot>
    <tr>
    </tfoot>
<td colspan="3">The total Employees</td> <td> 3 </td>     </tr>
</table>
```

Employee Details

ID	NAME	AGE	Address
101	Johan	20	Ganapathy
102	Mini	19	
103	Ivan	23	Coimbatore
The total Employees			3

HTML LIST TAGS

TEKNOTURF

Creates an ordered or unordered list for the contents of the web page

Ordered list

- Tag for defining the list item :
- Attribute : type
  - 1 -- numbered with number
  - A -- numbered with Capital case alphabets
  - a -- numbered with smaller case alphabets
  - I -- numbered with upper case roman letters
  - i -- numbered with lower case roman letters

EXAMPLE FOR LIST TAGS

```
<html>
<body>
    basic Tags
    <ol type="1">
        <li>paragraph</li>
        <li>heading</li>
        <li>line break</li>
        <li>horizontal break</li>
    </ol>
    formatting tags
    <ul style="list-style-type: square">
        <li>bold</li>
        <li>strong</li>
        <li>del</li>
        <li>code</li>
    </ul>
</body>
</html>
```

basic Tags

1. paragraph
2. heading
3. line break
4. horizontal break

formatting tags

- bold
- strong
- del
- code

Unordered list :

- list-style-type:disc -bullets (default)
- list-style-type:circle- circles
- list-style-type:square-squares
- list-style-type:none – nothing will appear

LINKS

TEKNOTURF

Hyperlink helps in navigation from one page to another web page or from one part of the page to the other part of the web page

<a> - anchor tag

- The hyperlink can be a text or image which is clickable

Attributes

- href: Defines the destination location
- Target : Defines where the targeted page must be opened

```
<body>
<a href="paragraph.html">Click Here</a>
</body>
```

LINKS – TARGET ATTRIBUTE

TEKNOTURF

Target Value	Description
_blank	Opens the linked page in a new window or tab
_self	Opens the linked page in the same frame from where it was clicked (default)
_parent	Opens the linked page in the parent frame
_top	Opens the linked page in the full body of the window
framename	Opens the linked page in a named frame

Internal links

Links can also be created inside large documents to simplify navigation.

HTML5 IMAGE TAG

<img>

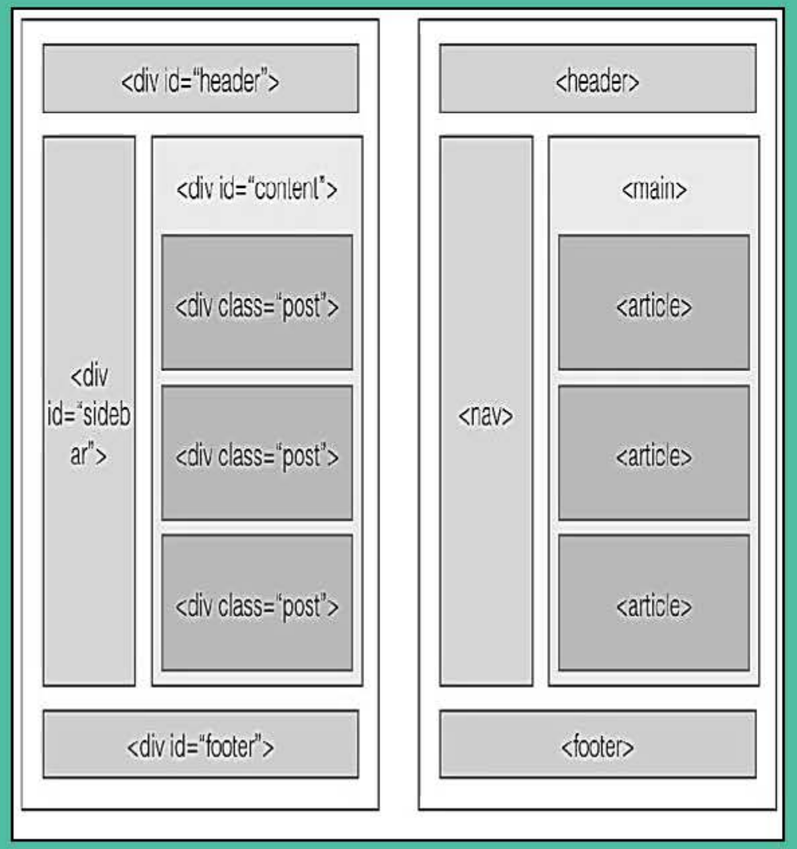
- Defines an image in HTML page
- Attributes
  - src – URL of the image
  - alt -- alternate text that is displayed, if the image is not displayed in the webpage
  - height – height of the image

```
<html>
<body>
    <img src="E:\HTML5\Tag.jpg" alt="Image can not be displayed" />
</body>
</html>
```

Image map is a map with clickable area
The **<map>** tag is used to define a client-side image-map.

SEMANTIC TAGS

TEKNOTURF



01 A Semantic element clearly describes its meaning to both the browser and the developer

02 Semantic elements: <form>, <table>, and <article>

03 Non-Semantic elements: and

04 The tag defines a division or a section in an HTML document.

HTML5 offers a set of new tags that provide the ability to mark up the sections of a document more descriptively than you could in HTML.

WHY USE SEMANTIC TAGS FOR A WEB PAGE?

TEKNOTURF

A semantically meaningful tag is way more powerful than a generic one

The browser can know which area of your site is the header or the footer.

The semantic elements are also used by search engines.

Site navigable for people with disabilities.

- People with learning difficulties might instruct their browser to always put the articles before the navigation

- It is easy to imagine Yahoo! giving lower weightage to content in footer elements, or extra weightage to content in the header.

Useful for code maintenance

NEW SEMANTIC ELEMENTS

TEKNOTURF

Specifies a header for a document or section

It represents a self-contained composition in a document, page, etc.

It groups links to other pages or to parts of the current page.

It defines a section that is tangentially related to the content around it.

```
<article>
   <header>  <h1>My website rocks !!!</h1> </header>
          <div id="content">
                <article>
    <header> <h2>This is the greatest page ever</h2> </header>
    Content of Article
                </article>
                <sidebar>
    <nav>
                        <ul> <li><a href="/home">Home Page</a></li>
                              <li><a href="/profile">Update Profile </a> </li>
                              <li><a href="/feedback">Feedback</a></li> </ul>
      </nav>
                 </sidebar>
                <aside> ..... </aside>
             </div>
            <footer>   Copyright 2012 Teknoturf Info Services Pvt., Ltd.,   </footer>
</article>
```

It is referred to the footer of a web page.

HTML FORMS

TEKNOTURF

1. HTML forms enable web applications to collect information from users.
2. Used to interact between a user and a web site or an application.
3. To build a form, the following are some of the HTML elements used

1. <form>
2. <label>
3. <input>
4. <textarea>
5. <button>

FORM – ELEMENTS

HTML forms always start with a **<form>** tag

Example

```
<form name="registration"    action="registered.html"    method="post"
          autocomplete="on"    novalidate="novalidate">
 </form>
```

Form tag has the following attributes

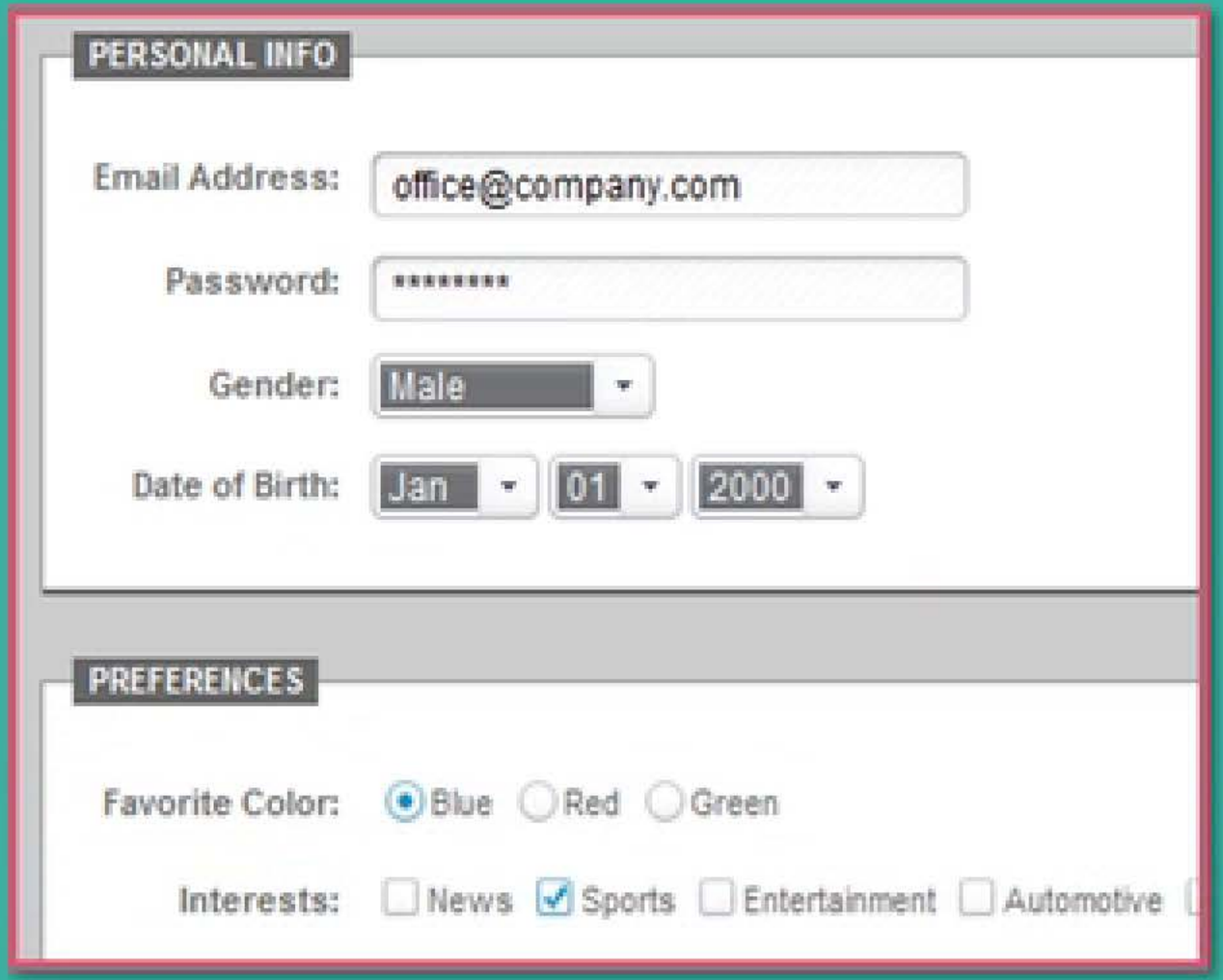
Attribute	Description
action	Specifies the destination on submission
method	Specifies the HTML method to be used on sending the form Get : the data is sent along the URL Post : the data is sent via the message body
autocomplete	Specifies whether a form should have auto complete on or off
novalidate	Specifies that the form should not be validated when submitted

FORM-INPUT ELEMENTS

TEKNOTURF

Form element contains

- <input>
- <textarea>
- <button>
- <select>
- <option>
- <optgroup>
- <fieldset>
- <label>



INPUT - ATTRIBUTES

TEKNOTURF

Type attribute is used to specify the type of the <input> element

Input types include

- text
- checkbox
- radio
- password
- button
- submit
- reset
- hidden
- file
- Image

The default type is text.

HTML5 FORM

TEKNOTURF

1. Form elements
2. Input types
3. Attributes
4. Form validation
5. Placeholder text

ATTRIBUTES

TEKNOTURF

Form Attributes:

- autocomplete
- novalidate

Input Attributes:

- autocomplete
- autofocus
- formaction
- formmethod
- formnovalidate
- height and width
- list
- min and max
- multiple
- pattern (regexp)
- placeholder
- required
- step

FORM ATTRIBUTES

TEKNOTURF

autocomplete

- Specifies whether a form or an input field should have autocomplete on or off.
- Automatically completes values based on values that a user has entered before.

novalidate

- Boolean attribute.
- Specifies that the form-data (input) should not be validated when submitted.

INPUT ATTRIBUTES – FORMACTION AND FORMMETHOD

TEKNOTURF

formaction	formmethod
• Specifies the URL of a file that will process the input control when the form is submitted. • Overrides the action attribute of the <form> element.	• This attribute defines the HTTP method for sending form-data to the action URL. • The formmethod attribute overrides the method attribute of the <form> element.

INPUT ATTRIBUTES – FORMNOVALIDATE AND AUTOFOCUS

TEKNOTURF

formnovalidate

- The formnovalidate attribute is a Boolean attribute.
- It specifies that the <input> element should not be validated when submitted.
- Overrides the novalidate attribute of the <form> element.

autoFocus

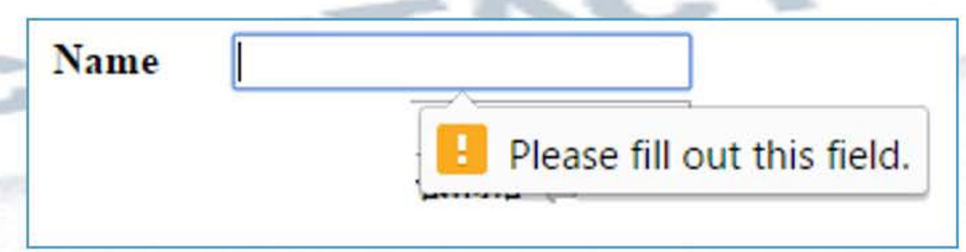
- The autofocus attribute is a Boolean attribute.
- It specifies that an <input> element should automatically get focus when the page loads.
- Only one form element can have autofocus in a given page.

INPUT ATTRIBUTE – REQUIRED

TEKNOTURF

required

- The Boolean required attribute tells the browser to submit the form only if the field in question is filled out correctly.
- If a required field is empty or invalid, the form will fail to submit, and focus will move to the first invalid form element.
- The required attribute can be set on any input type except button, range, color, and hidden, all of which generally have a default value.



LIST AND DATALIST

TEKNOTURF

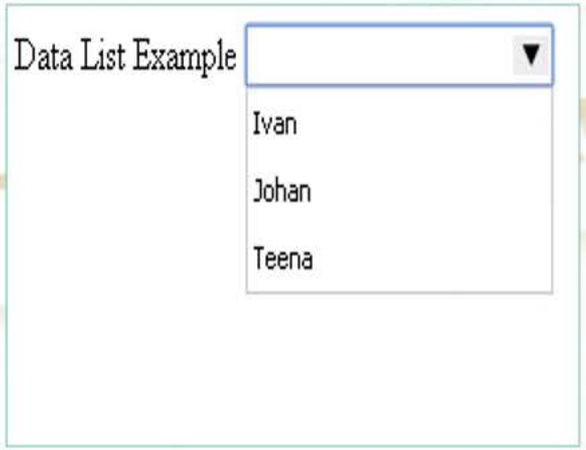
List

Used to bind the datalist created with the input element

Datalist

- The <datalist> tag specifies a list of pre-defined options for an <input> element.
- The <datalist> tag is used to provide an "autocomplete" feature on <input> elements. Users will see a drop-down list of pre-defined options as they input data.

```
<html>
<body>
Data List Example
<datalist id="names">
          <option value="Ivan"></option>
          <option value="Johan"></option>
          <option value="Teena"></option>
 </datalist>
 <input list="names" name="name" />
</body>
</html>
```



INPUT ATTRIBUTE – PLACEHOLDER

TEKNOTURF

1. The placeholder attribute allows a short hint to be displayed inside the form element, telling the user what data should be entered in that field

2. The placeholder text disappears when the field gains focus and reappears on blur if no data was entered.

INPUT ATTRIBUTE – MULTIPLE

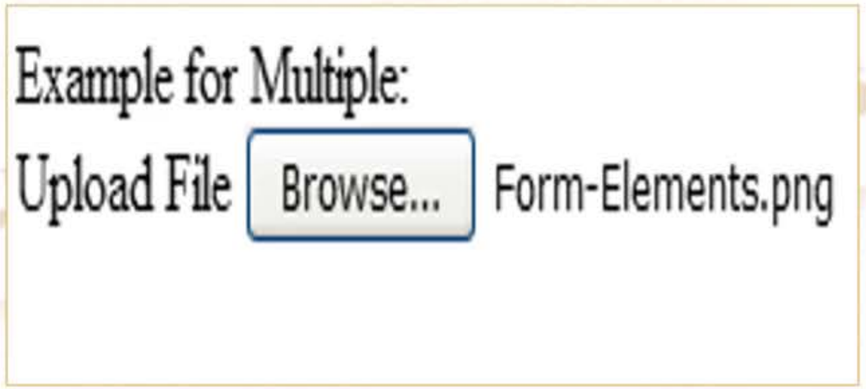
TEKNOTURF

multiple

- If present, a user can select more than one file when the input type is file and can include several comma-separated email addresses when the input type is email.

- While it was available in previous versions of HTML, it could be applied to select element only.

```
<html>
<body>
Example for Multiple: <br>
 Upload File <input type="file" name="img" multiple="multiple"/>
</body>
</html>
```



VALUES OF TYPE ATTRIBUTE

HTML5 gives us input types that provide for more data-specific UI elements and native data validation.

HTML5 has a total of 13 new input types:

- search
- email
- url
- tel

- text
- date
- month

- week
- time
- datetime-local

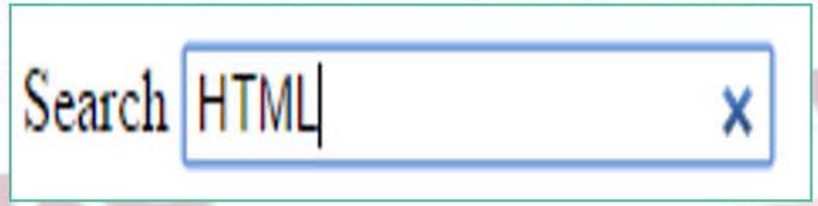
- number
- range
- color

VALUES OF TYPE ATTRIBUTE

search

- The search type is used for search fields
- Search type is only supported in Chrome, Opera, and safari

Search <input type="search"/>



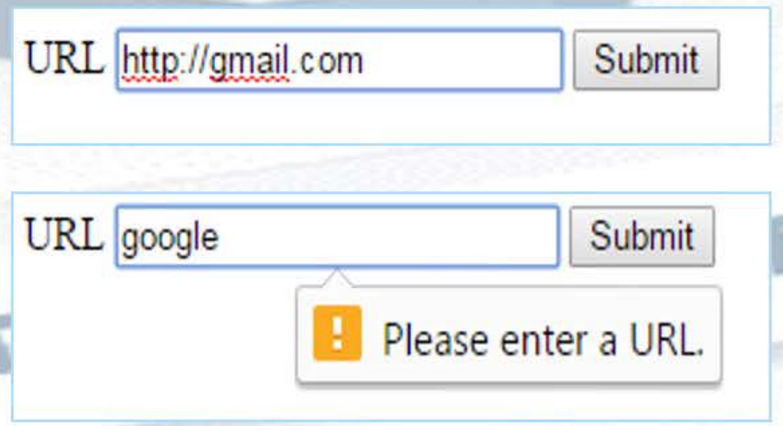
VALUES OF TYPE ATTRIBUTE

TEKNOTURF

url

- The url type is used for input fields that should contain a URL address.
- The value of the url field is automatically validated wher

```
URL <input type="url" />
<input type="submit" />
```



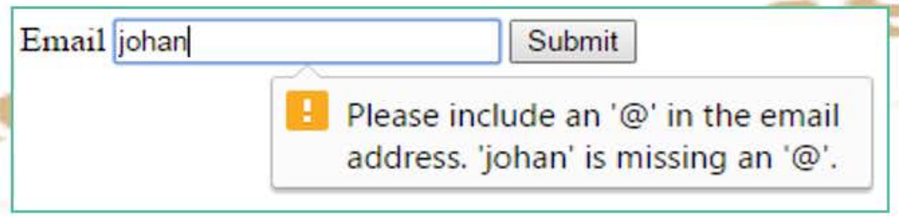
VALUES OF TYPE ATTRIBUTE

TEKNOTURF

email

- The email type (type="email") is used for specifying one or more email addresses
- Supports the Boolean multiple attributes, allowing for multiple, comma-separated email addresses

```
Email <input type="email" />
      <input type="submit" />
```



VALUES OF TYPE ATTRIBUTE

TEKNOTURF

tel

(type="tel") is used to accept telephone numbers

Unlike the url and email types, the tel type doesn't enforce a particular syntax or pattern

Letters and numbers—indeed, any character other than new lines or carriage returns—are valid

```
contact no : <input type="tel" />
<input type="submit" value="submit" />
```

VALUES OF TYPE ATTRIBUTE

number

Restricts the user to input numbers only

```
Mark <input type="number" />
```

range

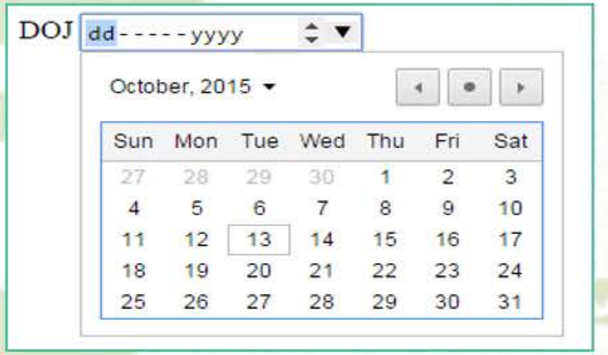
Restricts the user to input a value within the specified range only

```
Rate(1 to 10) <input
type="range" min="1"
max="10"/>
```

VALUES OF TYPE ATTRIBUTE

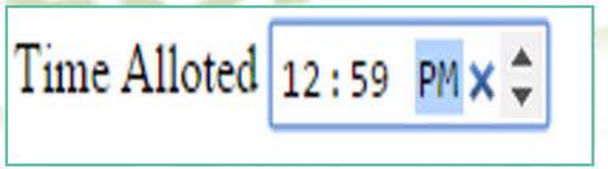
date

```
DOJ <input type="date"/>
```



time

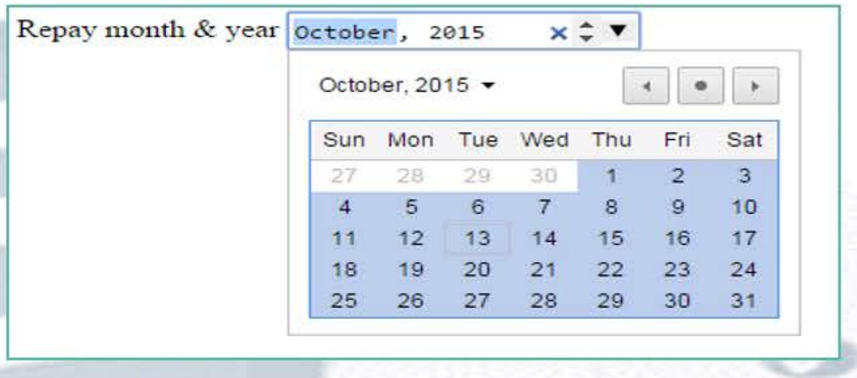
```
Time Allotted <input type="time"/>
```



VALUES OF TYPE ATTRIBUTE

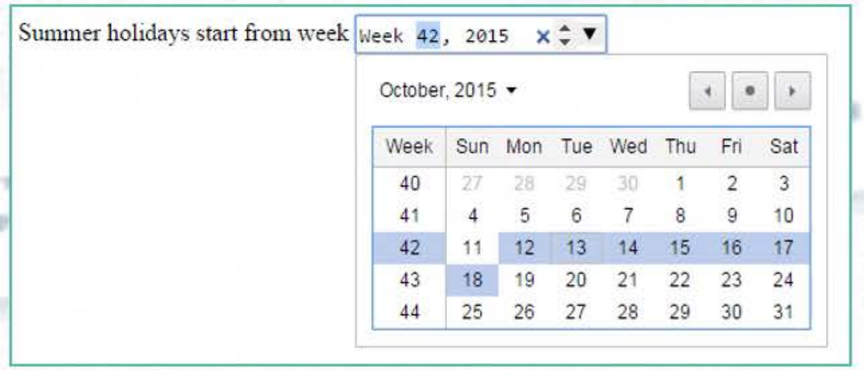
month

Repay month & year <input type="month"/>



week

Summer holidays start from week <input type="week"/>

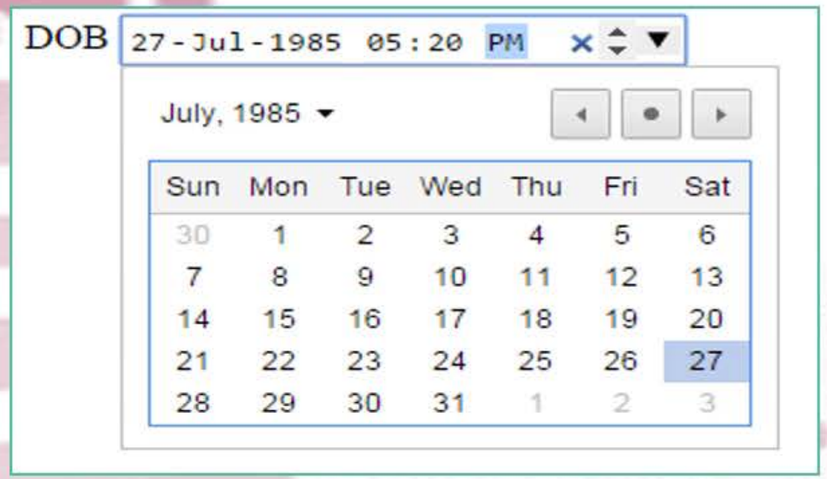


VALUES OF TYPE ATTRIBUTE

datetime-local

allows the user to select a date and time

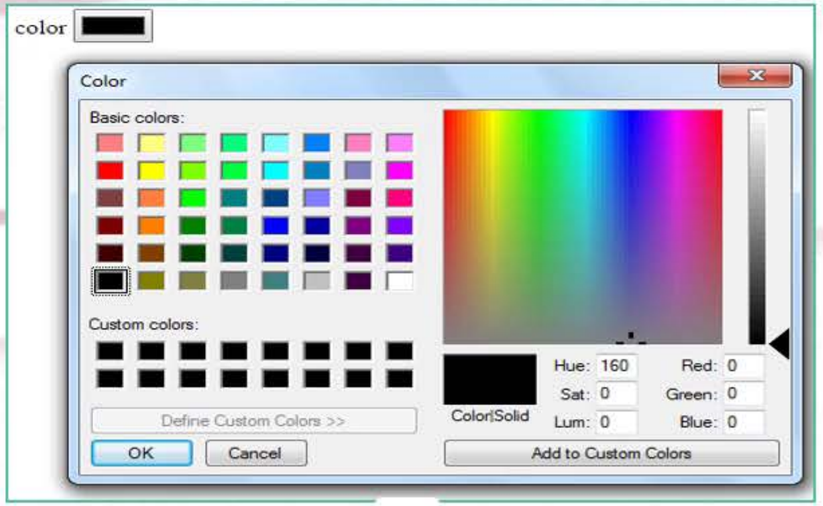
DOB <input type="datetime-local"/>



color

allows the user to select a color

color <input type="color"/>



RESTRICTIONS ON VARIOUS INPUT TYPES

TEKNOTURF

Restrictions that can be given on input elements

Attribute	Description
disabled	Specifies that an input field should be disabled
max	Specifies the maximum value for an input field
maxlength	Specifies the maximum number of characters for an input field
min	Specifies the minimum value for an input field
pattern	Specifies a regular expression to check the input value against
readonly	Specifies that an input field is read only (cannot be changed)
required	Specifies that an input field is required (must be filled out)
step	Specifies the legal number intervals for an input field
value	Specifies the default value for an input field

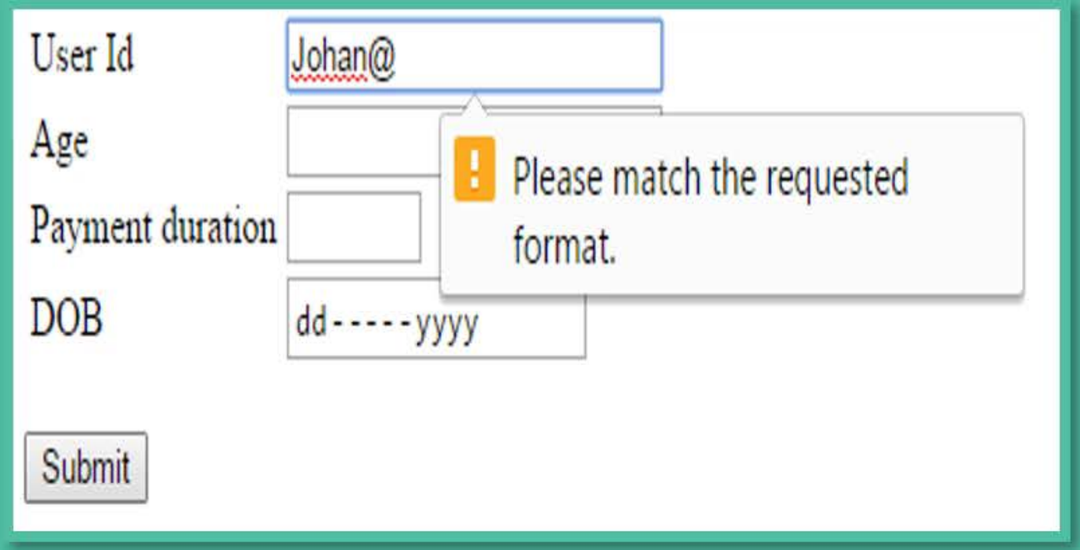
RESTRICTIONS ON VARIOUS INPUT TYPES

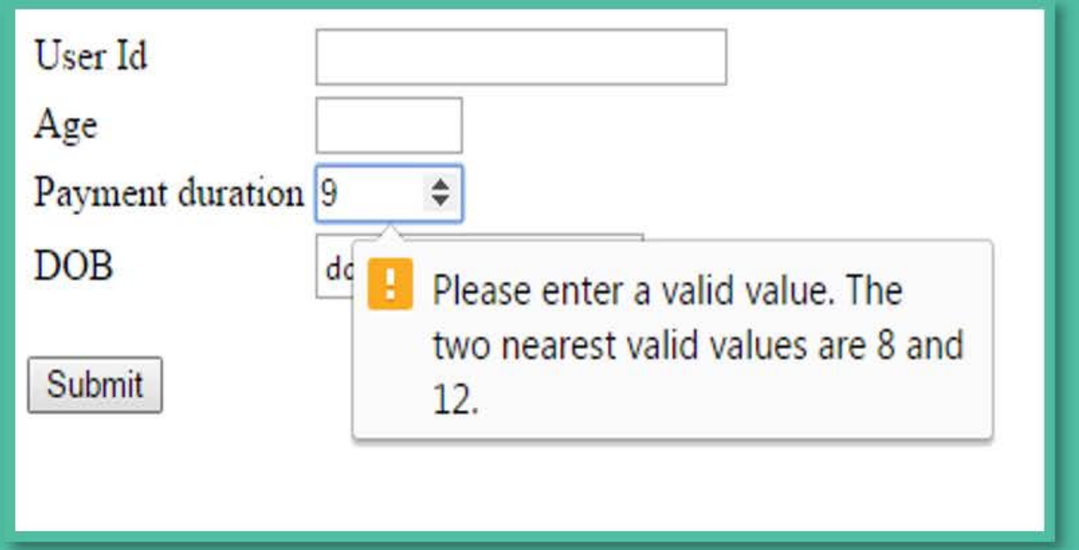
TEKNOTURF

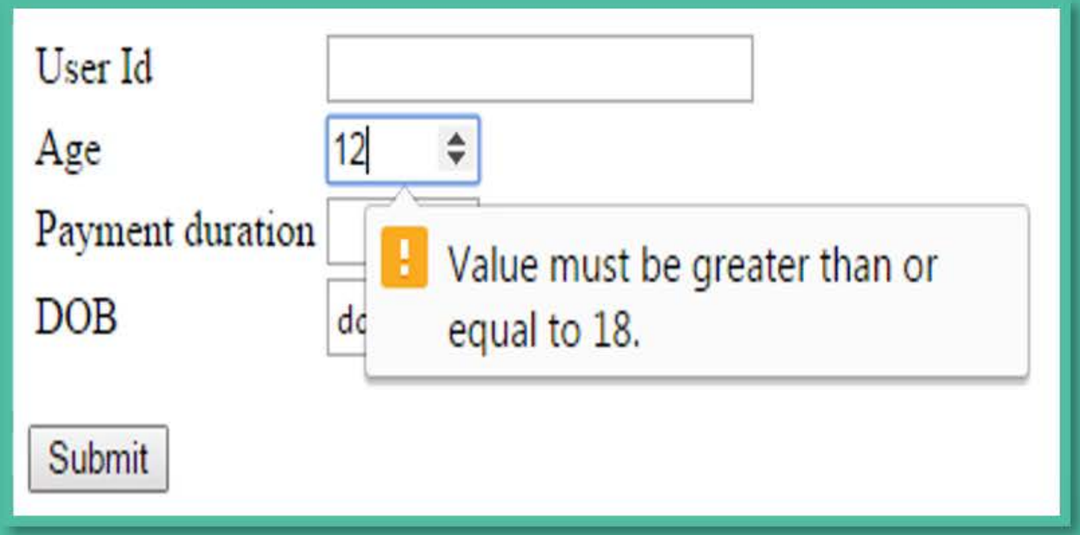
```
<table>
<tr><td>User Id</td><td><input type="text" pattern="[A-Ba-b0-9_]" /></td></tr>
<tr><td>Age</td><td><input type="number" min=18 max=58/></td>
<tr><td>Payment duration</td><td><input type="number" step="4" min="4" max="12"/></td>
<tr><td>DOB</td><td><input type="date" min="1965-12-31" max="1995-01-31"/></td></tr>
</table>
```

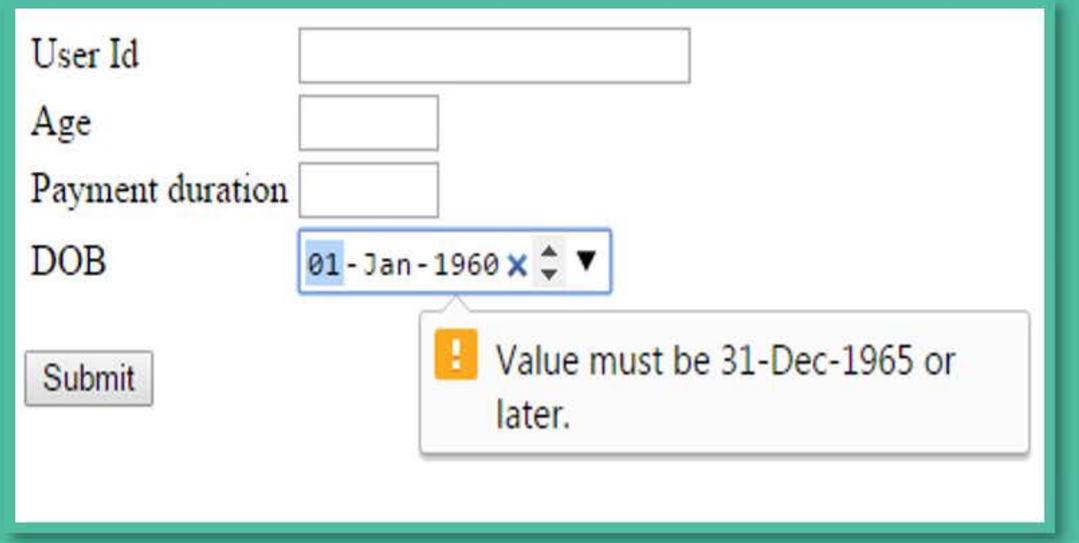
RESTRICTIONS ON VARIOUS INPUT TYPES

TEKNOTURF









AUDIO TAG

TEKNOTURF

<audio>

Audio files are played through a plug-in.

HTML5 defines a new element which specifies a standard way to embed an audio file on a web page: the <audio> element.

```
<audio src="http://songserver/english/batman3/song1.mp3"></audio>
```

Currently, there are 3 supported file formats for the <audio> element: MP3, Wav, and Ogg:

Browser	MP3	Wav	Ogg
Internet Explorer 9	YES	NO	NO
Firefox 4.0	NO	YES	YES
Google Chrome 6	YES	YES	YES
Apple Safari 5	YES	YES	NO
Opera 10.6	NO	YES	YES

09:32

VIDEO TAG

<video>

- Most video files are played through a plug-in (like flash). However, different browsers may have different plug-ins.
- HTML5 defines a new element, which specifies a standard way to embed a video file on a web page: the <video> element.

```
<video  src="http://songserver/english/song1.mp3"></video>
```

Currently, there are 3 supported video formats for the <video> element: MP4, WebM, and Ogg:

Browser	MP4	WebM	Ogg
Internet Explorer 9	YES	NO	NO
Firefox 4.0	NO	YES	YES
Google Chrome 6	YES	YES	YES
Apple Safari 5	YES	NO	NO
Opera 10.6	NO	YES	YES

HTML <IFRAME> TAG

TEKNOTURF

<iframe>

- The <iframe> tag specifies an inline frame.
- An inline frame is used to embed another document within the current HTML document.
- The iframe tag defines a rectangular region within the document in which the browser can display a separate document.
- You may use the height and width attributes to specify the size of the iframe.

Practical application:

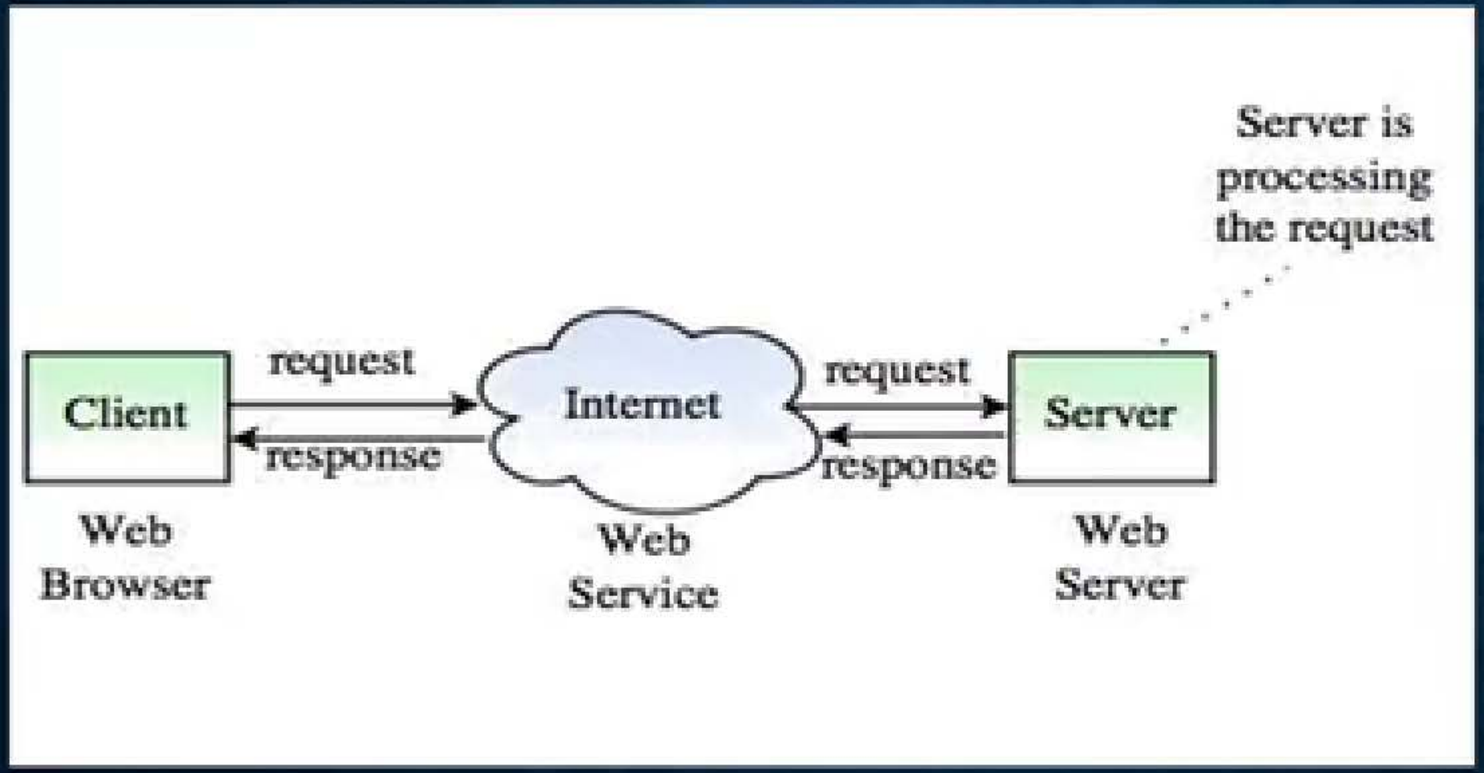
- Inline frame finds its usage in online advertising, where the contents of the <iframe> is an ad from an external party.

INTRODUCTION TO SCRIPTING LANGUAGE

TEKNOTURF

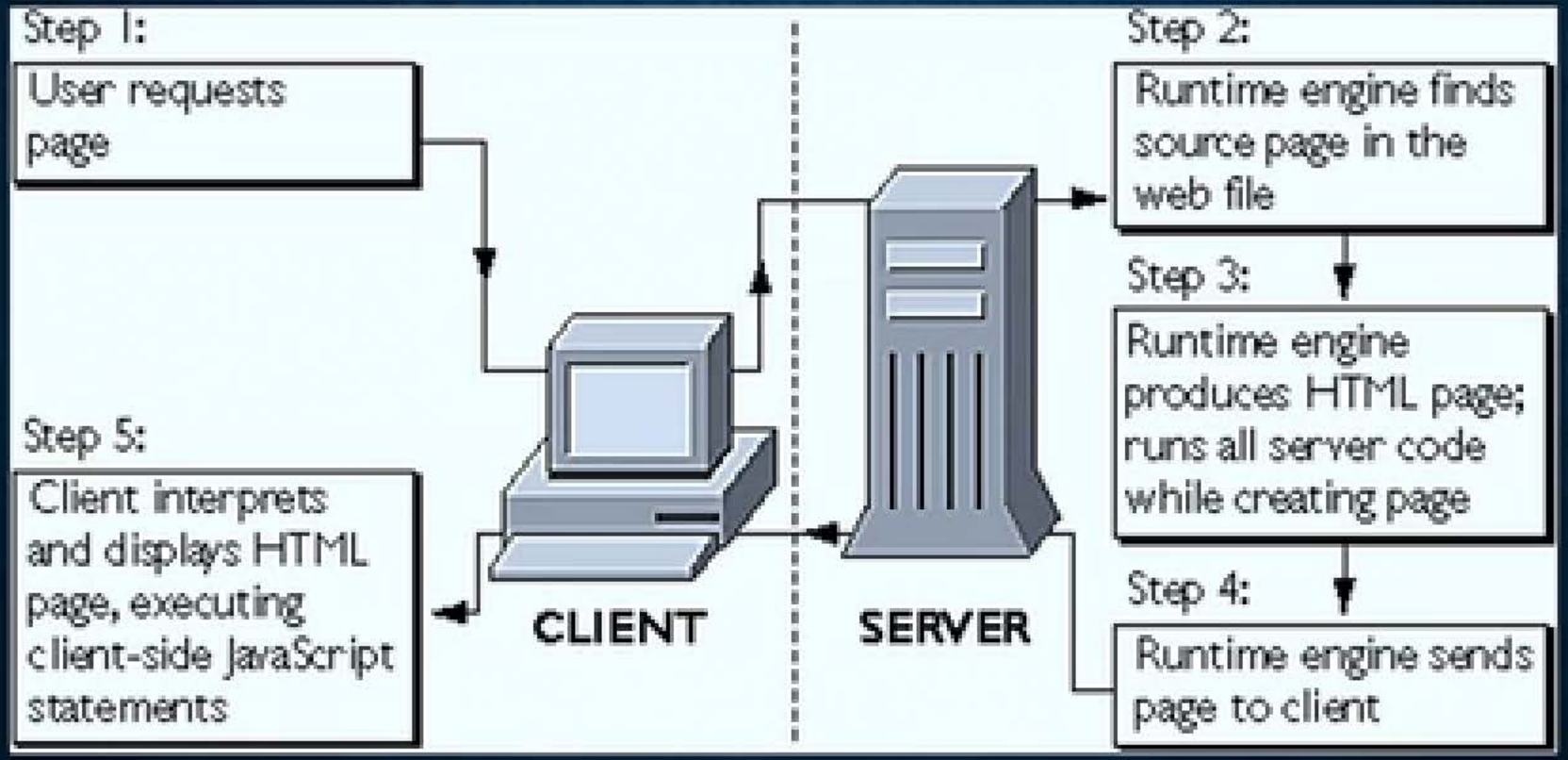
- Computer language with a series of commands within a file.
- Capable of being executed without being compiled.
- Script provides changes to the webpage.
- Two types of Scripting
  - Server-side scripting
  - Client-side scripting

SERVER-SIDE SCRIPTING



CLIENT-SIDE SCRIPTING

TEKNOTURF



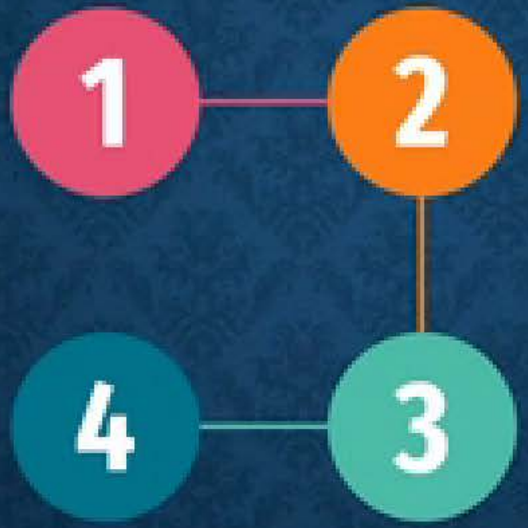
JAVASCRIPT INTRODUCTION

TEKNOTURF

1. JavaScript can be used in client and server side.
2. Traditionally on client side
   a) processing user input
   b) validations
3. Latest Implementation
   a) Client-side/Single page application
4. On Server-Side using Node JS
   a) It can create standalone application, Network application, Web Application
   b) REST services, service layers
5. From traditional validation module to
   a) JavaScript libraries like jQuery, DOJO, React and so on
   b) JavaScript frameworks like Angular, Ember and so on

ADVANTAGES OF JAVASCRIPT

TEKNOTURF



1. Improves transaction response time
2. Reduces server load
3. Less server interaction – Can validate user input before sending the page off to the server.
4. Provides immediate feedback

EMBEDDING JAVASCRIPT IN HTML

TEKNOTURF

There are three methods to embed JavaScript into **HTML code.**

1.Inline Script 2.Internal Script 3.External Script

Internal JavaScript

1

```
<script type="text/javascript">
function funDefinition(){
alert('You clicked me');
}
</script>
```

External JavaScript

2

```
<script src="ext.js"></script>
```

ext.js file

```
function funDefinition(){
alert("You clicked me");
}
```

Inline JavaScript

3

```
<button
onclick="alert('You
clicked me');">
Click Me
</button>
```

EXTERNAL JAVASCRIPT

TEKNOTURF

1. JavaScript can be put in a separate .js file

- Includes the external java script file in a HTML file using the code

```
<script src="myJavaScriptFile.js">
</script>
```

2. An external .js file (here, myJavaScriptFile.js) can be used in multiple HTML pages wherever necessary. This file cannot itself contain a <script> tag

EXECUTION OF JAVASCRIPT

TEKNOTURF

Locations into which we can attach JavaScript

- Directly into the **head** of the page.
- Directly into the **body** of the page.
- From an **event handler**.

Ex: Internal JavaScript that's not inside a function & interacts with the html page content directly :

```
<!DOCTYPE html>
<html><body>
<h3>Welcome</h3>
<div id="result"></div>
<script>
document.getElementById("result").innerHTML="Hello World";
</script>
</body></html>
```

Output:
Welcome
Hello World
When <script></script> is placed right above <h3> & below <body>, only 'Welcome' is displayed.

EXECUTION OF JAVASCRIPT

TEKNOTURF

You can invoke and implement your javascript directly upon an event handler, as in Ex 1. Or invoke your javascript defined in a function within script, from the event handler, as in Ex 2.

Example 1

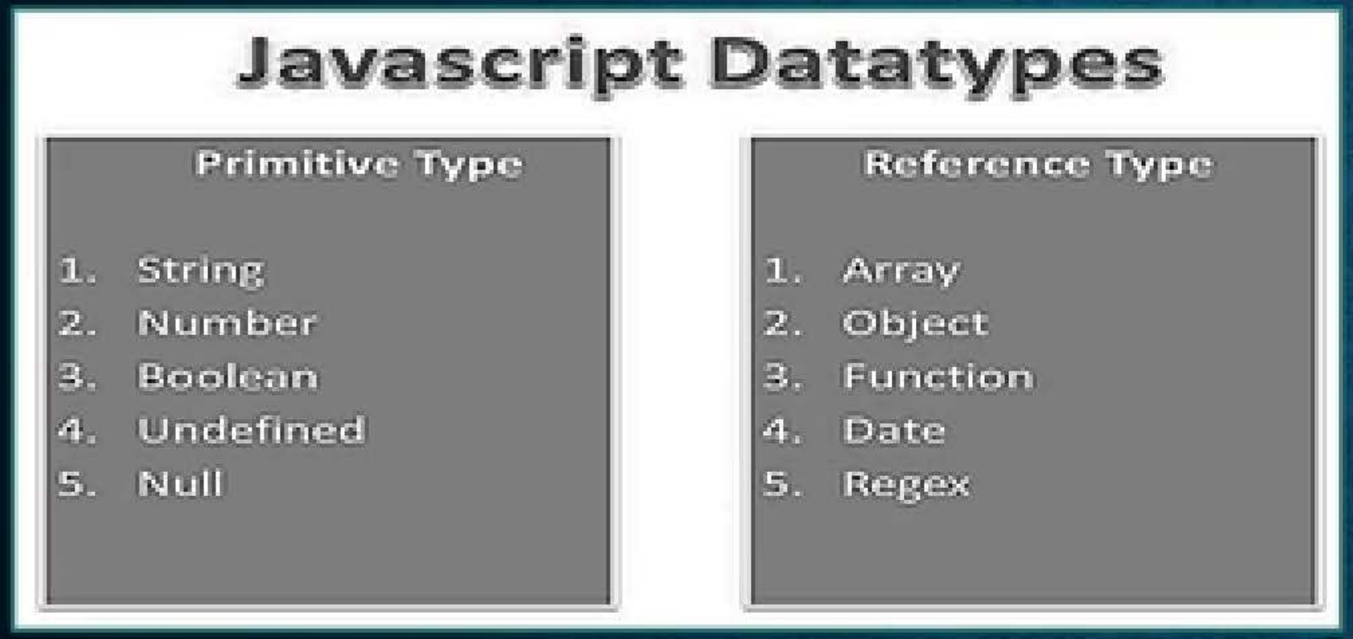
```
<button
onclick="alert('You
clicked me');">
Click Me
</button>
```

Example 2

```
<!DOCTYPE html>
<html><body>
<script>
function fun(){
alert('You clicked me');
}
</script>
<button onclick="fun()">Click
me</button>
</body></html>
```

DATA TYPES

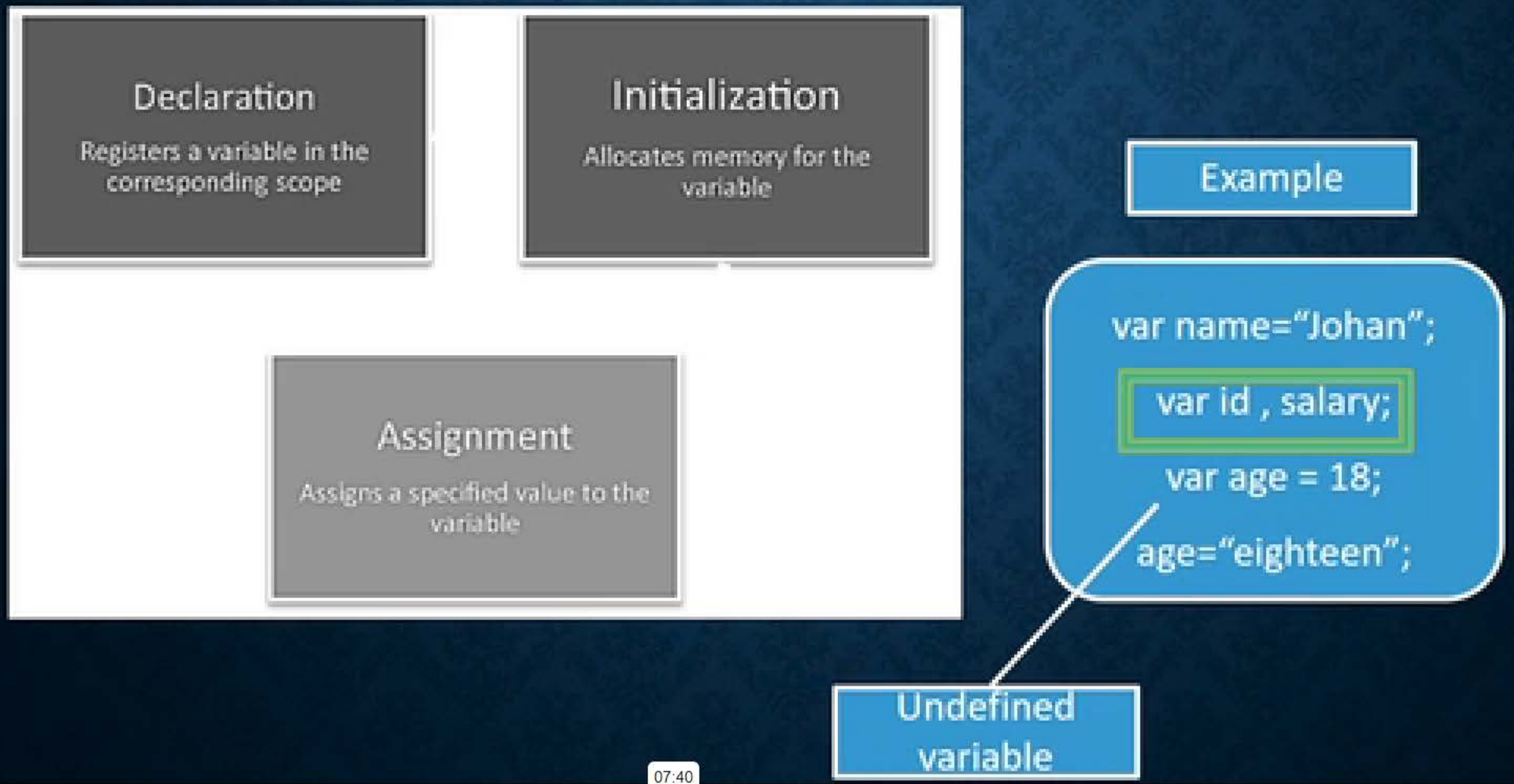
TEKNOTURF



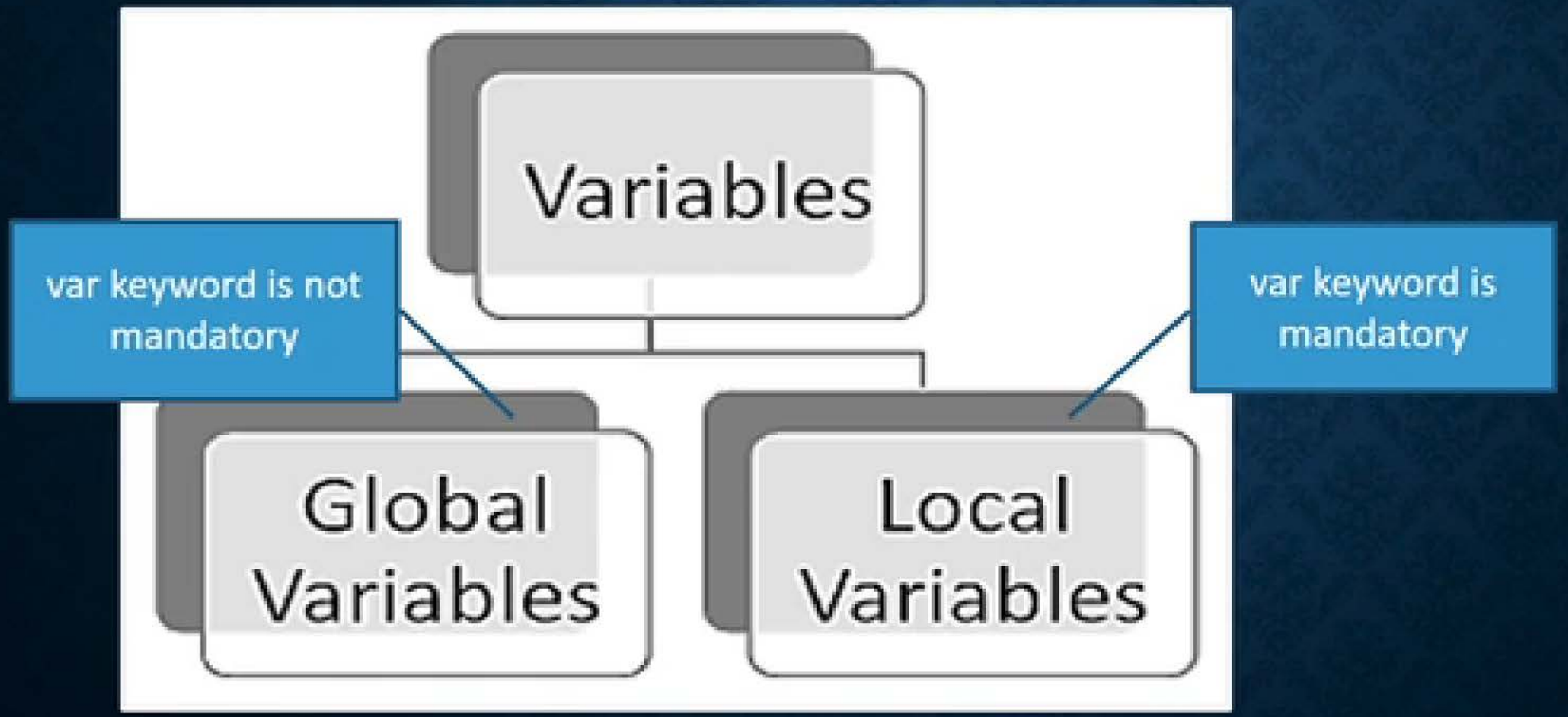

```
var  age = 16;   // Number
var name = "John";    // String
var ids = [101, 102, 103];   // Array
var x = {id:101, name:"John", salary:40000}; //Object
```

06:46

JAVASCRIPT VARIABLES



SCOPE OF VARIABLES



SCOPE OF VARIABLES

```
<!DOCTYPE html>
<html>
<body>
<script>
var message = 'Hi';
function say() {
    var message = 'Hello';
    document.write(message);
}
say();
document.write(message);
</script>
</body>
</html>
```

Global variable

Local variable

Function call

Output : HelloHi

When the first line within the js function say() is changed as : message = 'Hello'; (var removed) , Output : HelloHello (global variable is changed from 'Hi' to 'Hello')

OPERATORS

TEKNOTURF

Arithmetic Operators

Operator	Description	Examples
+	Addition	var z = 5 + 2;
-	Subtraction	var z = 5 - 2;
*	Multiplication	var z = 5 * 2;
/	Division	var z = 5 / 2;
%	Modulus	var z = 5 % 2;
++	Increment	var x = 5; z = x++;
--	Decrement	var x = 5; z = x--;

Assignment Operators

Operator	Description	Examples
=	x = y	x = y
+=	x += y	x = x + y
-=	x -= y	x = x - y
*=	x *= y	x = x * y
/=	x /= y	x = x / y
%=	x %= y	x = x % y

OPERATORS CONTD..

TEKNOTURF

Relational and Logical Operators

Operator	Description	Comparing	Returns
==	equal to	x == 8	false
		x == 5	true
		x == "5"	true
===	equal value and equal type	x === 5	false
		x === "5"	true
!=	not equal	x != 8	true
!==	not equal value or not equal type	x !== "5"	false

OPERATORS CONTD..

TEKNOTURF

Conditional Operators

Operator	Description	Example
? : (Conditional)	If Condition is true? Then value X : Otherwise value Y	var x=10 , y=20; var z=(x<y)?x:y;

OPERATORS CONTD..

TEKNOTURF

Types of Operator

1. The types of operator is a unary operator
2. It returns String

Example

```
var data=10;

var result = typeof data;

document.write(result);
```

Type	Return data
Number	"number"
String	"string"
Boolean	"boolean"
Object	"object"

COMMENTS IN JAVASCRIPT

TEKNOTURF

1. Comments are used to provide additional information about the JavaScript code and make it more readable.

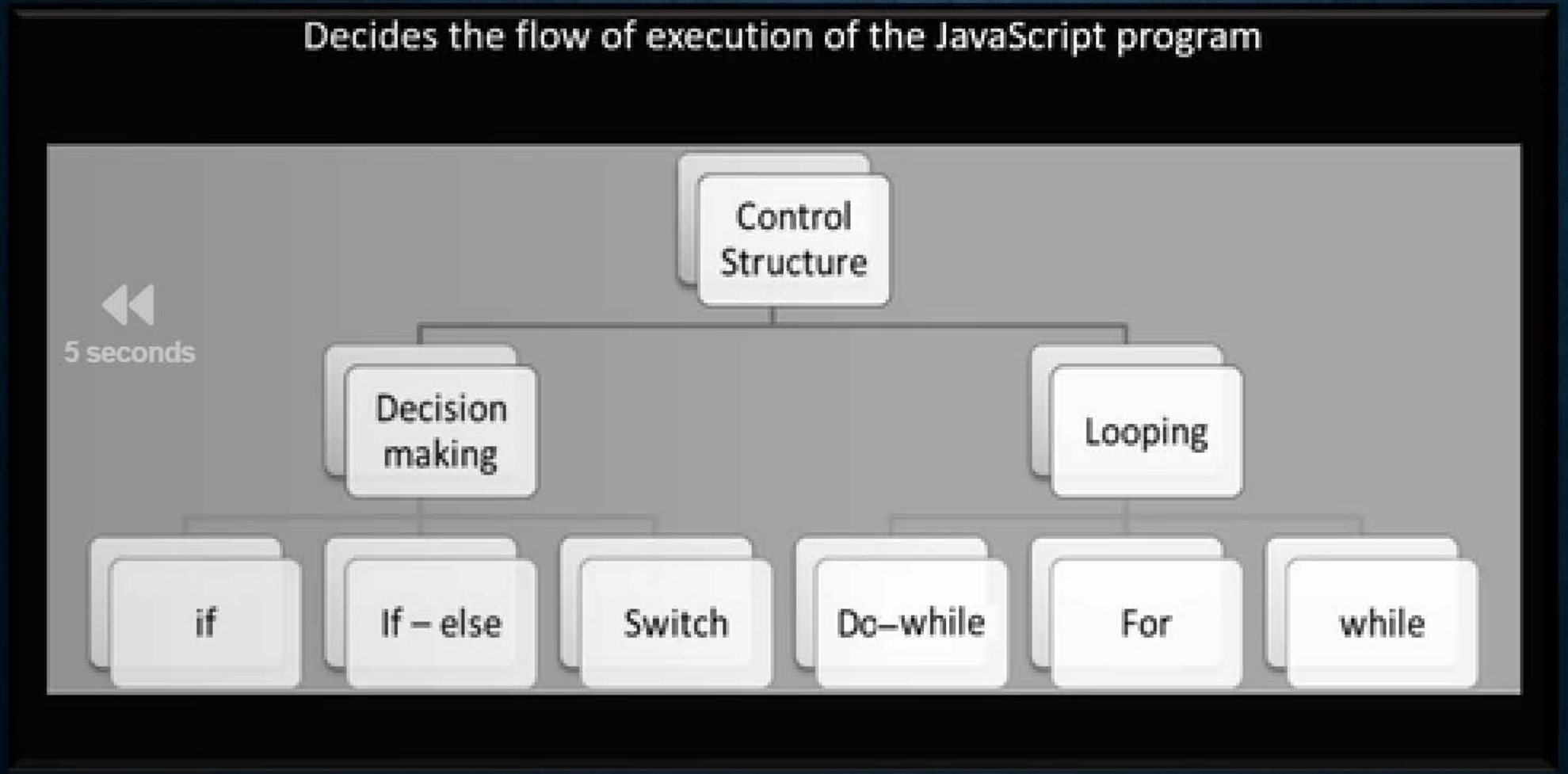
2. These statements will not get executed by the browser.

3. Single line comment : represented as **//commented statement**
Ex : // variable x is initialized
```
var  x = 5;
```

4. Multi line comment : represented as **/*commented statement(s)*/**
Ex : /* The code below initializes variable a with value 5
and variable b with value 6 */
```
var  a = 5;
var b = 6;
```

PROGRAMMING CONSTRUCTS IN JAVASCRIPT

TEKNOTURF



10:51

FOR .. IN LOOP

TEKNOTURF

```
for (variablename in object) {
    // statement or block to
execute
}
```

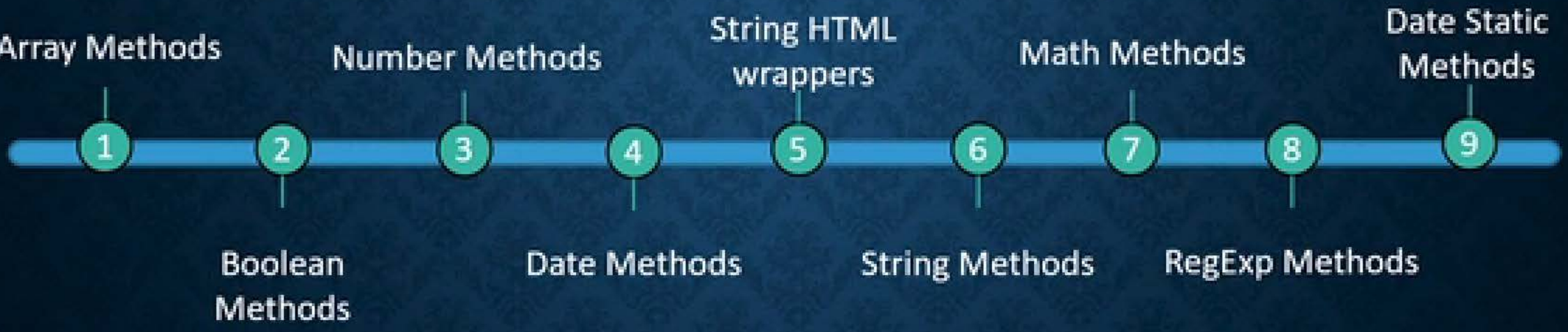
Syntax

Example

```
<script type="text/javascript">
      var ids = [101,102,103];
      var data;
      for(data in ids) {
document.write(ids[data]+" ");
      }
</script>
```

BUILT-IN FUNCTIONS

TEKNOTURF



BUILT-IN FUNCTIONS

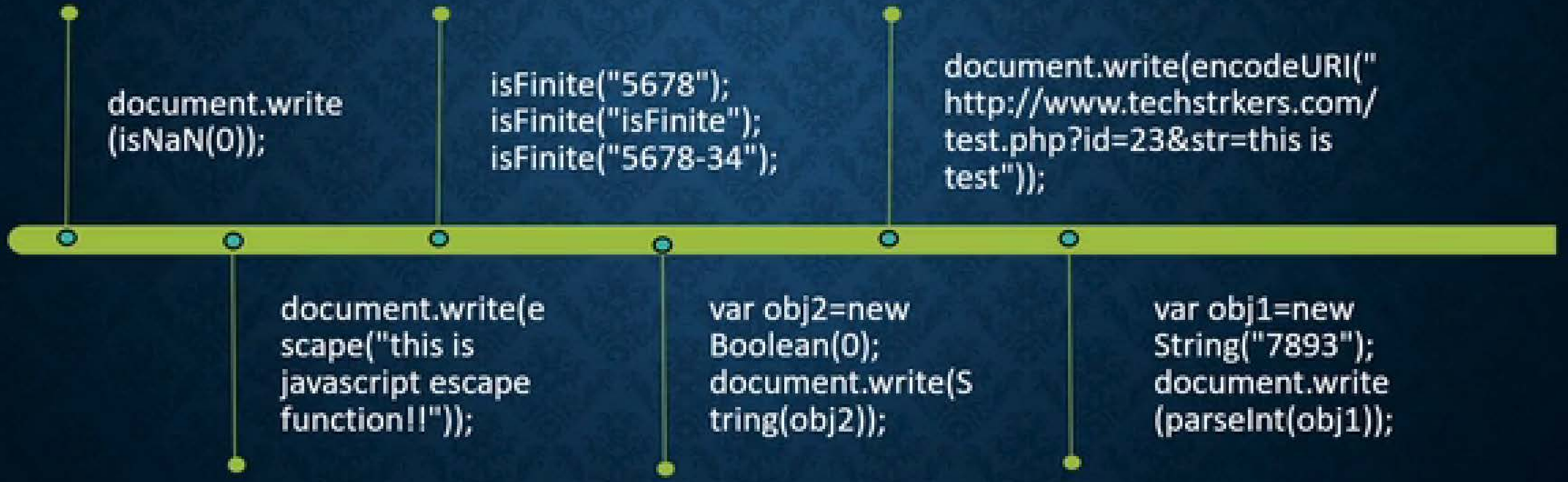
TEKNOTURF

Function	Description
isNan	Determines whether value is a legal number or not.
isFinite	To find whether a number is a finite legal number.
eval	Executes JavaScript source code.
number	Converts object to the corresponding number value.
string	Converts object to the corresponding string value.

Function	Description
parseInt	Converts string value to integer.
parseFloat	Converts string value to floating point number.
escape	Encodes the string value into world wide acceptable format.
encodeURI	To encode URI.
decodeURI	To decode URI.
encodeURIComponent	To encode URI component.
decodeURIComponent	To decode URI component.

BUILT-IN FUNCTIONS

TEKNOTURF



EXAMPLE

TEKNOTURF

```
function functionname(parameter-
list) {
    //statements
}
```

Syntax

5 seconds

Example

```
<script type="text/javascript">
function display()
{
    document.write("Welcome to
      JavaScript");
}
display();
</script>
```

Function call

16:41

JAVASCRIPT FUNCTIONS

TEKNOTURF

Functions with parameters and without return data

```
<script type="text/javascript">
function add(no1 , no2)
{
    var sum = no1 + no2;
    document.write(sum);
}
add(10,20);
</script>
```

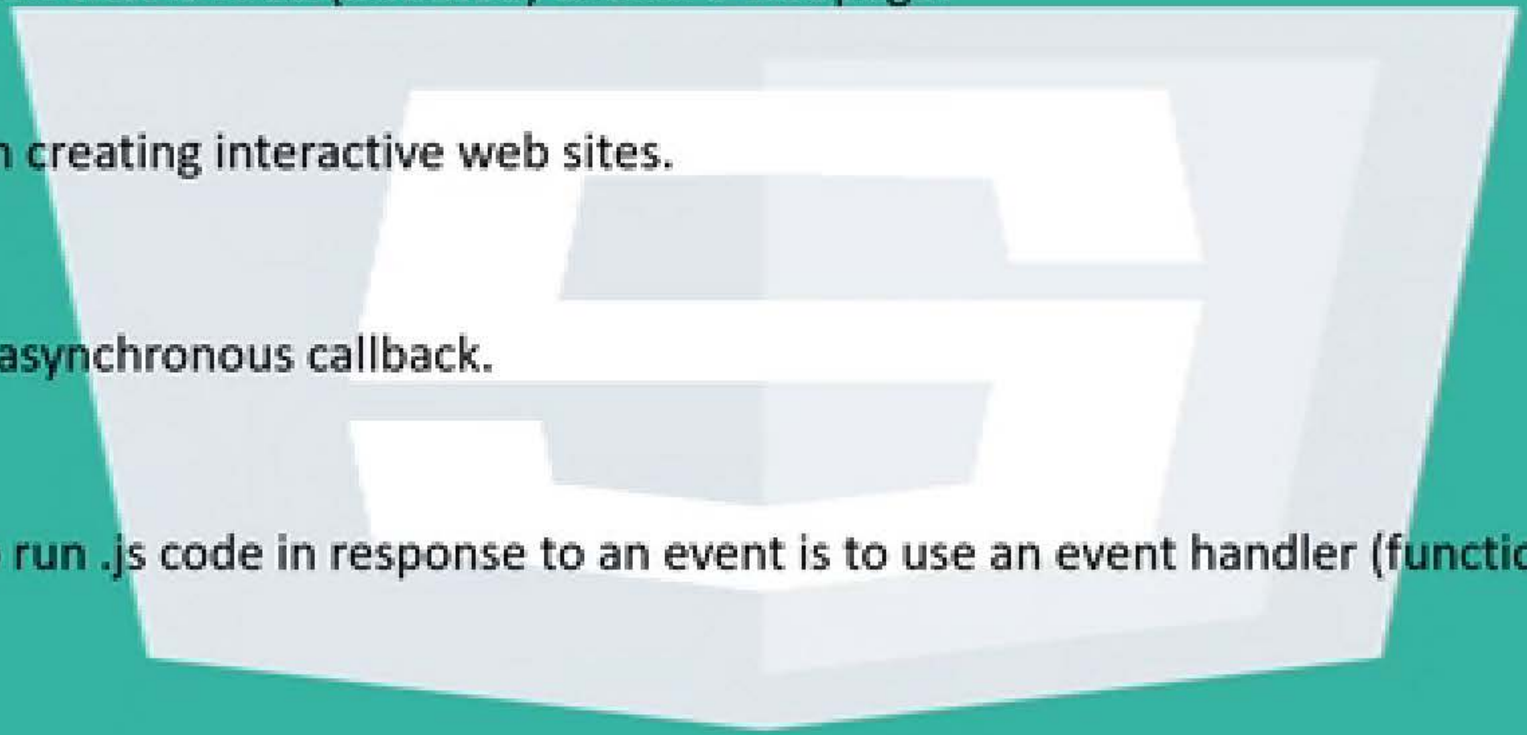
Function with parameters and return data

```
<script type="text/javascript">
function add(no1 , no2)
{
     var sum = no1 + no2;
      return sum;
}
var sum =add(10,20);
document.write(sum);
</script>
```

EVENT HANDLING

TEKNOTURF

1. Event – an action that is fired (initiated) within a webpage.
2. Event – an action that is fired (initiated) within a webpage.
3. It is so useful in creating interactive web sites.
4. JavaScript uses asynchronous callback.
5. Simplest way to run .js code in response to an event is to use an event handler (function)



EVENT HANDLING

TEKNOTURF

Event	Description
onchange	Script runs when the element changes
onsubmit	Script runs when the form is submitted
onreset	Script runs when the form is reset
onselect	Script runs when the element is selected
onblur	Script runs when the element loses focus
onfocus	Script runs when the element gets focus
onkeydown	Script runs when key is pressed
onkeypress	Script runs when key is pressed and released
onkeyup	Script runs when key is released
onclick	Script runs on a mouse click
ondblclick	Script runs on a mouse double-click
onmousedown	Script runs when mouse button is pressed
onmousemove	Script runs when mouse pointer moves
onmouseout	Script runs when mouse pointer moves out of an element
onmouseover	Script runs when mouse pointer moves over an element
onmouseup	Script runs when mouse button is released

EVENT HANDLING

TEKNOTURF

EXAMPLE :

```
<form>
    <input type="button" name="test" value="Click
me"
                onclick="inform()">
</form>
```

```
<script>
function inform()
{
   alert("You have activated me by clicking the grey

button!”)
}
</script>
```

When the user clicks the button, “inform()” will be called.

5 seconds

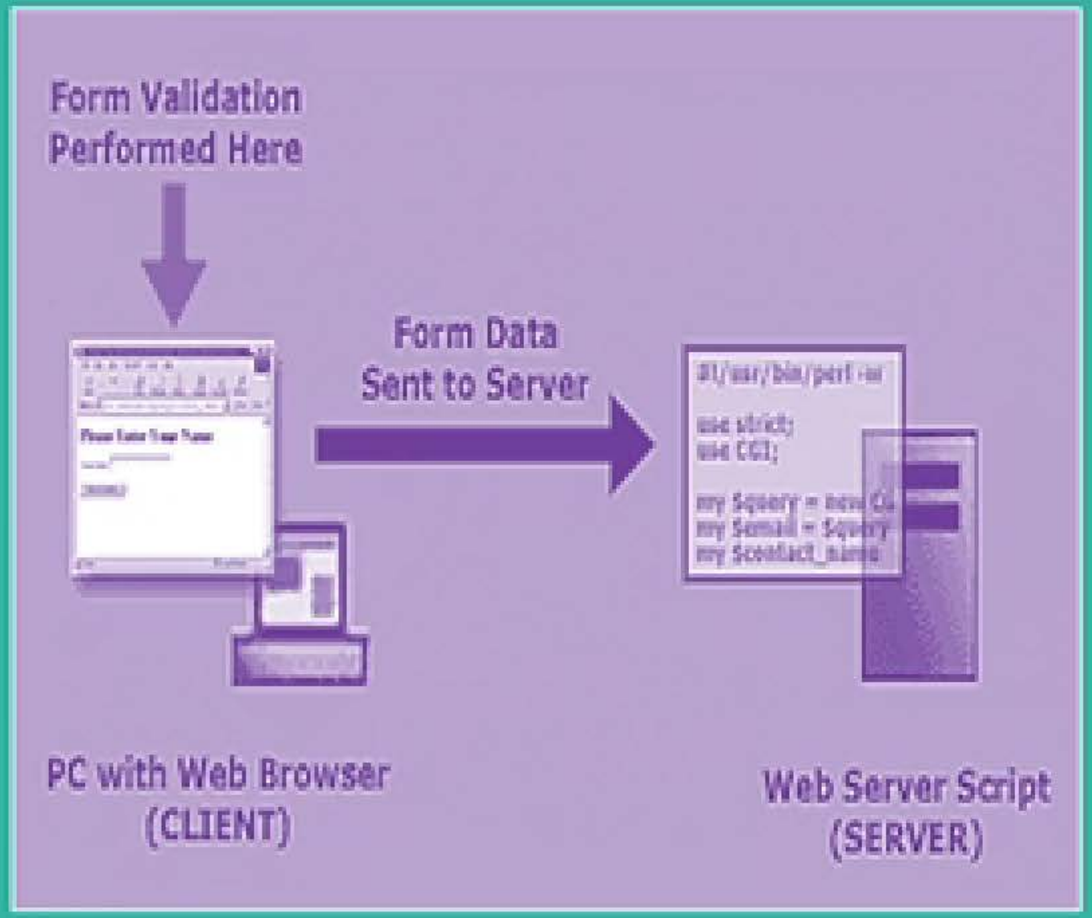
00:57

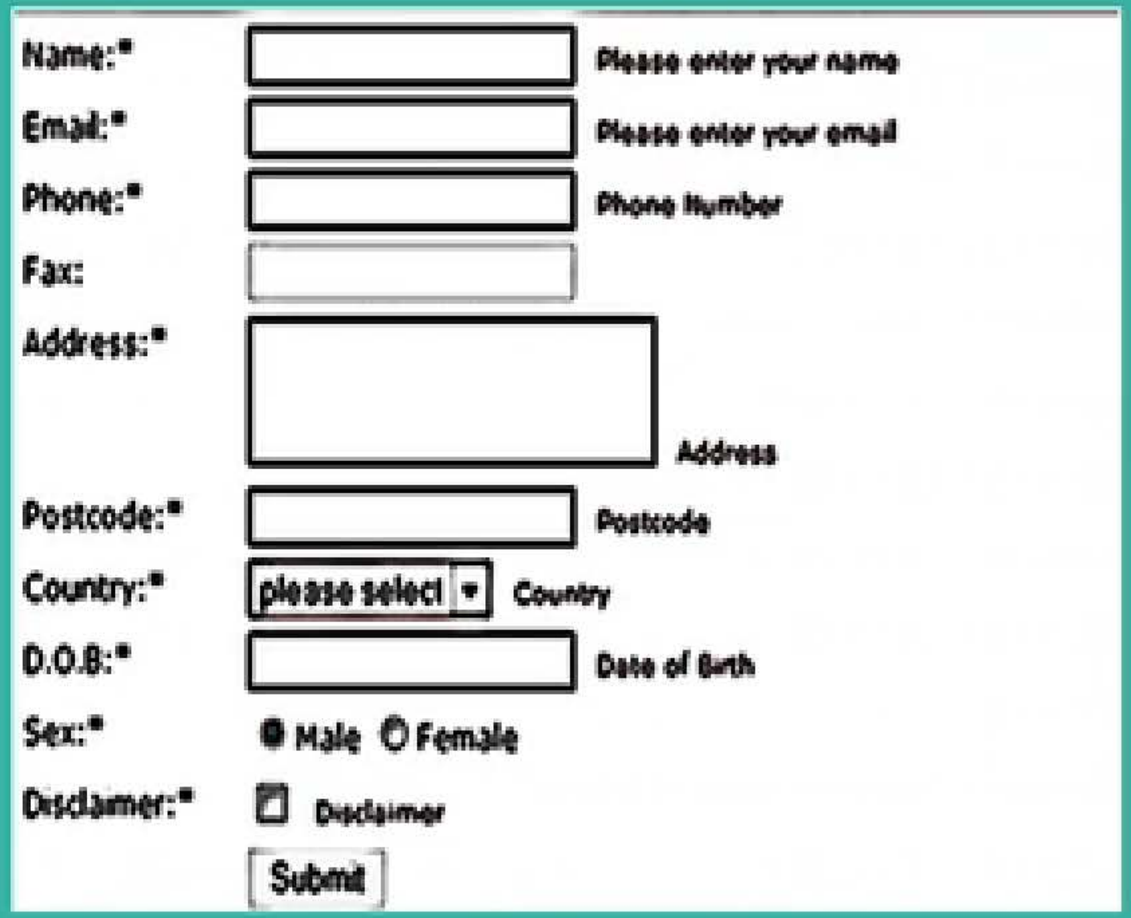
JAVASCRIPT VALIDATION

TEKNOTURF

- JavaScript data validation happens before form is submitted.
- Server-side application validation happens after the form is submitted to the application server.
- Form validation performs the following functions :
  - ❑ Basic Validation
  - ❑ Data Format Validation

JAVASCRIPT VALIDATION





JAVASCRIPT VALIDATION

```
<form name="register" action="#" method="post">
First Name <input type="text" name="fname" > <br/>
Last Name <input type="text" name="lname" > <br/>
<input type="button" value="Register"
onclick="validateData()"> <br/>
</form>
```

```
var fname=document.register.fname.value;
var lname=document.register.lname.value;
if(fname==null || fname== " "|| lname==null || lname.trim()=="")
{
    document.getElementById("msg").innerHTML +="Enter value for name <br />";
}
```

JAVASCRIPT VALIDATION

TEKNOTURF

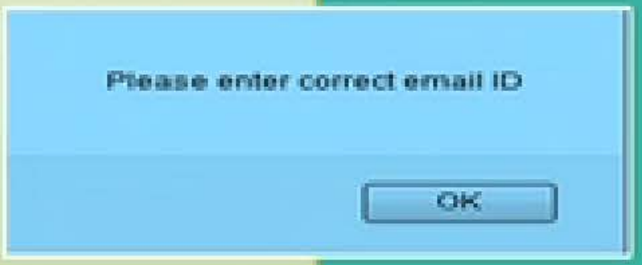
1. The data entered in a form can be validated for its format.

2. Our code must include appropriate logic to test the correctness of data.

```
<body>

 <form name="myForm" action="welcomePage.html" method="post" >

Email:<input type="text" name="email" id="email" onBlur="return validateEmail()"/>

<input type="submit" value="Submit" onsubmit="return validateEmail()">

</form>
```

JAVASCRIPT VALIDATION

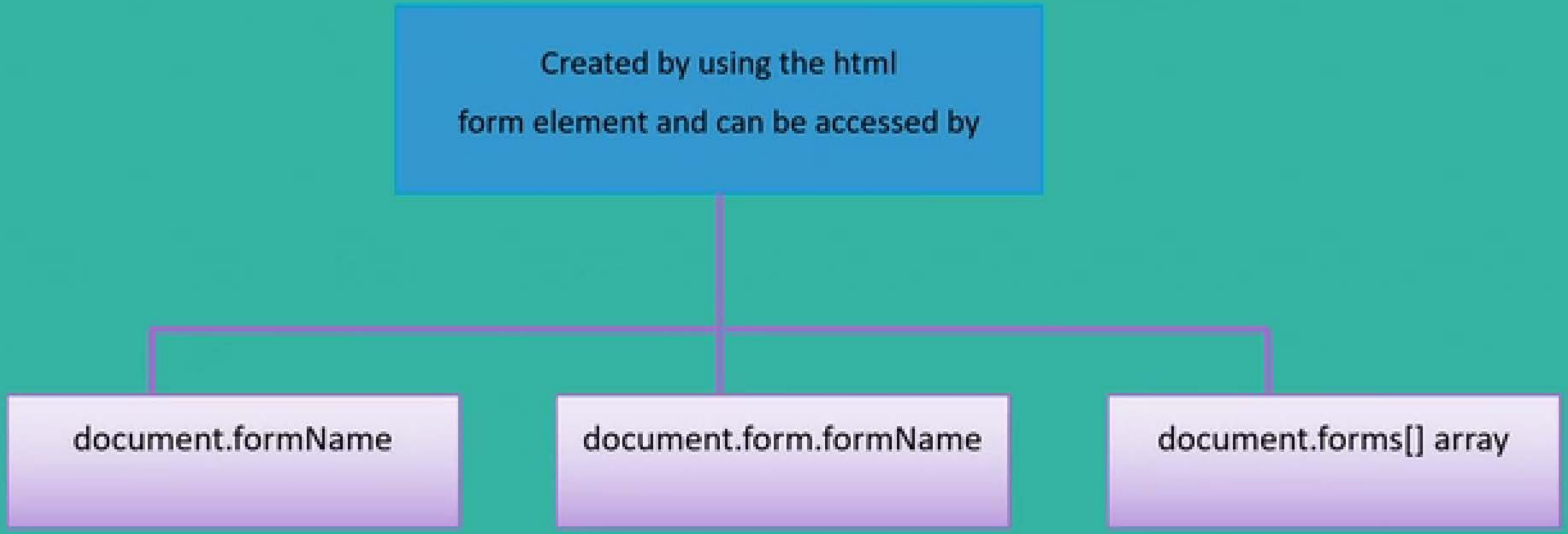
```
<script type="text/javascript">
    function validateEmail()
          {
          var emailID = document.myForm.EMail.value;
          atpos = emailID.indexOf("@");
          dotpos = emailID.lastIndexOf(".");
          if(atpos < 1 || ( dotpos - atpos < 2 ))
          {
                    alert("Please enter correct email ID");
                    document.myForm.EMail.value="" ;
                    document.getElementById("email").focus() ;
                    return false;
          }
      return true ;
    }
</script>
</body>
```



5 seconds

02:55

FORM OBJECT


Created by using the html
form element and can be accessed by
document.formName
document.form.formName
document.forms[] array

FORM OBJECT

TEKNOTURF

Property	Description
action	Presents the action attribute.
autocomplete	Presents the autocomplete attribute (on / off)
encoding	Presents the enctype attribute.
length	Presents the number of elements on a form.
method	Presents the forms method attribute.
name	Presents the name attribute of the form.
noValidate	Presents if form data needs to be validated or not (true / false)
target	Presents the target attribute of the form. It represents the name of the frame or window to which the form submission response is sent by the server.

FORM OBJECT

```
< <form id="myForm" action="homepage.html" >
<table>
<tr><td>User name </td><td><input type="text" name="uname"></td></tr>
<tr><td>Password</td><td> <input type="password" name="pwd"></td></tr>
<tr><td colspan="2"><input type="button" value="Submit"  onClick = "changeAction()" >
</td> </tr>
</table>  </form>
<div id="msg" ></div>
<script>
        function changeAction()
        {
                document.getElementById("myForm").action = "form_action.asp";
                document.getElementById("myForm").autocomplete = "off;
                document.getElementById("msg").innerHTML = "The value of the action
    attribute was changed";
    }
</script>
```

FORM OBJECT - METHODS

```
<body>
<form name="register" action="registerUser.html">
  First name: <input type="text" name="fname"><br>
  Last name: <input type="text" name="lname"><br>
  <input type="button" onclick="myFunction()" value="Submit form">
</form>
<script>
function myFunction()
{
   document.register.submit();
}
</script>
</body>
```

FORM EVENT HANDLER

```
<body >

<form name="register" onreset="return display()" onsubmit="return
 displayValues()">
  First name: <input type="text" name="fname"   onblur = "alert(this.value)“ ><br>
  Last name: <input type="text" name="lname"><br>
  <input type="submit" >
  <input type="reset" >
</form>

<script>

function display()
{
   document.register.fname.value="Pearson";
   document.register.lname.value="David";
   return false;
}
```

Form Events onset, onreset and onblur used

Accessing a textbox value from a form

FORM EVENT HANDLER

TEKNOTURF

Executed when onsubmit event occurs

```
function displayValues()
{
var fname=document.register.fname.value;
var lname=document.register.lname.value;
alert("First name is "+fname+" Last name is
"+lname);
return false;
}
</script>
</body>
```

TEXT OBJECT

TEKNOTURF

Text object represents a single-line text input field in a HTML form object.

```
<input type="text" name="firstname">
```

Properties

- defaultValue
- form
- name
- type
- value

Methods

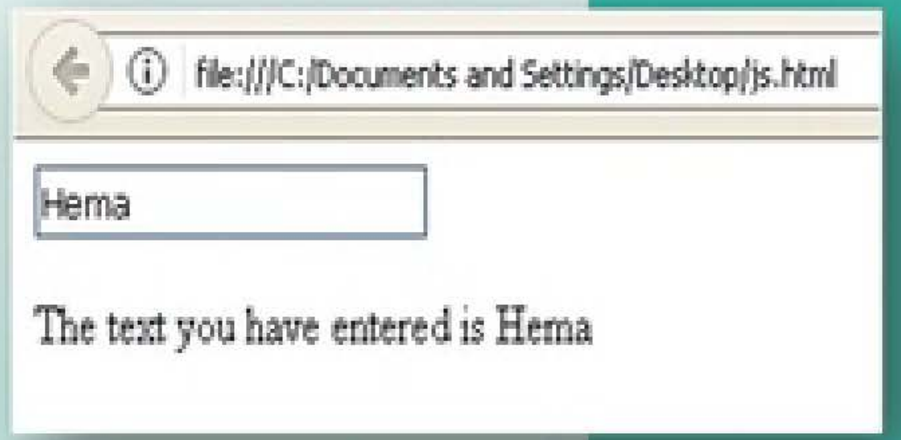
- blur()
- focus()
- select()

Events

- onBlur
- onChange
- onFocus
- onSelect

TEXT OBJECT

```
<html>
<body>
<form>
<input type="text" onblur="display(this)"/>
<p id="demo"></p>
<script type="text/javascript">
function display(str)
{
    document.getElementById("demo").innerHTML  =  "The text you have entered is "
                                                          +str.value;
}
</script>
</body>  </html>
```



BUTTON OBJECT

TEKNOTURF

```
<input type="button" name="myButton" value="Press This" onClick="clickFunction">
```

The input type could be "button" or "submit" or "reset".

Properties

- ✓ disabled
- ✓ form
- ✓ name
- ✓ type
- ✓ value

Methods

- ✓ blur() - Takes the focus away from the radio button.
- ✓ click() - This function acts as if the user clicked the button.
- ✓ focus() - Gives the focus to the checkbox.

Events

- ✓ onBlur
- ✓ onClick
- ✓ onFocus

CHECKBOX OBJECT

TEKNOTURF

<INPUT TYPE="checkbox" NAME="Name1" VALUE="1" CHECKED onClick="clickFunction">

The option "CHECKED" sets the button so it is selected when it is initially displayed

Properties

- checked
- defaultChecked
- form
- name
- type
- value

CHECKBOX OBJECT

TEKNOTURF

Methods

- blur()
- click()
- focus()

Events

- onBlur
- onClick
- onFocus

RADIO OBJECT

TEKNOTURF

Represents an HTML <input> element with type="radio"

Properties

- checked
- defaultChecked
- form
- name
- type
- value

Methods

- blur()
- click()
- focus()

RADIO OBJECT

TEKNOTURF

Events

- onBlur
- onClick
- onFocus

```
<body>
  <form name="form1">
  <p><input type=radio name="seats" value="sleeper">Sleeper</p>
  <p><input type=radio name="seats" value="semisleeper">Semi Sleeper</p>
  <p><input type=radio name="seats" value="normal">Normal</p>
  <p><input type=button value="Show Selected Seat"
                          onClick="getSelectedSeat(this.form.seats)"></p>
  <p><div id="msg"></div></p>
```

SELECT OBJECT

TEKNOTURF

Represents HTML <select> element

Properties

- form
- length
- name
- options
- selectedIndex
- type

Methods

- blur()
- focus()
- add()
- remove()

SELECT OBJECT

TEKNOTURF

Events

- onBlur
- onClick
- onFocus

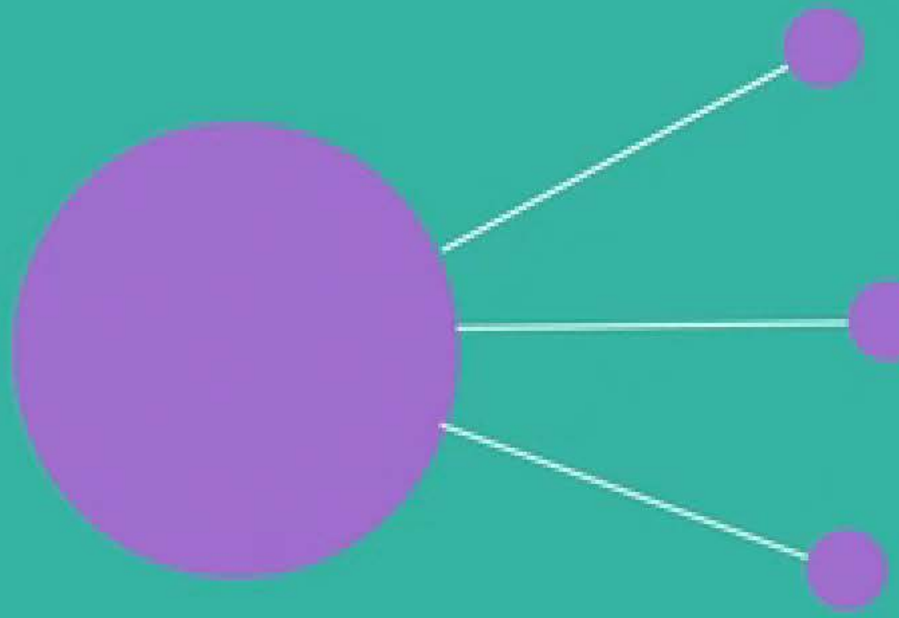
```
<body>
<form>
<select id="technology"    onchange="selectMethodsDemo()" >
 <option>HTML</option>
 <option>CSS3</option>
 <option>Javascript</option>
 <option>JQuery</option>
</select>
</form>
```

IMPLICIT OBJECTS IN JAVASCRIPT

TEKNOTURF

1. Object-based Language
2. Has State and Behaviour
3. Template based
4. Array, String, Number, Boolean, Date, Math are few implicit objects in JavaScript.

STRING



ARRAY OF CHARACTERS

```
var name= "Teknoturf";
var name= new String("Teknoturf");
```

PROPERTY

name.length 9

SPECIAL CHARACTERS

```
name="John \"David\"";
```

Code	Outputs
\'	single quote
\"	double quote
\\	backslash

STRING METHODS

TEKNOTURF

Method	Description
charAt()	Returns the character at the specified index (position)
concat()	Joins two or more strings, and returns a copy of the joined strings
indexOf()	Returns the position of the first occurrence of a specified value in a string
lastIndexOf()	Returns the position of the last occurrence of a specified value in a string
localeCompare()	Compares two strings in the current locale
replace()	Searches within a string for a value and returns a new string with the replaced value
search()	Searches a string for a value and returns the position of the match

URL STRING ENCODING AND DECODING

TEKNOTURF

- The encodeURI() function is used to encode a URI.

- This function encodes special characters, except: , / ? : @ & = + $ #

```
<p>Click the button to encode a URI.</p>
<button onclick="testEncode()">Try it</button>
<p id="uridemo"></p>

<script>
function testEncode ()
{
   var uri = "teknoturf infoservices?name=Tinå&location=USA";
   var res = encodeURI(uri);
   document.getElementById(“uridemo").innerHTML = res;
}
</script>
```

teknoturf%20infoservices?name=Tin%C3%A5&location=USAc

URL STRING ENCODING AND DECODING

TEKNOTURF

- The decodeURI() function is used to decode a URI.

```
<p>Click the button to decode a URI after encoding it.</p>
<button onclick="testDecode()">Try it</button>
<p id="uridemo"></p>
<script>
function testDecode() {
    var uri = "teknoturf infoservices?name=Tinå&location=USA";
    var encode = encodeURI(uri);
    var decode = decodeURI(encode);
    var res = "Encoded URI: " + encode + "<br>" + "Decoded URI: " + decode;
    document.getElementById("uridemo").innerHTML = res;
}
</script>
```

Encoded URI: teknoturf%20infoservices?name=Tin%C3%A5&location=USA
Decoded URI: teknoturf infoservices?name=Tinå&location=USA

MATH OBJECT

TEKNOTURF

Properties	Description	Example
PI	Returns PI (approx. 3.14)	document.write(Math.PI); //returns 3.141592653589793
SQRT1_2	Returns the square root of 1/2 (approx. 0.707)	document.write(Math.SQRT1_2); //returns 0.7071067811865476
SQRT2	Returns the square root of 2 (approx. 1.414)	document.write(Math.SQRT2); //returns 1.4142135623730951

MATH FUNCTIONS

TEKNOTURF

Function	Description	Example
abs(x)	Returns the absolute value of x	Math.abs(-7.25); // 7.25
ceil(x)	Returns x, rounded upwards to the nearest integer	Math.ceil(1.4) //2
floor(x)	Returns x, rounded downwards to the nearest integer	Math.floor(1.6); //1
max(x,y,z,...,n)	Returns the number with the highest value	Math.max(5, 10); //10
min(x,y,z,...,n)	Returns the number with the lowest value	Math.min(5, 10); //5
pow(x,y)	Returns the value of x to the power of y	Math.pow(4, 3); //64

DATE OBJECT

TEKNOTURF

1. Helps to work with dates in JavaScript

2. Can be created in many ways
   - new Date()
   - new Date(milliseconds)
   - new Date(dateString)
   - new Date(year, month, day, hours, minutes, seconds, milliseconds)

3. Examples

```
<script type="text/javascript">
    var  date1 = new Date();
    var  date2 = new Date("January 24 , 2017 11:59:00");
    var  date3 = new Date(2017,0,24,12,02,10,15,20);
    document.write(date1+"<br />"+date2+"<br />"+date3);
</script>
```

Tue Jan 24 2017 11:59:10 GMT+0530 (India Standard Time)
Tue Jan 24 2017 11:59:00 GMT+0530 (India Standard Time)
Tue Jan 24 2017 12:02:10 GMT+0530 (India Standard Time)

DATE OBJECT

TEKNOTURF

Date Formats – can be used while constructing a date object.

ISO Format - (YYYY-MM-DD) -- DD and MM are optional
Long Date - (MMM DD YYYY) -- year, month, and day can be in any order
Short Date - (MM/DD/YYYY) -- Either "/" or "-" can be used as a separator

Method	Description
getDate()	Get the day as a number (1-31)
getDay()	Get the weekday as a number (0-6)
getFullYear()	Get the four digit year (yyyy)
getHours()	Get the hour (0-23)
getMilliseconds()	Get the milliseconds (0-999)
getMinutes()	Get the minutes (0-59)
getMonth()	Get the month (0-11)

DATE FUNCTION

TEKNOTURF

Print current Date

```
<script type="text/javascript">
 var currDate= new Date()
var year=currDate.getFullYear()
var month=currDate.getMonth()+1
var day=currDate.getDate()

document.write("Today's date is: ")
document.write(year+"/"+month+"/"+day)
</script>
```

OUTPUT:

Today's date is: 2022/12/2

Print the day of the week - when date is given

```
birthday = new Date(1998,2,14)
weekDay = birthday.getDay()



alert(weekDay) //alerts 6
```

ARRAYS

TEKNOTURF

1. Array is a data structure consisting of similar data items sharing a common name
2. At each individual location is an element, which is accessed by its position or index
3. The first element in every array is at 0th position
4. In general, the nth element of an array c is referred to as c[n-1]
5. JavaScript arrays are “dynamic” entities where the size can be dynamically changed

ARRAYS - EXAMPLE

TEKNOTURF

```
<script type="text/javascript">
  var n1=new Array(5);
  var n2=new Array();
  var n3=[0,1,2,3,4];


  for(var i=0;i<n1.length;i++)
        n1[i]=i;
  for(var i=0;i<n1.length;i++)
        n2[i]=1;

  displayArray(n1);
  displayArray(n2);
```

Operator new creates an array that can hold 5 elements, under the name n1

Operator new creates an empty array under the name n2

A new array n3 is created with values 0,1,2,3 and 4

```
function displayArray(array) {
  for(var i=0;i<array.length;i++)
    document.writeln(array[i]+"  ");
}
```

ARRAYS

TEKNOTURF

Method	Description
concat()	Joins two or more arrays, and returns a copy of the joined arrays
indexOf()	Searches the array for an element and returns its position
join()	Joins all elements of an array into a string
lastIndexOf()	Searches the array for an element, starting at the end, and returns its position
pop()	Removes the last element of an array, and returns that element
push()	Adds new elements to the end of an array, and returns the new length
reverse()	Reverses the order of the elements in an array
shift()	Removes the first element of an array, and returns that element

ARRAYS

```
<script type="text/javascript">
var names=["John","Pinky","George"];
document.write("<b>Concatenated Names : </b>"+names.join(" ")+"<br />");
document.write("<b>Index of \"Pinky\" : </b>"+names.indexOf("Pinky")+"<br />");
document.write("<b>Pop last element : </b>"+names.pop()+"<br />");
document.write("<b>Push \"Tom\" element at last </b>"+names.push("Tom")+"<br />");
document.write("<b>Reverse Elements : </b>"+names.reverse()+"<br />");
document.write("<b>Shift elements from 1st : </b>"+names.shift()+"<br />");
document.write("<b>Unshift elements to 1st : </b>"+names.unshift("Tom")+"<br />");
document.write("<b>Slice element from 0 to 2 : </b>"+names.slice(0,2)+"<br />");
document.write("<b>Sort Array : </b>"+names.sort()+"<br />");
document.write("<b>Splice element at pos 1 :
        </b>"+names.splice(1,0,"Prem","Caesar"));
document.write(names+"<br />");
document.write("<b>Delete an element at position 0 : </b>"+names.splice(0,1)+"<br />");
document.write("<b>Name List : </b>"+names);
</script>
```

BOOLEAN OBJECT

TEKNOTURF

Syntax

```
var x = new Boolean(expression);
```

Function	Description
toSource	Returns a string which represents the source code of a boolean object.
toString	Returns a string representing the specified boolean object.
valueof	Returns the primitive value of a boolean object.

Initial Boolean value : false

```
new Boolean (false);
new Boolean ();
new Boolean ("");
new Boolean (0);
new Boolean (null);
```

Boolean Methods

BOOLEAN OBJECT

TEKNOTURF

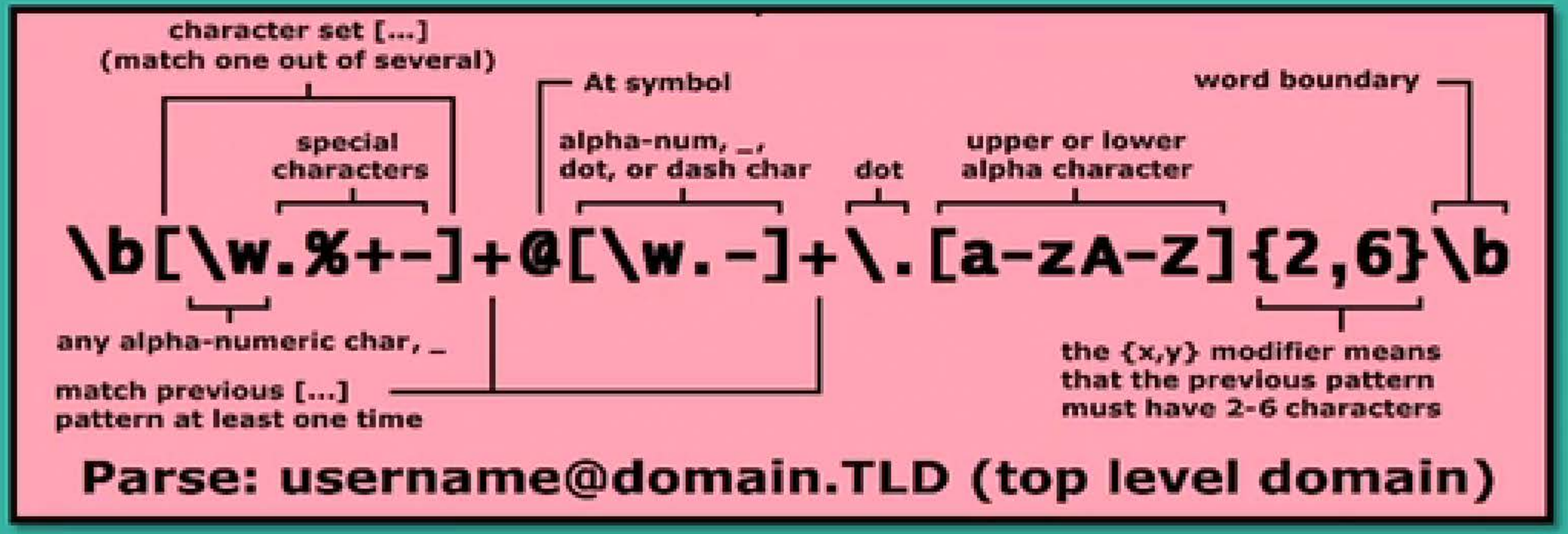
Example

```
<html>
  <head>
    <title>JavaScript toString() Method</title>
  </head>
  <body>
    <script type="text/javascript">
      var flag = new Boolean(false);
      document.write( "flag.toString is : " +
flag.toString()+"<br>" );
      document.write( "flag.valueOf is : " +
flag.valueOf()+"<br>");
      document.write( "flag source is : " + flag.toSource());
    </script>
  </body>
</html>
```

```
flag.toString is : false
flag.valueOf is : false
flag source is : (new Boolean(false))
```

REGEXP OBJECT

A regular expression is an object that describes a pattern of characters.



REGEXP OBJECT

TEKNOTURF

1. A regular expression could be defined with the RegExp() constructor, as follows:
var pat = new RegExp(pattern, modifiers); OR var pat = /pattern/modifiers;

2. Pattern : A string that specifies the pattern of the regular expression or another regular expression.

5 seconds

3. modifiers : Specifies global, case-insensitive and multiline matches

4. A modifier can be : "g" – Finds all global matches
"i" – Does case-insensitive matches
"m" – Does multiline matches

REGEXP OBJECT

TEKNOTURF

In Regular expressions, brackets have a special meaning.

They are used to find a range of characters.

Expression	Description
[abc]	Finds any one character between the brackets.
[^abc]	Finds any one character NOT between the brackets.
[0-9]	It matches any decimal digit from 0 through 9.
[^0-9]	It matches any decimal digit not from 0 through 9.
[a-z]	It matches any character from lowercase a through lowercase z.
[A-Z]	It matches any character from uppercase A through uppercase Z.
[a-Z]	It matches any character from lowercase a through uppercase Z.
(x/y)	Finds any of the alternatives specified . x or y

REGEXP OBJECT METHODS

TEKNOTURF

Here is a list of methods associated with RegExp along with their description.

Method	Description
exec()	Returns the first match. If no match returns null Syntax: RegExpObject.exec(string);
test()	Returns true or false Syntax: RegExpObject.test(string);
toString()	Syntax: RegExpObject.toString ( );

REGEXP OBJECT

TEKNOTURF

```
<body>
<p> To do a global case-insensitive search for the characters "a","i" and "s" in
the string
<h4 style="color:red" >"I saw Susie sitting in a shoeshine shop“ </h4> </p>
<script>
   var str = "I saw Susie sitting in a shoeshine shop";
   var patt1 = /[ais]/gi;    //or var patt1 = new RegExp(/[ais]/,"gi");
   var result = str.match(patt1);
   document.write(result);
</script>
</body>
```

To do a global case-insensitive search for the characters "a","i" and "s" in the string

"I saw Susie sitting in a shoeshine shop“

I,s,a,S,s,i,s,i,i,i,a,s,s,i,s

JAVASCRIPT DOCUMENT OBJECT MODEL - DOM

TEKNOTURF

Every web page displayed inside a browser window can be considered as an object

The DOM defines the logical structure of objects and the way in which an object is accessed and manipulated

JavaScript arranges objects in a Document Object Model or DOM

JAVASCRIPT DOCUMENT OBJECT MODEL - DOM

TEKNOTURF

Object	JavaScript Object Name
A frame within the browser window	frame
The history list combining the Web pages the user has already visited in the current session	history
The Web browser being run by the user	navigator
The URL of the current Web page	location
The Web page currently shown in the browser window	window
A hyperlink on the current Web page	link
A target or anchor on the current Web page	anchor
A form on the current Web page	form

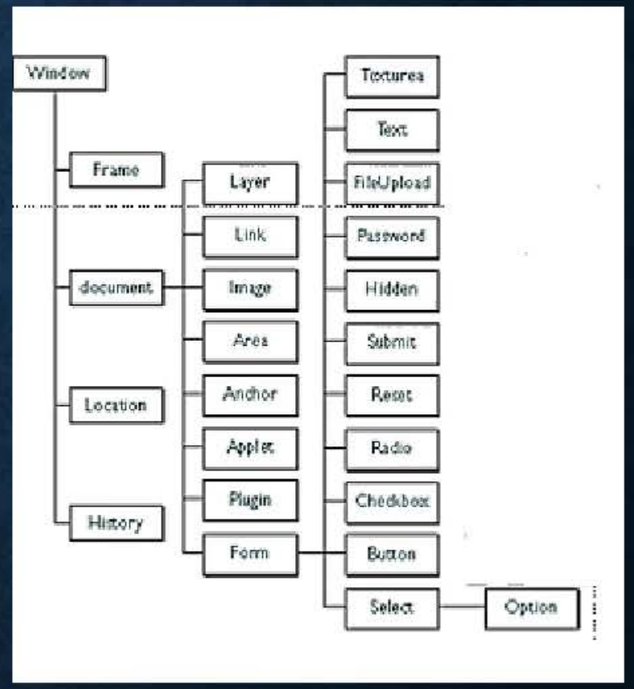
DOM HIERARCHY

TEKNOTURF

01 JavaScript uses Document Object Model(DOM) to navigate the HTML document in a hierarchy

02 The document object model can be thought of as a hierarchy moving from the most general object to the most specific.

03 Example: To access the text field element or the text object, use : window.document.form.text



DOM PROPERTIES

TEKNOTURF

01 There are several ways of working with properties:

- the value of a property can be changed
- property's value can be stored in a variable
- you can test whether the property equals a specified value using an If...then expression

02 Some properties are read - only, which means you can read the property value, but cannot modify it.

03 The syntax for changing the value of a property is : **object.property = value/expression** (Ex : document.bgColor="#FFFFFF")

WINDOW OBJECT

TEKNOTURF

Window object is the top level object in DOM

Window Object Properties:

- Window.innerHeight – represents the inner height of the browser window
- Window.innerWidth – represents the inner width of the browser window
- Note : It works for IE, Chrome, Firefox, Opera and Safari

Window Object Methods:

- Window.open() – opens a new window
- Window.close() – closes the current window
- Window.moveTo() – moves the current window
- Window.resizeTo() – resizes the current window

WINDOW OBJECT EXAMPLE

TEKNOTURF

```
<!DOCTYPE html>
<html>
<body>
<button onclick="openWindow()">Open gmail.com in a new  window </button>
<button onclick="closeWindow()">Close the window mail.com</button>
<script>
var myWindow;
function openWindow()
{
   myWindow = window.open("http://www.gmail.com", "_blank",   "width=500, height=500");
}
function closeWindow() {
   myWindow.close();
}
</script>
</body>
</html>
```

_blank : Loads the URL in a new window

_self : URL replaces the current page

WINDOW OBJECT EXAMPLE

TEKNOTURF

```
<!DOCTYPE html>
<html><body>
<script>
var newWindow = window.open("","msgWindow", "width=300,height=100,
                                                menubar=yes,  status=yes,resizable=yes");

newWindow.document.write("<html><head><title>New Window
</title></head><body><h1>Hello World</h1></body></html>");
</script>
</body>
</html>
```

FRAME OBJECT

TEKNOTURF

- It is the representation of HTML frame which belongs to a HTML frameset.
- Is a property of the window object.
- Properties
  - frames - An array of frames in a window or frame set.
  - name - Name of the frame defined using the name attribute of the <frame> tag.
  - length - Length of the frames array.
  - parent - Parent of the current frame.
  - self - Current frame.

FRAME OBJECT

TEKNOTURF

01 **METHODS**

- blur()
- focus()
- setInterval()
- clearInterval()
- setTimeout(expression, milliseconds) - Makes a timeout value and returns a timeout ID. After the number of milliseconds expire, the expression is run.
- clearTimeout(timeout)

02 **EVENTS**

- onBlur
- onFocus

NAVIGATOR OBJECT AND ITS METHODS

TEKNOTURF

01 It is an object that has information about the browser and does not come under window object.

02 Properties

- appName : The name of the browser ex: Mozilla Firefox
- appVersion : The version of the browser which may include a compatibility value and operating system name.
- cookieEnabled : Boolean value depending on whether cookies are enabled in the browser.
- mimeTypes : An array of MIME type descriptive strings that are supported by the browser. Internet Explorer supports the mimeTypes collection, but it is always empty
- userAgent : Describes the browser associated user agent header

NAVIGATOR OBJECT AND ITS METHODS

TEKNOTURF

```
<body>

<button onclick="getBrowserName()">  Click Here to know the Navigator
Properties</button>

<p id="navigatorProperties"></p>

<script>

function getBrowserName() {

   var name = "Name of the browser is " + navigator.appName;

   name = name+"<br />Version info " + navigator.appVersion;

   name = name+"<br />Cookies enabled " + navigator.cookieEnabled;

   name = name+"<br />User-agent header sent: " + navigator.userAgent;



   document.getElementById(" navigatorProperties").innerHTML = name;

}

</script>
```

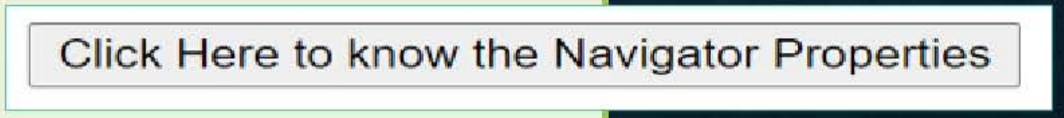
Methods

javaEnabled()

Returns a boolean indicating whether javascript is enabled or not in the browser

taintEnabled()

Returns a boolean indicating whether the tainting is enabled or not in the browser. This method is removed in JavaScript version 1.2. Tainting is a security protection mechanism for data.



DOCUMENT OBJECT

TEKNOTURF

Document is the property of the window object.

Each page has only one document object

In HTML, DOM is a node.

Document object is the root node

Provides properties and methods

Document object refers the html document's <body> tag

DOCUMENT OBJECT - PROPERTIES

TEKNOTURF

```
<body>
    <h2>Learning Objects in JavaScript</h2>
      <script type="text/javascript">
            document.title="JavaScript Object Example";
            document.bgColor="grey";
            document.fgColor="white";
 </script>
</body>
```

DOCUMENT OBJECT - METHODS

TEKNOTURF

Method	Description
document.getElementById()	Finding an element by element id
document.getElementsByTagName()	Finding elements by tag name
document.getElementsByClassName()	Finding elements by class name
document.forms[]	Finding the form element with id passed as argument

5 seconds

Inside the document object, we have a form object created like the one below.

```
<form name="userlogin">
 User name : <input type="text" id="userId" name="userName">
</form>
```

03:55

THE GETELEMENTBYID() METHOD

TEKNOTURF

getElementById() is a function that helps to access or set value into the document elements directly

Following is the syntax to **get** the text from between container tags like <p>, ,,<td> etc.

Syntax:

document.getElementById("elementId").value;

5 seconds

Following is the syntax to **set** the text between container tags like <p>, ,,<td> etc.

Syntax:

document.getElementById("elementId").**innerText**="value";

Here, "**elementId**" represents the value of the "**id**" attribute of the form element like textbox, radio button, check box, text area, etc.

THE GETELEMENTBYID() METHOD

TEKNOTURF

According to the DOM, we can access the value inside textbox using JavaScript in several ways like:

```
<!DOCTYPE html>
<html>
<body>
<form id="formid" name="formname">
<input type="hidden" id="componentid" name="componentname" value="Hello">
</form>
<script>
document.write("1"+document.getElementById("componentid").value+"<br/>");
document.write("2.1"+document.formname.componentid.value+"<br/>");
document.write("2.2"+document.formname.componentname.value+"<br/>");
document.write("3.1"+document.forms.formname.componentid.value+"<br/>");
document.write("3.2"+document.forms.formname.componentname.value+"<br/>");
document.write("4.1"+document.forms["formid"]["componentid"].value+"<br/>");
document.write("4.2"+document.forms["formname"]["componentname"].value+"<br/>");
</script>
</body>
</html>
```

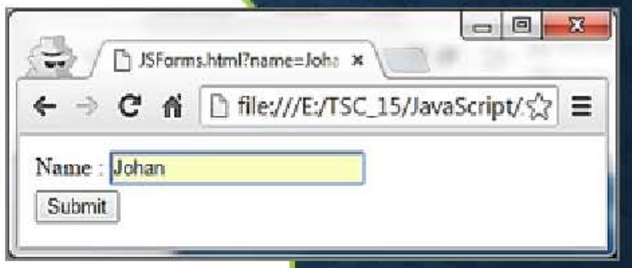
1Hello
2.1Hello
2.2Hello
3.1Hello
3.2Hello
4.1Hello
4.2Hello

Note: id should be unique

ACCESSING A FORM ELEMENT BY FIELD NAME ON INVOKING A FUNCTION

TEKNOTURF

```
<head>
 <script type="text/javascript">
   function getData(){
        var name  =  document.myForm.name.value;
        alert(name);
   }
 </script>
</head>
<body>
 <form name="myForm" onSubmit="getData()">
   <table>
        <tr><td>Name :</td><td><input type="text" name="name"/></td></tr>
        <tr><td colspan="2"><input type="submit" value="Submit"/></td></tr>
    </table>
 </form>
</body>
```



THE GETELEMENTSBYCLASSNAME() METHOD

TEKNOTURF

Example: var options = document.getElementsByClassName("msgCheckbox");
//This gets all the elements from within the checkbox component, with the class name "msgChecBox" - as a collection

This method is used to access an element using its class name.

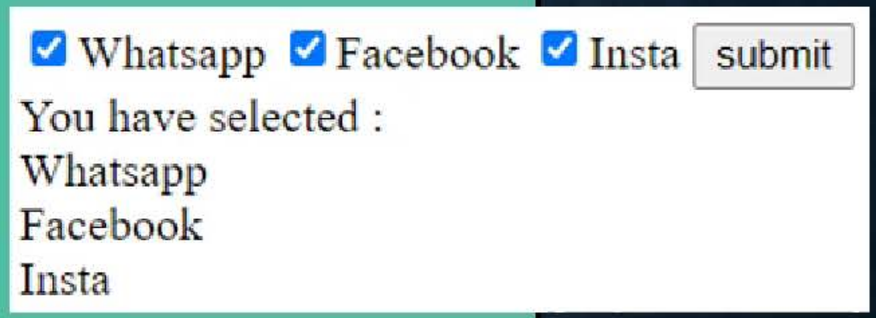
Individual elements can be accessed using index.
options[0] .value - represents the first element in the collection

Syntax: var variablename = document.getElementsByClassName("classname");

THE GETELEMENTSBYCLASSNAME() METHOD

TEKNOTURF

```
<!DOCTYPE html>
<html>
<body>
<form onsubmit="return check()">
<input type="checkbox" id="id1" class="msgCheckbox" value="Whatsapp">Whatsapp
<input type="checkbox" id="id2" class="msgCheckbox"  value="Facebook">Facebook
<input type="checkbox" id="id3" class="msgCheckbox" value="Insta">Insta
<input type="submit" id="submit" value="submit">
<div id="div1"><div></form>
<script>
function check()
{
    var values=document.getElementsByClassName("msgCheckbox");
    document.getElementById("div1").innerHTML="You have selected : <br/>";
    for(var  i=0 ; i < values.length ; i++) {
                    if(values[i].checked) document.getElementById("div1").innerHTML += values[i].value+"<br/>";
    }
return false;
}
</script>
</body>
```



WORKING WITH REGULAR EXPRESSIONS

TEKNOTURF

In JavaScript, regular expressions are also objects. These patterns are used with the exec and test methods of RegExp.

Regular expressions are patterns used to match character combinations in strings.

Can be used with match, replace, search and split methods of string.

QUANTIFIERS

TEKNOTURF

Expression	Description
+	Represents "One or more", same as {1,}.
*	Represents "Zero or more", same as {0,}.
?	Represents "Zero or one", same as {0,1}.
$	Represents value at the end
^	Represents value at the beginning. But if it is inside [^] – Not in the given range

QUANTIFIERS

TEKNOTURF

Expression	Description
k+	It matches any string containing at least one occurrence of ‘k’.
k*	It matches any string containing zero or more k’s.
k?	It matches any string containing zero or one k's.
k{N}	It matches any string containing a sequence of N k's
k{2,3}	It matches any string containing a sequence of two or three k's
k{2,}	It matches any string containing a sequence of at least two k's.
k$	It matches any string with k at the end of it.
^k	It matches any string with k at the beginning of it.
?=k	Matches any string that is followed by a specific string k
?!k	Matches any string that is not followed by a specific string k

QUANTIFIERS

TEKNOTURF

Example

```
<body>

<p> To do a global search for an "e" followed by zero or more "t" in the string

<h3 style="color:red" >"The secret of getting ahead is getting started."</p>

<p id="demo" style="color:green"></p> </h3>

<script>

   var str = "The secret of getting ahead is getting started.";

   var patt1 = /et*/g;

   var result = str.match(patt1);

   document.getElementById("demo").innerHTML = result;

</script>

</body>
```

5 seconds

To do a global search for an "e" followed by zero or more "t" in the string

"The secret of getting ahead is getting started."

e,e,et,ett,e,ett,e

07:57

META CHARACTERS

TEKNOTURF

Expression	Description
.	Finds a single character
\s	Finds a whitespace character (space, tab, newline)
\S	Finds a non-whitespace character
\d	Finds a digit (0-9)
\D	Finds a non-digit
\w	Finds a word character (a-z, A-Z, 0-9, _)
\W	Finds a non-word character
\b	Finds a match at the beginning/end of a word
\B	Finds a match not at the beginning/end of a word
\0	Finds a NULL character

META CHARACTERS - EXAMPLE

TEKNOTURF

```
var str="5678941982384";
   var patt1 = /[1-5]/g;
   var result = str.match(patt1);
   document.writeln("<br /> [1-5]/g in \""+str+"\" is "+result);

//output  :  5,4,1,2,3,4
-----------------------------------------------------------------------------
   var str="Hello7815";
   var patt1 = /\d/g;
   var result = str.match(patt1);
   document.writeln("<br /> "+result);

//output  :  7,8,1,5
-----------------------------------------------------------------------------
   var str="People\nwho\nlive\nin\nglass\nhouses\nshouldn't\nthrow\nstones";
   var patt1 = /es$/gm;
   var result = str.match(patt1);
   document.writeln(result);

//output : es,es
```

REGEXP MODIFIERS

TEKNOTURF

Using “i” modifier to do a **case–insensitive** search for characters in the string

```
<body>
<p>Example to do a case-insensitive search for "Asia" in the string <br/>"Indians are the Italians of
Asia and vice versa." .</p>
<p id="demo" style="color:red"></p>
<script>
var str = "Indians are the Italians of Asia and vice versa.";
var patt1  =  /asia/i ;
var result = str.match(patt1);
if(result!=null)
document.getElementById("demo").innerHTML =  "Asia string found in \""+str+"\"";
 else
document.getElementById("demo").innerHTML =  "Asia string not found  in \""+str+"\"";
</script>
</body>
```

We can also create RegExp object as: var patt1=new RegExp(“/Asia/", "i")

Example to do a case-insensitive search for "Asia" in the string "Indians are the Italians of Asia and vice versa." .

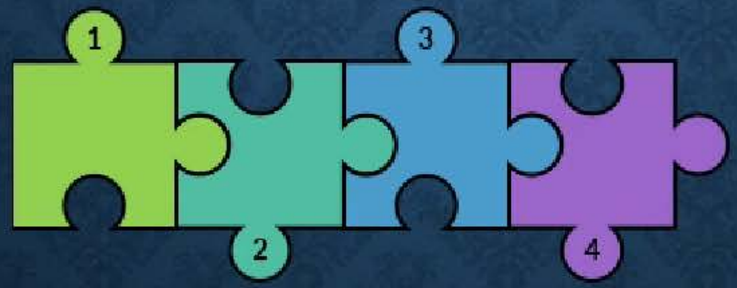
Asia string found in "Indians are the Italians of Asia and vice versa."

REGEXP USING STRING METHODS

TEKNOTURF

In JavaScript, regular expressions are often used with following **string methods**: match() , search(), replace()

The search() method uses an expression to search for a match, and returns the position of the first occurrence of match. **Syntax:** string.search(regexp);



The match () This method is used to retrievethe matches when matching a string against a regular expression. **Syntax:** string.match (param/regexp)

The replace() method returns a modified string where the pattern is replaced. **Syntax**: string.replace(regexp/substr, newSubStr/function[, RegExp flags]);

INTRODUCTION TO JQUERY

TEKNOTURF

jQuery is a JavaScript Library created by John Resig in 2006.

jQuery is a fast, small, and feature-rich JavaScript library

It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers*

WHY JQUERY?

TEKNOTURF

- Cross Browser Support
- Extensibility through plug-ins
- DOM manipulation
- Event Handling
- AJAX Processing
- Creating effects / animations

USING JQUERY LIBRARIES

TEKNOTURF

- jQuery can be used in two ways

Local Installation - can be downloaded directly to the local machne and use the script tag to refer the file location in the HTML document[2]

CDN Based Version - jQuery Content Delivery Network(CDN) can be used just by referencing the file in the script tag directly from the jQuery CDN domain[1]

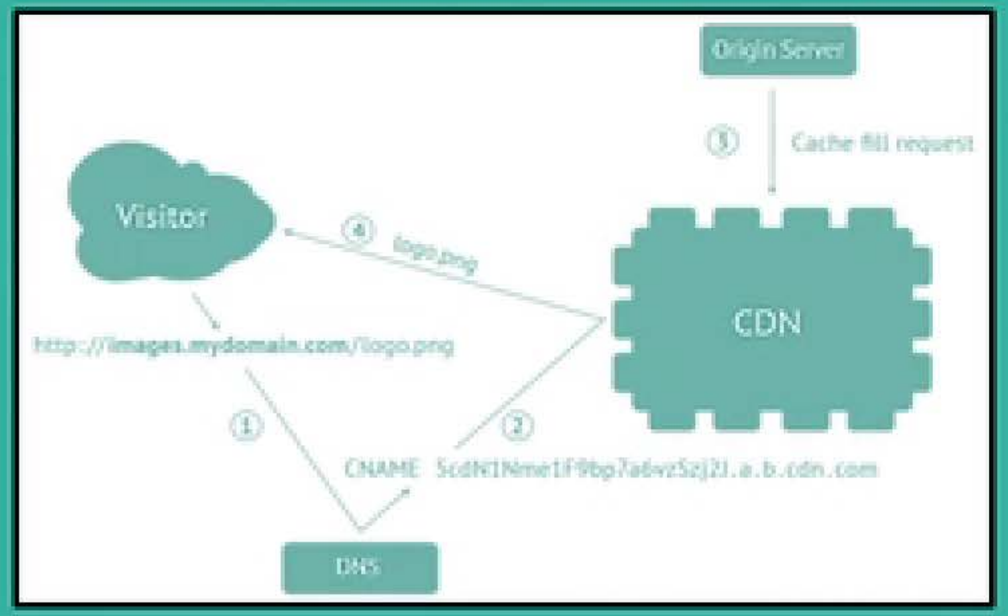
CDN INSTALLATION

TEKNOTURF

❖ Following are the releases of Google and Microsoft host compressed and uncompressed versions of jQuery

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
```

```
<script src="https://ajax.aspnetcdn.com/ajax/jQuery/jquery-3.6.0.min.js"></script>
```



Production version (compressed): The production version is meant to be used in a working application.

we can download jquery.min.js and access it locally using relative address mode

WORKING WITH JQUERY

TEKNOTURF

All the JavaScript functions that needs to be performed, happens only after the document is ready.

Example - adding events

```
$(document).ready(function()
{
        //JavaScript code
} );
```

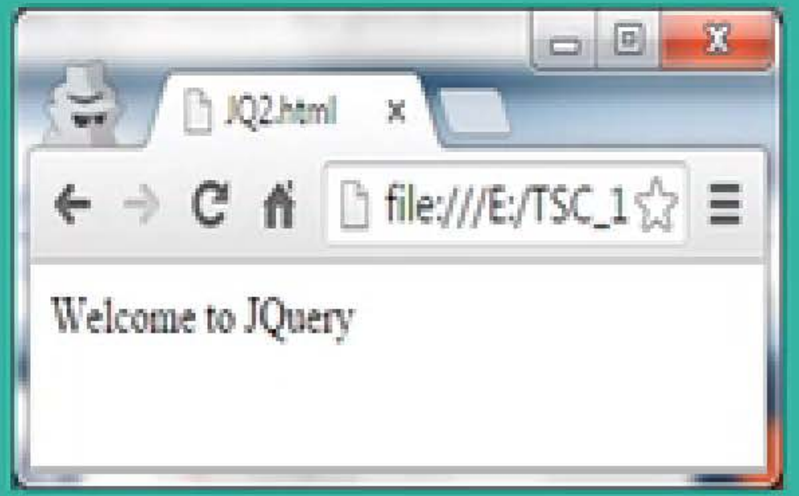
Everything inside it will load as soon as the DOM is loaded, and before the page contents are loaded.

EXAMPLE

```
<html>
<head>
<script type = "text/javascript" src =
"https://ajax.googleapis.com/ajax/libs/jquery/
3.6.0/jquery.min.js"></script>
<script type = "text/javascript" language =
"javascript">
        $(document).ready(function() {
        document.write("Welcome to JQuery");
</script>
</head>
<body>
</body>
</html>
```

Here, **'document'** refers to the HTML element and **ready()** is the action on the selected element.

Output



JQUERY SELECTORS

TEKNOTURF

- Selectors are used to select one or more HTML elements for manipulation using jQuery.
- jQuery selectors start with the dollar sign and parentheses – $()

Selectors can be :

•Tag name - $('name of the tag') - $('p')

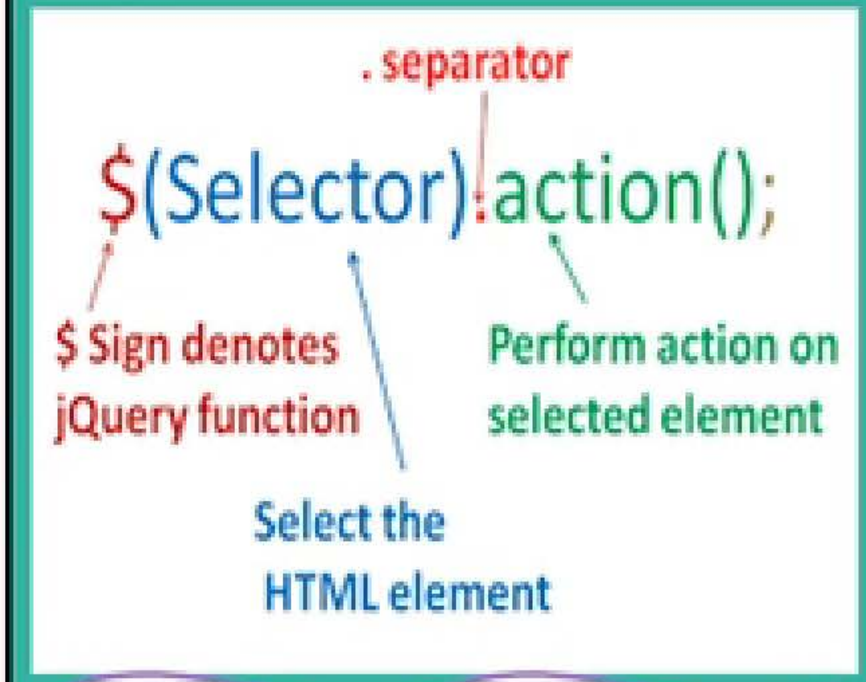
Selects all elements which match with the given element Name.

•Id - $('#p_id')

Selects a single element which matches with the given ID.

•Class - $('.p_class')

Selects all elements which match with the given Class.

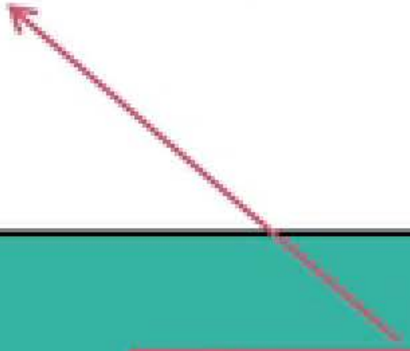


$() is a factory function and it is equivqlent to **jQuery()** function.
jQuery() can be used as an alternative to **$()**.

USING TAG NAME SELECTOR

TEKNOTURF

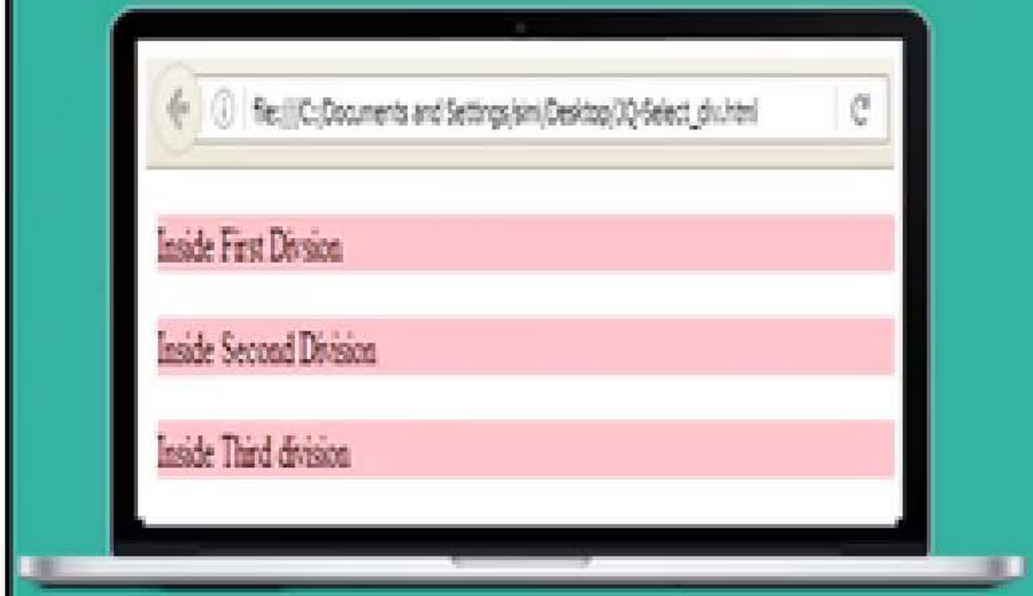
```
<html>
<head>
<script type = "text/javascript" src =
"https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script type = "text/javascript" language = "javascript">
        $(document).ready(function()  {
        $("div").css("background-color", "pink"); });
 </script>
</head>
```



This would select all the div tags in the html file

USING TAG NAME SELECTOR

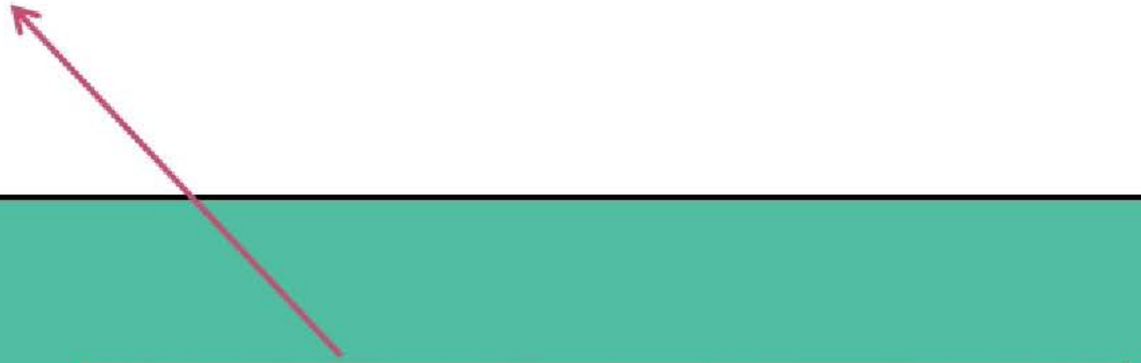
```
<body>
   <div class = "big" id = "div1">
      <p>Inside First Division</p>
   </div>
   <div class = "medium" id = "div2">
      <p>Inside Second Division</p>
   </div>
   <div class = "small" id = "div3">
      <p>Inside Third division</p>
   </div>
</body>
</html>
```



USING #ID SELECTOR

TEKNOTURF

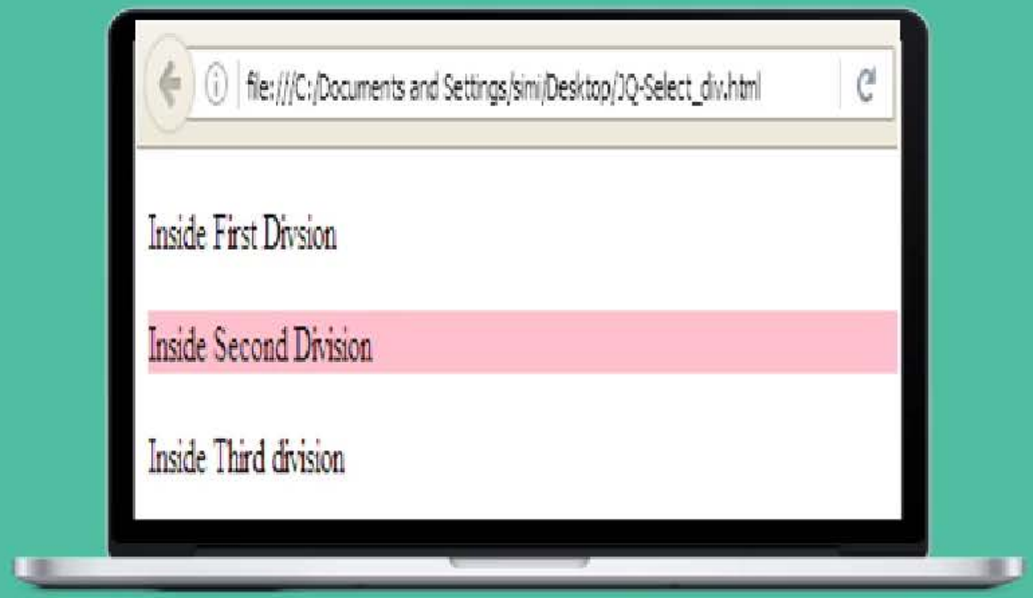
```
<html>
<head>
<script type = "text/javascript" src =
"https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script type = "text/javascript" language = "javascript">
        $(document).ready(function()  {
        $("#div2").css("background-color", "pink"); });
 </script>
</head>
```



This would select the div tag which has the id div2

USING #ID SELECTOR

```
<body>
   <div class = "big" id = "div1">
      <p>Inside First Division</p>
   </div>
   <div class = "medium" id = "div2">
      <p>Inside Second Division</p>
   </div>
   <div class = "small" id = "div3">
      <p>Inside Third division</p>
   </div>
  </body>
</html>
```



USING .CLASS SELECTOR

TEKNOTURF

```
<html>
<head>
<script type = "text/javascript" src =
"https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></scri
pt>
<script type = "text/javascript" language = "javascript">
        $(document).ready(function()  {
        $(“.big").css("background-color", "pink"); });
 </script>
</head>
</html>
```

.css() is an action which sets the css attributes to the specified elements selected by the selector

This would select only the division which belongs to the class ‘big’

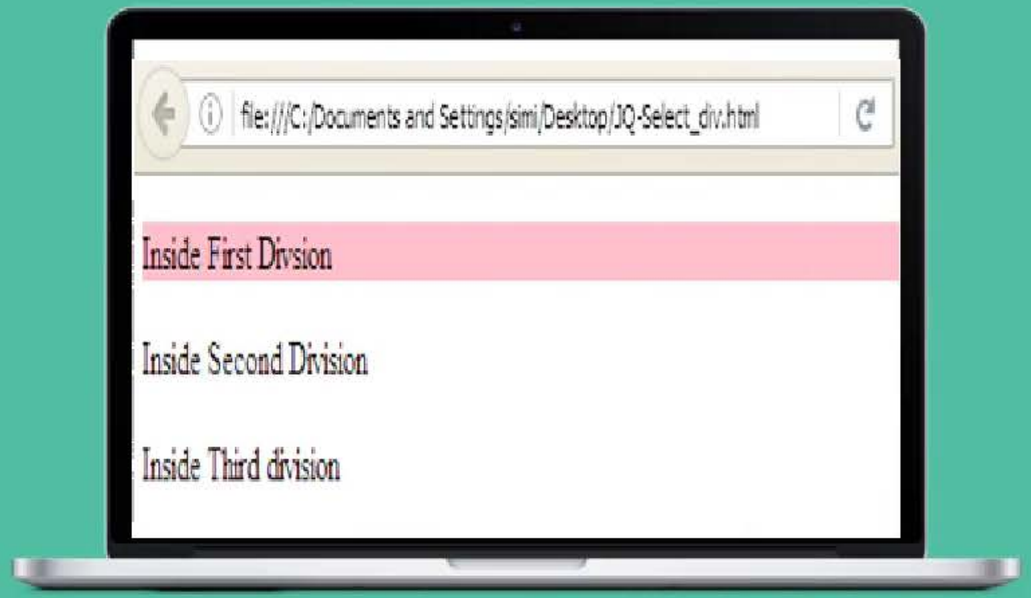
5 seconds

09:01

USING .CLASS SELECTOR

```
<body>
   <div class = "big" id = "div1">
      <p>Inside First Division</p>
   </div>
   <div class = "medium" id = "div2">
      <p>Inside Second Division</p>
   </div>
   <div class = "small" id = "div3">
      <p>Inside Third division</p>
   </div>
 </body>
</html>
```

5 seconds



09:11

USING ATTRIBUTE SELECTOR

TEKNOTURF

- The attribute selector is used to select an element by one of its HTML attributes, such as a link's target attribute or an input's type attribute, etc.
- It selects elements that have the specified attribute with a value exactly equal to a certain value

Example: The following jQuery code will select and highlight all the text inputs :

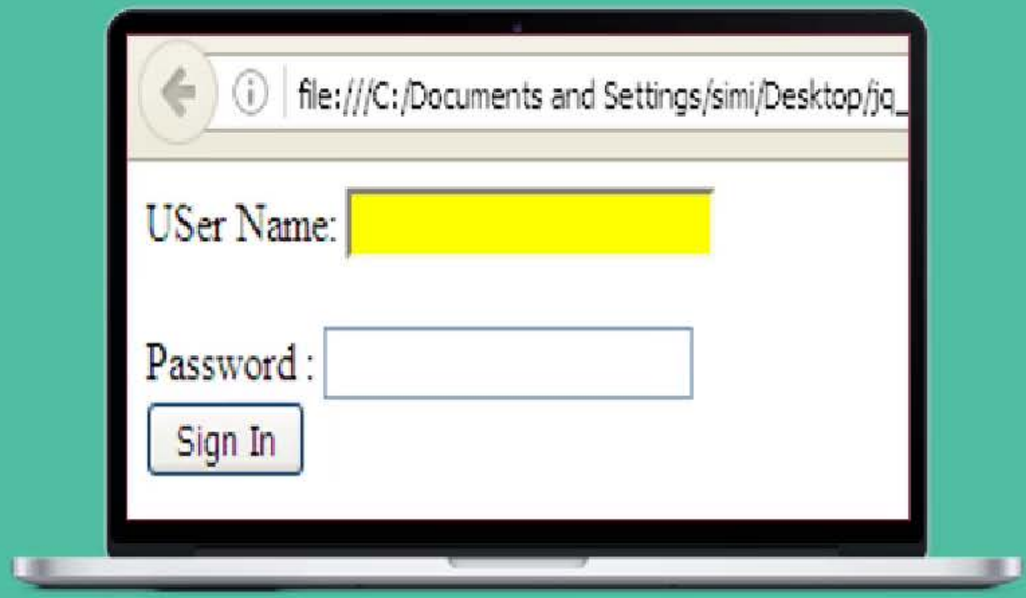
```
$('input[type="text"]')
```

<input> elements with the type="text", when the document is ready.

USING ATTRIBUTE SELECTOR

TEKNOTURF

```
<html > <head>
<<script src="
https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.
js"></script>
<script type="text/javascript">
$(document).ready(function(){
   // Highlight paragraph elements
   $('input[type="text"]').css("background", "yellow");
});
</script> </head>
<body>    <form>
      <label>USer Name: <input type="text"></label>
      <label>Password: <input type="password"></label>
      <input type="submit" value="Sign In">
   </form> </body> </html>
```



.css() is an action which sets the css attributes to the specified elements selected by the selector

FORM ELEMENT SELECTORS

TEKNOTURF

01 **INPUT**

This selector basically selects all form controls

Selects all input, text area, select and button elements

02 **CHECKBOX**

Selects all elements of type checkbox

03 **RADIO SELECTOR**

Selects all elements of type radio

04 **SUBMIT SELECTOR**

Selects all elements of type submit

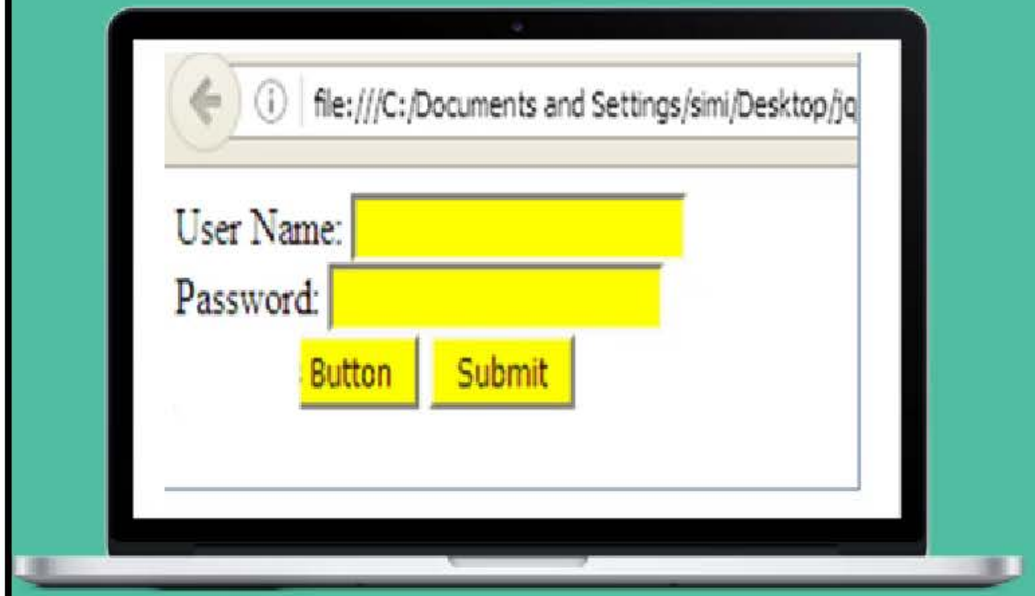
05 **TEXT SELECTOR**

Selects all input elements of type text.

USING ELEMENT SELECTOR

TEKNOTURF

```
<html><head> <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js
">
</script>
<script>
$(document).ready(function(){
   $(":input").css("background-color", "yellow"); });
</script></head>
<body> <form action="">
 Username: <input type="text" name="user"><br>
 Password: <input type="password" name="password"><br>
 <button type="button">Button</button>
    <input type="submit" value="Submit"><br>
</form></body></html>
```



OTHER JQUERY SELECTORS

TEKNOTURF

Selector	Description
$('*')	This selector selects all elements in the document
$("p > *")	This selector selects all elements that are children of a paragraph element
$("li:not(.myclass)")	Selects all elements matched by that do not have class="myclass"
$("p a.specialClass")	This selector matches links with a class of specialClass declared within <p> elements
$("ul li:first")	This selector gets only the first element of the .(*deprecated)
$(":empty")	Selects all elements that have no children
$("p:empty")	Selects all elements matched by <p> that have no children
$("div[p]")	Selects all elements matched by that contain an element matched by <p>

OTHER JQUERY SELECTORS

TEKNOTURF

Selector	Description
$("li:even")	Selects all elements matched by that have an even index value.(*deprecated)
$("tr:odd")	Selects all elements matched by <tr> that have an odd index value.(*deprecated)
$("li:first")	Selects the first element.(*deprecated)
$("li:last")	Selects the last element.(*deprecated)
$("li:visible")	Selects all elements matched by that are visible.
$("li:hidden")	Selects all elements matched by that are hidden.
$(":radio")	Selects all radio buttons in the form.
$(":checked")	Selects all checked boxes in the form.

JQUERY SELECTOR:CONTENT FILTER

TEKNOTURF

jQuery selector:CONTENT Filter

```
<html ><head>
 <style>
 div{ background-color: #5ac5c5;
  width: 50%;  }
 h3{ color:#b8860b; }  </style>
 <script src="https://code.jquery.com/jquery-3.5.0.js"></script>
</head><body>
  <div>
    <h3>Dog Breed</h3>
    <ul>
      <li>Akbash</li>
      <li>Papipoo</li>
      <li></li>
      <li>Bassador</li>
      <li></li>
    </ul>
  </div> <script>
$( "li:empty" )
 .text( "No dog breed listed here" )
 .css( "color", "#FF0000" );</script></body>
</html>
```

Output

Dog Breed

- Akbash
- Papipoo
- No dog breed listed here
- Bassador
- No dog breed listed here

selects all the li element that have no children of the div element from the DOM.The element includes text nodes.

JQUERY SELECTOR:FILTER

TEKNOTURF

```
<html>
<head>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.1/jquery.min.js"></script>
  <script>
    $(document).ready(function(){
      $("p").filter(".intro").css("background-color", "yellow");
      $("li").filter(":even").css("background-color", "red");
    });
  </script>
</head>
<body>
  <h1>Welcome to My Homepage</h1>
  <p>Esther Saradha</p>
  <p class="intro">Corporate Trainer</p>
  <p>My Favourite Foods:</p>

  <ul>
    <li>Pizza</li>
    <li>Club Sandwich</li>
    <li>Open-face Sandwich</li>
    <li>Hamburger</li>
  </ul>
</body>
<html>
```

JQUERY DOM

TEKNOTURF

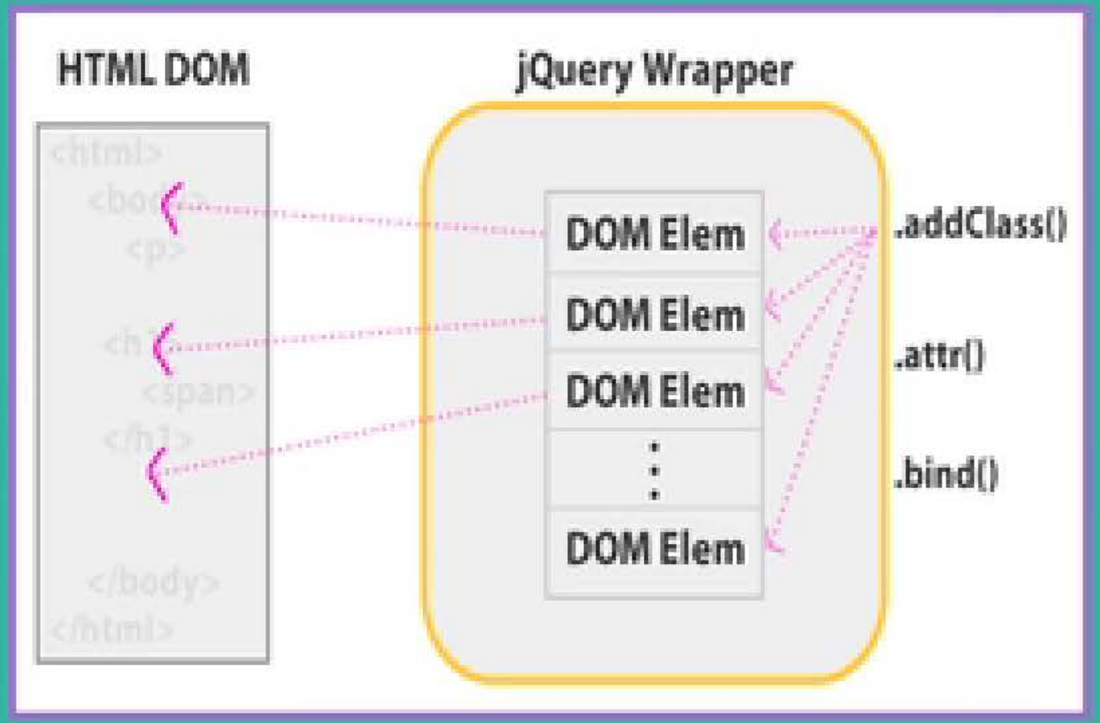
jQuery contains methods for changing and manipulating DOM elements and attributes

Getting content from html

- **text()** - Sets or returns the text content of selected elements

- **html()** - Sets or returns the content of selected elements (including HTML markup)

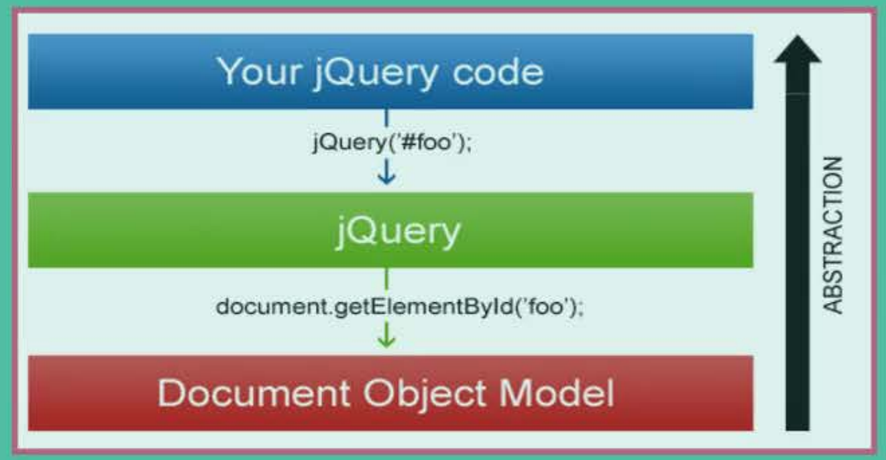
- **val()** - Sets or returns the value of form fields



JQUERY HTML : VAL()

TEKNOTURF

```
<script>
$(document).ready(function(){
   $("button").click(function(){
      alert("Name Value: " +
$("#name").val());
   });
});
</script>
</head>
<body>
         Name: <input type="text"
id="name" value="Krishna">
         <button>Show Value</button>
</body>
```



JQUERY HTML: ATTR METHOD

TEKNOTURF

```
<html>
<head>
   <script>
        $(document).ready(function(){
            $("button").click(function(){

        alert($("#name").attr("value"));
                    });
        });
  </script>
</head>
<body>
    <p><input type="text" id="name"
value="Johan"> </p>
    <button>Show Name</button>
</body>
</html>
```

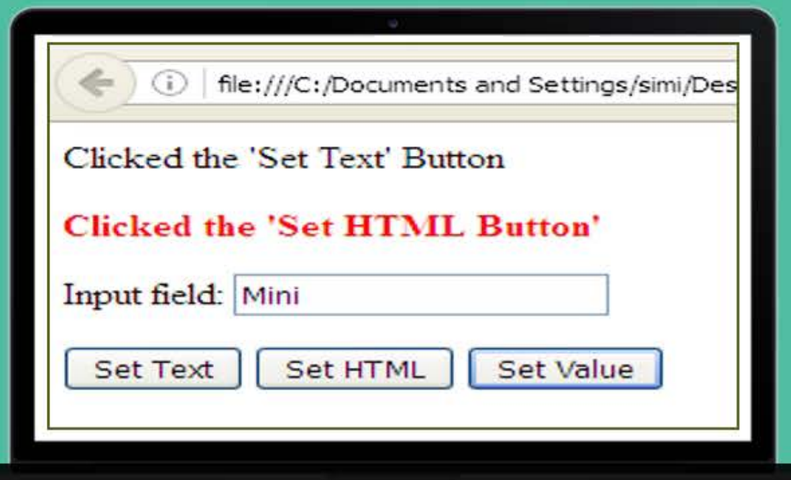
Attribute name of the input field

JQUERY HTML: SET METHOD

TEKNOTURF

```
<script>
$(document).ready(function(){
   $("#b1").click(function(){
      $("#p1").text("Clicked the 'Set Text'
Button");
   });
   $("#b2").click(function(){
      $("#p2").html("<font
color='red'><b>Clicked the 'Set HTML
Button'</b></font>") ;
   });
   $("#b3").click(function(){
      $("#name").val("Mini");
   });   });</script>
```

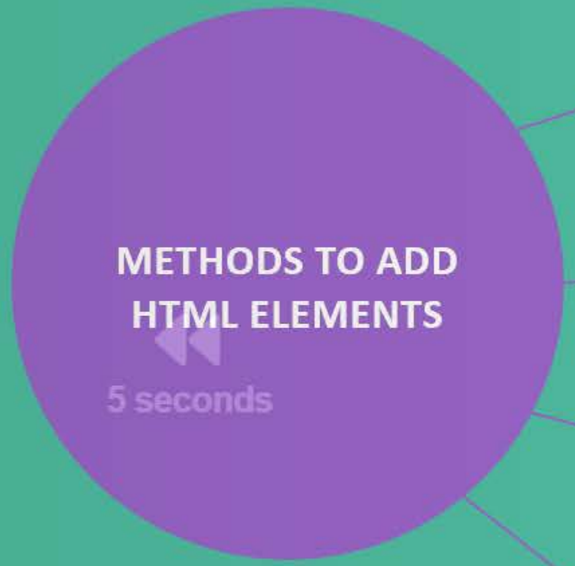
```
<body>
<p id=“p1">This is a paragraph.</p>
<p id=“p2">This is another paragraph.</p>
<p>Input field: <input type="text"
id="name" value="Johan"></p>
<button id="b1">Set Text</button>
<button id="b2">Set HTML</button>
<button id="b3">Set Value</button>
</body>
```



01:26

JQUERY DOM INSERTIONS

TEKNOTURF



1

APPEND()

Inserts content at the end of the selected elements

2

PREPEND()

Inserts content at the beginning of the selected elements.

3

AFTER()

Inserts content after the selected elements.

4

BEFORE()

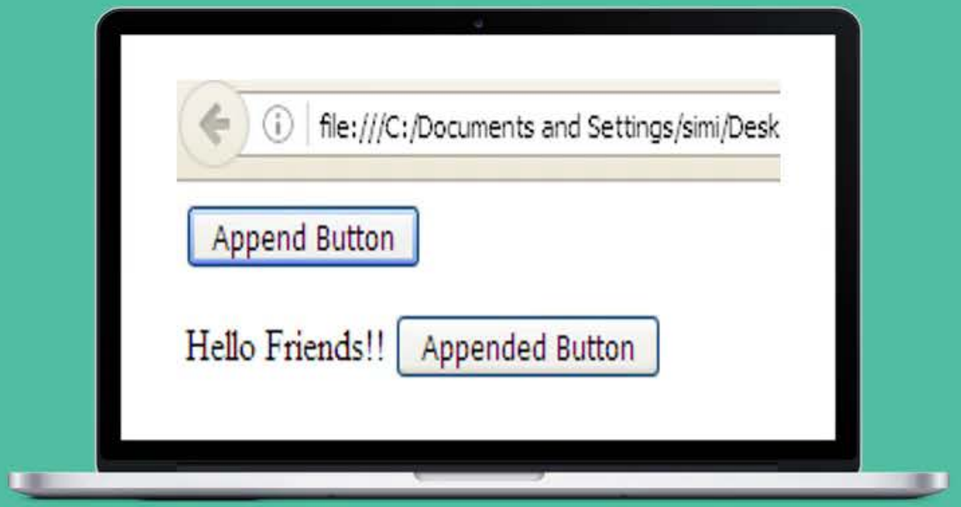
Inserts content before the selected elements

JQUERY HTML : APPEND

TEKNOTURF

```
<html>
<head>
<script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
   $("button").click(function(){
      $("p").append(" <button>Appended Button</button>");
   });
});
</script>
</head>
<body>
                <button>Append Button</button>
                <p> Hello Friends!!</p>
</body> </html>
```

This line will add the button after the <p> tag



02:01

JQUERY HTML : PREPEND

TEKNOTURF

```
<html><head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
   $("#btn2").click(function(){
      $("ol").prepend("<li>ME</li>");
   }); });
</script> </head>
<body>
<ol>
  <li> BE</li>
  <li>BTech</li>
  <li>MTech</li>
</ol>
<button id="btn2">Prepend list item</button>
</body> </html>
```

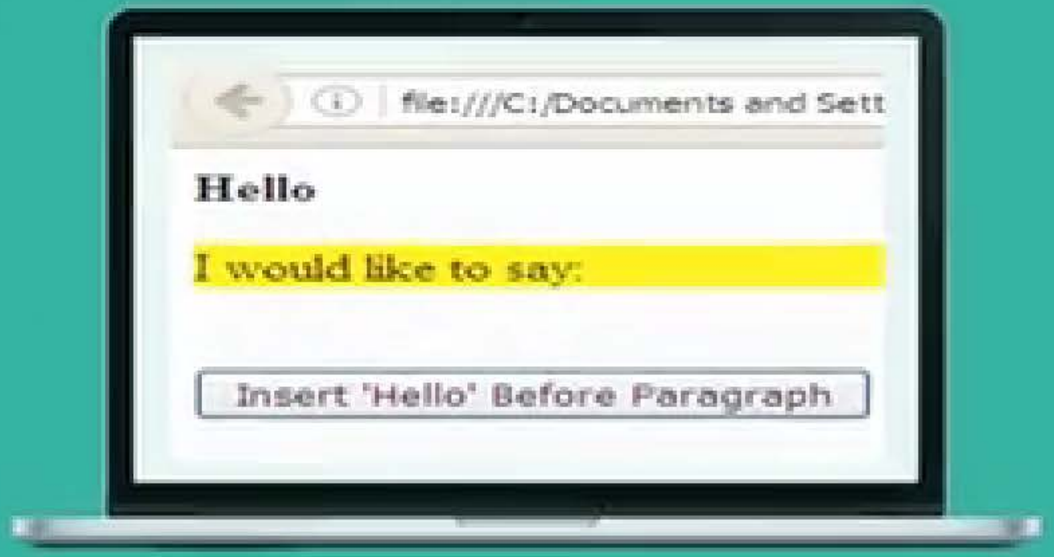
This line will insert ME to the beginning of the list.

This line will add “ME” as the list element at beginning of the list

JQUERY HTML : BEFORE

```
<html><head>
    <style>
        p {  background: yellow;   }
   </style>
<head> <script
src="https://ajax.googleapis.com/ajax/libs/jque
ry/3.1.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
                $("#b1").click(function(){

                $("p").before("<b>Hello</b>");
                });
</script></head>
<body>
<p>I would like to say: </p><br>
<button id="b1"> Insert 'Hello' Before
Paragraph</button>
</body> </html>
```

This function will add "Hello" at the beginning of the paragraph when you click the button

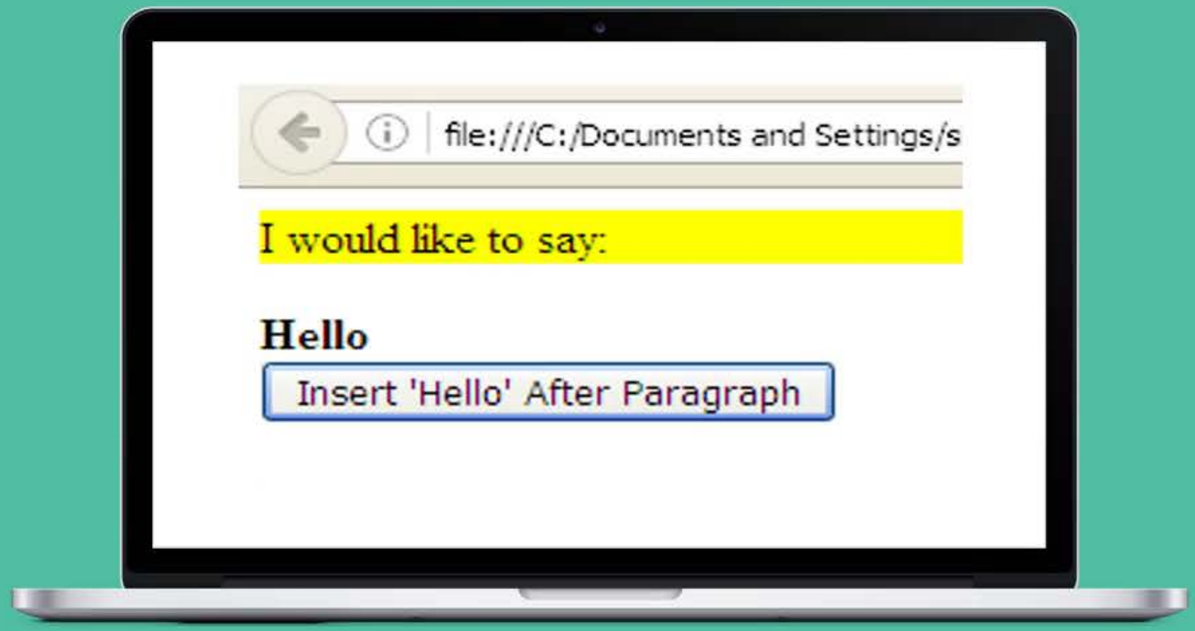


02:16

JQUERY HTML : AFTER

```
<html><head>
 <style>
 p {  background: yellow;  }
 </style>
<head> <script
src="https://ajax.googleapis.com/ajax/libs/jque
ry/3.1.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
                $("#b2").click(function(){

                $("p").after("<b>Hello</b>");
                }); });
</script></head>
<body> <p>I would like to say: </p>
<button id="b2"> Insert 'Hello' After
Paragraph</button>
 </body></html>
```

This function will add “Hello” at the end of the paragraph when you click the button



JQUERY DOM REMOVAL

TEKNOTURF

Remove Elements/Content

- remove() - Removes the selected element (and its child elements)
- empty() - Removes the child elements from the selected element

```
<script>
$(document).ready(function(){
   $("button").click(function(){
      $("#name").remove();
   });
});
</script>
</head>
<body>
                <input type="text" id="name">
                <button>Remove text
box</button>
</body>
```

This will remove the text field when you click on the button

ATTACH EVENTS

TEKNOTURF

01 Using the jQuery Event Model, we can establish event handlers on DOM elements with the on() command

02 .on() method , attaches an event handler function for one or more events to the selected elements.

syntax:

$(Selector/listener).on(eventName [, selector] [, data])

ATTACH EVENTS

TEKNOTURF

jQuery also provides a shortcut method for most common browser and Ajax events.

jQuery(listener/selector).eventName(handlerFunction)

Example: Attaching a click event, with and without the shortcut:

```
// Using on()
 jQuery('div').on('click',function(e){...});

// Using the shortcut
 jQuery('div').click(function(e){...});
```

JQUERY HTML : ALERT

TEKNOTURF

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.
min.js"></script>
<script>
$(document).ready(function(){
   $("p").on("click", function(){
      alert(“Welcome Everybody");
   });
});
</script>
</head>
<body>
<p>Click on the statement.</p>
</body>  </html>
```

When you click on the paragraph, it shows an alert box With the message “ Welcome Everybody’

ATTACH

TEKNOTURF

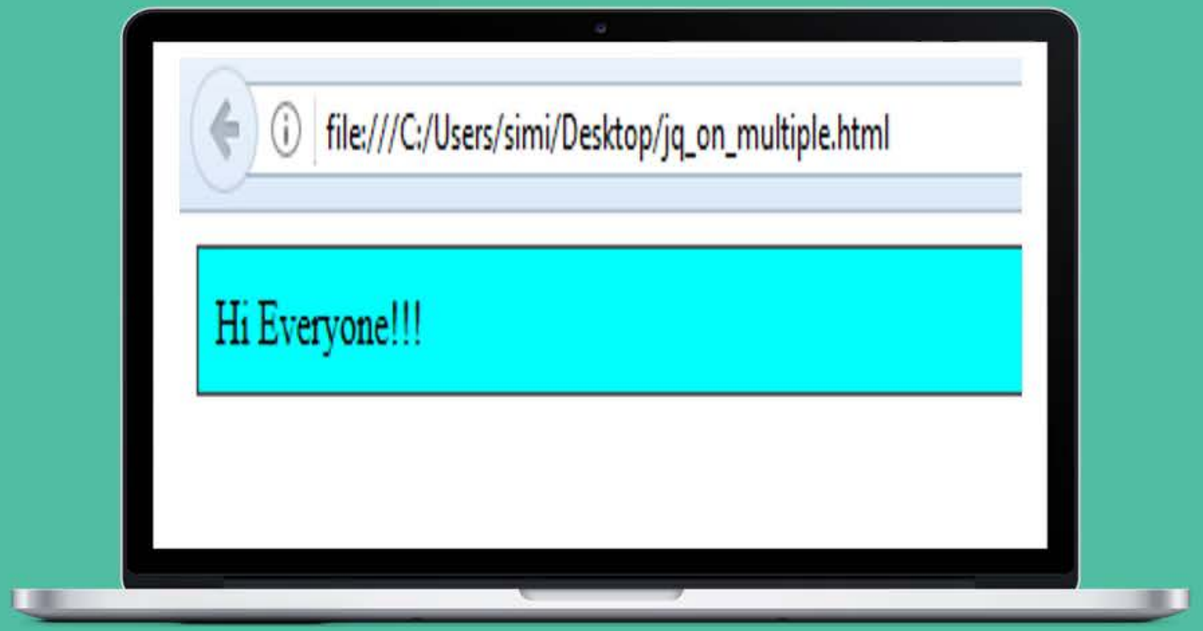
Attaching a Handler to Many Events:

```
<!doctype html> <html lang="en"><head>
<style>
.test {
color: #000;
padding: .5em;
border: 1px solid #444;
}
.active {
color: #900;
}
.inside {
background-color: aqua;
}  </style>
```

ATTACH EVENTS

TEKNOTURF

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/
3.1.1/jquery.min.js"></script>
</head>
<body>
<div class="test">Hi Everyone!!!</div>
<script>
$( "div.test" ).on({
  click: function() {
    $( this ).toggleClass( "active" );
  }, mouseenter: function() {
    $( this ).addClass( "inside" );
  }, mouseleave: function() {
    $( this ).removeClass( "inside" );
  }
});
</script> </body></html>
```



DETACH

TEKNOTURF

off() : This method removes event handlers that were attached with .on()

Calling .off() with no arguments it removes all handlers attached to the elements

Syntax: .off(events [, selector] [, handler])

Events : One or more space-separated event types

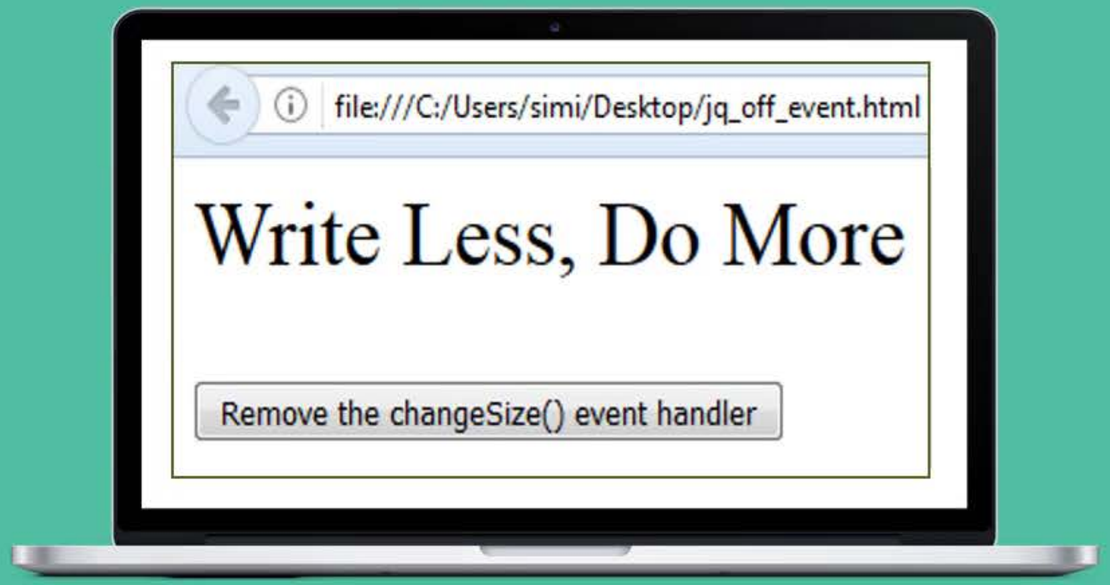
Handler: A handler function previously attached for the event(s), or the special value false.

Selector : A selector which should match the one originally passed to .on() when attaching event handlers.

DETACH EVENTS

```
<html><head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/
jquery.min.js">
</script>
<script>
function changeSize() {
   $(this).animate({fontSize: "+=10px"});
}
  $(document).ready(function(){
    $("p").on("click", changeSize);
       $("button").click(function(){
        $("p").off("click", changeSize);
    });   });
</script> </head>
<body>
<p>Write Less, Do More</p>
<button>Remove the changeSize() event ndler</button>
</body></html>
```

In this, when we click on the paragraph, its size increases and once you click the button, the changeSize() event handler is removed from the paragraph element.



JQUERY EVENTS

TEKNOTURF

- The activities that are recognized by your web application are known as events.
- The term "fires/fired" is often used with events.
- The click event is fired, the moment you click a button.

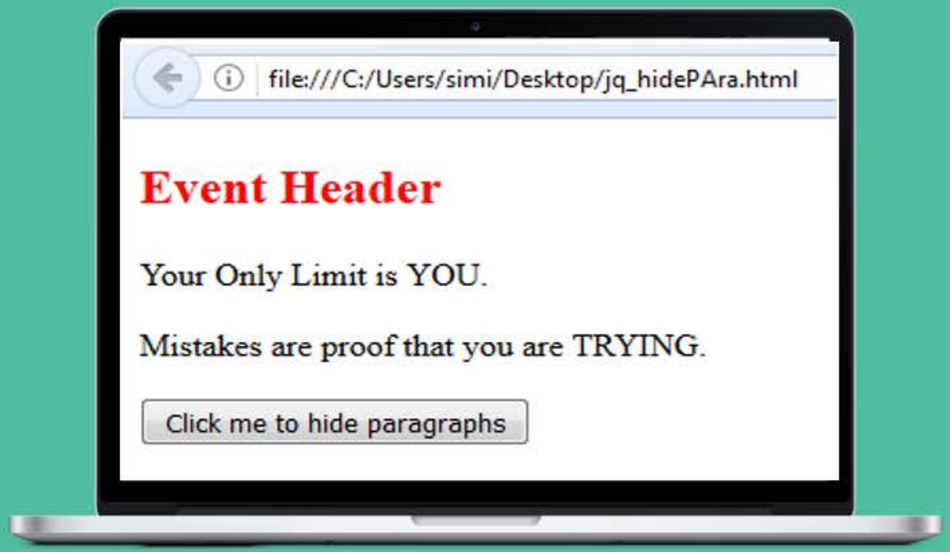
JQUERY EVENTS

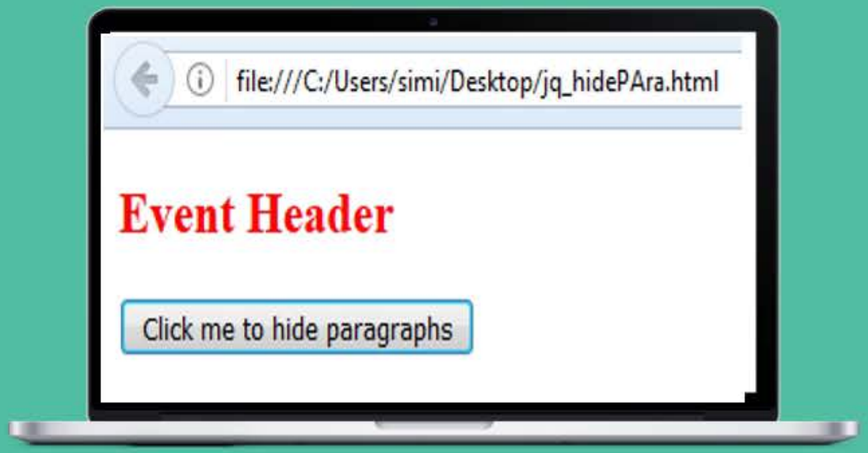
TEKNOTURF

Mouse Events	Keyboard Events	Form Events	Document/Window Events
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
Mouseleave		blur	unload
mouseover			

EVENTS:SAMPLE PROGRAM

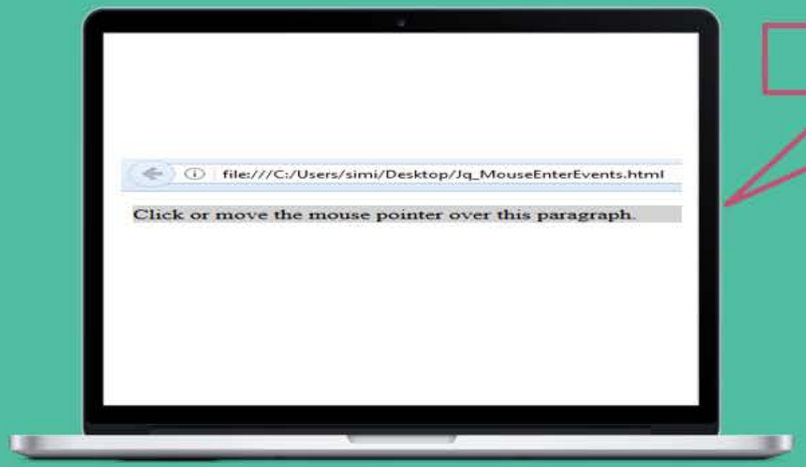
```
<!DOCTYPE html>
<html><head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js ">
</script>
<script>
        $(document).ready(function(){
          $("button").click(function(){
            $("p").hide();
          });
        });
</script></head>
<body>
        <h2><font color="red"> Event Header</font></h2>
        <p>Your Only Limit is YOU.</p>
        <p>Mistakes are proof that you are TRYING.</p>
        <button>Click me to hide paragraphs</button>
</body>
</html>
```



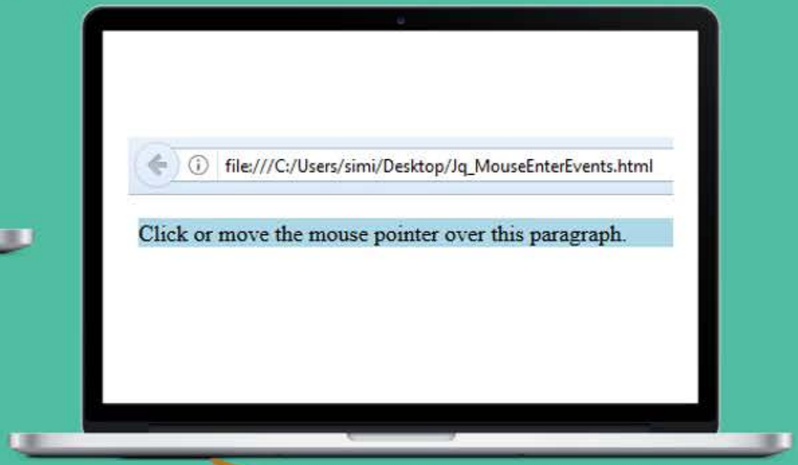


EVENTS:SAMPLE PROGRAM

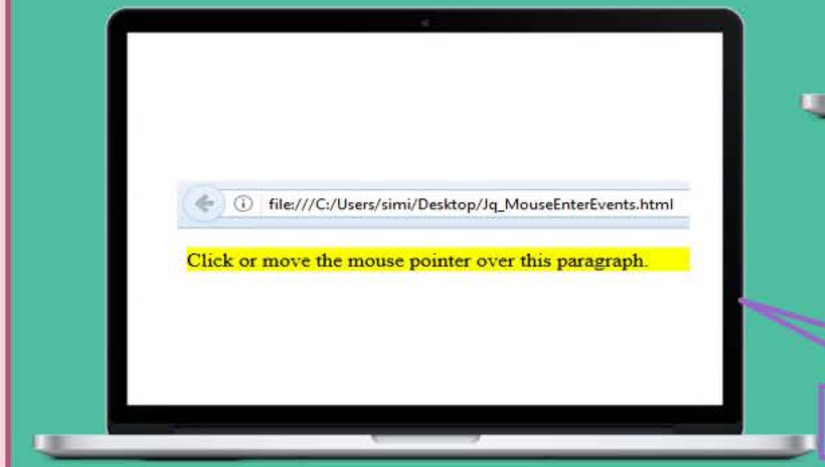
```
html><head>
<script
src="https://ajax.googleapis.com/ajax/libs
/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
   $("p").on({
      mouseenter: function(){
         $(this).css("background-color",
"lightgray");    },
      mouseleave: function(){
         $(this).css("background-color",
"lightblue");    },
      click: function(){
         $(this).css("background-color",
"yellow");  },
      });
});
</script></head> <body><p>Click or
move the mouse pointer over this
paragraph.</p></body></html>
```



Mouse enter event



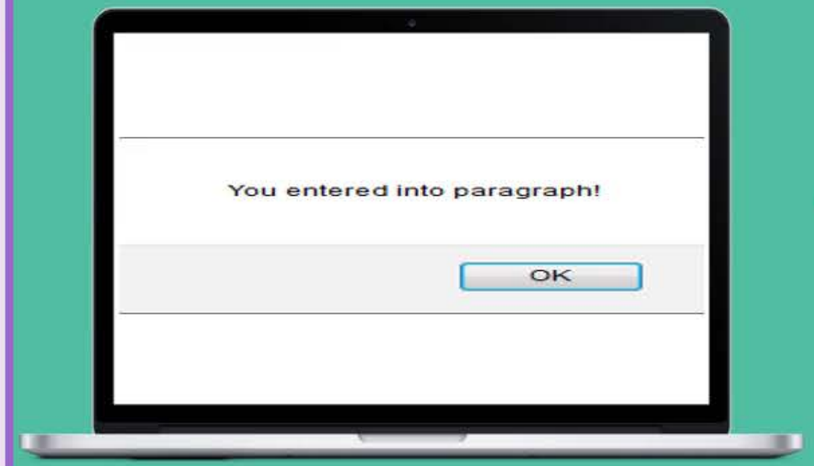
Mouse leave event



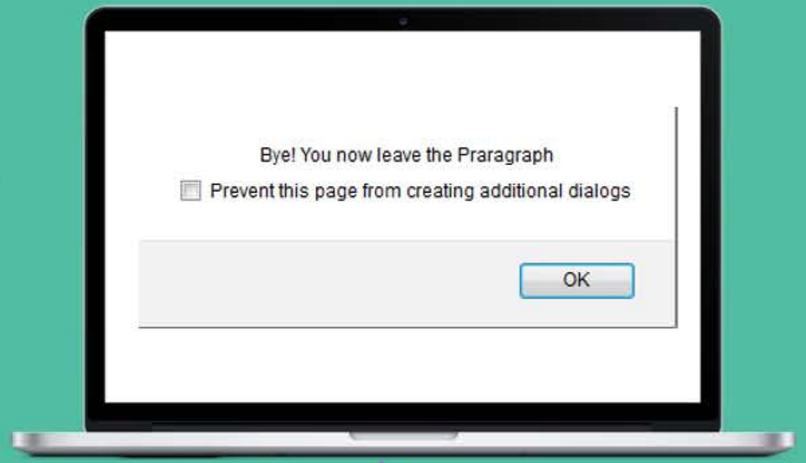
Mouse click event

EVENTS:SAMPLE PROGRAM

```
<!DOCTYPE html>
<html><head>
<script src=
"https://ajax.googleapis.com/ajax/libs/jquery/
3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
   $("#p1").hover(function(){
      alert("You entered into paragraph!");
   },
   function(){
      alert("Bye! You now leave the
Paragraph");
   });
});</script></head>
<body>
<p id="p1">Work Until Your IDOLS Become
Your RIVALS</p>
</body></html>
```



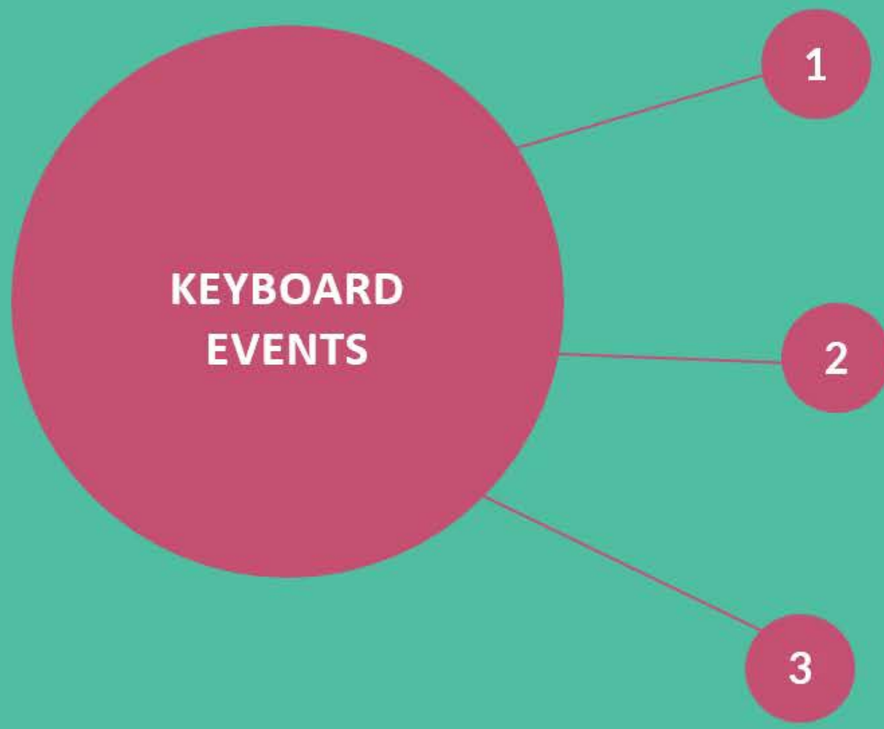
hover() method takes two functions and is a combination of the mouseenter() and mouseleave() methods.



When the mouse leaves the paragraph

JQUERY KEYBOARD EVENTS

TEKNOTURF



KEYDOWN

The key is on its way down

KEYPRESS

The key is pressed down

The keypress() method triggers the keypress event, or attaches a function to run when a keypress event occurs

The keypress event is similar to the keydown event

The event occurs when a button is pressed down

KEYUP

The key is released

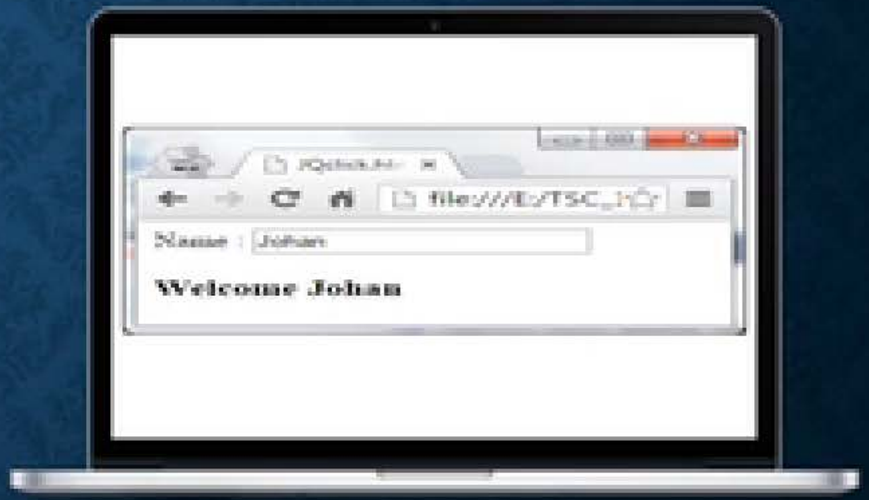
JQUERY KEYBOARD EVENTS

TEKNOTURF

```
html><head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js">
</script>
<script>
            i = 0;
            $(document).ready(function(){
               $("input").keypress(function(){
                  $("span").text(i += 1);
               });
            });
</script>
</head><body>
Enter your name: <input type="text">
<p>Keypresses: <span>0</span></p>
</body></html>
```

JQUERY FORM EVENTS

```
<head>
<script type="text/javascript"
src="
https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.
js "></script>
<script type="text/javascript" >
        $(document).ready(function() {
            $('input').change(function(){
                    var
name=document.myForm.name.value;
                    $('div').append("<h3>Welcome
"+name+"</h3>");
                    });
        });
</script> </head>
<body>
        <form name="myForm">
            Name : <input type="text" name="name">
            <div></div>
        </form>
</body>
```



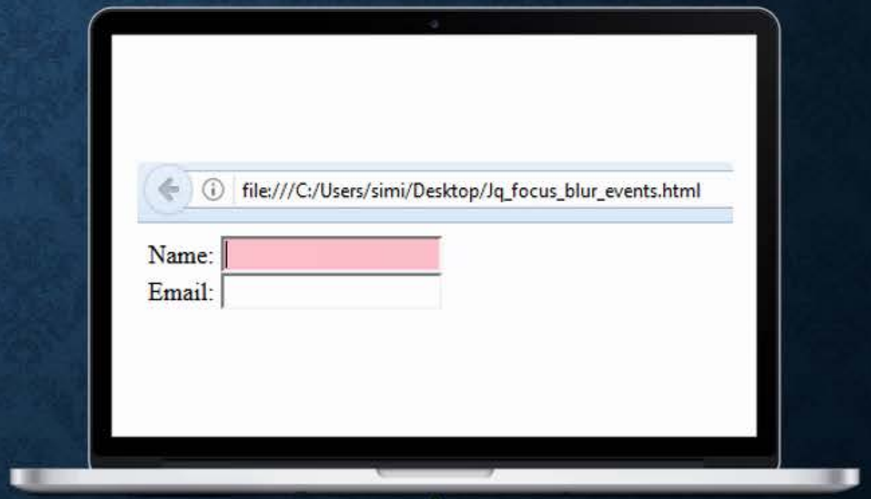
When you type /change the name in the input box, it displays the word Welcome' along with that name.

JQUERY FORM EVENTS

TEKNOTURF

```
<!DOCTYPE html>
<html><head>
<script src=
"https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js">
</script>
<script>
        $(document).ready(function(){
          $("input").focus(function(){
            $(this).css("background-color", "pink");
          });
          $("input").blur(function(){
            $(this).css("background-color", "#ffffff");
          });
        });
</script>
</head>
<body>
Name: <input type="text" name="fullname"><br>
Email: <input type="text" name="email“>
</body>
</html>
```

- Focus() is executed when the form field gets focus.
- blur() is executed when the form field loses focus.



When the mouse pointer focus on the input field, it turns to pink, when exits, turns to default color.

JQUERY SCROLL EVENTS

TEKNOTURF

```
<html><head>
<script src=" https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js "></script>
<script>
        $(document).ready(function(){
        $(txtscroll).scroll(function(){
        alert ("Scroll event occured!");
        });
                                x = 0;
$("div").scroll(function(){
  $("span").text( x+= 1);
    }); });
</script></head>
<body><p><font color="Green"> TEXT AREA </font></p>
<textarea cols="20" rows="10" id="txtscroll">
    jQuery is a fast, small, and feature-rich JavaScript library.
    <br><br>
```

5 seconds

The scroll event occurs when the user scrolls in the specified element
$(selector).scroll(function)

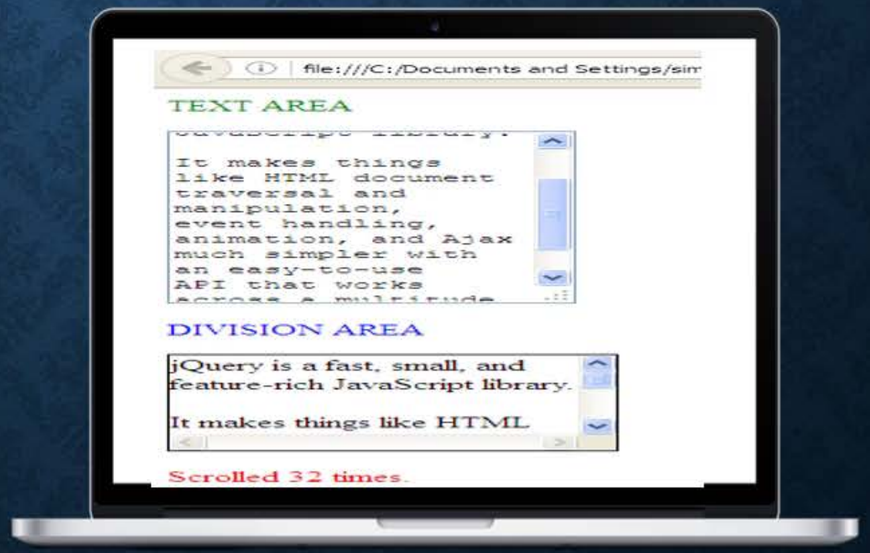
02:06

JQUERY SCROLL EVENTS

TEKNOTURF

- It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use.
- API that works across a multitude of browsers.

```
</textarea>
<P><font color="blue"> DIVISION AREA </font></p>
<div style="border:1px solid
black;width:200px;height:100px;
    overflow:scroll;">
   JQuery is a fast, small, and feature-rich JavaScript library.
   <br><br>
   It makes things like HTML document traversal and
   manipulation, event handling, animation, and Ajax
   much simpler with an easy-to-use
   API that works across a multitude of browsers.
</div>
<p><font color="red">Scrolled <span>0</span>
times.</font></p>
</body></html>
```



5 seconds

02:36

JQUERY RESIZE EVENTS

TEKNOTURF

- The resize event fires when the document view (window) has been resized.
- This is useful for many situations like adjusting page elements depending on window size in order to avoid scroll bars ,etc.
- Resize events are triggered using the resize() method, which also attaches a function that will be executed when a resize event happens.

Syntax:

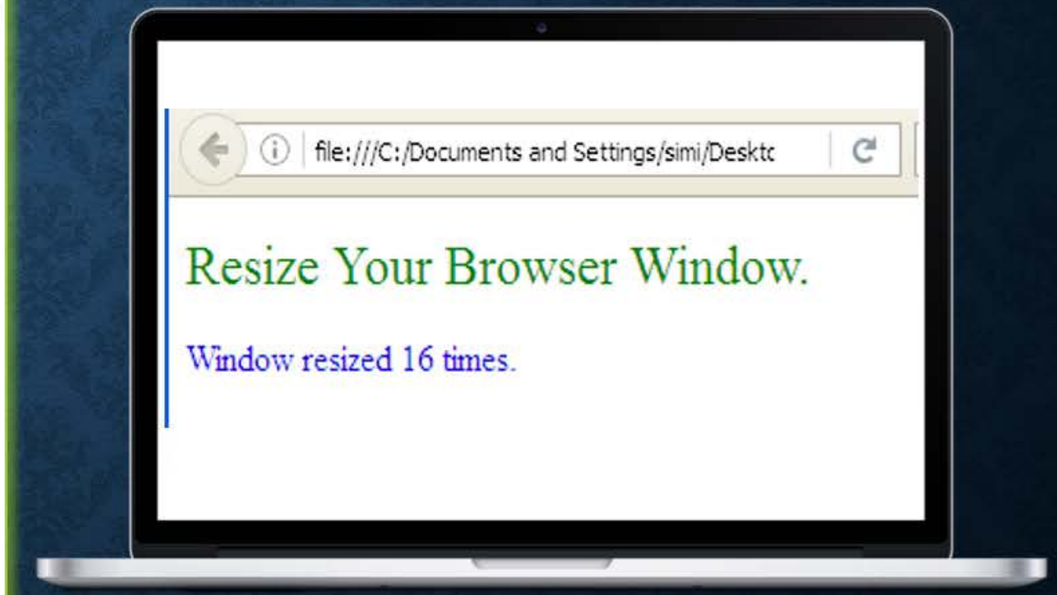
$(*selector*).resize()

$(*selector*).resize(handler function)

JQUERY RESIZE EVENTS

TEKNOTURF

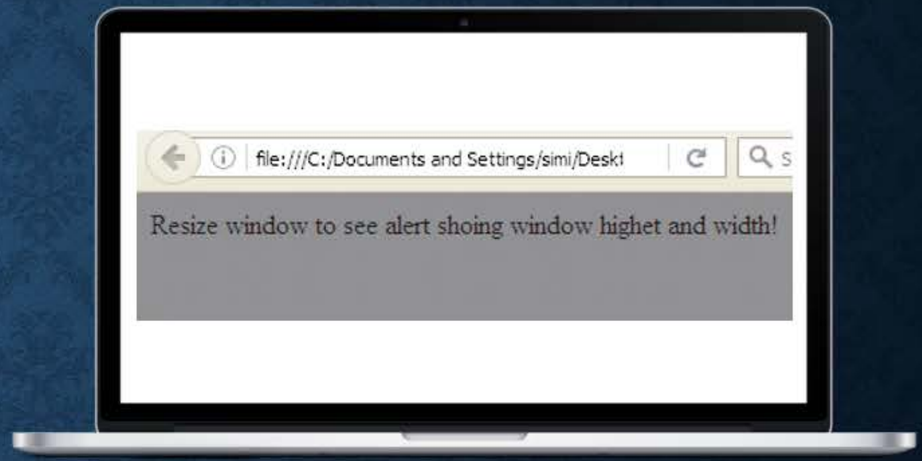
```
<!DOCTYPE html>
<html><head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.m
in.js">
</script>
<script>
            x = 0;
   $(document).ready(function(){
        $(window).resize(function(){
                    $("span").text(x += 1);
        });
            });</script> </head>
<body><p><font color="green" size="5">Resize Your Browser
Window.</font></p>
<p><font color="blue"> Window resized <span>0</span>
times.</font></p></body> </html>
```

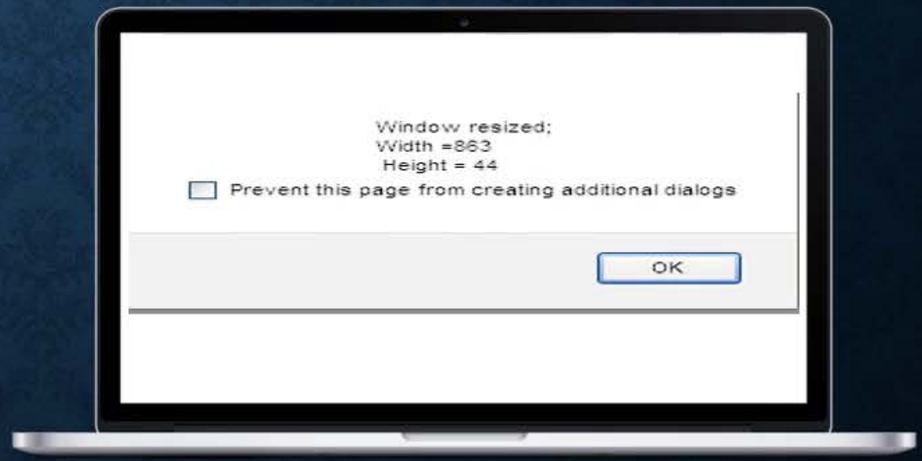


JQUERY RESIZE

TEKNOTURF

```
<html><head>
 <script src="
https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js ">
</script>
 <script>
 $(document).ready(function(){
   $(window).resize(function(){
                alert ("Window resized;" + "\n" + "Width =" +
$(window).width() + "\n"    + " Height = " + $(window).height());
  });
 });
 </script> </head>
 <body>
 <p>Resize window to see alert showing window height and width!</p>
 </body>
 </html>
```





MOUSE EVENTS

TEKNOTURF

```
<script>
    $(document).ready(function()
    {
            $("p").on({
                    mouseenter: function(){
                    $(this).css("background-color", "lightgray"); },
                    mouseleave: function(){
                    $(this).css("background-color", "lightblue");},
                    click: function(){
                    $(this).css("background-color", "yellow");  },
                    });
    });
</script>
```

The **on()** method attaches one or more event handlers for the selected elements

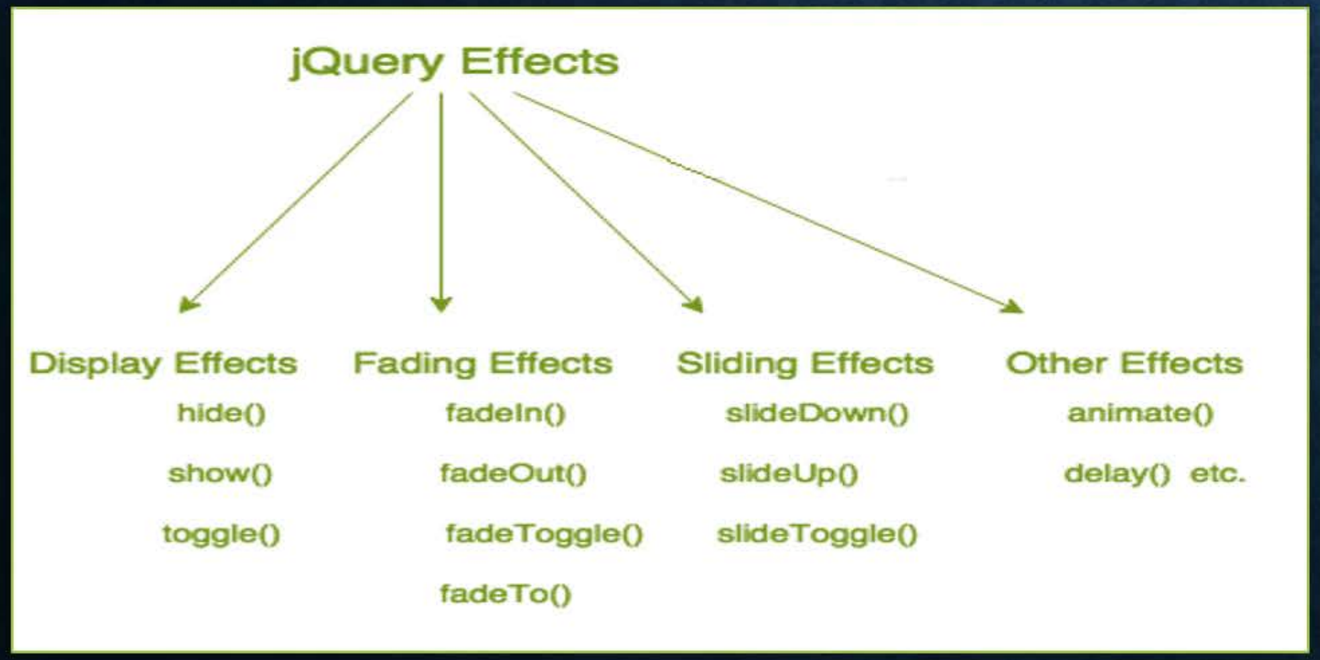
```
<script>
    $(document).ready(function()
    {
            $("#p1").hover(function(){
                    alert("You entered into paragraph!");
                },
        function(){
                alert("Bye! You now leave the Praragraph");
            });
    });
</script>
```

hover() method takes two functions and is a combination of the **mouseenter()** and **mouseleave()** methods.

JQUERY EFFECTS

TEKNOTURF

- jQuery enables us to add effects on a web page.
- jQuery effects can be categorized into display, fading, sliding, hiding/showing and other animation effects.



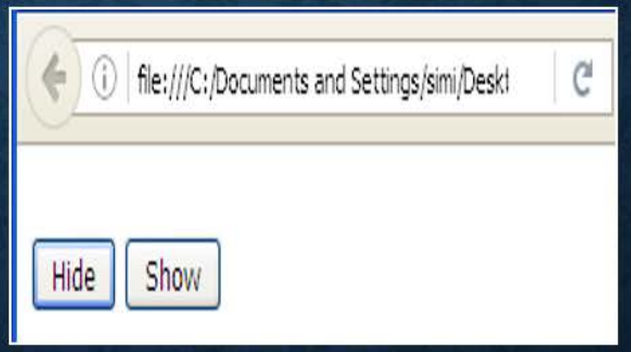
DISPLAY EFFECTS

TEKNOTURF

```
<script type="text/javascript" >
    $(document).ready(function()
    {
        $('#hide').click(function(){
            $('img').hide();
        });
        $('#show').click(function(){
            $('img').show();
        });
    });
</script>
```

Hide and show can take an optional time in milliseconds to have transition

After clicking the hide button, image is hidden




```
<script type = "text/javascript" language = "javascript">
    $(document).ready(function()
    {
                $(".clickme").click(function(event){
                    $(".target").toggle('slow', function(){
                });  }); });
</script>
```

toggles() method toggles between the hide() and show() methods.

FADING EFFECTS

TEKNOTURF

- **fadeIn() - Syntax**
  - $(selector).fadeIn([speed, callback]);

Specifies the duration of the effect

It is a function to be executed after fading completes

- **fadeOut() - Syntax**
  - $(selector).fadeOut([speed, callback]);
- **fadeToggle() - Syntax**
  - $(selector).fadeToggle([speed, callback]);
- **fadeTo() - Syntax**
  - $(selector).fadeTo([speed, opacity, callback]);

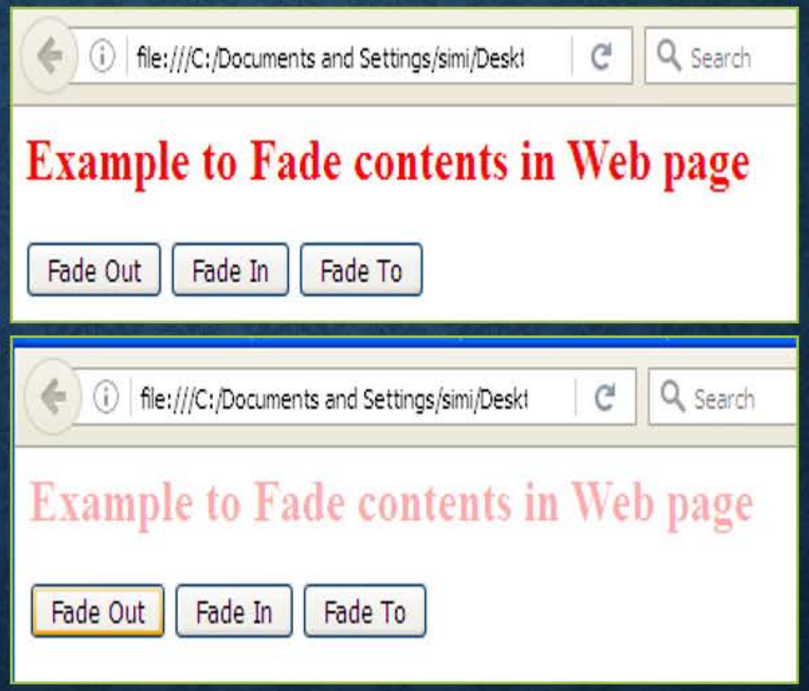
Opacity value between 0 and 1

FADING EFFECTS - EXAMPLE

TEKNOTURF

```
<script type="text/javascript" >
    $(document).ready(function() {
        $('#fadeout').click(function(){
            $('h3').fadeOut(1000);
        });
        $('#fadein').click(function(){
            $('h3').fadeIn(1000);
        });
        $('#fadeto').click(function(){
            $('h3').fadeTo(1000,.5);
        });
    });
</script>
```

Output



SLIDING EFFECTS

TEKNOTURF

- **slideDown() - Syntax**
  - $(selector).slideDown([speed, callback]);
- **slideUp() - Syntax**
  - $(selector).slideUp([speed, callback]);
- **slideToggle() - Syntax**
  - $(selector).slideToggle([speed, callback]);

5 seconds

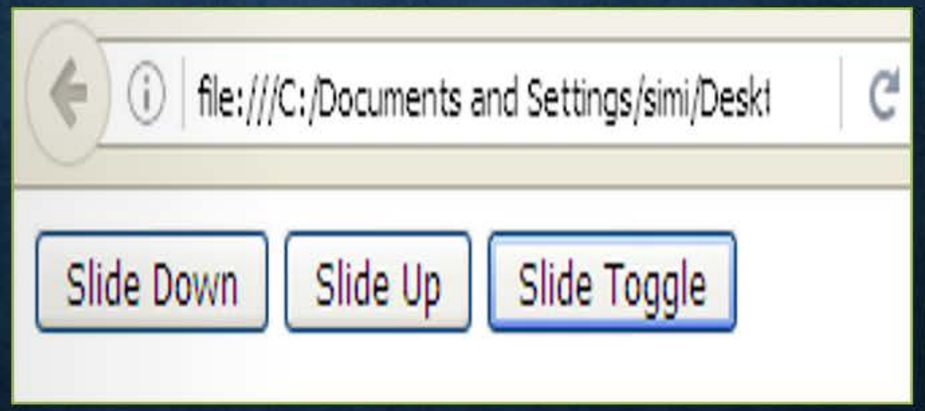
SLIDING EFFECTS - EXAMPLE

TEKNOTURF

```
<script type="text/javascript" >
    $(document).ready(function()
    {
        $('#slidedown').click(function(){
        $('h3').slideDown("slow");
    });
    $('#slideup').click(function(){
        $('h3').slideUp("fast");
    });
    $('#slidetoggle').click(function(){
        $('h3').slideToggle(1000);
    });});
</script>
```

5 seconds

Output



08:51

ANIMATE() - EXAMPLE

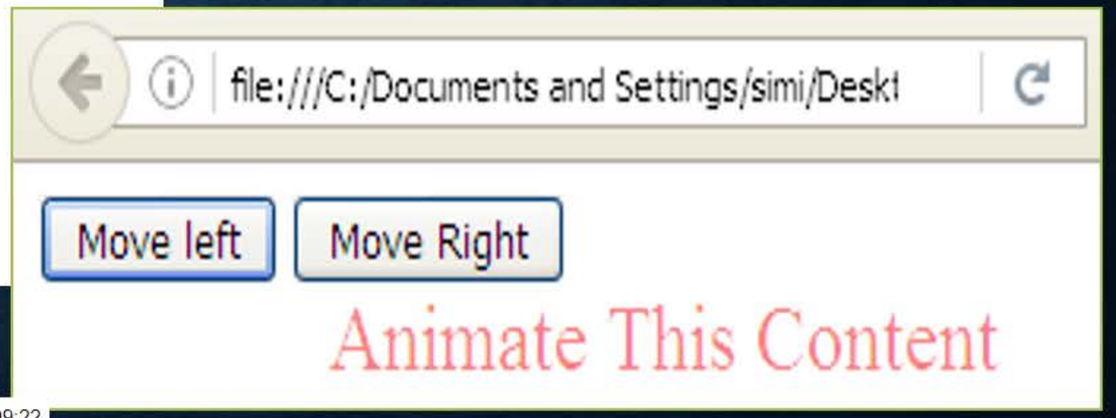
TEKNOTURF

```
<script>
    $(document).ready(function()
    {
                $('#left').click(function()
        {
        $('div').animate( {left: "-=50px" , height:'toggle', opacity:0.5}, "slow");
            });
            $('#right').click(function(){
        $('div').animate( {left: "+=50px", width : 'toggle',opacity:1}, "slow");
    }); });
</script>
<head>
<body>
    <button id="left">Move left</button>
    <button id="right">Move Right</button>
    <div style="left: 100px ; position: absolute ">
          <font size=5 color="red">Animate This Content</font></div>
</body>
</html>
```

Syntax

```
$(selector).animate({params}, speed, callback);
```

Output



09:22

JQUERY WITH JSON

TEKNOTURF

JSON

[JSON stands for “JavaScript Object Notation”]

01 It is a syntax for storing and exchanging data

02 It is lightweight data-interchange format

03 "self-describing" and easy to understand

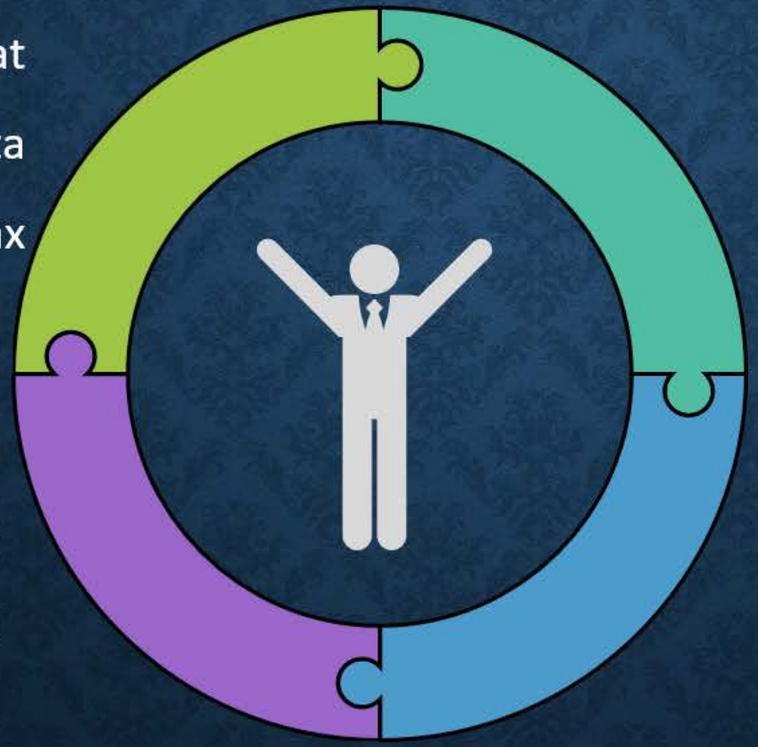
04 It is Language independent

WHY USE JSON

TEKNOTURF

STANDARD STRUCTURE

JavaScript Object Notation (JSON)
is a standard text-based format
for representing structured data
based on JavaScript object syntax



LIGHT WEIGHT

JSON is light weighted; it
becomes easier to get and load
the requested data quickly

SCALABLE

JSON is language independent

JSON SYNTAX RULES

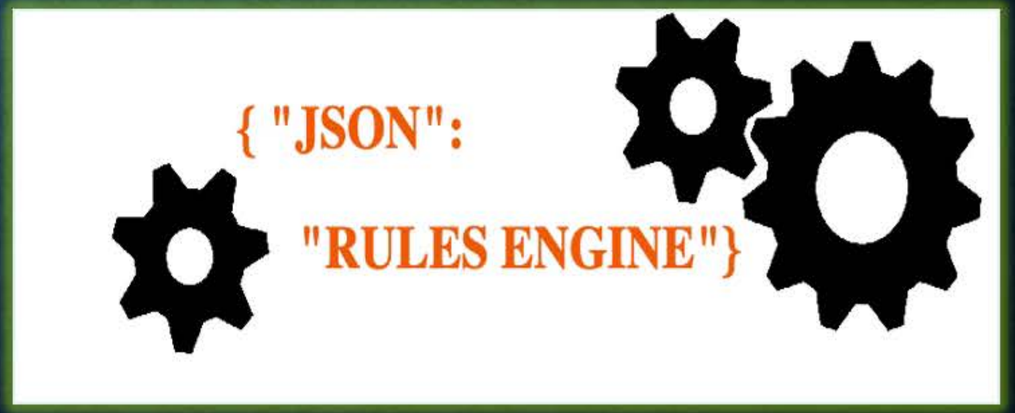
TEKNOTURF

JSON syntax is derived from JavaScript object notation syntax:

- JSON Data is in name/value pairs
- Data is separated by commas
- Curly braces hold objects
- Square brackets hold arrays

A name/value pair consists of a field name (in double quotes), followed by a colon, followed by a value:

Example : "name":"John"



JSON DATA TYPES

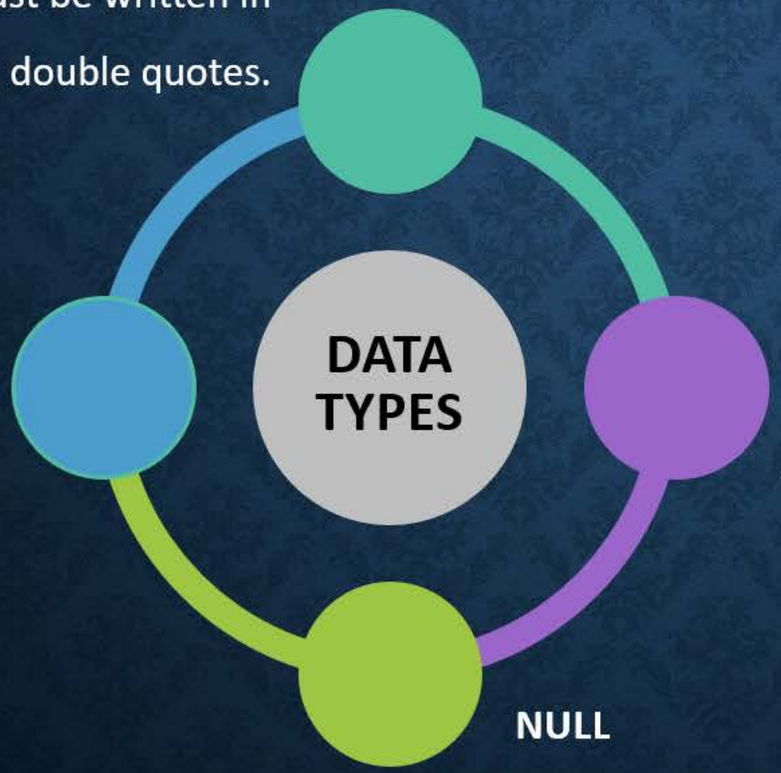
TEKNOTURF

STRING

Strings in JSON must be written in double quotes.

NUMBER

Numbers in JSON must be an integer or a floating point.



BOOLEAN

Values in JSON can be true/false.

NULL

Values in JSON can be null.

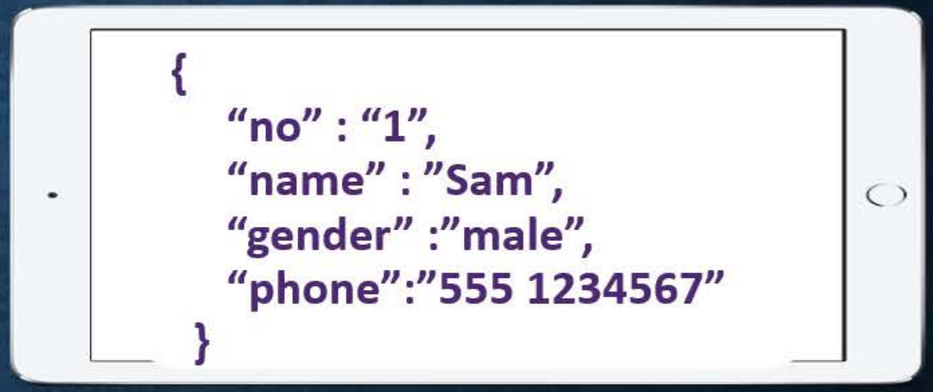
- **Numbers** `{ "id": 5, "balance":2942.59 }`
- **Strings** `{ "name": "Joe", "city": "Morrisville" }`
- **Booleans** `{ "premium": true, "balance":2942.59 }`
- **Null** `{ "last_address": null }`
- **Arrays** `{ "hobbies": ["tennis", "skiing", "lego"]}`
- **Objects** `{ address: {"street": "1, Main street", "city": Morrisville, "state":"CA", "zip":"94824"}}`

JSON OBJECTS

TEKNOTURF

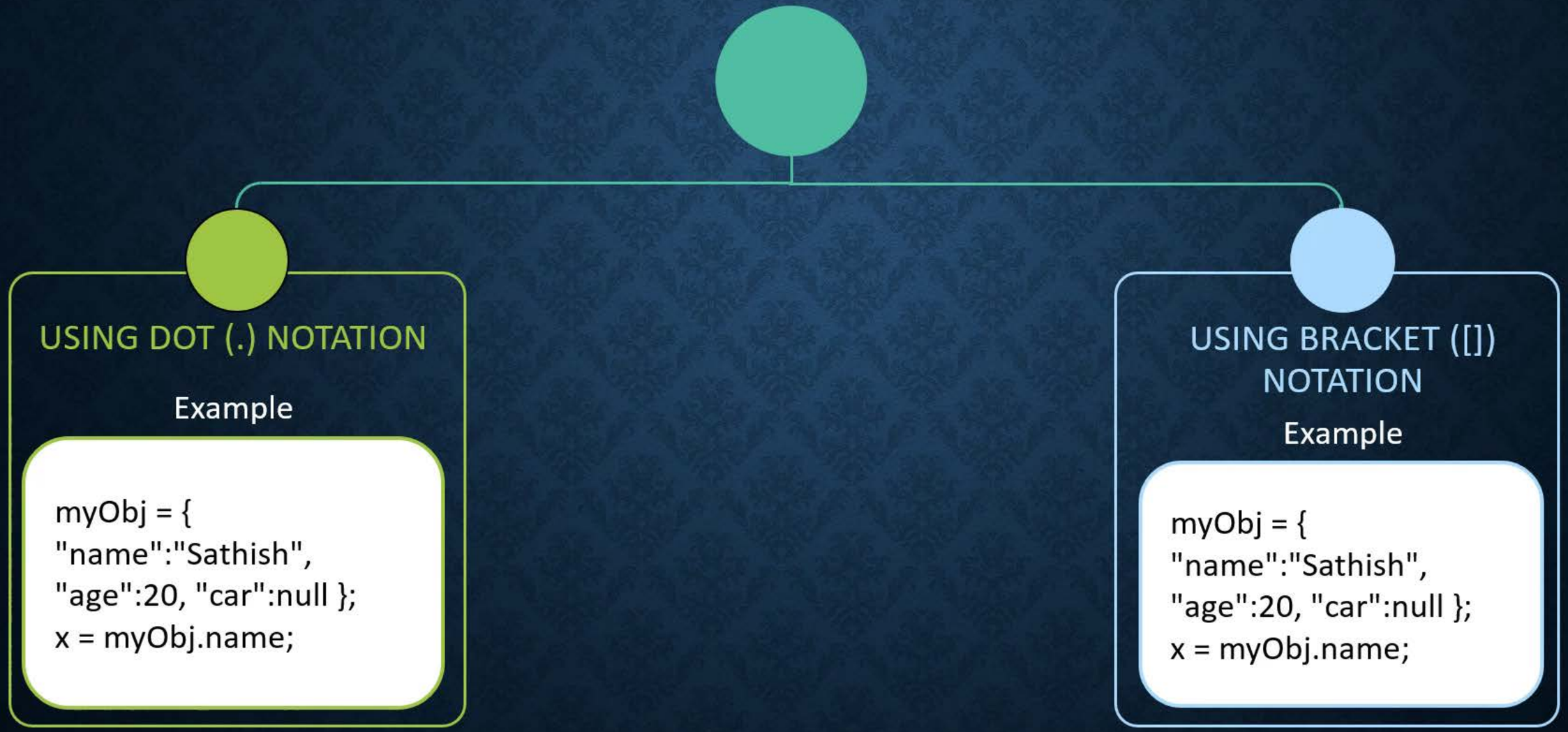
A JavaScript object is an unordered set of name/value pairs

- The pairs are enclosed within braces { }
- Keys must be strings, and values must be a valid JSON data type (string, number, object, array, Boolean or null).
- There is a colon between the name and the value
- Pairs are separated by commas



ACCESSING JSON OBJECTS

TEKNOTURF



NESTED JSON OBJECTS

TEKNOTURF

NESTED JSON OBJECTS :
VALUES IN A JSON OBJECT CAN BE ANOTHER JSON OBJECT.

```
myObj = {
   "name":"Sathish",
   "age":20,
   "cars": {
      "car1":"Ford",
      "car2":"BMW",
      "car3":"Fiat"
   }
}
```

Example

```
x = myObj.cars.car2;
//or:
x = myObj.cars["car2"];
```

MODIFY/DELETE OBJECTS

TEKNOTURF

Modify Object Values : You can use the dot notation to modify any value in a JSON object.

Example:

- myObj.cars.car2 = "Mercedes";
- myObj.cars["car2"] = "Mercedes";

Delete Object Properties: Use the ‘delete’ keyword to delete properties from a JSON object.

Example:

```
delete myObj.cars.car2;
```

JSON ARRAYS

TEKNOTURF

An Array is an ordered collection of values:

- The values are enclosed within brackets [ ]
- Values are separated by commas

Example:

```
[ {
    "name": "html",
    "years": 5
  },
  {
    "name": "jquery",
    "years": 6
  } ]
```

USING JSON IN JQUERY

TEKNOTURF

jQuery.getJSON(url, [data], [callback]) :

- jQuery helps to load JSON-encoded data

from a remote source using $.getJSON()

- This jquery method loads JSON-encoded data

from the server using a GET HTTP request

- The method returns XMLHttpRequest object

Syntax: $.getJSON(url, [data], [callback]

The three parameters correspond to:

01 URL

This parameter is a string containing the URL to which the request is sent.

02 DATA

The optional data parameter is either an object or a string that is sent to the server with the request.

03 CALL BACK

This optional parameter represents a function to be executed whenever the data is loaded successfully.

USING JSON IN JQUERY

TEKNOTURF

Example: for JSON file: bookData.json

```
{
    "books":
    [
     {
     "bookId": "B101",
     "bookName": "C++ Programming",
                "bookPrice":7890.80
      },
      {
     "bookId": "B102",
     "bookName": "Java Programming",
               "bookPrice":800
      },
      {
     "bookId": "B103",
     "bookName": "Python Programming",
               "bookPrice":600
      }
    ]   }
```

$.each(array, callback):
A generic iterator function, which can be used to iterate over both objects and arrays.

USING TAG NAME SELECTOR

TEKNOTURF

```
<html> <head>
<script type = "text/javascript"
    src = "https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js">
</script>
<script type="text/javascript">
  $(function(){
    $.getJSON('bookData.json',function(data){
      $.each(data.books,function(key,val){
       $('div').append('<p>'+val.bookId+" "+val.bookName+" "+val.bookPrice+
'</p>' );
          });
      });    });
</script> </head>
<body>
  <div></div>
</body>
</html>
```