basic stuff :

- world wide web : series of web pages following http protocol
- http : protocol for transmission of files between clients and server
- scheme specifiers : specifiy protocol of communicates fr request/reply.ex: `http:/`
- dns : convert domain names to ip adress
- IPV4 : 4 no.s(each no in binary has 8 bits) separated by 3 decimal pts (ex:70.74.251.42)
- IPV6 : 8 hex no.s separated by colons

DNS LOOKUP - domain to ip

- user types URL(example.com) into web browser , query travels to internet and received by DNS recursive resolver
- resolver queries DNS root nameserver which responds to resolver with address of TLD(top level domain) DNS server (like .com/.net)
- resolver makes request to .com TLD
- TLD server responds with IP address of domains name server example.com
- recursive resolver sends query to domain's name server which returns IP address of example.com to resolver
- DNS resolver responds to web browser with ip address of domain requested initially.the browser then makes HTTP request to IP address and loads the contents of the web page

HTTP PROTOCOL

- protocol for transmitting hypertext messages between client(usually browsers) and server
- uses TCP to communicate to servers
- stateless (server maintains no info abt previous requests)
- HTTP/2 : encapsulating HTTP messages into frames and reduced complexity for newcomers.
- default behaviour of HTTP/1.0 is to open a separate TCP connection for each HTTP request/response pair
- https : Hypertext Transfer Protocol over Secure Socket Layer : websites with https have SSL certified and use TLS/SSL to encrypt http requests
- non persistent http : at most one object sent over TCP connection connection is then closed (default mode for http/1.0)
- persistent http : the server leaves the TCP connection open after sending responses and server closes the connection only when it is not used for a certain configurable amount of time (default for http/1.1)

- http request format :

```
request line : <method> <uri> <version>
request headers
CRLF(\r\n)
message body
```

- http response format :

```
response line : <version> <status code> <status msg>
response headers
Content-Type : MIME
CRLF
message body
```

- http methods :
  - GET: Retrieve static or dynamic content
  - POST: Send content to server through request body
  - OPTIONS: shows HTTP methods available for a URL
  - HEAD: Fetches only header field without any response body
  - PUT: Write a file to the server
  - DELETE: Delete a file on the server
- status codes :
  - 100-199 : informational response (everything ok so far)
  - 200-299 : successful response (ex: 200 ok)
  - 300-399 :redirection message (301 moved permanently)
  - 400-499 : client error response (400 bad request - invalid syntax,403 forbidden,404 not found)
  - 500-599 : server error

note :

- The **HTTP** `PUT` **request method** creates a new resource or replaces a representation of the target resource with the request payload.(idempotent so callin it several times has no side effect)

some extra stuff which might be helpful but not mentioned :

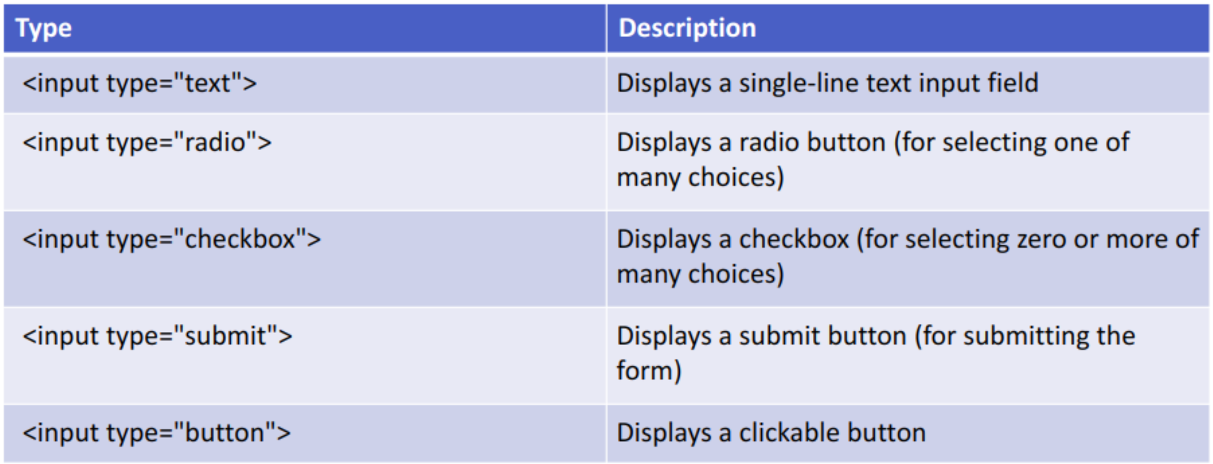
- MIME - multipurpose internet mail extension - specifies type of file like (image/png , text/html)

- http is stateless but states can be maintained using cookies (ts a small piece of data that a server sends to a user's web browser.the browser stores and sends back to server and this way it can tell if two requests comes from same browser) like log in
- The `Set-Cookie` HTTP response header sends cookies from the server to the user agent. A simple cookie is set like this

```
Set-Cookie: <cookie-name>=<cookie-value>
```

- CORS - cross origin resource sharing (allows a server to indicate any origins (domain, scheme, or port) other than its own from which a browser should permit loading resources). (one document can have an image from domaina.com and another image from imageb.com)

HTML

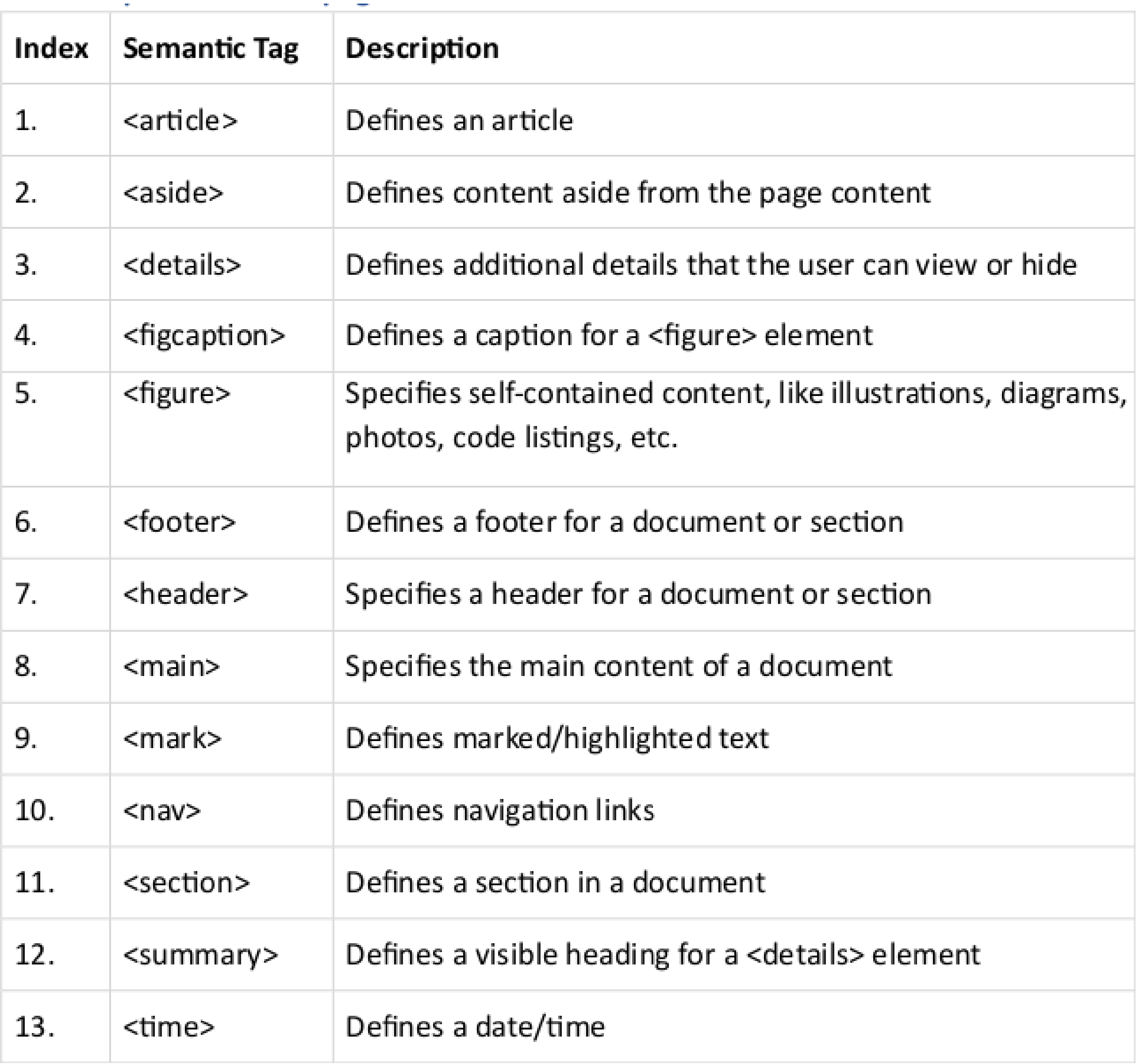


Type	Description
<input type="text">	Displays a single-line text input field
<input type="radio">	Displays a radio button (for selecting one of many choices)
<input type="checkbox">	Displays a checkbox (for selecting zero or more of many choices)
<input type="submit">	Displays a submit button (for submitting the form)
<input type="button">	Displays a clickable button

tag	attribute	description
<img>	**alt**	alternate text to describe image
	border	alternate text to describe image
	height , **width**	height/width
	src	location of image
<TEXTAREA><TEXTAREA/>	**COLS = "" , ROWS= ""**	multi line
<FORM> <FORM/>	**METHOD = ""**	HTTP METHOD
	ACTION = ""	location of script which process form data
<dl><dt>key</dt><dd>value</dd></dl>		definition list (like ul but with key value pairs)

tag	attribute	description
<select > <option> </option> </select>		menu

- note use <div align="top"><div/> to align image. to align text wrapped use align attribute inside <img>
- value attribute in input type is used to show initial value
- set checked = "checked" to check radiobutton / checkbox
- <em> means italic, strong means bold
- html5 semantic tags

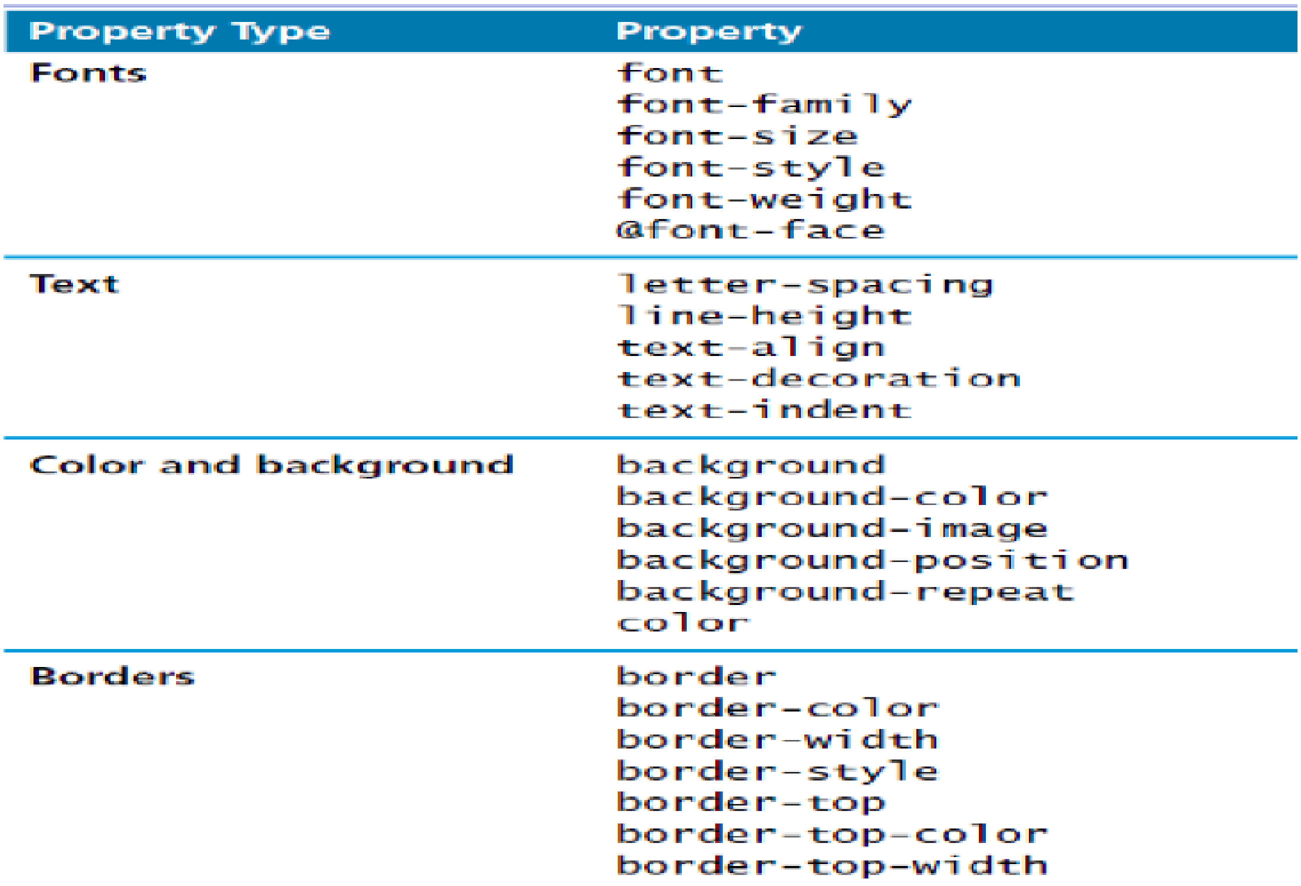


Index	Semantic Tag	Description
1.	<article>	Defines an article
2.	<aside>	Defines content aside from the page content
3.	<details>	Defines additional details that the user can view or hide
4.	<figcaption>	Defines a caption for a <figure> element
5.	<figure>	Specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.
6.	<footer>	Defines a footer for a document or section
7.	<header>	Specifies a header for a document or section
8.	<main>	Specifies the main content of a document
9.	<mark>	Defines marked/highlighted text
10.	<nav>	Defines navigation links
11.	<section>	Defines a section in a document
12.	<summary>	Defines a visible heading for a <details> element
13.	<time>	Defines a date/time

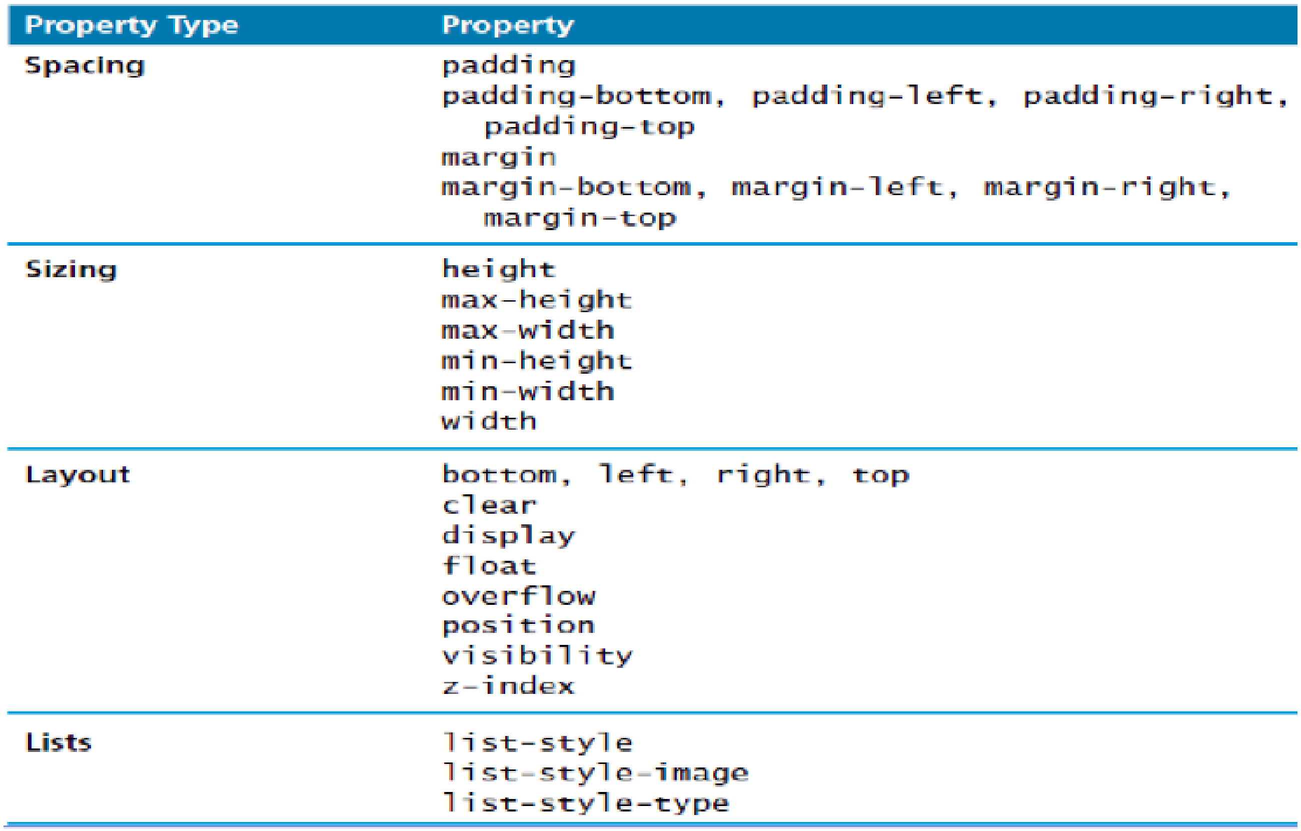
- cascading in css : inline > internal > external > default
-

CSS

- syntax : selector { property : value } (without any prefix select refers to tag name, # id name , '.' prefix indicates class property)
- '>' selects direct descendant while tag1 tag2 selects all tag2 descendants of tag1

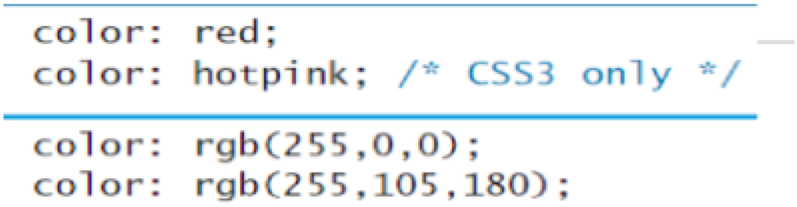


Property Type	Property
Fonts	font font-family font-size font-style font-weight @font-face
Text	letter-spacing line-height text-align text-decoration text-indent
Color and background	background background-color background-image background-position background-repeat color
Borders	border border-color border-width border-style border-top border-top-color border-top-width



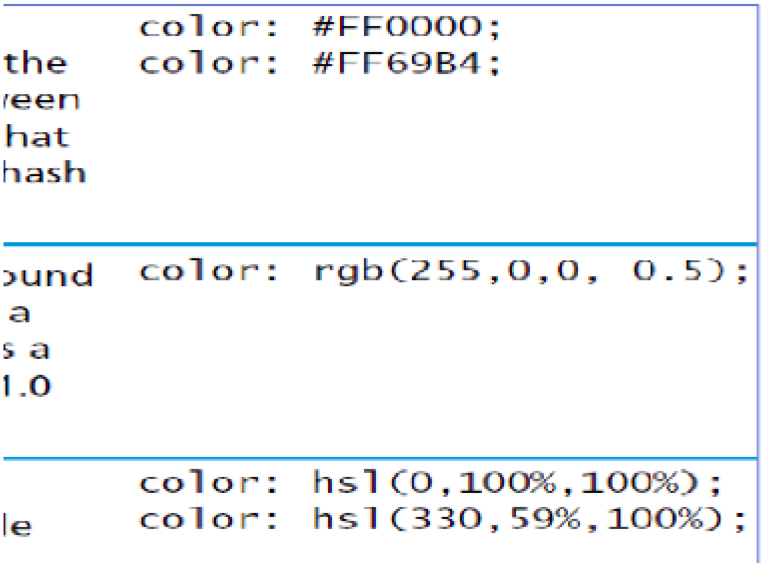
Property Type	Property
Spacing	padding padding-bottom, padding-left, padding-right, padding-top margin margin-bottom, margin-left, margin-right, margin-top
Sizing	height max-height max-width min-height min-width width
Layout	bottom, left, right, top clear display float overflow position visibility z-index
Lists	list-style list-style-image list-style-type

- color values :



```
color: red;
color: hotpink; /* CSS3 only */

color: rgb(255,0,0);
color: rgb(255,105,180);
```



```
        color: #FF0000;
the     color: #FF69B4;
een
hat
hash

ound    color: rgb(255,0,0, 0.5);
 a
s a
1.0

        color: hsl(0,100%,100%);
le      color: hsl(330,59%,100%);
```

- pseudo class selectors
  - other pseudo class selectors

    - :nth-child() - selects elements based on algebraic expression (ex \nth-child(2n) for even positions)

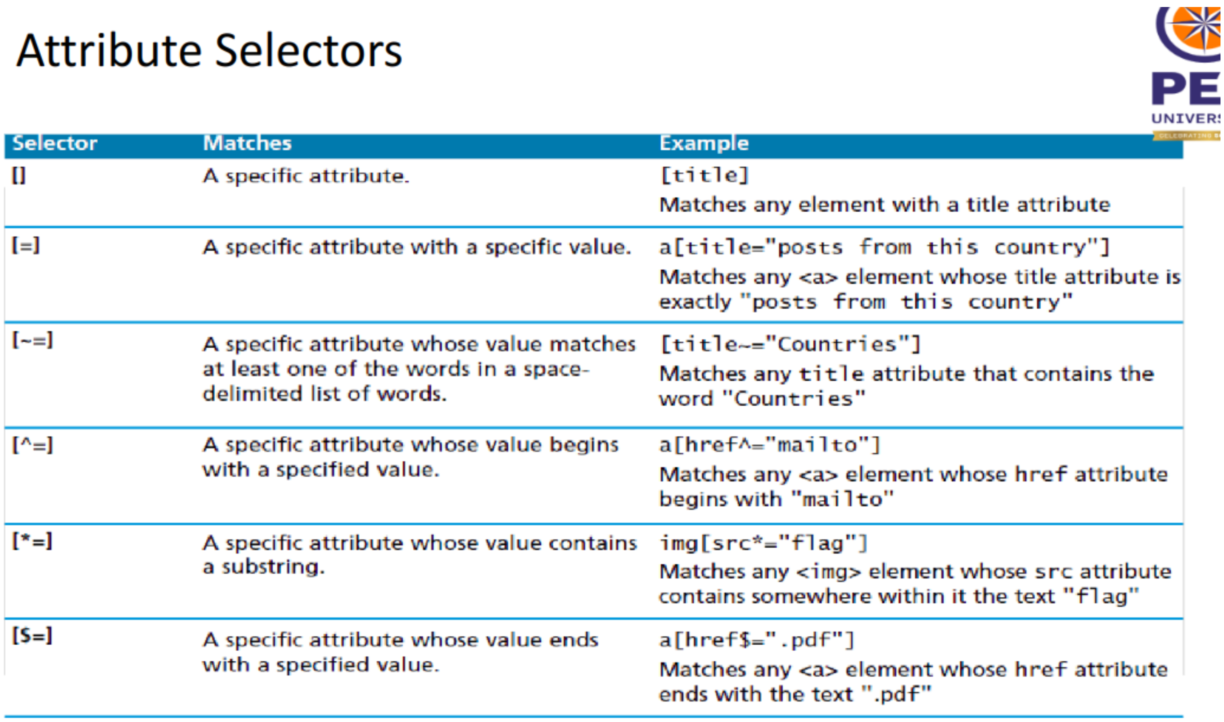
    - first-of-type : selects first element of its type

    - last-of-type : selects last element of its type

    - nth-of-type like nth-child but all elements at same level

    - '>' used for direct descandant (meant to include along css selector)

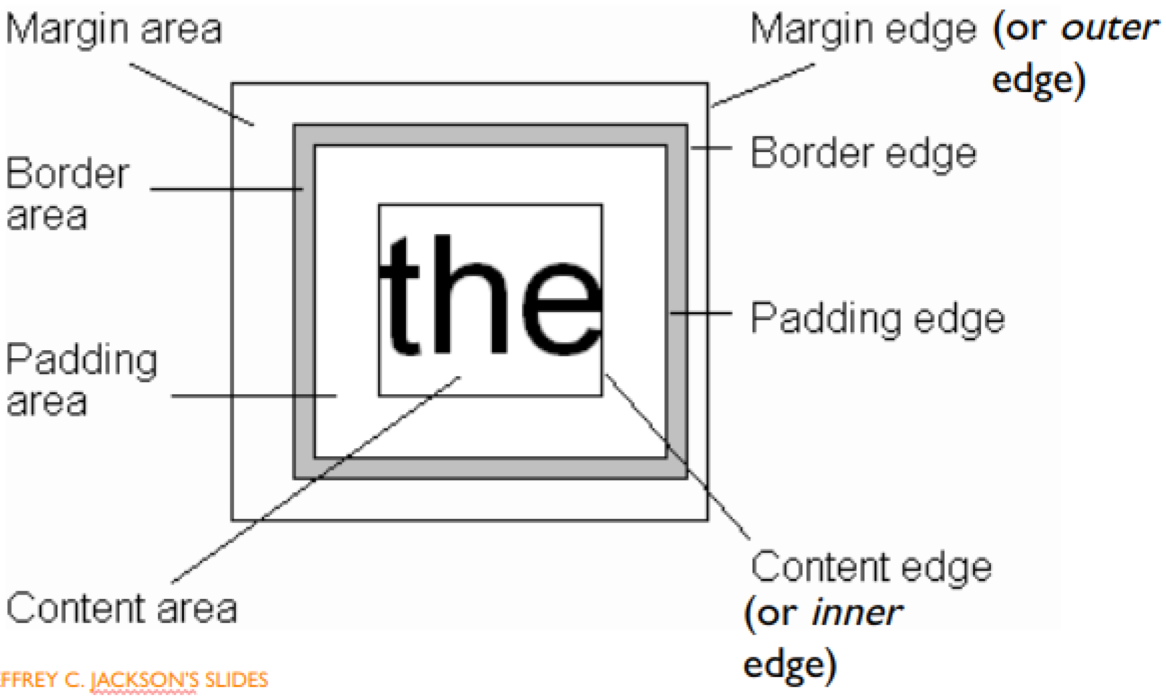
`a:link`	pseudo-class	Selects links that have not been visited
`a:visited`	pseudo-class	Selects links that have been visited
`:focus`	pseudo-class	Selects elements (such as text boxes or list boxes) that have the input focus.
`:hover`	pseudo-class	Selects elements that the mouse pointer is currently above.
`:active`	pseudo-class	Selects an element that is being activated by the user. A typical example is a link that is being clicked.
`:checked`	pseudo-class	Selects a form element that is currently checked. A typical example might be a radio button or a check box.
`:first-child`	pseudo-class	Selects an element that is the first child of its parent. A common use is to provide different styling to the first element in a list.
`:first-letter`	pseudo-element	Selects the first letter of an element. Useful for adding drop-caps to a paragraph.
`:first-line`	pseudo-element	Selects the first line of an element.



Attribute Selectors

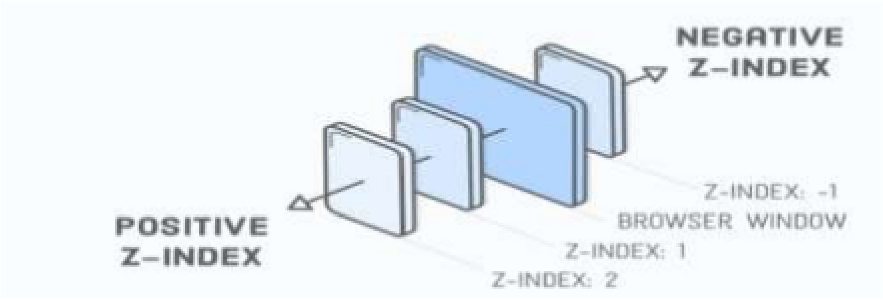
Selector	Matches	Example
[]	A specific attribute.	`[title]` Matches any element with a title attribute
[=]	A specific attribute with a specific value.	`a[title="posts from this country"]` Matches any <a> element whose title attribute is exactly "posts from this country"
[~=]	A specific attribute whose value matches at least one of the words in a space-delimited list of words.	`[title~="Countries"]` Matches any `title` attribute that contains the word "Countries"
[^=]	A specific attribute whose value begins with a specified value.	`a[href^="mailto"]` Matches any <a> element whose href attribute begins with "mailto"
[*=]	A specific attribute whose value contains a substring.	`img[src*="flag"]` Matches any <img> element whose src attribute contains somewhere within it the text "flag"
[$=]	A specific attribute whose value ends with a specified value.	`a[href$=".pdf"]` Matches any <a> element whose href attribute ends with the text ".pdf"

- pseudo elements
  - ::before - This enables us to add content before a certain element. For example,adding an opening quote before a blockquote.
  - ::after - This enables us to add content after a certain element. For example, a closing quotes to a blockquote.
  - ::first-letter - This is used to add a style to the first letter of the specified selector.
- box model in css

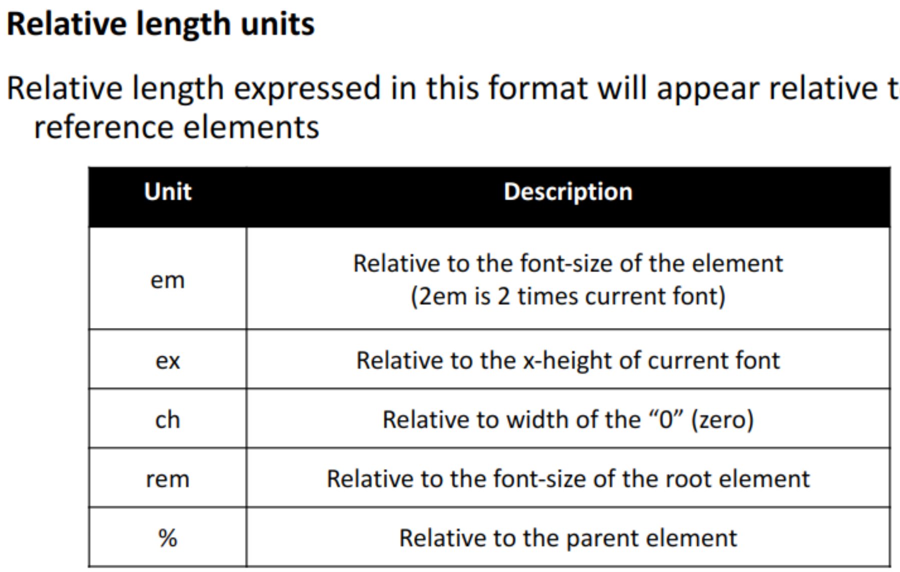


ON JEFFREY C. JACKSON'S SLIDES

- The total width of an element is calculated as:
  - Total element width = width + left padding + right padding + left border + right border + left margin + right margin
- The total height of an element is calculated as:
  - Total element height = height + top padding + bottom padding + top border + bottom border + top margin + bottom margin
- z index : element with higher z index stacked on top of element with lower z index



- positioning (position : property)
  - static : boxes positioned acc to normal document flow
  - relative : new position relative to its normal/old position
  - absolute : element positioned relativ to parent
  - fixed : relative to <html> , not affected by scrolling
  - sticky : It behaves until a declared point like position: relative, after that it changes its behaviour to position: fixed
- to get rounded border set border-radius to value
- display property allows to decide whether element rendered or not (display:none)
- text-shadow : horizontal_offset vertical_offset blur radius color



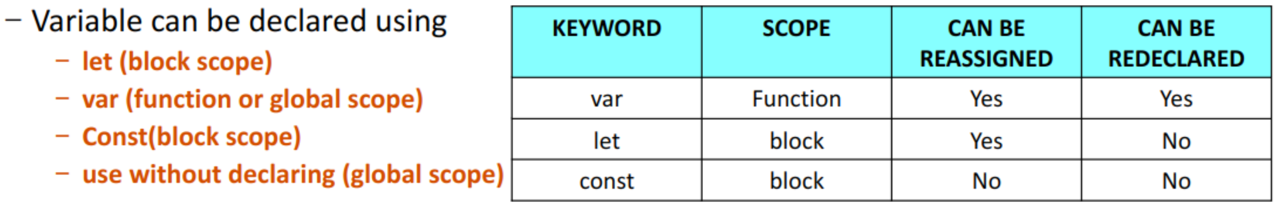
Relative length units

Relative length expressed in this format will appear relative to other reference elements

Unit	Description
em	Relative to the font-size of the element (2em is 2 times current font)
ex	Relative to the x-height of current font
ch	Relative to width of the “0” (zero)
rem	Relative to the font-size of the root element
%	Relative to the parent element

JAVASCRIPT

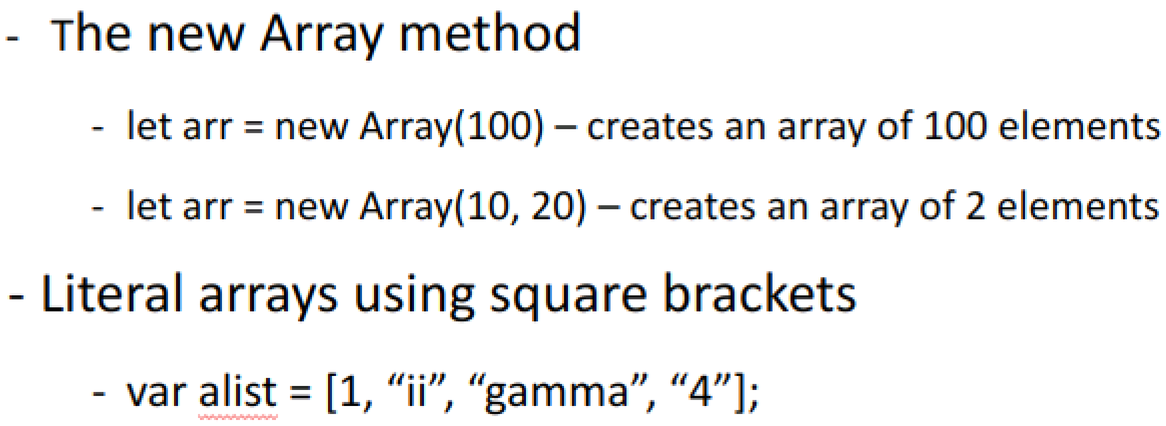
- loosely typed



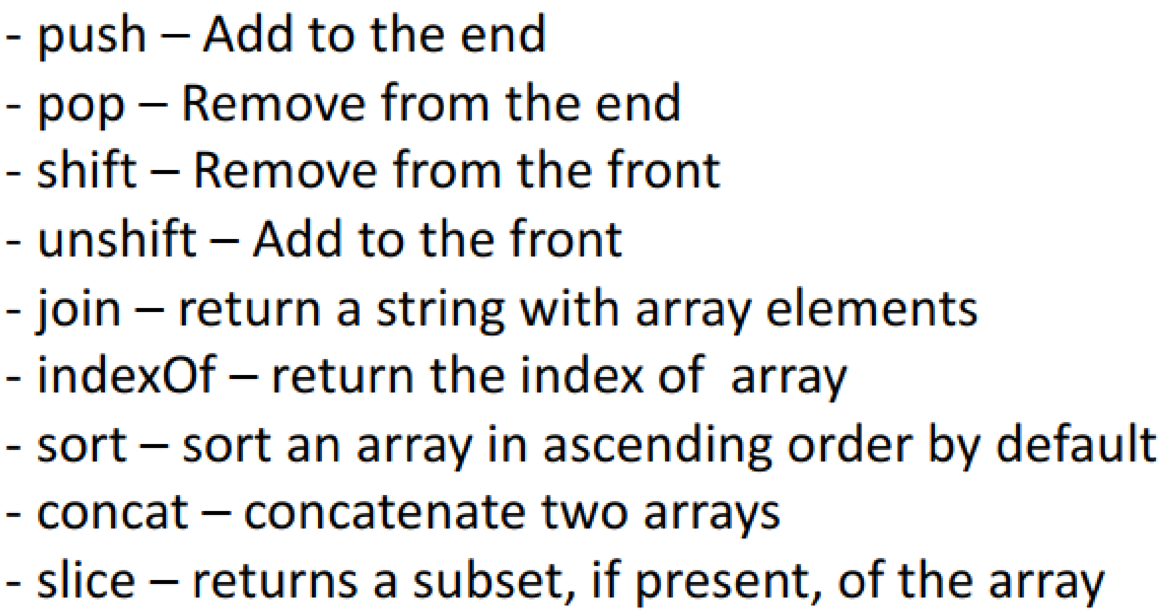
– Variable can be declared using
  - let (block scope)
  - var (function or global scope)
  - Const(block scope)
  - use without declaring (global scope)

KEYWORD	SCOPE	CAN BE REASSIGNED	CAN BE REDECLARED
var	Function	Yes	Yes
let	block	Yes	No
const	block	No	No

- when adding no and string , js treat it as a string
- hoisting : variavle nd function declaration moved to top of current scope
- array :



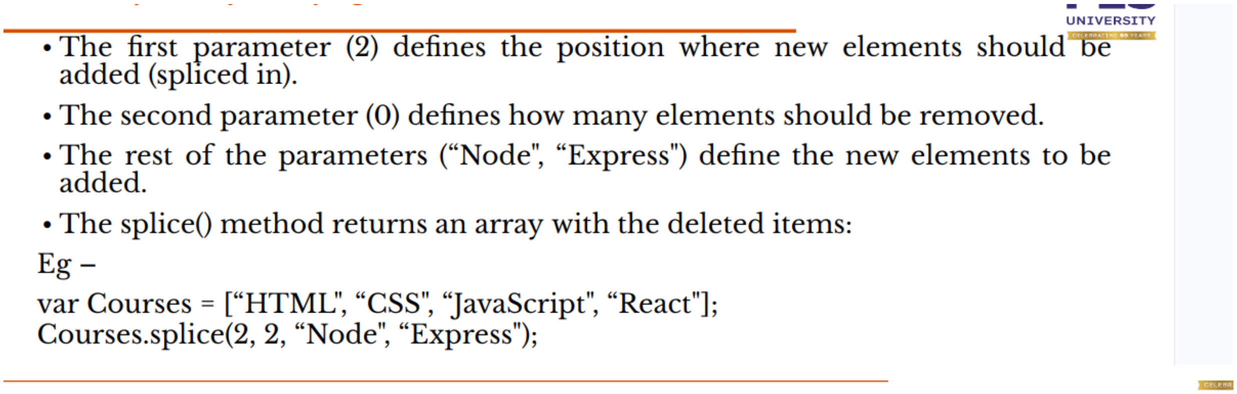
- The new Array method
  - let arr = new Array(100) – creates an array of 100 elements
  - let arr = new Array(10, 20) – creates an array of 2 elements
- Literal arrays using square brackets
  - var alist = [1, “ii”, “gamma”, “4”];



- push – Add to the end
- pop – Remove from the end
- shift – Remove from the front
- unshift – Add to the front
- join – return a string with array elements
- indexOf – return the index of array
- sort – sort an array in ascending order by default
- concat – concatenate two arrays
- slice – returns a subset, if present, of the array

- arr.sort([compare function])
  - **
- compareFunction takes two parameters say a and b and returns

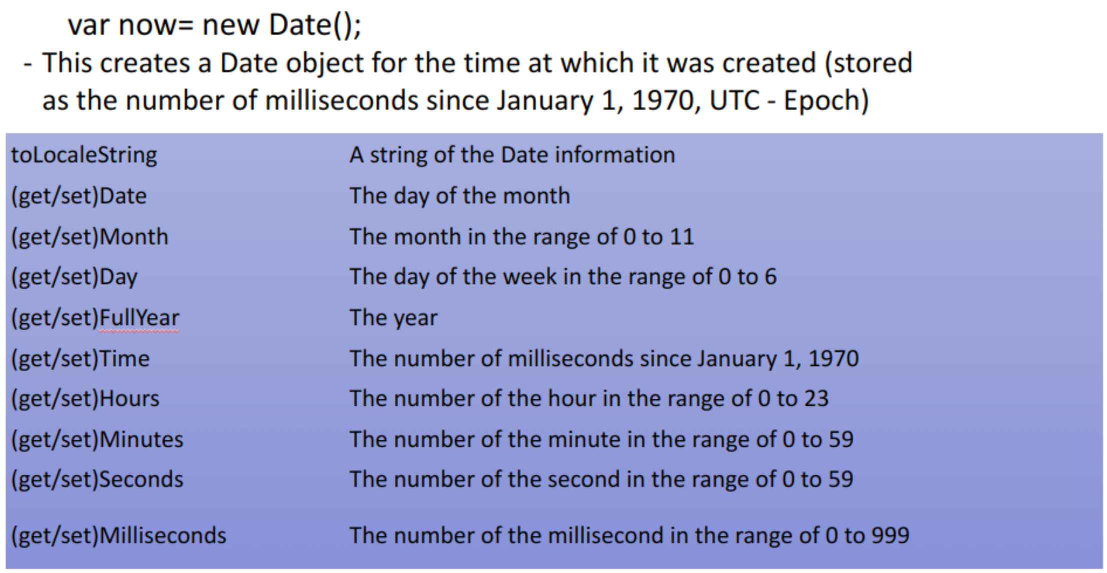
- 1 if a > b
- 0 if a = b
- -1 if a < b



- The first parameter (2) defines the position where new elements should be added (spliced in).
- The second parameter (0) defines how many elements should be removed.
- The rest of the parameters (“Node", “Express") define the new elements to be added.
- The splice() method returns an array with the deleted items:

Eg –

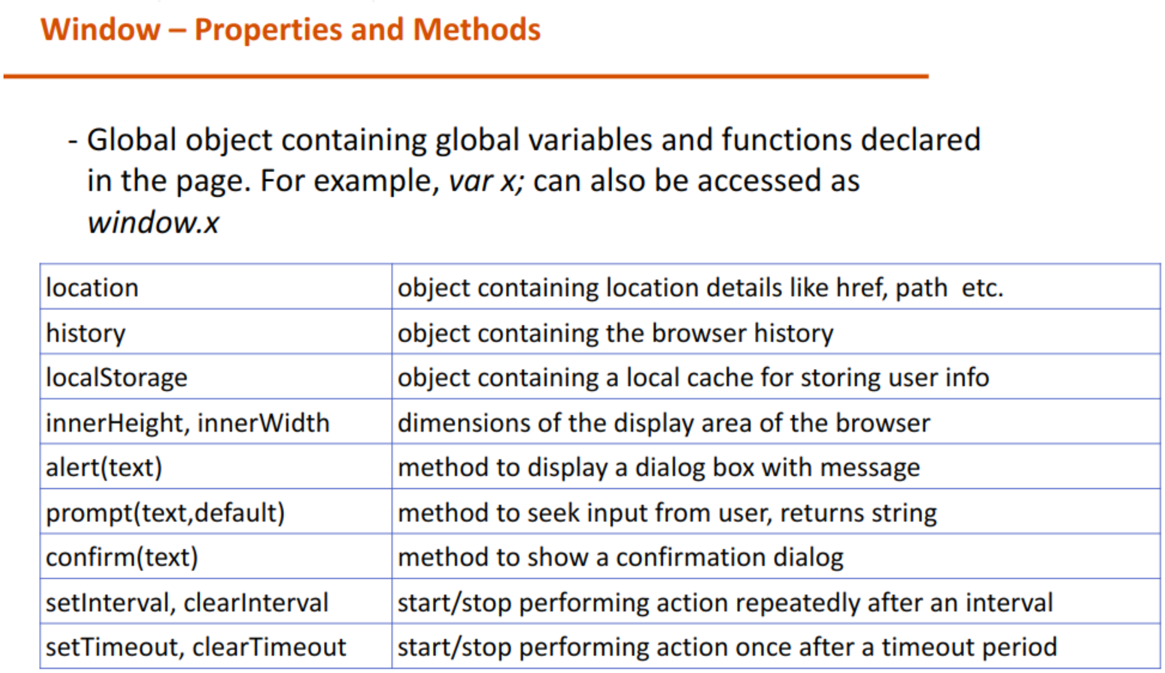
```
var Courses = [“HTML", “CSS", “JavaScript", “React"];
Courses.splice(2, 2, “Node", “Express");
```



```
var now= new Date();
```

- This creates a Date object for the time at which it was created (stored as the number of milliseconds since January 1, 1970, UTC - Epoch)

toLocaleString	A string of the Date information
(get/set)Date	The day of the month
(get/set)Month	The month in the range of 0 to 11
(get/set)Day	The day of the week in the range of 0 to 6
(get/set)FullYear	The year
(get/set)Time	The number of milliseconds since January 1, 1970
(get/set)Hours	The number of the hour in the range of 0 to 23
(get/set)Minutes	The number of the minute in the range of 0 to 59
(get/set)Seconds	The number of the second in the range of 0 to 59
(get/set)Milliseconds	The number of the millisecond in the range of 0 to 999



Window – Properties and Methods

- Global object containing global variables and functions declared in the page. For example, *var x;* can also be accessed as *window.x*

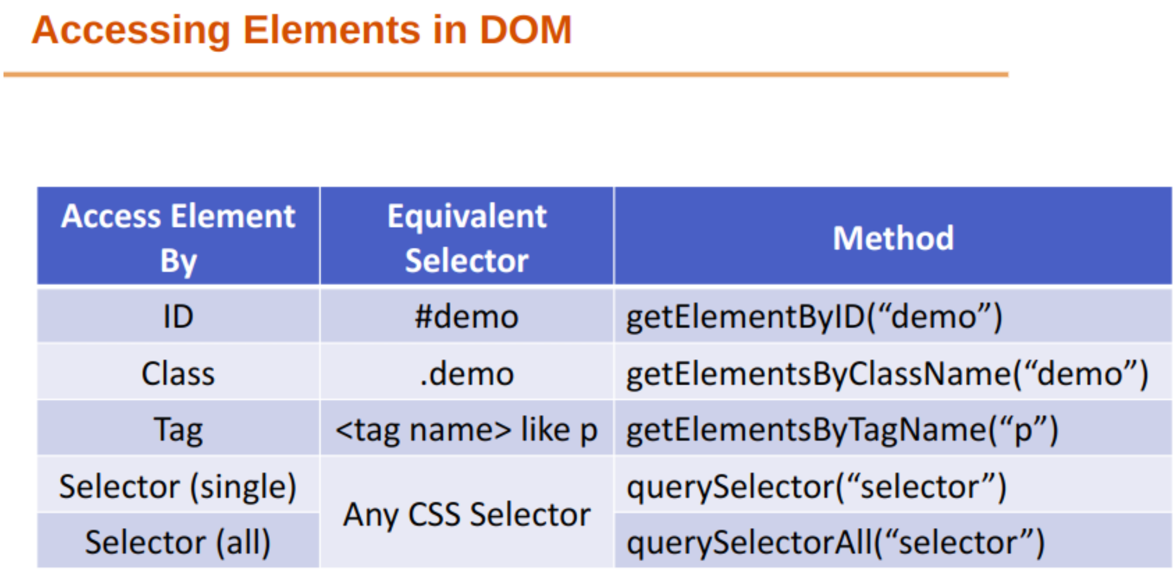
location	object containing location details like href, path etc.
history	object containing the browser history
localStorage	object containing a local cache for storing user info
innerHeight, innerWidth	dimensions of the display area of the browser
alert(text)	method to display a dialog box with message
prompt(text,default)	method to seek input from user, returns string
confirm(text)	method to show a confirmation dialog
setInterval, clearInterval	start/stop performing action repeatedly after an interval
setTimeout, clearTimeout	start/stop performing action once after a timeout period

- setTimeout(function,interval,function arguments)

- x=setInterval(function,interval)
- clearInterval(x)
- js functions are objects themselves.once function constructor created,create object of function using new keyword (fn nam uses camel case here)
- every object has prototype property.adding vars to prototype is liek class variable (object.prototype.var=var)
- object.create(prototype_object, propertiesObject)Returns a new object with the specified prototype object

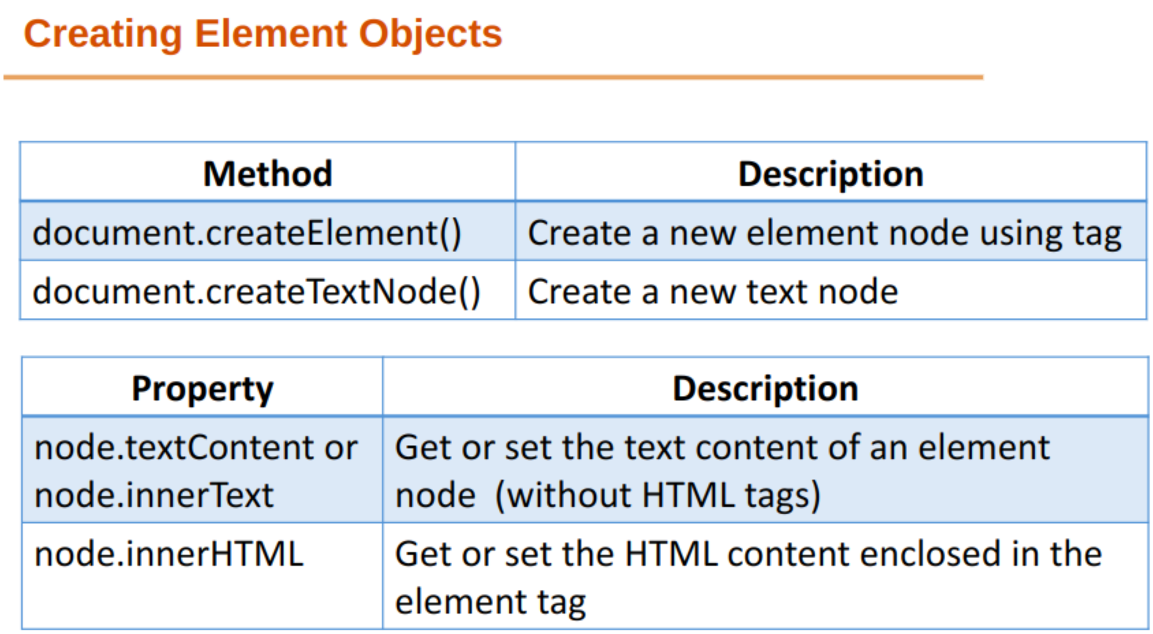
DOM - MANIPULATIONS

- dom - object oriented repre of web page constructed from browser
- The Document Fragment interface is a lightweight version of the Document that stores a piece of document structure like a standard document. However, a Document Fragment is not part of the active DOM tree.
  If you make changes to the document fragment, it doesn't affect the document or incurs any performance.
- To create a new document fragment, you use the Document Fragment constructor like this:
  - let fragment = new Document Fragment();
  - Or you can use the createDocumentFragment() method of the Document object:
    - let fragment = document.createDocumentFragment();



Accessing Elements in DOM

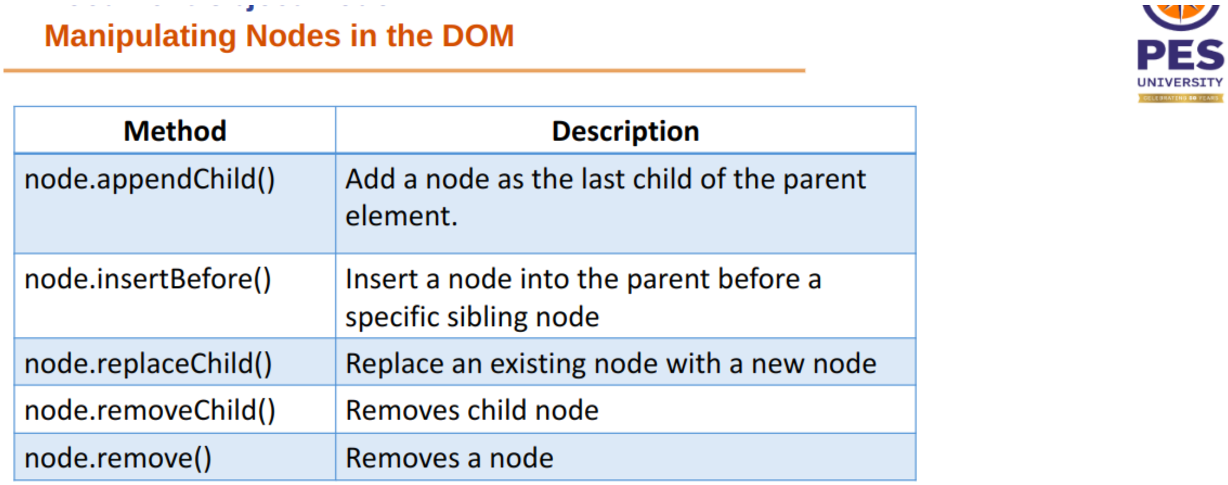
Access Element By	Equivalent Selector	Method
ID	#demo	getElementByID("demo")
Class	.demo	getElementsByClassName("demo")
Tag	<tag name> like p	getElementsByTagName("p")
Selector (single)	Any CSS Selector	querySelector("selector")
Selector (all)		querySelectorAll("selector")



Creating Element Objects

Method	Description
document.createElement()	Create a new element node using tag
document.createTextNode()	Create a new text node

Property	Description
node.textContent or node.innerText	Get or set the text content of an element node (without HTML tags)
node.innerHTML	Get or set the HTML content enclosed in the element tag



Manipulating Nodes in the DOM

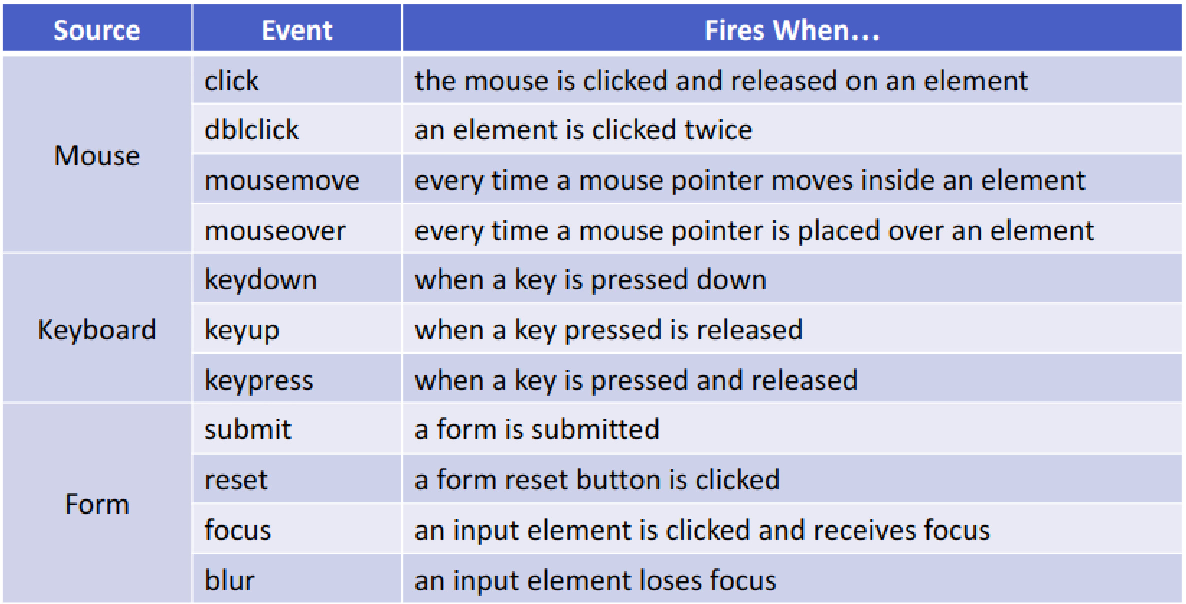
Method	Description
node.appendChild()	Add a node as the last child of the parent element.
node.insertBefore()	Insert a node into the parent before a specific sibling node
node.replaceChild()	Replace an existing node with a new node
node.removeChild()	Removes child node
node.remove()	Removes a node

- traversing elements
    - to get parent node : let parent = node.parentNode;
    - to get nextsibling : let nextSibling = currentNode.nextElementSibling;
    - to get privious sibling : let prevSibling = currentNode.previousElementSibling;
    - to delete dom node : elementNode.parentNode.removeChild(elementNode);
    - to set attribute : Element.setAttribute(name, value);
    - to get attribute : let attribute = element.getAttribute(attributeName);

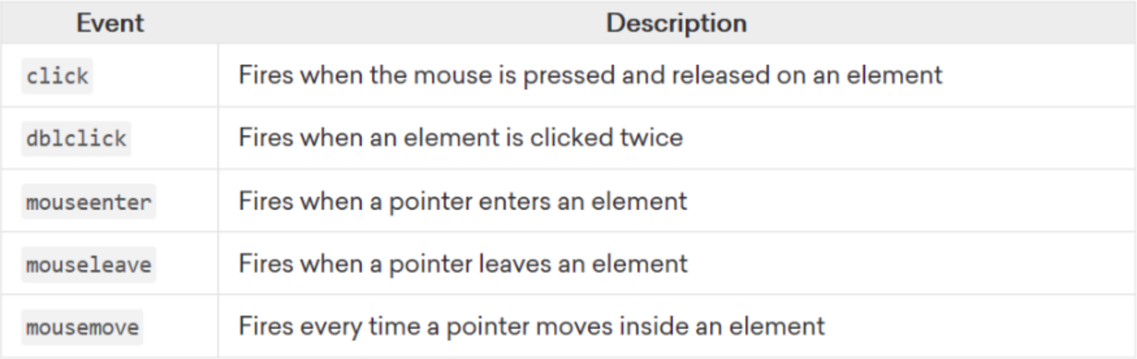
EVENTS

- events are created by activities associated with specific HTML elements.
- The process of connecting an event handler to an event is called registration.
- 3 ways to assign events to elements :

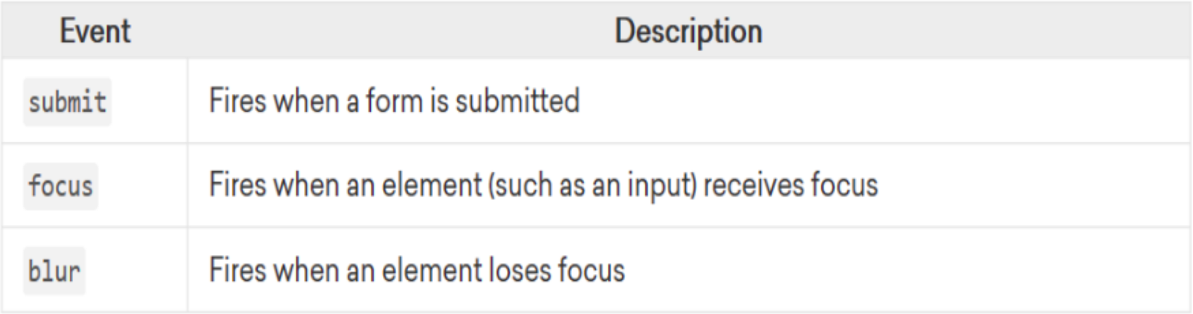
- inline event handler : onclick="function()"
- event handler properties : element.on<event>=handler(function())
- event listener : element.addEventListener(event,handler)



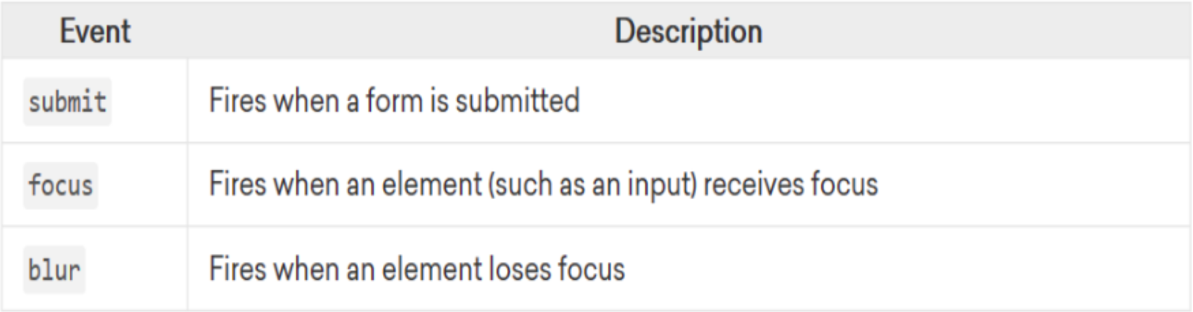
Source	Event	Fires When…
Mouse	click	the mouse is clicked and released on an element
	dblclick	an element is clicked twice
	mousemove	every time a mouse pointer moves inside an element
	mouseover	every time a mouse pointer is placed over an element
Keyboard	keydown	when a key is pressed down
	keyup	when a key pressed is released
	keypress	when a key is pressed and released
Form	submit	a form is submitted
	reset	a form reset button is clicked
	focus	an input element is clicked and receives focus
	blur	an input element loses focus



Event	Description
`click`	Fires when the mouse is pressed and released on an element
`dblclick`	Fires when an element is clicked twice
`mouseenter`	Fires when a pointer enters an element
`mouseleave`	Fires when a pointer leaves an element
`mousemove`	Fires every time a pointer moves inside an element

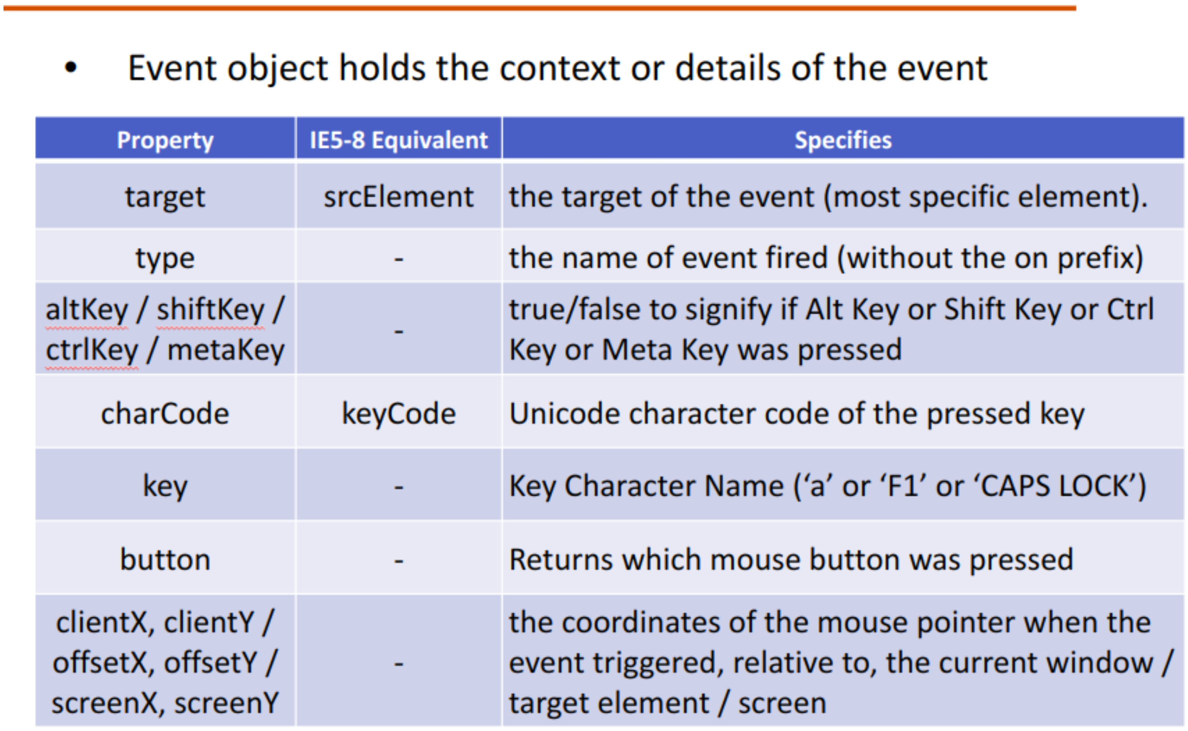


Event	Description
`submit`	Fires when a form is submitted
`focus`	Fires when an element (such as an input) receives focus
`blur`	Fires when an element loses focus



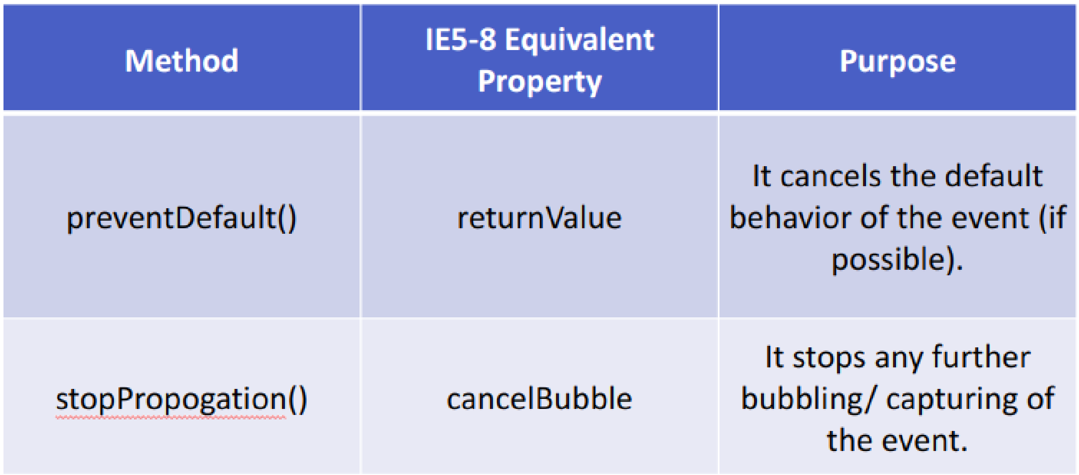
Event	Description
`submit`	Fires when a form is submitted
`focus`	Fires when an element (such as an input) receives focus
`blur`	Fires when an element loses focus

- event object properties



- Event object holds the context or details of the event

Property	IE5-8 Equivalent	Specifies
target	srcElement	the target of the event (most specific element).
type	-	the name of event fired (without the on prefix)
altKey / shiftKey / ctrlKey / metaKey	-	true/false to signify if Alt Key or Shift Key or Ctrl Key or Meta Key was pressed
charCode	keyCode	Unicode character code of the pressed key
key	-	Key Character Name ('a' or 'F1' or 'CAPS LOCK')
button	-	Returns which mouse button was pressed
clientX, clientY / offsetX, offsetY / screenX, screenY	-	the coordinates of the mouse pointer when the event triggered, relative to, the current window / target element / screen



Method	IE5-8 Equivalent Property	Purpose
preventDefault()	returnValue	It cancels the default behavior of the event (if possible).
stopPropogation()	cancelBubble	It stops any further bubbling/ capturing of the event.

- event propagation : defining element order when event occurs
  - bubbling
    - inner most element's event is handled first and then the outer: the <p> element's click event is handled first, then the <div> element's click event.
- capturing
  - outer most element's event is handled first and then the inner: the <div> element's click event will be handled first, then the <p> element's click event.
- elem.addEventListener(“event”, func_ref, flag);
  - flag = true :=> Handler registered for Capturing phase
  - flag = false:=> Handler registered for Bubbling phase (default)