PAGE No.
DATE / /

* Polymorphism :

- The term "poly" means "Many" and "morphos" means "Forms".
- It is ability of component to change its behaviour according to state and situation.
- Technically it is single base class object using memory of multiple derived classes
- In JS a single object can work for various classes

Java Script Browser Events

PAGE No.
DATE / /

- Event is a message sent by sender to its subscriber in order to notify change
- Event follows the design pattern called "observer" Also called as Communication pattern

function InsertClick() ← Subscriber
{

}

Sender

```
<button onclick = "InsertClick()"></button>
```

onclick = "InsertClick()" → Event handler

Event → called as Delegate - Function Pointer

- Sender sends notification
- Subscriber contains action to perform
- Event uses a Delegate mechanism [Function Pointer].

onclick ⇒ Event
onclick = "InsertClick()" ⇒ Event handler

- Every event handler contains default argumen

button → a) this : Contains information about object
onclick → b) event : Contains information about event.

Object info contains
name, id, class, value etc.

Event info contains
clientX, clientY, keyCode, charCode, which etc.

* Event Arguments :

- Default arguments [this, event]
- Custom arguments

PAGE No.
DATE / /

Various Catagories of Browser Events

1) Keyboard Events
2) Mouse Events
3) Clipboard Events
4) Button Events
5) Timer Events
6) Touch Events etc...

1) Keyboard events

- Handles actions regarding to keys
- They specify actions to perform while user is keying in the keys..

Event	Description
onkeyup	Actions when key is released
onkeydown	Actions when user holds down a key
onkeypress	Actions when user keys in another key

Key event related Properties

- keyCode	Code of every key ASCII
- ~~char~~code charCode	Code as per UTF standards [only character keys]
- which	Similar to keyCode [Enhanced keyboard]
- shiftKey - ctrlKey - altKey	Return true or false

* onkeypress will function when character is finished. i.e.

if we type "john" then keypress will work for "joh" only.
Since courser is after "n" so the "n" character is not finished.

Note:

onkeypress should be used when we are working on "charCode" because onkeyup and onkeydown will not return correct value of "charCode"/"keyCode".

If you are dealing with codes then always use keypress

If you are dealing with any other actions then use keyup or keydown.

PAGE No.
DATE / /

Mouse Events

Specify actions to perform with regard to mouse interactions

Event	Description
onmouseover	Actions when pointer is over element
onmouse out	Actions when pointer moves out
onmouse down	Actions when mouse button is down
onmouse up	Actions when mouse button is up
onmouse move	Actions when mouse ptr. is moving over element

Event Properties

clientX	Gets x position
clientY	Gets y position

Clipboard Events :

oncut	Removed and kept in Clipboard
oncopy	Copied to Clipboard
onpaste	Inserted from clipboard

FAQ: How to disable right click, cut, copy and paste like operations in page?

Ans: By disabling the event handler

You can disable event handler by returning false

ex: oncut = return false

Note:

oncontextmenu	: for right click
onselect	: for selection
onselectstart	: for extending selection [for dragging

PAGE No.
DATE / /

Button Events.

onclick : Actions on Single Click
ondbl click : Action on double click
oncontexmenu : Actions on right click

FAQ: How to handle click event for <a> element?

Ans: <a onclick = "function()"> // Invalid

<a href = "javascript:function()"> // Valid

Form Events

onsubmit : Defines actions when submitted
onreset : Defines actions when form resets

Note: These are events written for <form> element but they require submit and reset button in form

* We have to write functionality / events in <form> element not for the buttons of submit and reset

<form onsubmit = "function()"
onreset = "function()">

FAQ: How form can be submitted on any another element ~~of~~ event other than submit button?

Ans: By using "form.submit()" method

FAQ: Can a form ~~ha~~ have multiple submit buttons?

Ans: Yes

FAQ: Why you need multiple submit buttons?

Ans: As all submit buttons do same ~~work~~ work?

Ans: You can submit data to various API's or you ~~can~~ can handle multiple functionalities

FAQ: Form can have only one "onsubmit" then How it will manage various functionalities

Ans: By accessing submit button value

FAQ: Can we have ~~multi~~ multiple forms in page?

Ans: Yes

PAGE No.
DATE / /

Focus Events

onfocus : When element gets focus
onblur : When element ~~gets~~ to lost focus

Miscellaneous Events :

onchange : When value changes
onselect : When selected

Touch Events

ontouchstart
ontouchend
ontouchmove

Timer Events

setInterval()
clearInterval()
setTimeout()
clearTimeout()

PAGE No.
DATE / /

setTimeout : It makes any task inactive for specific duration of time and, invokes after time interval

It can execute the given task only once for every object

Syntax:

```
setTimeout (functionName, interval);
```

clearTimeout : Clears task from memory and will not allow to execute.

Syntax:

```
clearTimeout (refference Of Function);
```

setInterval : It performs specified task at regular intervals.

It is kept in the memory and sent to process

It is not removed from momory hence it repeats until cleared from memory

Syntax:

```
setInterval (FunctionName, Interval);
```

clearInterval() : Clears task from memory

Syntax:

clearInterval(reference of Function);

Note:

Example for reference of function

reference of function → var m1 = setInterval(FnName, interval);

clearInterval(m1);

Browser Objects

- JavaScript ⇒ BOM is Browser Object Model which provides a set of objects to control browser window.

1) Window
2) Location
3) ~~Navigator~~ Navigator
4) History
5) Document [DOM]

1) window:

- Provides properties ~~to control~~ and methods that are used to control browser window

1) open() : Opens the specified path data in new window
2) close() = Closes current window
3) print() = Prints current window

Syntax:

```
<button onclick = "window.close()">
<button onclick = "window.print()">
<button onclick = "window.open(
    'path', 'title', 'features')">
```

2) location :

. Provides properties and methods to control browser location details

host : Returns server name or IP address
port : Returns port no.
protocol : Returns protocol
href : gets & sets url dynamically
hash : gets current reference ID
search : gets the query string
pathname : gets current file path

Syntax: location.host

Methods :

location.reload : refresh / reloads current location details

FAQ: How to access current location details ?
Ans:

href : Complete URL
path : Current file path
hash : Current reference ID
search : Query String

location.hash : is used to access current location by using "id" reference

It can access named location within the page

3) Navigator Object :

- It is used to access current browser details
- It includes

Browser Family
version
plugins
Mime types
Geo location etc...

- It uses properties

appName
language
platform
plugins []
mimeTypes []
userAgent etc...
cookie Enabled

FAQ: How to get the status of JS in browser

Ans: By using HTML element

```
<noscript>
```

ex.

```
<h2> Javascript </h2>
<noscript>
    Javascript is desabled please
    enable it
</noscript>
```

FAQ: How to get MIME types?

Ans:

```
<script>
    Function f1() {
        for(var item of navigator.mimeTypes)
        {
            document.write(item.type +
                "<br>");
        }
    }
    f1();
</script>
```

FAQ: How to access Geo Location ?

Ans: By using

" navigator.geolocation.getCurrentPosition () "

Ex.:

```
<script>
    function f1(){

        navigator.geolocation.getCurrentPosition(
        function($position){

            document.write(`
            Latitude : ${position.coords.
                latitude} <br>
            Longitude: ${position.coords.
                longitude} <br>
            `);
        }
    }
    f1();
</script>
```

4) History Object

- Used to access browser history details
- Allows :

history.length : Returns total count of pages in history
history.back() : Moves to previous page in history
history.forward() : Moves to nxt page
history.go() : Moves to specific page.

Syntax:

```
history.go('home.html');
history.go(1)      // One page forward
history.go(-1)     // One page back
```

jQuery

PAGE No.
DATE / /

- It is a JavaScript library
- Library is set of values and functions
- You can import and implement in any application
- "Write less and Do More"
- Light weight and faster
- Reduces browser compatibility issues
- Introduced in 2006 by "John Resig".
- Like jQuery there are several JS based libraries
  - React.JS
  - Ember.js
  - Vue.js
  - Redux.js
  - Rx.js etc.
- jQuery is used for
  - Client side interactions
  - DOM manipulations
  - Interacting with API

Setup jQuery For project

> npm install jquery

Values , Functions

Factory

String Array Math History

[Library] Factory uses Single call technique

Service

Application

[Framework] Service uses Single ton

We have to create object for each call

We can call multiple times using single object

$ Represents Factory

Accessing jQuery in Web page

1) Import jQuery Script - core library

<script src = " path / .jquery.js></script>

2) Access jquery Factory and return all Functions

<script>

$(Function(){

});

</script>

For Everything

OR

<script>

$(document).ready(function(){

});

</script>

For DOM manipulation

jQuery Reference techniques

- jQuery can refer HTML elements by using all CSS selectors

```
$("p")          Type selector
$("#para1")     ID selector
$(".para2")     Class selector
```

jQuery DOM methods:

```
html()          - innerHTML
text()          - innerText
val()           - value
attr()          - attribute
prop()          - property
append()        - Add after
appendTo()      - Add
prepend()       - Add before
before()        - Add above
after()  etc.   - Add below
```

jQuery Events

- All JavaScript events are same in jQuery

JS		jQuery
onclick	→	click
onchange	→	change
onblur	→	blur

- No event Handler:

As we used in JS

```
<button onclick = "InsertClic()">
```

In jQuery No Need
In jQuery we write as

```
$("button").click(function(){})
```

In jQuery we don't write any thing in HTML

All program is in the script

- Only "event" is allowed in event argument
- There is no "this" key word.

```
$("button").click(function(event){
```

Event properties:
- event.clientX;
- event.clientY;

Object Properties:
- event.target.name;
- event.target.id;

```
});
```

jQuery Iterations

PAGE No.
DATE / /

$.each ()

:Syntax:

```
$.each (data, function (key,val
{

})
```

Similar to

```
for(var key in data)
{

}
```

&

```
for (var value of data)
{
}
```

But Combined

jQuery DOM

PAGE No.
DATE / /

Manipulation methods

html() : Similar to innerHTML
text() : Similar to innerTEXT
append() : Similar to append [after]
prepend() : Similar to prepend [before]
before() : Similar to before [up]
after() : Similar to after [down]
appendTo() : Similar to appendTo
attr() : Similar to attribute
prop() : Similar to property
each() : Similar to for loop
etc.

jQuery Animation Effects

show() : Similar to display: block
hide() : Similar to display: hide
toggle() : Similar to ~~then~~ onclick: fade
slide() : Similar to transition: slide
slideToggle() : Similar to show & hide by slide animation
Fade() : Similar to ~~show~~ by fade animation
fadeToggle() : Similar to show & hide by fade animation.

Syntax: $("#container").toggle(interval);

jQuery UI Components

- Effects
- Widgets
- Components
- Interactions

It is similar to bootstrap with pre-defined templates and interactions

* Download and Setup jQuery UI

① Visit https://jqueryui.com/
② Download UI library
③ Extract file and copy them into your project folder [node_modules]

"node_modules/jquery-ui"

④ Import following files into ~~web~~ web page

```
<link rel = "stylesheet" href = "path/jquery-ui.css">
<script src = "path/jquery.js"></script>
<script src = "path/jquery-ui.js"></script>
```

* jQuery Interactions

- draggable()
- resizable()
- droppable()
- sortable()
- selectable

PAGE No.
DATE / /

jQuery Widgets

accordion ()
datepicker ()
dialog ()
menu ()
etc

jQuery Ajax

- Asynchronous JavaScript and XML
- Asynchronous allows to handle tasks using 'unblocking' technique
- Unblocking technique allows the current task to run along with other tasks
- Ajax provides techniques that allows "partial post back"
- Only specific portion of page can post back
- Improved performance of page
- jQuery provides various Ajax methods.

$.ajax()
$.getJSON()
$.ajaxSuccess()
$.ajaxComplete()
$.ajaxStart()
$.ajaxFailure()
$.ajaxEnd() etc

FAQ: What is difference between Fetch() & "$.ajax()" ?

Ans:

- fetch() is window method
- It reads data in binary format.
- You have to convert data into Json
- It is only for "GET" not for other actions [POST, PUT, DELETE

- jQuery Ajax methods are imported and used from jQuery library
- Returns data directly in JSON format
- It can handle actions like GET, POST, PUT, DELETE
- It provides error object that can easily track the error in page request.

FAQ: What are issues in jQuery Ajax method

Ans:

- CORS [Cross Origin Resource Sharing]
- jQuery Ajax can not handle CORS issues
- jQuery Ajax request can be blocked by browser.
- Explicitly you have to enable and configure for CORS.

- Ajax is asynchronous but it needs external async techniques to handle various interactions

FAQ: What is solution?

Ans: You can try third party Ajax libraries
- HttpClient with RxJS [Angular]
- Axios [React]
- whatwg Fetch [WHATWG group]

What is use of these third parties

- They can handle CORS implicitly
- They are async
- They can handle multiple requests with single Ajax method

Syntax for Ajax:

```
$.ajax({ajaxOptions}).then(successFunction)
.then(failureFunction)
```

- Ajax in jQuery provides life cycle methods which can track the status of ajax request at every phase

```
ajaxStart()
ajaxComplete()
ajaxSuccess()
ajaxError()
```

- Error Object is defined in ajaxError that can catch the errors and display

status : code [ex 404 error]
statusText : Message [Not found]

Syntax:

```
.ajaxError(function(event, xhr
{
    xhr.Error.status;
    xhr.Error.statusText;
})
```

- Ajax "$.getJSON()" can be used if you are fetching data from JSON file or API that returns JSON

Syntax:

```
$.getJSON("url").then(function
(data) {})
```

SASS

- Syntactically Awesome Style Sheet
- It is CSS pre-compiler
- A precompiler can't directly compile on browser
- Sass files are compiled and translated into CSS

Browser : CSS
Developer : SASS

- Developer writes code in Sass ⇒ Compiles into CSS ⇒ Links CSS to page
- Sass provides all programming features that allow reusability, seperation, extensibility etc.
- Scss files are compatible with every web technology (React, Angular, Vue, JS, Java, .Net etc)

* Setup environment for Sass

① Download & install Sass CLI for your PC

> npm install -g sass

② Create new Sass program with extension scss

③ Trans compile scss into css

> sass file.scss file.css

Sass Hierarchy

- Sass allow to access and use HTML elements in same hierarchy how they are defined in page

- You don't need rational selectors [child, dependent]

Sass Variables

- Variables are storage locations in memory where you can store a value and use it as part of any expression

- Sass variables are declared by using '$'
- Sass variables are implicitly typed
- All datta types are same as JS types
- Sass variable allows atomic changes [Changes will effect various locations

Syntax:

```
$varName : value;

{
    property : $varName;
}
```

- Variable can be declared in block scope or global

Sass Functions

- Sass provides predefined functions to handle various actions implicitly
- Suppors functions that are used for
  a) Conditions
  b) Loops
  c) Iterations

Conditions in SASS

- Usess conditional functions for decision making
- Can execute set of statements based on condition
- Conditions are configured by using @if.... @else if ... @else

Syntax:

```
@if condition
{
    effects on true
}
@elseif condition
{
    effects on true
}
@else
{

}
```

DOMS | Page No.
Date / /

Loops in Sass

- Looping is the process of executing set of statements repeatedly until condition is satisfied

- Sass loops are defined by using

@ For

- Sass loops require

"from ... through"

- "from" defines starting counter
- "through" defines ending counter

- In function you can embed expressions by using

#{}

Syntax:

```
@ For $var from <start> through <end>
{

}
```

Sass Iterator function

@each ()

- Iterator is sass design pattern
- Iterator can read elements for collection in sequential order
- It doesn't require initialization, condition, counter

Syntax:

@each $var in/of collection

Mixins

- Similar to method in JavaScript
- They configure set of statements to execute but will not return a value

- Syntax:

@mixin referenceName (Params)
{
—
}

Method (it-> @include referenceName (values)

- Mixin enables reusability of statements

Functions

- Function is similar to ~~metmixins~~ but it is configure will a return value

Syntax:

@ function Name (params)
{
@ return value;
}

fn Call → referenceName (params);

Sass Module

- Module system allows to build a library
- Sass modules comprises of variables, functions and mixins
- You can import module and implement at any location
- Sass modules are defined in sass file with prefix "_".

ex. _moduleName.sass

- You can import module by using @ import statement.

@ import "moduleName"

Note:

For Sass module underscore as prefix is mandatory