jQuery

->Jquery is javascript library .
-> current version 3.7.1
-> it is simplify html , css , javascript code.

Ex. event(mouse , click) ,ajax request , animation ,Dom manipulation

Adding jquery in webpage

->download jquery
->include cdn link

Two version of jquery:
->production version
->development version

Difference between javascript and jQuery:
Javascript:
->it is programming language.
->it's syntax is complex
->it's speed is faster
Jquery:
->it is javascript library.
->it's syntax is short.
->it's speed is slow because library load.

Syntax of jQuery

$(selector).action()

$ is define jquery.
Selector find html element,tag,anchor tag etc.
Action is perform to element.

Selector use in jQuery:
->element selector
->id selector
->class selector

ex.
->* ->select all elements
-> this select the current html element
->p.btn select p element with class name btn
->p:first select first p element
->[href] select all element with anchor tag

Events:

Mouse event
->click
->dblclick
->mouse enter
->mouse leave

keyboard event
->keyup
->keypress
->keydown

Form event
->submit
->change
->focus
->blur

window event
->load
->resize
->scroll
->unload

Q.how to fire event programmatically?
1.trigger method:
Button clicked! (Handled by button)
Parent Div clicked! (Handled by parent)

2.triggerhandler
Button clicked! (Handled by button)

Custom logic on event fire:
1.basic example
2.condition with basic example
3.event fire with dynamic content

Validation with validator:
It is in include in message , rules , handler function , error class.

All condition rules:

Required , email , url , minlength , maxlength
Rangelength , min , max , number
Digit, accept , pattern , letters only
Alphanumeric , date , time

Jquery traversing:
->it is like dom tree.
parent():it is return direct parent element of selected element.

parents():it is return all parent element of the selected element.

parentsUntill():it is return parent element between two given argument.

children():it is return all direct children element of the select element.

find():it is return descendant element of the selected element.

siblings():it is return the all sibling element of the selected element.

next(): it is return next sibling element of the selected element.

nextAll():it is return all next sibling element of the selected element.

nextUntill():itis return next sibling element between two given argument.

prev():it is return prev sibling element of the select elemet.

prevAll():it is return all prev element of the select element.

prevUntill():it is return prev sibling element between two given elements.

first():it is return a first element of specified element.

last(): it is return last element of specified element.

eq():it is return element with a specific index number of selected element

filter():it is use a specify condition
Ex. p element is not match for condition so p element is remove and other element is return.

not():it is return all element they dont match condition

Filter and not method are opposite.

jQuery function:
map():
Applies a function to each item in an array or element collection and returns a new array.

Syntax:

```
$(selector).map(function(index, element) {
    // Transformation logic
    return newValue;
}).get(); // Optional to get a plain array
```

grep():
To filter specific items from an array or list.
Syntax:

```
$.grep(array, function(value, index) {
    return condition; // true values will be included
});
```

Extend:
To merge the properties of one object into another.
Syntax:

```
$.extend(target, object1, object2, ...);
```

Each:
Applies a function to every element or item in an array.
Syntax:

```
$.each(arrayOrObject, function(index, value) {
    // Logic for each item
});
```

Merge:
Combine two array
Syntax:

```
$.merge(array1, array2);
```

Regular expression in jquery:

i means incasesensive match
d means digit match
g means global match

m multiline match
D non-digit match
a+ more a occurs (aaa)
a? (“” , a) only

use JavaScript methods like `test()`, `match()`, or `replace()` in combination with jQuery to manipulate or validate strings.

Common use case:
->email validation
->password validation
->replace

Call back function in jquery:
callback function is a function passed as an argument to another function, which gets executed once the main function is completed.

->animation call back
->ajax call back
->custom call back

Advantages of call back:
->reusebality

Deferred and promise object:

Deferred and Promise are used in jQuery to handle asynchronous operations
Their main purpose is to help manage asynchronous tasks (such as AJAX requests) so that we can easily handle the result once it's ready.

Deferred:
A Deferred is an object that manages the result of an asynchronous operation. When the asynchronous task completes, it either resolves or rejects
You can set callback functions that will execute when the response is ready.

Key:
resolve():Called when the task is successful
reject(): Called when the task fail
done():This function is called when the task completes successfully
fail():This function is called if the task fails

->Calling deferred.resolve() triggers the done() callback.
->Calling deferred.reject() triggers the fail() callback.

Promise:
A Promise is an object that represents the result of an asynchronous operation.
A Promise resolves or rejects a value, and then callback functions are executed when the result is ready.

Key:

Promises have methods like then() , catch() and finally() to handle success, failure, or completion cases.
->Calling resolve() triggers the then() function.
->Calling reject() triggers the catch() function.

What is ajax??

->ajax is asynchronous javascript and XML.
->ajax is technique it's give to permission to data send and data receive without reload the webpage.
->it is not programming language.
Ex.gmail,google,youtube

Why use ajax??
->better user experience(without reload webpage)
->fast performance

Use of ajax:
It is use to create dynamic content.

Types of requests in ajax:
Synchronous
Asynchronous

Synchronous:
When you send a synchronous request, JavaScript execution is blocked until the server responds.

->After sending the request, your code execution stops.
->Once the server responds, the rest of the code is executed.
->The UI is blocked, and no interaction is possible.
->performance is slow because code block
->Code execution is blocked
->Waits for the server response, then handles errors

Asynchronous:

When you send an asynchronous request, JavaScript code continues to run without blocking

->After sending the request, the code continues executing without waiting for the response.
->The UI remains responsive, and you can interact with it.
->Code execution is not blocked
->The callback function is called once the server responds.
->Faster, as the code doesn't wait and execution continues
->Errors are handled asynchronously

By default ajax code behaviour is asynchronous.

Basic syntax:

```
$.ajax({
  url: "url",  // server url
  type: "GET", // req-type-get/post
  Data:{} // send data to the server
  success: function(response) {
    console.log(response);
  },
```

```
  error: function(error) {
    console.log("Error: ", error);
  }
});
```

Functions:

->get:

$.get()

Send a GET request to the server (retrieve data).

Syntax:

```
$.get(url, data, successCallback);
$.get("https://jsonplaceholder.typicode.com/posts/1",
function(data) {
  console.log(data);
});
```

->post:

$.post()

Send a POST request to the server (send data)

Syntax:

```
$.post(url, data, successCallback);
$.post("https://jsonplaceholder.typicode.com/posts", { title: "Test"
}, function(data) {
  console.log(data);
});
```

->load:

dynamically loading content.

$.load()

$('#element').load(url, data, successCallback);
$("#content").load("example.html");

->ajax:
$ajax()
General-purpose method for making AJAX requests (GET, POST, PUT, DELETE, etc.)

Json:
JSON is a format for storing and transporting data.
Json stands for javascript object notation.
Json structure is key value pair format.
Syntax:

```
{
  "name": "parth",
  "age": 25,
  "email": "parth@example.com",
  "isStudent": false,
  "address": {
    "street": "123 Mvadi",
    "city": "rajkot"
  },
  "hobbies": ["reading", "cycling", "swimming"]
}
```

It’s store only data . not store function,method etc.

Use of json:
->Data exchange between server and client
->Storing data (e.g., browser's localStorage, databases)

Serialization:
Converting data into a format (usually a string) so that it can be sent to the server.
The jQuery serialize() method converts form data into a query string format that can be sent to the server.
This serialized data is then sent to the server through the AJAX request.

Ex.

```
it is use to convert javascript object to string format(json)
        let obj = { name: "parth", department: "CSE", marks: 90 };
        let serializedData = JSON.stringify(obj);
        console.log("serialization data",serializedData);
```

o/p: {"name":"parth","department":"CSE","marks":90}

DeSerialization:
Converting serialized data back into its original form (like an object)

```
let jsonString = '{"name":"parth","department":"CSE" , "marks":"90"}';
        let deserializedData = JSON.parse(jsonString);
        console.log("Deserialization data",deserializedData.name);
```

o/p:parth CSE 90