

Q: Difference between let and Var in JavaScript?

Q: What is the difference between var a =90 and a=90?

Q: What is the difference between manifestation and bundling?

Q: How to optimize sql queries?

Q: Find the 2nd highest salary from number type column in sql?

Q: Can we use multiple catch with single try statement in c#?

Q: Is JavaScript case sensitive? Give an example?

**Q: What does the following statement declares?**

```
var myArray = [[[]]];
```

It declares a three dimensional array.

**NOTE: NULL IS CONSIDERED AS AN OBJECT IN JAVASCRIPT.**

**Q: What is the use of isNaN function?**

**NaN == not a number**

isNaN function returns true if the argument is not a number otherwise it is false.

**Q: What will be the output of this code?**

```
var x = 21;
var girl = function () {
  console.log(x);
  var x = 20;
};
girl ();
```

Neither 21, nor 20, the result is **undefined** It's because JavaScript initialization is not hoisted.

(Why doesn't it show the global value of 21? The reason is that when the function is executed, it checks that there's a local x variable present but doesn't yet declare it, so it won't look for global one.)

**Q: How do you add an element at the beginning of an array? How do you add one at the end?**

```
var myArray = ['a', 'b', 'c', 'd'];
myArray.push('end');
myArray.unshift('start');
console.log(myArray); // ["start", "a", "b", "c", "d", "end"]
```

**Q: Imagine you have this code:**

```
var a = [1, 2, 3];
```

**a) Will this result in a crash?**

```
a[10] = 99;
```

**b) What will this output?**

```
console.log(a[6]);
```

a) It will not crash. The JavaScript engine will make array slots 3 through 9 be “empty slots.”

b) Here, a[6] will output undefined, but the slot still remains empty rather than filled with undefined.

**Q: What would following code return?**

```
console.log(typeof typeof 1);
```

string

typeof 1 will return "number" and typeof "number" will return string

**Q: What is the difference between == and === operator?**

== in JavaScript is used for comparing two variables, but it ignores the datatype of variable.

=== is used for comparing two variables, but this operator also checks datatype and compares two values.

**Q: Can you assign an anonymous function to a variable and pass it as an argument to another function?**

Yes! An anonymous function can be assigned to a variable. It can also be passed as an argument to another function.

**NOTE: IMPORTANT! In JavaScript, functions are objects. Because of this, functions can take functions as arguments, and can be returned by other functions.**

**In C programming you can only pass variables as parameter to function. You cannot pass function to another function as parameter. But, you can pass function reference to another function using function pointers (delegates)**

**Q: What is Closure? Give an example.**

Closures are created whenever a variable that is defined outside the current scope is accessed from within some inner scope. It gives you access to an outer function's scope from an inner function. In JavaScript, closures are created every time a function is created. To use a closure, simply define a function inside another function and expose it.

**Q: List out the different ways an HTML element can be accessed in a JavaScript code.**

Here are the list of ways an HTML element can be accessed in a Javascript code:

(i) getElementById('idname'): Gets an element by its ID name

(ii) getElementsByClass('classname'): Gets all the elements that have the given classname.

(iii) getElementsByTagName('tagname'): Gets all the elements that have the given tag name.

**Q: What are the ways to define a variable in JavaScript?**

1. Var
2. Const
3. Let

**NOTE:** undefined and null are two distinct types: undefined is a type itself (undefined) while null is an object.

***Q: What is the 'Strict' mode in JavaScript and how can it be enabled?***

"use strict" is a ES5 feature in JavaScript that makes our code in Strict Mode in functions or entire scripts. Strict mode is a way to introduce better error-checking into your code. You can enable strict mode by adding "use strict" at the beginning of a file, a program, or a function.

***Q: Which type of variable among global and local, takes precedence over other if names are same?***

A local variable takes precedence over a global variable with the same name.

***Q: What is the difference between undefined and not defined in JavaScript?***

In JavaScript, if you try to use a variable that doesn't exist and has not been declared, then JavaScript will throw an error var name is not defined and script will stop executing. However if you have declared a variable but not initialized it then it will give 'undefined' error on its access.

***Q: What is the prototype of an object?***

A prototype in simplest terms is a blueprint of an object. It is used as a fallback for properties and methods if it does exist in the current object. It's the way to share properties and functionality between objects. All JavaScript objects inherit properties and methods from a prototype.

***Q: what is bitwise AND& operator in javascript?***

### Q: What are Promises?

Promises are just like callbacks but they are a cleaner way of doing that. Promises are used to handle asynchronous operations in JavaScript. They are easy to manage when dealing with multiple asynchronous operations where callbacks can create callback hell leading to unmanageable code.

1. **fulfilled**: Action related to the promise succeeded
2. **rejected**: Action related to the promise failed
3. **pending**: Promise is still pending i.e not fulfilled or rejected yet
4. **settled**: Promise has fulfilled or rejected

```
let p = new Promise((resolve, reject) => {
  let a = 1 + 1;
  if (a == 2) {
    resolve('success');
  }
  else {
    reject('rejected');
  }
});
p.then((message) => {
  console.log('This is in the then ' + message);
}).catch((message) => {
  console.log('This is in the catch ' + message);
});
var userLeft = true, catMeme = true;
function promiseExample() {
  return new Promise((resolve, reject) => {
    if (userLeft) {
      reject({
        name: "User left",
        message: "Your user has left you :('"
      })
    }
    else if (catMeme) {
      reject({
        name: "your user is watching cat meme",
        message: "Cats are better than you, duhh"
      })
    }
    else {
      resolve({
        name: "Your user is watching your video.",
        message: "Your user is with you."
      })
    }
  })
}
```

```
    }  
  });  
}  
promiseExample().then((message) => {  
  console.log('Success: ' + message.message + ' '+ message.name);  
}).catch((error) => {  
  console.log(error.name + ' ' + error.message);  
});
```

# **AJAX & jQuery**

## ***Q: What is AJAX?***

AJAX stands for Asynchronous JavaScript and XML. It is a group of related technologies used to display data asynchronously. What this means is that we can send data to the server and get data from the server without reloading the web page.

## ***Q: What is the difference between synchronous and asynchronous requests?***

Synchronous request blocks the user until a response is retrieved whereas asynchronous doesn't block the user.

## ***Q: What are the types of post back in AJAX?***

There are two types of post back in AJAX.

- Synchronous Postback - It blocks the client until the operation completes.
- Asynchronous Postback - It doesn't block the client.

## ***Q: How can you test the AJAX code?***

JUnit is the open source unit testing framework for client-side JavaScript. It is required to create test cases. The unit test case is a code which ensures that the program logic works as expected.

## ***Q: Which request is better for AJAX, Get or Post?***

AJAX requests should use an HTTP GET request where the data does not change for a given URL requested.

An HTTP POST should be used when state is updated on the server. This is highly recommended for a consistent web application architecture.

get, post and load are the wrapper ajax methods, that call ajax method inside them.

## ***Q: How to select an element with attribute name, in jquery?***

`$("[attribute]")` - selects a set of elements that have the specified attribute.

`$("[attribute='value']")` - selects a set of elements that have the specified attribute with the specified value.

**`$("[attribute*='value']")`** - selects a set of elements that have the specified attribute with the value that contains the specified substring. A substring can be a part of another string anywhere inside it, so the phrase you specify as the value does not have to match the whole.

**`$("[attribute$='value']")`** - selects a set of elements that have the specified attribute with the specified value at the end.

**`$("[attribute^='value']")`** - selects a set of elements that have the specified attribute with the specified value at the start.

**`$('input[type=checkbox]:checked')`** - selects all checked elements.

**`$(":disabled")`** - selects all disabled elements.

#### ***Q: What is the difference between `find()` and `children()` methods?***

Both the methods are used to filter the child of the matched elements. `Find()` method takes the element that it wants to search in its argument, then it will find all levels down the DOM tree but `children` method searches only a single level down the DOM tree.

#### ***Q: What is the difference between `onload()` and `document.ready()` methods?***

`Body.Onload()` event will be called only after the DOM and associated resources like images get loaded, but jQuery's `document.ready()` event will be called once the DOM is loaded and it does not wait for the resources such as images to be loaded.

```
$("ul").children().css({"color": "red", "border": "2px solid red"});           // this will apply color to its //immediate child that in most of the cases would be li element.
```

```
$("ul").find("span").css({"color": "red", "border": "2px solid red"});       // this will find span element in all the children of ul element and apply css to it.
```

#### ***Q: What is jQuery connect?***

A 'jQuery connect' is a plug-in used to connect or bind a function with another function. Connect is used to execute a function whenever a function from another object or plug-in is executed.

#### ***Q: What is the difference between `jquery.min.js` and `jquery.js`?***

`jquery.min.js` is a compressed version of `jquery.js` (whitespaces and comments are removed, shorter variable names are used and so on) in order to preserve bandwidth. In terms of functionality, they are absolutely the same. It is recommended to use this compressed version in the production environment. The efficiency of web page increases when the minimized version of jQuery is used.

**Q: Differentiate among .empty() VS .remove() VS .detach() in jQuery.**

**remove()** removes the element as well as its child elements. Data from DOM can be restored; however, event handlers can't be restored.

**empty()** does not remove the element, however, remove its content and the associated child elements

**detach()** removes the element and all the associated child elements, but retains the data and event handlers of the removed element to be reused later.

```
<p>This is a paragraph</p>
x = $("p").detach();
$("body").append(x);
```

**Q: What is the method chaining in jQuery, and what are the advantages?**

With chaining, multiple jQuery commands on a particular element can be executed in one go. It helps in implementing various actions on an element at once rather than executing them one after the other.

```
$("#h2").css("color", "blue").animate({left: '100px'}).slideDown(1000);
```

**Q: Difference between prop and attr?**

Both attr() and prop() can be used to set or get an element's value, however attr() returns the original (default) value whereas prop() returns the most recent (current) value. For example, if a text input had an initial value of 'Male,' and later it was changed by the user to 'female,' attr() will return the value as 'Male' whereas prop() will return the value as 'female.'

**Q: Can you use any other name in place of \$ (dollar sign) in jQuery?**

Yes, instead of \$ (dollar sign) we can use jQuery as a function name. For example:

```
jQuery(document).ready(function() {
  jQuery("p").css("background-color", "pink");
});
```

**Q: Whether our own specific characters are used in place of \$ in jQuery?**

Yes, we can use our own variable in place of \$ by using the method called no Conflict () method.

```
var sample = $.noConflict()
var jq = $.noConflict();
jq(document).ready(function(){
  jq("button").click(function(){
    jq("p").text("jQuery is still working!");
  });
});
```



'NoConflict' method is used to reference a jQuery and save it in a variable. That variable can be used instead of Sign.

***Q: Explain what the following code will do:***

```
$( "div#first, div.first, ol#items > [name$='first']" )
```

This code performs a query to retrieve any <div> element with the id first. It also includes all <div> elements with the class first and all elements which are children of the <ol id="items"> element whose name attribute ends with the string "first". This shows how to use multiple selectors at once. The function will return a jQuery object containing the results of the query.