Q1. What is the difference between Java & JavaScript?

Java	JavaScript
Java is an OOP programming language.	JavaScript is an OOP scripting language.
It creates applications that run in a virtual machine or browser.	The code is run on a browser only.
Java code needs to be compiled.	JavaScript code are all in the form of text.

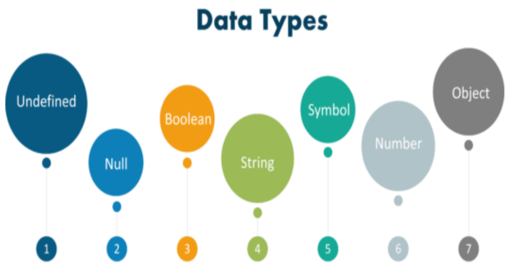
Q2. What is JavaScript?

JavaScript is a lightweight, interpreted programming language with object-oriented capabilities that allows you to build interactivity into otherwise static HTML pages. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

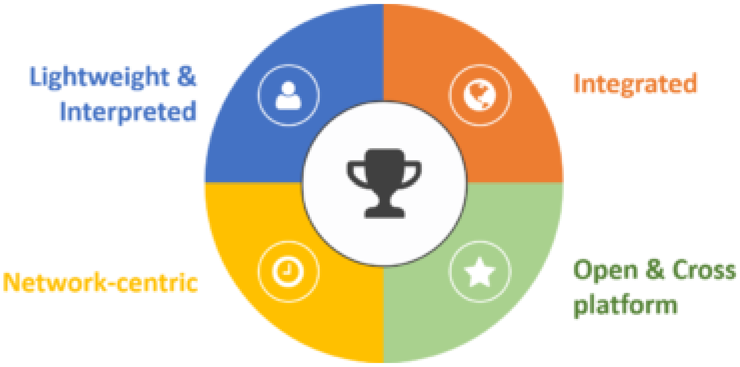
Q3. What are the data types supported by JavaScript?

The data types supported by JavaScript are:

- Undefined
- Null
- Boolean
- String
- Symbol
- Number
- Object



Q4. What are the features of JavaScript?



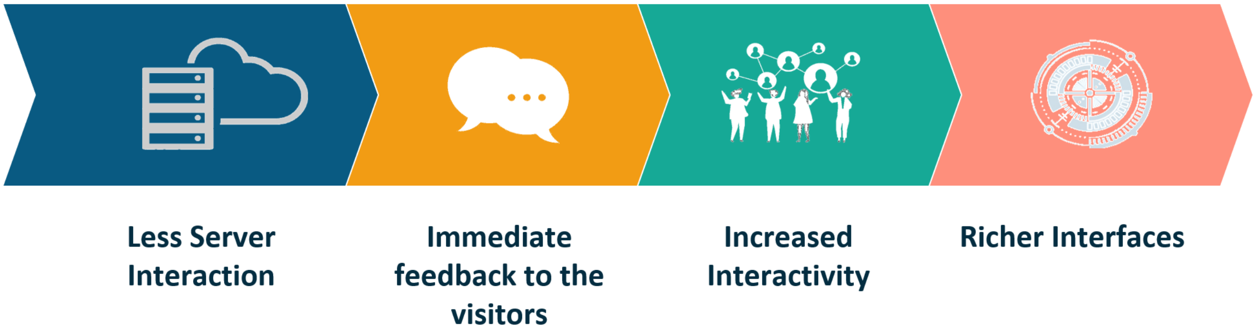
Following are the features of JavaScript:

- It is a lightweight, interpreted programming language.
- It is designed for creating network-centric applications.
- It is complementary to and integrated with Java.
- It is an open and cross-platform scripting language.

Q5. Is JavaScript a case-sensitive language?

Yes, JavaScript is a case sensitive language. The language keywords, variables, function names, and any other identifiers must always be typed with a consistent capitalization of letters.

Q6. What are the advantages of JavaScript?



Following are the advantages of using JavaScript −

- Less server interaction − You can validate user input before sending the page off to the server. This saves server traffic, which means less load on your server.

- Immediate feedback to the visitors − They don’t have to wait for a page reload to see if they have forgotten to enter something.
- Increased interactivity − You can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.
- Richer interfaces − You can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to your site visitors.

Q7. How can you create an object in JavaScript?

JavaScript supports Object concept very well. You can create an object using the object literal as follows −

```
var emp = {
name: "Daniel",
age: 23
};
```

Q8. How can you create an Array in JavaScript?

You can define arrays using the array literal as follows-

```
var x = [];
var y = [1, 2, 3, 4, 5];
```

Q9. What is a name function in JavaScript & how to define it?

A named function declares a name as soon as it is defined. It can be defined using function keyword as :

```
function named(){
// write code here
}
```

Q10. 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.

Q11. What is argument objects in JavaScript & how to get the type of arguments passed to a function?

JavaScript variable arguments represents the arguments that are passed to a function. Using typeof operator, we can get the type of arguments passed to a function. For example −

```
function func(x){
console.log(typeof x, arguments.length);
}
func(); //==> "undefined", 0
func(7); //==> "number", 1
func("1", "2", "3"); //==> "string", 3
```

Q12. What are the scopes of a variable in JavaScript?

The scope of a variable is the region of your program in which it is defined. JavaScript variable will have only two scopes.

- Global Variables − A global variable has global scope which means it is visible everywhere in your JavaScript code.
- Local Variables − A local variable will be visible only within a function where it is defined. Function parameters are always local to that function.

Q13. What is the purpose of ‘This’ operator in JavaScript?

The JavaScript this keyword refers to the object it belongs to. This has different values depending on where it is used. In a method, this refers to the owner object and in a function, this refers to the global object.

Q14. What is Callback?

A callback is a plain JavaScript function passed to some method as an argument or option. It is a function that is to be executed after another function has finished executing, hence the name ‘call back‘. In JavaScript, functions are objects. Because of this, functions can take functions as arguments, and can be returned by other functions.

Q15. 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.

Q16. Name some of the built-in methods and the values returned by them.

Built-in Method	Values
CharAt()	It returns the character at the specified index.
Concat()	It joins two or more strings.
forEach()	It calls a function for each element in the array.
indexOf()	It returns the index within the calling String object of the first occurrence of the specified value.
length()	It returns the length of the string.
pop()	It removes the last element from an array and returns that element.
push()	It adds one or more elements to end of an array and returns the new length of the array.
reverse()	It reverses the order of the Elements of an array.

Q17. What are the variable naming conventions in JavaScript?

The following rules are to be followed while naming variables in JavaScript:

1. You should not use any of the JavaScript reserved keyword as variable name. For example, break or boolean variable names are not valid.
2. JavaScript variable names should not start with a numeral (0-9). They must begin with a letter or the underscore character. For example, 123name is an invalid variable name but _123name or name123 is a valid one.
3. JavaScript variable names are case sensitive. For example, Test and test are two different variables.

Q18. How does TypeOf Operator work?

The typeof operator is used to get the data type of its operand. The operand can be either a literal or a data structure such as a variable, a function, or an object. It is a unary operator that is placed before its single operand, which can be of any type. Its value is a string indicating the data type of the operand.

Q19. How to create a cookie using JavaScript?

The simplest way to create a cookie is to assign a string value to the document.cookie object, which looks like this-

Syntax :

```
document.cookie = "key1 = value1; key2 = value2; expires = date";
```

Q20. How to read a cookie using JavaScript?

Reading a cookie is just as simple as writing one, because the value of the document.cookie object is the cookie. So you can use this string whenever you want to access the cookie.

- The document.cookie string will keep a list of name = value pairs separated by semicolons, where name is the name of a cookie and value is its string value.
- You can use strings' split() function to break the string into key and values.

Q21. How to delete a cookie using JavaScript?

If you want to delete a cookie so that subsequent attempts to read the cookie in JavaScript return nothing, you just need to set the expiration date to a time in the past. You should define the cookie path to ensure that you delete the right cookie. Some browsers will not let you delete a cookie if you don't specify the path.