# Q-1 What is JavaScript?

## - JavaScript is a lightweight programming language that web developers commonly

## use to create more dynamic interactions when developing web pages, applications, servers, and or even games.

## 

## - Developers generally use JavaScript alongside HTML and CSS The scripting language

## works well with CSS in formatting HTML elements. However, JavaScript still maintains user interaction, something that CSS cannot do by itself.

## 

## - JavaScript’s implementations within the web, mobile application, and

## game development make the scripting language worth learning.

## 

# Q-2: What is the use of isNaN function?

## - In JavaScript NaN is short for "Not-a-Number".

## - The isNaN() method returns true if a value is NaN.

## - The isNaN() method converts the value to a number before testing it.

## [Examples] :

## - console.log(isNaN(12)); Output : false

## - console.log(isNaN(0 / 0)); Output : true

## - console.log(isNaN(12.3)); Output : false

## - console.log(isNaN("Devanshu")); Output : true

## - console.log(isNaN("25/05/2023")); Output : true

## - console.log(isNaN(-46)); Output : false

## - console.log(isNaN(NaN)); Output : true

## 

## 

## Q-3: What is negative Infinity?

## - The negative infinity in JavaScript is a constant value that is used to represent a value that is the lowest available. This means that no other number is lesser than this value. It can be generated using a self-made function or by an arithmetic operation.

## 

## - JavaScript shows the NEGATIVE\_INFINITY value as -Infinity.

## 

## Example :

## 

## let x = Number.NEGATIVE\_INFINITY;

## Output : -Infinity

## 

## Negative infinity is different from mathematical infinity in the following ways:

## - Negative infinity results in -0(different from 0 ) when divided by

## any other number.

## - When divided by itself or positive infinity, negative infinity return NaN

## Negative infinity, when divided by any positive number (apart from positive infinity) is negative infinity.

## - Negative infinity, divided by any negative number

## (apart from negative infinity) is positive infinity.

## - If we multiply negative infinity with NaN, we will get NaN as a result.

## - The product of 0 and negative infinity is Nan.

## - The product of two negative infinities is always a positive infinity.

## - The product of both positive and negative infinity is always negative infinity.

## 

## 

## 

## Q 4-Which company developed JavaScript?

## - JavaScript was invented by Brendan Eich in 1995.

## 

## - It was developed for Netscape 2, and became the ECMA-262 standard in 1997.

## 

## - After Netscape handed JavaScript over to ECMA,

## the Mozilla foundation continued to develop JavaScript for the Firefox browser.

## 

## 

## 

## 

## Q 5-What are undeclared and undefined variables?

## 

## Undefined:

## - It occurs when a variable has been declared but has not been assigned any value.

## - Undefined is not a keyword.

## 

## Example of Undefined

## :

## - let a;

## console.log(a)

## Console gives Error like 'a' is undefined.

## 

## 

## Undeclared:

## - It occurs when we try to access any variable that is not initialized or

## sdeclared earlier using the var or const keyword.

## - If we use ‘typeof’ operator to get the value of an undeclared variable,

## we will face the runtime error with the return value as “undefined”.

## - The scope of the undeclared variables is always global.

## 

## 

## Example of Undeclared:

## - console.log(a)

## //ReferenceError: a is not declared

## 

## 

## Q-6: Write the code for adding new elements dynamically?

## - Javascript is a very important language when it comes to learning how the browser works. Often there are times we would like to add dynamic elements/content to our web pages. This post deals with all of that.

## 

## - Creation of new element: New elements can be created in JS by using the createElement() method.

## 

## Syntax:

## document.createElement("<tagName>");

## 

## - // Where <tagName> can be any HTML

## - // tagName like div, ul, button, etc.

## 

## // newDiv element has been created

## For Eg: let newDiv = document.createElement("div");

## Q-7 Differences between ViewState and SessionState:

|  |  |
| --- | --- |
| **ViewState** | **SessionState** |
| Maintained at page level only. | Maintained at session level. |
| View state can only be visible from a single page and not multiple pages. | Session state value availability is across all pages available in a user session. |
| It will retain values in the event of a postback operation occurring. | In session state, user data remains in the server. Data is available to user until the browser is closed or there is session expiration. |
| Information is stored on the client’s end only. | Information is stored on the server. |
| used to allow the persistence of page-instance-specific data. | used for the persistence of user-specific data on the server’s end. |
| ViewState values are lost/cleared when new page is loaded. | SessionState can be cleared by programmer or user or in case of timeouts. |

# (Q-8) What is === operator?

- In JavaScript, the === operator is a strict equality comparison operator that returns false

for values that are not of a similar type.

- This operator performs type casting for equality.

- If we compare 2 with “2” using ===, then it will return a false value.

- The == operator is an equality operator that checks whether its two operands are

the same or not by changing expression from one data type to others.

- You can use the == operator in order to compare the identity of two operands even

though they are not of a similar type.

# (Q-9) : How can the style/class of an element be changed?

- We can change, add or remove any CSS property from an HTML element on

the occurrence of any event with the help of JavaScript.

- There are two common approaches that allow us to achieve this task.

- (1) style.property

- (2) Changing the class itself

# (Q-10) : What is a bootstrap card and how would you create one?

### To create a basic Bootstrap card, you need to :

- Add the .card class to a element.

- Inside the element, add another element with the .card-body class.

- Add text inside the inner element.

- To create a more complex card, you can use the following classes:

- .card-title: to add a heading element to the card.

- .card-text: to add text elements to the card.

- .card-header: to create a header inside the card.

- .card-footer: to create a footer inside the card.

- .card-img-top or .card-img-down: to align an image within the card.

## Example :

<div class="card" style="width: 18rem;">

<img src="Module - 11 - WD - Bootstrap Basic & Advanced\profile-logo.jpg" alt="...">

<div class="card-body">

<h5 class="card-title">Card title</h5>

<p class="card-text">Some quick example text to build on the card title and make up the bulk of the card's content.</p>

<a href="#" class="btn btn-danger">Click Me</a>

</div>

</div>