



USN

--	--	--	--	--	--	--	--	--	--

Internal Assessment Test II – August 2023

Sub:	Web Technologies						Sub Code:	22MCA24	
Date:	29/08/2023	Duration:	90 min's	Max Marks:	50	Sem:	II	Branch:	MCA

Note : Answer FIVE FULL Questions, choosing ONE full question from each Module

PART I		MARKS	OBE	
			CO	RBT
1	What is Bootstrap? Explain file structure with a neat diagram. Give an example of basic HTML template using Bootstrap OR	[10]	CO6	L2
2	Discuss Grid systems and Containers of Bootstrap with example.	[10]	CO6	L3
PART II				
3	Explain various types of forms in Bootstrap? Explain their use with code snippet. OR	[10]	CO6	L3
4	How to create a bootstrap table illustrate it with an example. Explain any 3 table classes with example.	[10]	CO6	L3

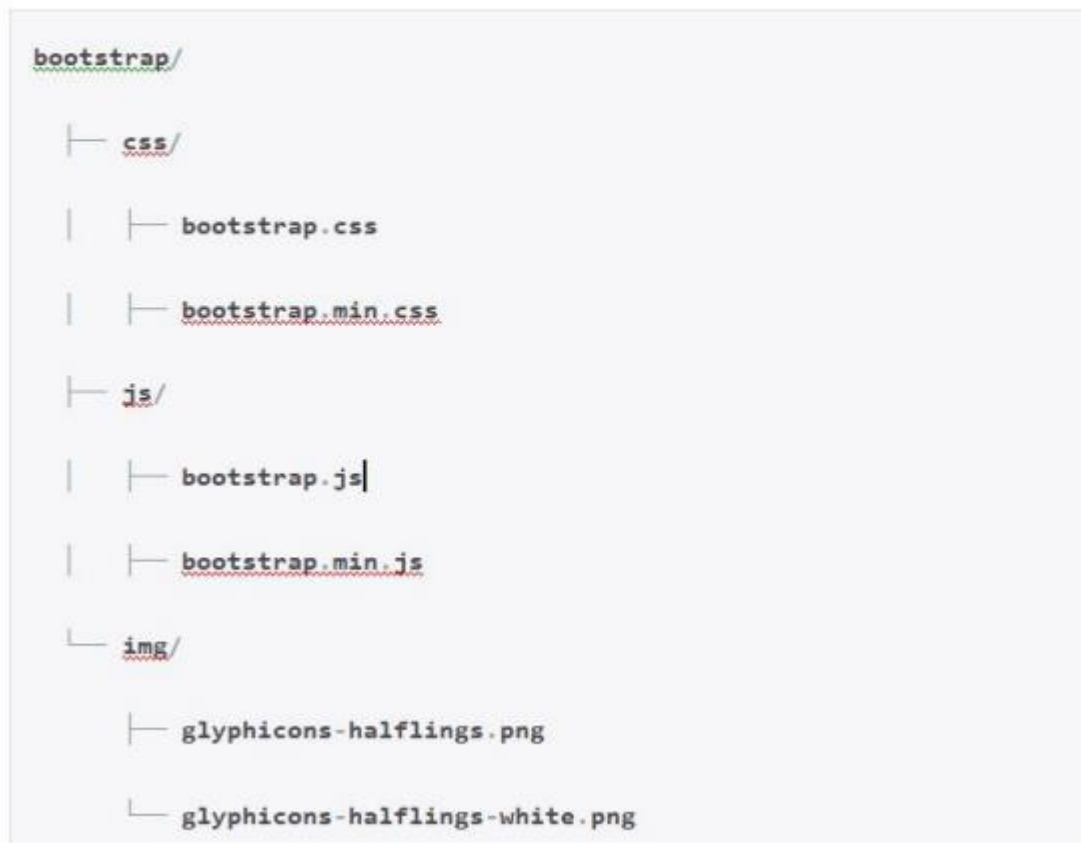
PART III				
5	Explain the following elements in Bootstrap a)Text Colors b)Images c)Progress bars OR	[10]	CO6	L3
6	Explain any three custom forms in Bootstrap	[10]	CO6	L4
PART IV				
7	Explain different ways of creating Array with example. Explain any 5 array functions in JavaScript with example. OR	[10]	CO2,CO6,CO7	L3
8	Explain screen output and keyboard input statements available in javascript	[10]	CO2,CO6,CO7	L3
PARTV				
9	Name the primitive datatypes in java script How are objects different from primitives. Explain. OR	[10]	CO2,CO6,CO7	L4
10	Explain Javascript code to find Median of a set of numbers stored in an array	[10]	CO2,CO6,CO7	L5

1. What is Bootstrap? Explain file structure with a neat diagram. Give an example of basic HTML template using Bootstrap

Bootstrap

- Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation and other interface components.
- Mark Otto and Jacob Thornton
- launched in August 2011.

Bootstrap File Structure bootstrap



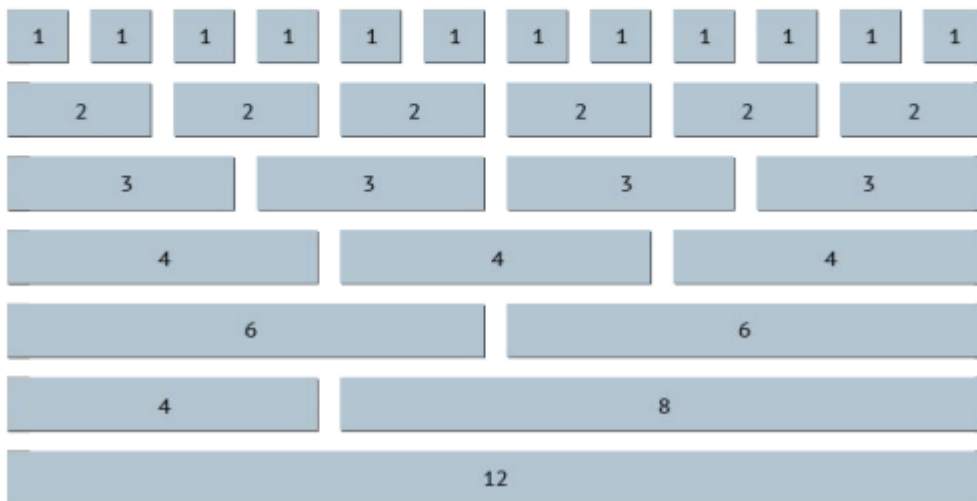
Basic HTML Template

```
<!DOCTYPE html>
<html>
  <head>
    <title>Bootstrap 101 Template</title>
    <link href="css/bootstrap.min.css" rel="stylesheet">
  </head>
  <body>
    <h1>Hello, world! </h1>
    <script src="js/bootstrap.min.js"></script>
  </body>
</html>
```

2. Discuss Grid systems and Containers of Bootstrap with example.

Default Grid System

- The default Bootstrap grid system utilizes 12 columns, making for a 940px-wide container without responsive features enabled.



Basic Grid HTML

```
<div class="row">  
  <div class="span8">...</div>  
  <div class="span4">...</div>  
</div>
```

- **Container Layouts**

```
<div class="container">...</div>  
<div class="container-fluid">...</div>
```

3. Explain various types of forms in Bootstrap? Explain their use with code snippet

Form Layout

Bootstrap provides you with following types of form layouts:

- Vertical (default) form
- Inline form
- Horizontal form

Vertical form (Default)

Username

Password

- Wrap labels and controls in a `<div>` with class `.form-group`. This is needed for optimum spacing.
- Add a class of `.form-control` to all textual `<input>`, `<textarea>`, and `<select>` elements.

INLINE FORM

To create a form where all of the elements are inline, left aligned and labels are alongside, add the class `.form-inline` to the `<form>` tag.

Inline form

Username Password

Horizontal form

Username

Password

- Add a class of `.form-horizontal` to the parent `<form>` element.
- Wrap labels and controls in a `<div>` with class `.form-group`.
- Add a class of `.control-label` to the labels.

4. How to create a bootstrap table illustrate it with an example. Explain any 3 table classes with example.

✓ Bootstrap 5 Tables

▪ Basic Table

A basic Bootstrap 5 table has a light padding and horizontal dividers. The `.table` class adds basic styling to a table:

Firstname	Lastname	Email
John	Doe	john@example.com
Mary	Moe	mary@example.com
July	Dooley	july@example.com

```
ex:
<table class="table">
  <thead>
    <tr>
      <th>Firstname</th>
      <th>Lastname</th>
      <th>Email</th>
    </tr>
  </thead>
```

Optional Table Classes

STRIPED TABLE

By adding the `.table-striped` class, you will get stripes on rows within the `<tbody>` as seen in the following example:

BORDERED TABLE

By adding the `.table-bordered` class, you will get borders surrounding every element and rounded corners around the entire table as seen in the following example:

HOVER TABLE

By adding the `.table-hover` class, a light gray background will be added to rows while the cursor hovers over them, as seen in the following example:

CONDENSED TABLE

By adding the `.table-condensed` class, row padding is cut in half to condense the table. as seen in the following example. This is useful if you want denser information.

Contextual classes

The Contextual classes shown in following table will allow you to change the background color of your table rows or individual cells.

Class	Description
<code>.active</code>	Applies the hover color to a particular row or cell
<code>.success</code>	Indicates a successful or positive action
<code>.warning</code>	Indicates a warning that might need attention
<code>.danger</code>	Indicates a dangerous or potentially negative action

These classes can be applied to `<tr>`, `<td>` or `<th>`.

5. Explain the following elements in Bootstrap

- Text Colors
- Images
- Progress bars

Text Colors

- Bootstrap 5 has some contextual classes that can be used to provide "meaning through colors".
- The classes for text colors are: `.text-muted`, `.text-primary`, `.text-success`, `.text-info`, `.text-warning`, `.text-danger`, `.text-secondary`, `.text-white`, `.text-dark`, `.text-body` (default body color/often black) and `.text-light`:

```
ex:
<div>
<p class="text-muted">This text is muted.</p>
<p class="text-primary">This text is important.</p>
<p class="text-success">This text indicate
success.</p>
<p class="text-info">This text represents some
information.</p>
<p class="text-warning">This text represents a
warning.</p>
```

This text is important.

This text indicates success.

This text represents some information.

This text represents a warning.

This text represents danger.

Secondary text.

Dark grey text.

Body text.

.text-primary

.text-secondary

.text-success

.text-warning

.text-danger

.text-info

.text-muted

Images

- `.img-rounded`: adds `border-radius:6px` to give the image rounded corners.
- `.img-circle`: makes the entire image round by adding `border-radius:500px`.
- `.img-thumbnail`: adds a bit of padding and a gray border:

The following example demonstrates this:

```



```



✓ Bootstrap 5 Progress Bars

A progress bar can be used to show how far a user is in a process.

To create a default progress bar, add a `.progress` class to a container element and add the `.progress-bar` class to its child element. Use the CSS width property to set the width of the progress bar

- The height of the progress bar is 1rem (usually 16px) by default. Use the CSS height property to change it

```
Ex: <div class="progress">
  <div class="progress-bar" style="width:70%"></div>
</div>
```



6. Explain any three custom forms in Bootstrap

Custom Checkbox

To create a custom checkbox, wrap a container element, like `<div>`, with a class of `.custom-control` and `.custom-checkbox` around the checkbox. Then add the `.custom-control-input` to the input with `type="checkbox"`.

Tip: If you use labels for accompanying text, add the `.custom-control-label` class to it. Note that the value of the `for` attribute should match the id of the checkbox:

Custom checkbox Default checkbox

Custom Switch

To create a custom "toggle switch", wrap a container element, like `<div>`, with a class of `.custom-control` and `.custom-switch` around a checkbox. Then add the `.custom-control-input` class to the checkbox:

Toggle me

Custom Radio buttons

To create a custom radio button, wrap a container element, like `<div>`, with a class of `.custom-control` and `.custom-radio` around the radio button. Then add the `.custom-control-input` to the input with `type="radio"`.

Tip: If you use labels for accompanying text, add the `.custom-control-label` class to it. Note that the value of the `for` attribute should match the id of the radio:

Custom radio Default radio

7. Explain different ways of creating Array with example. Explain any 5 array functions in JavaScript with example.

Array OBJECT CREATION

The usual way to create any object is with the `new` operator and a call to a constructor. In the case of arrays, the constructor is named `Array`:

The second way to create an `Array` object is with a literal array value, which is a list of values enclosed in brackets: `var my_list_2 = [1, 2, "three", "four"];`

CHARACTERISTICS OF Array OBJECTS

The lowest index of every JavaScript array is zero. Access to the elements of an array is specified with numeric subscript expressions placed in brackets. The length of an array is the highest subscript to which a value has been assigned, plus 1. For example, if `my_list` is an array with four elements and the following statement is executed, the new length of `my_list` will be 48.

`my_list[47] = 2222;`

The length of an array is both read and write accessible through the `length` property, which is created for every array object by the `Array` constructor.

For example, `my_list.length = 1002;`

An array is lengthened by setting its `length` property to a larger value, shortened by setting its `length` property to a smaller value. The next example, `insert_names.js`, illustrates JavaScript arrays. This script has an array of names, which are in alphabetical order. It uses `prompt` to get new names, one at a time, and inserts them into the existing array. Notice that each new name causes the array to grow by one element.

Array METHODS

Array objects have a collection of useful methods, most of which are described in this section.

- The **join** method converts all of the elements of an array to strings and concatenates them into a single string. If no parameter is provided to `join`, the values in the new string are separated by commas. If a string parameter is provided, it is used as the element separator. Consider the following example:

```
var names = new Array("Mary", "Murray",  
                      "Murphy", "Max");
```

```
...
```

```
var name_string = names.join(" : ");
```

The value of `name_string` is now "Mary : Murray : Murphy : Max".

- The **reverse** method reverses the order of the elements of the Array object through which it is called.

- The **sort** method coerces the elements of the array to become strings if they are not already strings and sorts them alphabetically
- The **concat** method concatenates its actual parameters to the end of the Array object on which it is called.

```
var names = new Array("Mary", "Murray",
                     "Murphy", "Max");
```

```
...
```

```
var new_names = names.concat("Moo", "Meow");
```

The `new_names` array now has length 6, with the elements of `names`, along with "Moo" and "Meow" as its fifth and sixth elements.

- The **slice** method does for arrays what the `substring` method does for strings, returning the part of the `Array` object specified by its parameters, which are used as subscripts. The array returned has the elements of the `Array` object through which it is called, from the first parameter up to, but not including, the second parameter.

```
var list = [2, 4, 6, 8, 10];
```

```
...
```

```
var list2 = list.slice(1, 3);
```

8. Explain screen output and keyboard input statements available in javascript

SCREEN OUTPUT AND KEYBOARD INPUT

- JavaScript models the XHTML document with the `Document` object.
- The window in which the browser displays an XHTML document is modelled with the `Window` object.
- The `Window` object includes two properties, `document` and `window`.
- The `document` property refers to the `Document` object.
- The `window` property is self-referential; it refers to the `Window` object.
- `write` is used to create XHTML code, the only useful punctuation in its parameter is in the form of XHTML tags. Therefore, the parameter of `write` often includes `
`.
- The `writeln` method implicitly adds "\n" to its parameter, but since browsers ignore line breaks when displaying XHTML, it has no effect on the output.
- The parameter of `write` can include any XHTML tags and content.
- The `write` method actually can take any number of parameters.
- Multiple parameters are concatenated and placed in the output.
- Example: `document.write("The result is: ", result, "
");`



- There are 3 types of pop-up boxes:
 - Alert
 - Confirm
 - Prompt
- The `alert` method opens a dialog window and displays its parameter in that window. It also displays an OK button.
- The string parameter of `alert` is not XHTML code; it is plain text. Therefore, the string parameter of `alert` may include `\n` but never should include `
`.

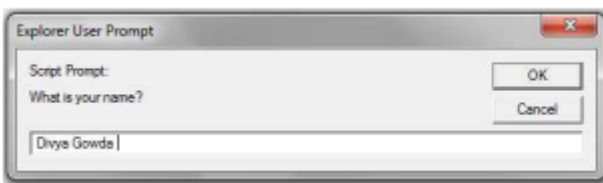
```
alert("The sum is:" + sum + "\n");
```



- The `confirm` method opens a dialog window in which the method displays its string parameter, along with two buttons: OK and Cancel.
 - `confirm` returns a Boolean value that indicates the user's button input: `true` for OK and `false` for Cancel. This method is often used to offer the user the choice of continuing some process.
- ```
var question = confirm("Do you want to continue this download?");
```
- After the user presses one of the buttons in the `confirm` dialog window, the script can test the variable, `question`, and react accordingly.



- The `prompt` method creates a dialog window that contains a text box used to collect a string of input from the user, which `prompt` returns as its value.



9. Name the primitive datatypes in java script How are objects different from primitives. Explain.

## Primitive Types

- JavaScript has five primitive types: Number, String, Boolean, Undefined, and Null.
- Each primitive value has one of these types.
- JavaScript includes predefined objects that are closely related to the Number, String, and Boolean types, named **Number**, **String**, and **Boolean**, respectively.
- These objects are called wrapper objects.
- Each contains a property that stores a value of the corresponding primitive type.
- The purpose of the wrapper objects is to provide properties and methods that are convenient for use with values of the primitive types.
- The difference between primitives and objects is shown in the following example.

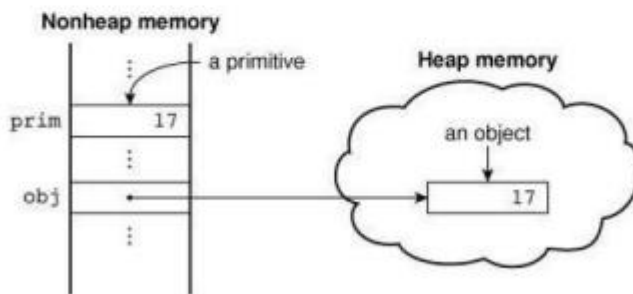


Figure 4.1 Primitives and objects

10. Explain Javascript code to find Median of a set of numbers stored in an array

```
// medians.js
// A function and a function tester
// Illustrates array operations

// Function median
// Parameter: An array of numbers
// Result: The median of the array
// Return value: none
function median(list) {
 list.sort(function (a, b) {return a - b;});
 var list_len = list.length;

 // Use the modulus operator to determine whether
 // the array's length is odd or even
 // Use Math.floor to truncate numbers
 // Use Math.round to round numbers
 if ((list_len % 2) == 1)
 return list[Math.floor(list_len / 2)];
 else
 return Math.round((list[list_len / 2 - 1] +
 list[list_len / 2]) / 2);
} // end of function median
```