Harsh Gupta
1809713039

KIT-401 IT-I (2nd year)

(Section-A)

(Answer-1)

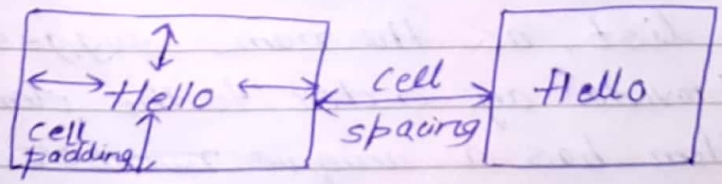
A block element always starts from a new line and provides a default line break after itself. A block element stretches to the 100% width of its parent container eg:-

An inline element however does not force a line break before & after itself. It also does not occupy 100% of available width but instead only expands as much as required eg:-

Answer-2

Cell spacing is used to define the space that must be present in between any two adjacent cells.

Cell padding on the other hand, defines the space between the border of any element and its content.



Answer-3

Hyperlinks in HTML are created using the anchor tags (<a>). Hyperlinks are of two types -
- intrapage link
- interpage link

Intrapage links, link one webpage to another while intrapage link, ~~links~~ use anchors to link one section of the same webpage to another.

eg:- Google
 Go to bottom

⋮

 <a name = "bottom">
 This is bottom

On linking first link, the user will get redirected to google.com while on clicking the second link the user will reach the bottom of the same page.

Answer-4

There are two types of lists in HTML
- ordered list
- unordered list

Order list, as the name suggests are those which provide an order to the elements i.e. each list item has a unique number or character assigned to it.

Unordered list on the other hand assign the same symbol to the element in the list

eg:-
 First
 Second


```
<ul>
    <li> Third </li>
    <li> Fourth </li>
</ul>
```

Output:-

1. First
2. Second
3. Third
4. Fourth

Answer -5

A CSS file can be integrated into a webpage in 3 ways

- Inline : The style attribute is used in the tag whose style has to be changed.
  eg:- `<body style = "background-color : red;">`

- External : The styles are written in a file with .css extension and imported into the html file using <link> tag.
  i.e. `<link rel = "stylesheet" type = "text/css" href = "style.css">`

- Internal : This type CSS is defined using <style> tag in the head section
  i.e.

```
<html> <head>
    <style>
    ----
    </style> </head>
    <body> </body>
    </html>
```

Answer - 6

A pseudo class is used to define a special state of an element. This is generally used to define style in response to user interaction with the element. For example.

- to change style on hover
- to change style on focus

The syntax of pseudo-class in:

```
selector: pseudo-class {
        property: value;
}
```

Answer - 7

CSS selectors are used to "find" (or select) the HTML elements you want to style.

We can divide CSS selectors into five categories

- Simple selector: selects elements based on name, id, class
- Combinator selector: selects elements based on a specific relationship
- Pseudo-class selectors: select elements based on a certain state
- Pseudo-elements selector: select and style a part of an element
- Attribute selectors: select elements based on an attribute or attribute value

Harsh Gupta
1802713039

Answer-8

Javascript, often abbreviated as JS, is a programming language that conforms to the ECMAScript specifications. Javascript is high-level, often just-in-time compiled and multi-paradigm.

The major benefits of JS are:

- It is fast
- It can be mixed easily with HTML
- Interpreted
- Loosely typed
- Object-based
- Event-driven.

Answer-9

The data type in Javascript are:-

- Null: A null value is one that has no value and means nothing. let name = null

- Numbers: Numbers are further divided into two types i.e. integer & floating point. Integers can be expressed in decimal (base 10), hexadecimal (base 16) and octal (base 8)
  let num=8, num=2.84.

- String: Strings contain alphanumeric data that are delincated by single or double quotation marks.

- Boolean: These can contain only true or false

- Object: const new_obj = new Object();

- Undefined: A value that is undefined is a value held by a variable after it has been created but before a value has been assigned to it.

Answer - 10

There are three types of pop-up boxes available in JavaScript:-

- Alert Box:- This is simple pop-up box which is used to display some text and has only one button (OK)
  i.e. alert("I am Harsh");

- Prompt Box:- prompt() is used to take input from the user as two buttons, Ok and cancel. Ok returns the input value while, cancels returns null.

- Confirm Box:- Confirm boxes are used to get a boolean value from user. It has the same two buttons as prompt - Ok & cancel, Ok returns true while cancel returns false.

(Section - B)

Answer - 12

```
<html>
<body>
<table border = 1>
```

Harsh Gupta
1802713039

PAGE No. 7
DATE : / /

```
<tr>
    <th> Roll Number </th>
    <th> Name </th>
    <th> Marks </th>
</tr>
<tr>
    <td> 181 </td>
    <td> Amit </td>
    <td> 92 </td>
</tr>
<tr>
    <td> 192 </td>
    <td> Himani </td>
    <td> 100 </td>
</tr>
<tr>
    <td> 193 </td>
    <td> Deepak </td>
    <td> 72 </td>
</tr>
<tr>
    <td> 194 </td>
    <td> Ishika </td>
    <td> 98 </td>
</tr>
<tr>
    <td> 200 </td>
    <td> Sparsh </td>
    <td> 34 </td>
</tr>
<tr>
    <td> 201 </td>
    <td> Deepanshu </td>
    <td> 05 </td>
</tr>
```

```
</table>
</body>
</html>
```

Answer - 14

a)
```
a:active , a:hover {
        background-color : yellow;
    }
```

b)
```
ul {
        list-style = square;
    }
```

c)
```
body {
        background = url("pago.gif");
    }
```

d)
```
body {
        border-style : dashed;
    }
```

Answer - 11

Frames are used to divide the browser window into multiple sections where each section can be used to open a separate html document. The window is divided into frames in the some way as tables i.e. in the ~~frame~~ form of rows & columns.

Harsh Gupta
1802713039

PAGE No. 9
DATE : / 201

main html

```
<html>
<frameset rows =
<frame src = "a.html">
<frame src = "b.html">
<frame src = "c.html">
</frameset>
</html>
```

a.html

```
<html>
<body>
<h1> Frame 1 </h1>
</body>
</html>
```

b.html

```
<html>
<body>
<h1> Frame 2 </h1>
</body>
</html>
```

c.html

```
<html>
<body>
<h1> Frame 3 </h1>
</body>
</html>
```

Output :-

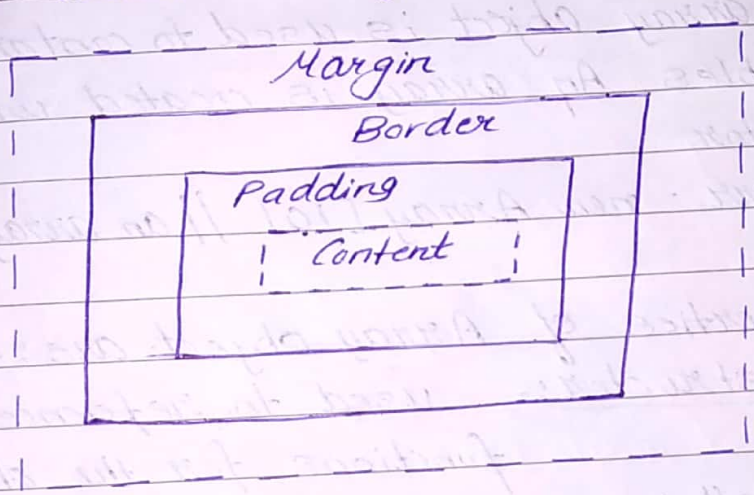
Frame 1
Frame 2
Frame 3

Harsh Gupta
1802713039

PAGE No 10
DATE : / 201

Answer - 813

All HTML elements can be considered as boxes. In CSS, the term 'box model' is used when talking about design & layout.
The CSS box model is essentially a box that wraps around every HTML element.




```
<html>
<head>
<style>
h1, h2 {
        padding : 0.5 em
        border style : groove;
        margin : 0.5 em;
      }
</style>
</head>
<body>
<h1> Hello </h1>
<h2> World </h2>
</body>
</ html >
```

Harsh Gupta
1802713039

Answer - 15

There are two build-in objects in JS :-

1) Array object
2) Date object

Array object:

The array object is used to contain an array of variables. An array is created with the Array Constructor

```
let arr = new Array [10] // an array of 10 elements
```

The properties of Array object are :-

(i) Constructor :- used to reference the constructor functions for the objects.

(ii) Length :- returns the numbers of elements in array.

The methods of Array object are :-

(1) Concat (2 arrays) - concatenates two arrays and returns the array thus formed

(2) join - joins all elements of arrays into a single string separated by specific delimeter.

(3) reverse() reverse the order of elements in the array

(4) sort() - sort the elements of array in ascending order.

5) to string - converts array into string with elements separated by commas (,)

Date object:

The Data object handles everything to do with date and time in JS. A Date object can be created in 4 ways using the constructor.

new Date()
new Date (milliseconds)

There is only one property of the date object

- constructor - used to reference the constructor functions for the object.

The various methods of Data objects are:

1) getDate() - retrieves the date in the month.
2) getDay() - retrieves the day of the week
3) getFullyear() - retrieves the year as a four digit number
4) getHours() - retrieves the hour of the day.
5) getMinutes() - retrieves the number of minutes.
6) getMonth() - retrieves the month.
7) getseconds() - retrieves the number of seconds.
8) getTime() - retrieves the number of milliseconds since January 1 1970 00:00:00
9) getyear() - retrieves the year as two digit number
10) setDate(date) - sets the date in the month to 'date'

Harsh Gupta
18027I3039


```
<html>
<head>
<script>
let x = parseInt (prompt("Enter a number");
let flag = 0;
for (let i=2; i<x ; i++)
{
    if (x%i == 0) {
      flag= 1;
  }
}
if (flag == 0)
{
   document.write (x+" is a prime number");
}
else
{
   document.write (x+" is not a prime number");
}
</script>
</head>
</html>
```

Harsh Gupta
1802713039 IT-1

KIT-401

(Section - C)

Answer - 20 (b)

There are many operators in Javascript, some of them are:

1) Arithmetic operators:
Arithmetic operators are used to perform arithmetic operations.

@ Addition: (+), This operator adds numbers

```
eg:- var x = 10;
     var y = 15;
     var z = x + y;
```

(b) subtraction: (-) This operator subtracts numbers

```
eg:- var x=10;
     var y=15;
     var z = y-x;
```

© Multiplication: (*) This operator multiplies numbers

```
eg:- var x = 10;
     var y = 15;
     var z = 10 * 15;
```

(d) Division: (/) This operator divides numbers

```
eg:- var x = 10;
     var y = 15;
     var z = ~~10/15~~ x/y;
```

Harsh Gupta
1802713039

PAGE No 15
DATE : / 201

(e) exponential : (**) This operators does the function of exponent
eg:- var z = 10**15;

(f) Modulus : (%) This operator gives the remainder
eg:- var z = 10%15;

(g) Increment : (++) This operator increment the value by 1.
eg:- 10++;

(h) Decrement : (--) This operator decrements the value by 1
eg:- 10--;

2) Assignment operators :
(a) Assignment operators assign values to Javascript variables.

(a) = : This operator assign the value to left operant ~~of~~ right operant.
eg:- var z = 10;

(b) additional assignment : (+=) This operator adds the values and assign them to the left operant.
eg:- var z = 10;
z += 5;

(c) subtraction assignment : (-=) This operator subtracts the values and assign them to left operant.
eg:- var z = 10; z -= 5;

(d) multiply assignment :- (*=) This operator multiplies the values and assign them to the left operand

eg:- z = 10;
z *= 5;

(e) division assignment :- (/=) This operator divides the values and assign them to the left operand

eg:- var z = 10;
z /= 5;

(f) modulus assignment : (%=) This operator takes the remainder and assign them to the left operand

eg:- var z = 10;
z %= 5;

(g) exponent assignment : (**=) This operator takes the exponent value and assign it to the left operand.

eg:- var z = 2;
z **= 2;

3) Comparison operators:
These operator are used to compare two operands, variables etc.

(a) equal to :- (==) This operator checks for the equality

10 == 10 (true)
10 == 5 (false)

(b) equal value and equal type : (===) This operator checks for equal value as well as equal types.

eg:- 5===5 (true)
5===5.1 (false)

(c) not equal : (!=) This operator check for not equality.

eg:- 5!=5 (false)
5!=10 (true)

(d) not equal value or not equal type : (!==) This operator check for different values and different types.

eg:- 5!==5 (false)
5!==5.1 (true)

(e) greater than : (>) This operator checks that is the left operand is greater or not

eg: 10>5 (true)
5>10 (false)

(f) less than : (<) This operator checks that is right operand is greater or not

eg:- 10<5 (false)
5<10 (true)

(g) greater than equal to : (>=) This operator, checks that is the left operand is greater or equal.

eg:- 10>=5 (true)
5>=5 (true)

(h) less than or equal to: (<=) This operator checks that is the right operand is greater or equal

eg:- 10<=5 (false)
5<=5 (true)

4) Logical operators:
These operators performs all the logical operations.

@ Logical and: (&&) This gives true value only when both the ~~operators~~ operands are true

true && true (true)
false && true (false)

(b) Logical or: (||) This gives true when any one of the two operands are true or both are true

true || false (true)
false || false (false)

(c) Logical not: (!) This gives the reverse value.

eg:- !true = false
!false = true

5) Bitwise operators:
Operators which works on the bits are called bitwise operators.

@ AND: (&) This operator gives one only when both the operands are 1

eg:- 1&1 = 1
0&1 = 0

(b) OR: (|) This operator gives 1 when either two operands are 1 or only one operand is 1.

eg:- 1|0 = 1

0|0 = 0

(c) NOT: (~) This operator gives the reverse value

eg: ~1 = 0

~0 = 1

(d) XOR: (^) This operator gives 0 when both operands are same & gives 1 when both operands are different

0^0 = 0

1^0 = 1

(e) Left-shift operator: (<<) This operator shifts the bits to left by the given number.

eg:- 0101<<1 = 1010

(f) Right-shift operator: (>>) This operator shifts the bits to right by the given number.

eg:- 0101>>1 = 0010

6) Type operators:

(a) typeof: Returns the type of a variable

(b) instanceof: Returns true if an object is an instance of an object type.

Harsh Gupta
1802713039

Answer-16 @

<DIV>	<SPAN>
(i) The tag is a block level element	The tag is an inline element.
(ii) It is best to attach it to a section of a web page	It is best to attach a CSS to a small section of a line in a web page.
(iii) It accepts align attribute	It does not accepts align attributes.
(iv) The tag should be used to wrap a section, for highlighting that section	This tag should be used to wrap any specific word that you want to highlight in your webpage.
(v) The div tag is used in HTML to make divisions of content on the webpage.	It is used to group elements for styling purposes.

Font properties:

property:	Uses:
(i) font-family:	This property can help us to use different-different fonts.
(ii) font-weight:	It helps us to control the boldness of the font.
(iii) font-size:	It helps us to control the font size
(iv) font-variant:	specifies the font variant.
(v) font-style:	specifies the font style.

Harsh Gupta
1802713039

Text properties:

property:	Uses:
• text-color:	allow us to use different colours
• text-align:	used to set the horizontal alignment
• text-decoration:	used to add or remove decoration
• text-transformation:	used to change the cases of text
• text-indentation:	used to indent the first line of paragraph

Answer-17 @

Absolute positioning means that the element is taken out of the normal flow of the page. The absolutely positioned element does not exist for the other elements on the page. The element is drawn on top of everything else at positioning specified by the top, bottom, left, right attributes.

Relative positioning on the other hand is just like starting no positioning at all, but the left, right, top, bottom attributes nudge the elements out of the page still get laid out as if the element was on its normal spot.

for example
```
<span> Span1 </span>
<span> Span2 </span>
<span> Span3 </span>
```

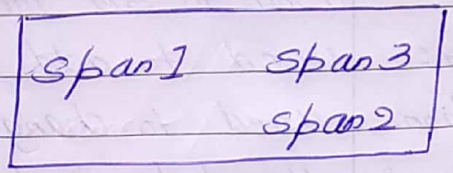
output

Span1 Span2 Span3

Harsh Gupta
1802713039

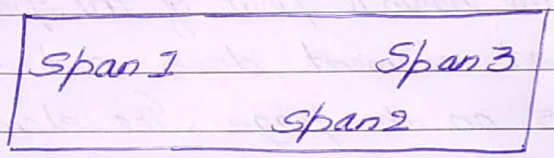

```
<span> span1 </span>
<span style = "position: absolute, top: 10px;
    left: 20px;"> span2 </span>
<span> span3 </span>
```

Output:




```
<span> Span1 </span>
<span style = "position: relative, top: 10px,
    left: 20px;"> Span2 </span>
<span> Span3 </span>
```

Output



The z-index property defines the position of elements in stack. Element having higher value of z-index will always be on top of elements having lower value of z-index. The default value of z-index of all elements is 1.

z-index property only works on positioned elements i.e. on those elements whose position property has been defined to absolute, relative, fixed or sticky.

Answer - 18(b)

```
<html>
<head>
<script>
```

```
const a = parseInt(prompt("Enter the value of a"));
const b=parseInt(prompt("Enter the value of b"));
const c=parseInt(prompt("Enter the value of c"));
const x = b**2 - 4*a*c;
if(x<0)
{
    alert("No real root exist");
}
else
{
    const d = Math.sqrt(x);
    const r1 = (-b+d)/(2*a);
    const r2 = (-b+d)/(2*a);
    alert("The roots are : "+r1+", "+r2);
    confirm("Do you want to continue?");
}
</script>
</head>
</html>
```

Answer -19(b)

```
<html>
<head>
<style>
p{
    border: 1px solid black;
}
table{
    border: 1px groove red;
}
</style>
```

Harsh Gupta
1802713039


```
</head>
<body>
<p> Hello world </p>
<table>
<tr>
    <th> S No </th>
    <th> Name </th>
</tr>
<tr>
    <td> 1 </td>
    <td> Harsh </td>
</tr>
</table>
</body>
</html>
```