

3

UNIT

Scripting and Networking

CONTENTS

- Part-1** : Scripting : Javascript : 3-2D to 3-10D
Introduction,
Documents, Forms
- Part-2** : Statements, Functions, 3-10D to 3-18D
Objects
- Part-3** : Introduction to AJAX 3-18D to 3-23D
- Part-4** : Networking : Internet 3-23D to 3-27D
Addressing, InetAddress,
Factory Methods,
Instance Methods
- Part-5** : TCP/IP Client Sockets, 3-27D to 3-32D
URL, URL Connection,
TCP/IP Server
Sockets, Datagram

PART-1*Scripting : JavaScript : Introduction, Documents, Forms.***Questions-Answers****Long Answer Type and Medium Answer Type Questions**

Que 3.1. What is JavaScript ? How it works ? What are the features of JavaScript ?

Answer

1. JavaScript is a scripting language which enables web authors to design interactive sites.
2. JavaScript can interact with HTML source code, enabling web authors to build their sites with dynamic content.
3. JavaScript is an open source language that anyone can use without purchasing a license.

Working of JavaScript :

1. When the browser loads a web page, the HTML parser creates the DOM.
2. Whenever parser encounters JavaScript directive, it is handed over the JavaScript engine and loads the external and inline code.
3. After HTML and CSS parsing is completed, JavaScript is executed in order they were found in web page and DOM is updated and rendered by the browser.

JavaScript has several features :

1. **Programming tool** : JavaScript is a scripting language with very simple syntax.
2. **Can produce dynamic text into an HTML page** : For instance, the JavaScript statement - `document.write("<h1>" + name + "</h1>");` results into the HTML output `<h1> Atul </h1>`, if the variable name contains the text Atul.
3. **Reaching to events** : JavaScript code executes when something happens such as when page has finished loading, when a user clicks on an HTML element.
4. **Reading and writing HTML elements** : JavaScript can read and change the content of an HTML element.
5. **Validate data** : JavaScript can be used to validate form data before it is submitted to a server, and thus saves the server from extra processing.

Que 3.2. What is the difference between Java and JavaScript ?

Describe the strengths and weakness of JavaScript.

OR

Compare and contrast Java and JavaScript.

AKTU 2016-17, Marks 10

Answer

Difference between Java and JavaScript :

S.No.	Java	JavaScript
1.	Java is an object-oriented programming language.	JavaScript is an object based scripting language.
2.	Java is strongly typed language and type checking.	JavaScript is very flexible in data type.
3.	Objects in Java are static.	Objects in JavaScript are dynamic.
4.	It can be used to create standalone application.	It cannot be used to create standalone application.
5.	Variables in Java are declared as : int num.	Variables in JavaScript are declared as : var myname.

Strengths/Advantages of JavaScript :

- An interpreted language :** JavaScript is an interpreted language, which requires no compilation steps.
- Quick development :** JavaScript does not require time consuming compilations, scripts can be developed in a short period of time.
- Performance :**
 - JavaScript can be written such that the HTML files are fairly compact and quite small.
 - It minimizes storage requirements on the web server and download time for the client.
- Easy debugging and testing :** Being an interpreted language, scripts in JavaScript are tested line by line and the errors are also listed as they are encountered.

Weakness/Disadvantages of JavaScript :

- There are issues of incompatibility of several scripting that result in website overloading.
- Different layout engines may render JavaScript differently resulting in inconsistency in terms of functionality and interface.
- JavaScript is also a common tool for the web hackers; they use scripts

injection to hack a site.

4. JavaScript is light, somehow too much of JavaScript can slow down the page loading of a website.

Que 3.3. Write short notes on JavaScript DOM.

Answer

1. A document object represents the HTML document that is displayed in the window.
2. DOM is an object oriented representation of an HTML document and acts as an interface between JavaScript and the document itself and allows the creation of dynamic web pages.
3. The hierarchical structure of object is applied for the organization of objects in a web document which include following object :
 - a. **Window object** : It is the top most element of the object hierarchy.
 - b. **Document object** : Each HTML document that gets loaded into a window becomes a document object which contains the contents of the page.
 - c. **Form object** : Everything enclosed in the `<form>...</form>` tags sets the form object. The form object contains all the elements defined for that object such as text fields, buttons, radio buttons, and checkboxes.
4. When a web page is loaded, the browser creates a Document Object Model of the page.
5. DOM supports navigation in any direction (*i.e.*, parent and previous sibling) and allows for arbitrary modifications.

Que 3.4. What is the difference between Java and JavaScript ?

Write a JavaScript function for e-mail address validation, that is to check if the content has the general syntax of an e-mail or not.

AKTU 2015-16, Marks 10

Answer

Difference between Java and JavaScript : Refer Q. 3.2, Page 3-3D, Unit-3.

JavaScript function for e-mail address validation :

```
<script type = "text/javascript">
```

```
function validateEmail()
```

```
{
```

```
    var i;
```

```
    var str=document.my_form.Email_txt.value;
```

```
var index_at=str.indexOf("@");
var len=str.length;
var index_dot=str.indexOf(".");
var emailID=document.my_form.Email_txt;
if((emailID.value==null) || (emailID.value == " "));
{
    alert("Please Enter your Email ID");
    emailID.focus();
    return false;
}
if (str.indexOf(".") == -1 || str.indexOf(".") == 0 ||
str.indexOf("@") == -1 || str.indexOf(".") == index_at
|| str.indexOf("@", (index_at+1)) != -1 ||
|| str.indexOf(" ") != -1 ||
{
    alert("Invalid Email ID");
    return false;
}
return true;
}
```

</script>

Que 3.5. How do you perform client-side validation using JavaScript? Illustrate with suitable example.

Answer

Following are the steps used to perform client-side validation using JavaScript :

1. First the user will enter the value in the form field.
2. Then, browser will ensure that the value provided by user is correct and is valid so that successful validation can be done.
3. JavaScript used in the web page uniquely defines all the special functionalities in the client browser.
4. By default, if there is a validation error then an error or pop-up message is shown by the browser.
5. If there is no error then the validation on client-side will be successfully

performed.

For example :

If a form field (fname) is empty, validateform function alerts a message, and returns false, to prevent the form from being submitted:

```
function validateForm() {  
    var x = document.forms["myForm"]["fname"].value;  
    if (x == " ") {  
        alert("Name must be filled out");  
        return false;  
    }  
}
```

The JavaScript function is called when the form is submitted :

```
<form name="myForm" action="/action_page.php" onsubmit="return  
validateForm()" method="post">  
Name: <input type="text" name="fname">  
<input type="submit" value="Submit">  
</form>
```

Que 3.6. What are scripting languages and why JavaScript is used ? Write a JavaScript function for validating form data like mandatory fields and email field.

AKTU 2017-18, Marks 05

Answer**Scripting language :**

1. A scripting language is a programming language designed for integrating and communicating with other programming languages.
2. Some of the most widely used scripting languages are JavaScript, VBScript, PHP, Perl, Python, Ruby, ASP.

JavaScript is used because :

- a. It is executed on client side.
- b. It saves bandwidth on web server.
- c. It is written into an HTML page.

JavaScript function for validating form data :

```
<script type="text/javascript">  
function validateform()  
{  
    var name = document.myform.name.value;  
    var password = document.myform.password.value;  
    var confirmpassword = document.myform.password2.value;
```



```
var email = document.myform.email.value;
# Username validation
if (name == null || name == ""){
    alert("Name can't be blank");
    return false;
}else if(password.length<6){
    alert("Password must be at least 6 characters long.");
    return false;
}
}
# Retype password validation
if (password == confirmpassword){
    return true;
}
else{
    alert("password must be same!");
    return false;
}
}
# Email validation
var emailErr = True;
if(email == "") {
    printError("emailErr", "Please enter your email address");
} else {
    // Regular expression for basic email validation
    var regex = /^[^\\S+@\\S+\\.\\S+$/;
    if(regex.test(email) === false) {
        printError("emailErr", "Please enter a valid email address");
    } else{
        printError("emailErr", "");
        emailErr = false;
    }
}
</script>
<body>
<form name="myform" method="post" onsubmit="return validateform()">
Username: <input type="text" name="name"><br/>
```

```
Password : <input type="password" name="password" /><br/>
Re-enter Password: <input type="password" name="password2"/><br/>
Email : <input type="text" name="email">
<div class="error" id="emailErr"></div>
<input type="submit" value="Register">
</form>
</body>
</html>
```

Que 3.7. Compare Java and JavaScript. Write a JavaScript program to define a user defined function for sorting the values in an array.

AKTU 2018-19, Marks 07

Answer

Comparison : Refer Q. 3.2, Page 3–3D, Unit-3.

Program to sort value using JavaScript :

```
<!DOCTYPE html>
<html><body>
<h2>JavaScript Array Sort</h2>
<p>The sort() method sorts an array alphabetically.</p>
<button onclick="myFunction()">Try it</button>
<p id="demo"></p>
<script>
var fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo").innerHTML = fruits;
function myFunction() {
fruits.sort();
document.getElementById("demo").innerHTML = fruits;
}
</script></body></html>
```

Que 3.8. Create an html page named as "String_Math.html" and within the script tag define some string variables and use different string function to demonstrate the use of the predefined functions.

Do the same for the math function.

AKTU 2018-19, Marks 07

Answer

```
<html><body>
```

```
<script type="text/javascript">
var str1="W3Schools is great!"
document.write(str1.substr(2,6))
document.write("<br><br>")
document.write(str1.substring(2,6))
// string length
var str2="Web Enabling Tools is Cool!"
document.write("<p>" + str2 + "</p>")
document.write("str.length")
// Square root of a number
var property_value = Math.SQRT1_2
document.write("Property Value is : " + property_value);
var value = Math.exp(1);
document.write("First Test Value : " + value );
var value = Math.exp(30);
document.write("<br />Second Test Value : " + value );
var value = Math.exp(- 1);
document.write("<br />Third Test Value : " + value );
// Find maximum of a number
var value = Math.max(10, 20, - 1, 100);
document.write("First Test Value : " + value );
var value = Math.max(- 1, - 3, - 40);
document.write("<br />Second Test Value : " + value );
</script>
```

Que 3.9. Explain the role of JavaScript to develop a web page.

Write a JavaScript function to check a textbox is either empty or not.

Answer

Role of JavaScript :

1. It makes our website dynamic.
2. It makes our webpage interactive which means that it can respond to mouse clicks, double clicks, hover and many other actions (called events).

3. It can also be used for modifying our html, content and styles, form validation.
4. JavaScript can interact with the website server to send and receive information to update UI in real-time.

JavaScript function to check whether a textbox is either empty or not :

```
<html>
<script>

function required(inputtx)
{
if (inputtext1.value.length == 0)
{
alert("Textbox is empty");
return false;
}
return true;
}

</script>
<body>
<form>
<input type = "text" name="text1">
</form>
</body></html>
```

PART-2

Statements, Functions, Objects.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 3.10. Explain conditional statements used in JavaScript with example.

Answer

There are following conditional statement used in JavaScript :

1. **If statement :** If statement is used if we want to execute some code only if a specified condition is true.

Syntax :

```
if (condition)
{
```

code to be executed if condition is true

```
}
```

For example :

```
<script type = "text/javascript">
var d = new Date();
var time = d.getHours();
if (time<10)
{
document.write ("<b>Good morning to all</b>");
}
</script>
```

2. **If...else statement :** If...else statement is used when we do not confirm about the condition that is true or not.

Syntax :

```
if (condition)
{
```

code to be executed if condition is true

```
}
```

else

```
{
```

code to be executed if condition is not true

```
}
```

For example :

```
<script type = "text/javascript">
//If the time is less than 10, print "Good Day" Otherwise "Good Night"
var d = new Date ();
var time = d.getHours ();
if (time < 10)
{
document.write ("Good Day");
}
else
{
document.write ("Good Night");
}
```

`</script>`

3. **If...else if...else statement :** We can use the if...else if...else statement if we want to select one from many sets of lines with different condition.

Syntax :

```
if (condition1)
{
    code to be executed if condition1 is true
}
else if (condition2)
{
    code to be executed if condition2 is true
}
else
{
    code to be executed if condition1 and condition2 are not true
}
```

For example :

```
<script type = "text/javascript">
var d = new Date();
var time = d.getHours();
if (time<10)
{
    document.write("<b>Good morning</b>");
}
else if (time>10 && time<16)
{
    document.write("<b>Good afternoon</b>");
}
else
{
    document.write ("<b>Good night</b>");
}
</script>
```

4. **Switch statement :** If we want to select one of many blocks of code then we use switch statement.

Syntax :

```
switch(n)
{
case 1 :
execute code block 1
break ;
case 2 :
execute code block 2
break ;
default ;
code to be executed if n is different from case 1 and 2
}
```

For example :

```
<script type = "text/javascript">
var d=new Date();
theDay=d.getDay();
switch(theDay)
{
case 1 :
document.write("Finally Monday");
break ;
case 2 :
document.write("Super Tuesday");
break ;
case 0 :
document.write("Sleepy Sunday");
break ;
default :
document.write("I'm looking forward to this weekend!");
}
</script>
```

Answer

Data types in JavaScript : JavaScript provides different data types to hold different types of values.

There are two types of data types in JavaScript :

1. **Primitive data type :** There are five types of primitive data types in JavaScript. They are as follows :

Data type	Description
String	It represents sequence of characters.
Number	It represents numeric.
Boolean	It represents boolean value either false or true
Undefined	It represents undefined value
Null	It represents null <i>i.e.</i> , no value at all

2. **Non-primitive (reference) data type :** The non-primitive data types are as follows :

Data Type	Description
Object	It represents instance through which we can access members
Array	It represents group of similar values
RegExp	It represents regular expression

Function in JavaScript :

1. Functions can be defined both in the <head> and in the <body> section of a document.
2. However, to assure that the function is read/loaded by the browser before it is called, it is needed to be defined in the <head> section.
3. **Syntax :**

```
function function_name (var1, var2, ... , varX)
{
    some code
}
```

var1, var2, etc., are variables or values passed into the function.

4. A function with no parameters must include the parentheses () after the function name :

```
function function_name ()
{
```



```
some code  
}
```

Que 3.12. Compare Java and JavaScript. Explain and demonstrate 5 different types of objects in JavaScript with example.

AKTU 2019-20, Marks 07

Answer

Comparison : Refer Q. 3.2, Page 3–3D, Unit-3.

Different object in JavaScript :

1. **Built-in objects :** These objects are used quite extensively for data processing in JavaScript. Following are some built-in object :

a. String object :

- The string object enables programs to work with and manipulate string.
- It provide methods such as : `big()`, `blink()`, `bold()`, `italics()`, `charAt()`, `toUpperCase()`, `toLowerCase()` and `substring()`.

b. Math object :

- The math object provides some commonly used methods such as : `sqrt(num)`, `abs(num)`, `sin(num)`, `cos(num)`, `tan(num)`, `exp(num)`, `min(a, b)`, `max(a, b)`, `log(num)`, `pow(a, b)`, `floor(num)`, `ceil(num)` etc.

c. Date object :

- The date object enables JavaScript programmers to create an object that contains information about a particular date and provides a set of methods to work with that information.
- Syntax :**
`var mydate = new Date(<parameters>);`
- If the parameter left empty, it indicates current date and time.
- The date object provides some methods which are : `getDate()`, `setDate()`, `getHours()`, `setHours()`, `getTime()`, `setTime()`, `getDay()`, `setDay()`, `getMinutes()`, `setMinutes()`, `getSecond()`, `setSecond()`.

d. Array object :

- The array object stores multiple values in a single variable.
- Syntax :**
`var fruits = new Array("apple", "orange", "mango");`

2. **User-defined objects :**

- A user-defined object is also associated with properties and methods,

which belong to it.

- b. The user-defined object would also require methods that will allow the storage of name, age and salary of the employee object.
- ```
function Employee(name, age, salary)
{
 this.name = name;
 this.age = age;
 this.salary = salary;
}
```
- c. In the given example, object Employee has three properties : name, age and salary. Here, this keyword refers to the current object in focus. In given example, this.name will refer to the name of the current object.

**For example :**

```
<html>
<head>
<title>JavaScript Object </title>
</head>
<body>
<script type="text/javascript">
var value = Math.acos(-1);
document.write("ACOS Test Value : " + value + "
");
// Math object
var d = new Date();
document.write("Locale String: " +
d.toLocaleString()+"
");
// Date Object
var str = "CareerRide Info"; //String Object
var s = str.split();
document.write("Char At: " + str.charAt(1)+"
");
document.write("CharCode At: " + str.charCodeAt(2)+"
");
document.write("Index of: " + str.indexOf("ide")+"
");
</script>
</body>
</html>
```

**Output:**

```
ACOS Test Value : 3.141592653589793
Locale String: 22/02/2020 11:14:06 AM
Char At: a
```

CharCode At: 114

Index of: 7

**Que 3.13. How do you associate functions with objects using JavaScript ?**

**Answer**

1. In JavaScript we can associate functions with objects by creating objects and assigning properties.
2. Every object in JavaScript has different property.
3. Property of an object is defined in a variable which is attached to an object.
4. Object properties are mostly the same as simple JavaScript variables.
5. JavaScript has a number of predefined objects. So, we can create our own objects and embed the function directly in the object.

**For example :**

```
var myCar = new Object();
myCar.make = "Nissan";
myCar.model = "N-54";
myCar.year = 1972;
<script type="text/javascript">
function user(name, email) {
this.name = name;
this.email = email;
//Custom method for object this.toString
= function userToString() {
return("Name: "+this.name+" Email: "+this.email);
}
}
var obj = new user("Aditya","aditya123@example.com");
document.write(obj.toString());
```

```
</script>
```

**Que 3.14. How do we handle event in JavaScript ? What is DHTML ?**

**Answer**

**Event handling in JavaScript :**

1. JavaScript allows us to change the default action associated with events. For example, event processing where a user clicks a hyperlink on a page.

Fig. 3.14.1 depicts how events occur and are handled in JavaScript.

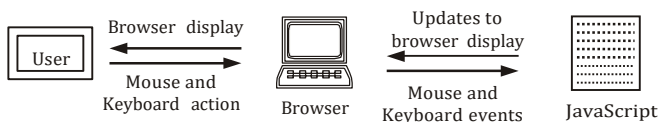


Fig. 3.14.1

event handlers

2. Clicking on the hyperlink generates an event and the default associated action, then browser loads and displays the page associated with that URL.
3. In JavaScript, we can change the default action and write our own event handler that will be associated with the hyperlink.
4. The following are a few other tasks that we can do with events using JavaScript event handlers :
  - a. Validate the data entered by a user in a form.
  - b. Shift the focus of controls from one field to another in a form.
  - c. Load and display animation when a user clicks a button.
  - d. Communicate with Java applets and browser plug-ins.
  - e. Display a dialog box when a user moves the mouse over a link.

DHTML : Refer Q. 2.34, Page 2-39D, Unit-2.

## PART-3

### Introduction to AJAX.

### Questions-Answers

#### Long Answer Type and Medium Answer Type Questions

Que 3.15. Write a short note on AJAX.

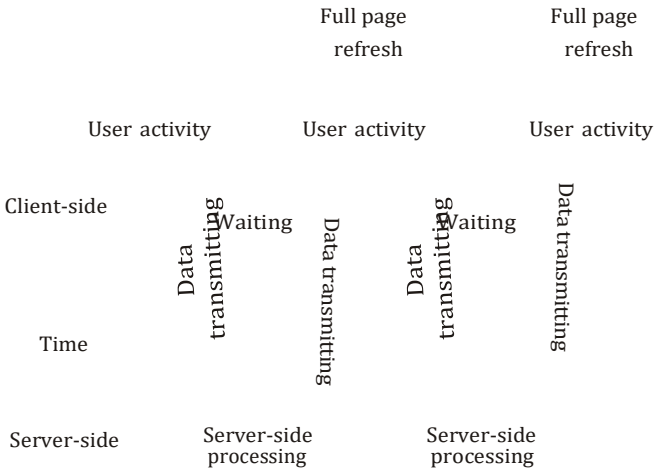
AKTU 2016-17, Marks 05

#### Answer

1. AJAX (Asynchronous JavaScript and XML) is a set of web development techniques for creating better, faster and more interactive web applications with the help of XML, HTML, CSS and JavaScript.
2. Traditional web applications tend to follow the pattern shown in Fig. 3.15.1.
3. First a page is loaded. Next, the user performs some action such as

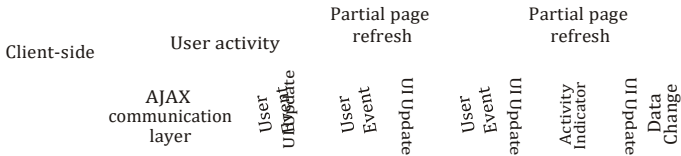
filling out a form or clicking a link.

- 4. The user activity is then submitted to a server-side program for processing while the user waits until final result is sent which reloads the entire page.
- 5. AJAX style applications use a significantly different model. Here user actions signal the server to fetch just the data needed to update the page in response to the submitted actions.
- 6. This process generally happens asynchronously, thus it allows the user to perform other actions within the browser while data is returned.



**Fig. 3.15.1.** Traditional web application communication flow.

- 7. Asynchronous requests allow more than one thing to happen at the same time.
- 8. Only the relevant portion of the page is changed when we use AJAX, as shown in Fig. 3.15.2.





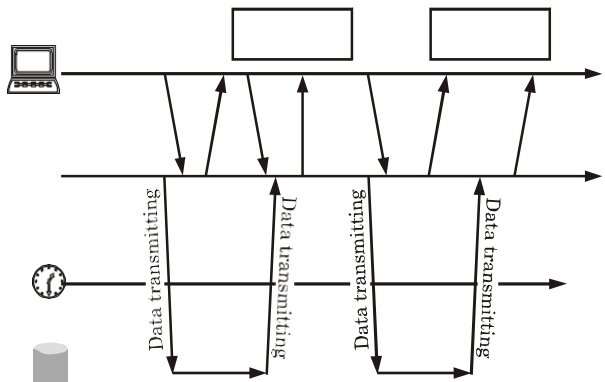
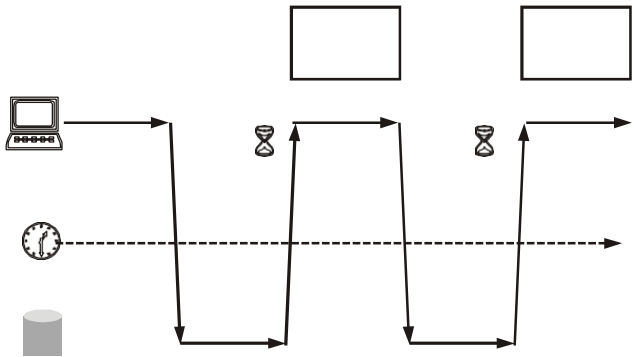
Time

Server-side

Server-side  
Processing

Server-side  
Processing

**Fig. 3.15.2. AJAX style communication flow.**



**Que 3.16. What is AJAX ? Explain its advantage and its working.**

**Explain with example.**

**AKTU 2017-18, Marks 10**

**Answer**

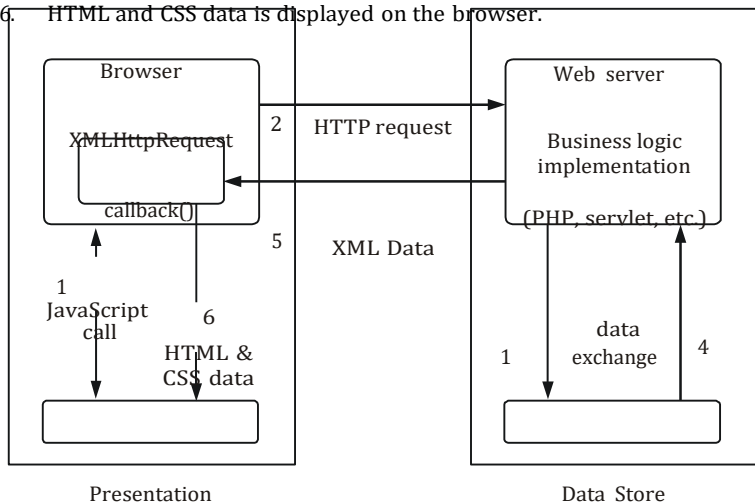
**AJAX :** Refer Q. 3.15, Page 3-18D, Unit-3.

**Advantages of AJAX :**

1. Reduces the server traffic and increases the speed.
2. Ajax is responsive and time for data transfer is also less.
3. Form validation
4. Bandwidth usage can be reduced.
5. Asynchronous calls can be made which reduces the time for data arrival.

**Working of AJAX :** XMLHttpRequest object plays an important role as AJAX communicates with the server using XMLHttpRequest object.

1. User sends a request from the UI and a JavaScript call goes to XMLHttpRequest object.
2. HTTP request is sent to the server by XMLHttpRequest object.
3. Server interacts with the database using JSP, PHP, Servlet, ASP.net etc.
4. Data is retrieved.
5. Server sends XML data or JSON data to the XMLHttpRequest callback function.
6. HTML and CSS data is displayed on the browser.



**Fig. 3.16.1.**

**For example :**

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<div id="demo">
```

```
<h2>The XMLHttpRequest Object</h2>
```

```
<button type="button" onclick="loadDoc()">Change Content</button>
```

```
</div>
```

```
<script>
```

```
function loadDoc() {
```

```
var xhttp = new XMLHttpRequest();
```

```
xhttp.onreadystatechange = function() {
```

```
if (this.readyState == 4 && this.status == 200) {
```

```
document.getElementById("demo").innerHTML = this.responseText;
```

```
}
```

```
};
```

```
xhttp.open("GET", "ajax_info.txt", true);
```

```
xhttp.send();
```

```
}
```

```
</script>
```

```
</body>
```

```
</html>
```

**Que 3.17. What is AJAX ? Explain the application of AJAX with the help of suitable examples.**

AKTU 2015-16, Marks 10

**OR**

**Discuss AJAX. Explain the application of AJAX with the help of suitable examples.**

AKTU 2019-20, Marks 07

**Answer**

**AJAX :** Refer Q. 3.15, Page 3–18D, Unit-3.

**Applications of AJAX are :**

1. AJAX is used to change the text without reloading the web page.
2. AJAX is a technique used for creating fast and dynamic web pages.
3. AJAX contains div section which is used to display information returned

from a server.

4. Major application of AJAX is in login forms where user can enter their login details directly on the original page.

**For example :**

```
<!DOCTYPE html>
<html>
<style>
table,th,td {
 border : 1px solid black;
 border-collapse: collapse;
}
th,td {
 padding: 5px;
}
</style>
<body>
<h1>The XMLHttpRequest Object</h1>
<form action="">
<select name="customers" onchange="showCustomer(this.value)">
<option value="">Select a customer:</option>
<option value="name1">Aakash Pandey</option>
<option value="name2">Mohan</option>
<option value="name3">Harshit Kumar</option>
</select>
</form>

<div id="txtHint">Customer info will be listed here...</div>
<script>
function showCustomer(str) {
 var xhttp;
 if (str == "") {
 document.getElementById("txtHint").innerHTML = "";
 return;
 }
 xhttp = new XMLHttpRequest();
```

```
xhttp.onreadystatechange = function() {
 if (this.readyState == 4 && this.status == 200) {
 document.getElementById("txtHint").innerHTML = this.responseText;
 }
};
xhttp.open("GET", "getcustomer.asp?q="+str, true);
xhttp.send();
}
</script>
</body>
</html>
```

## PART-4

*Networking : Internet Addressing, InetAddress, Factory Methods,  
Instance Methods.*

### Questions-Answers

#### Long Answer Type and Medium Answer Type Questions

**Que 3.18. Explain IP addressing.**

**Answer**

1. The IP address is a network layer address that uniquely identifies each computer on network.
2. Each TCP/IP host is identified by a logical IP address.
3. The IP address identifies a system's location on the network. An IP address must be globally unique and have a uniform format.
4. Each IP address includes a network ID and a host ID.
  - i. The network ID (also known as a network address) identifies the systems that are located on the same physical network ID. The network ID must be unique to the internetwork.
  - ii. The host ID (also known as a host address) identifies a workstation, server, router, or other TCP/IP host within a network. The address for each host must be unique to the network ID.
5. The use of the term network ID refers to any IP network ID, whether it is class-based, a subnet, or a supernet.

An internet address is made of four bytes (32 bits) that define a host's connection to a network.

Class type	Netid	Hostid
------------	-------	--------

**Fig. 3.18.1.**

- An IP address is 32 bits long. It is a common practice to segment the 32 bits of the IP address into four 8-bit fields called octets.
- Each octet is converted to a decimal number (the base 10 numbering system) in the range 0-255 and separated by a period (a dot). This format is called dotted decimal notation.

**Que 3.19. Give the classification of different IP address.**

**Answer**

**IP address is classified as :**

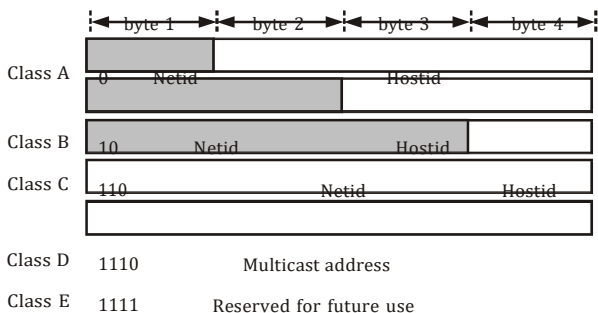
**1. Class A :**

- Class A addresses are assigned to networks with a very large number of hosts.
- The high-order bit in a class A address is always set to zero.
- The next seven bits (completing the first octet) complete network ID. The remaining 24 bits (the last three octets) represent the host ID.
- This allows for 126 networks and 16,777,214 *i.e.*,  $2^{24}$  hosts per network.

**2. Class B :**

- Class B addresses are assigned to medium-sized to large-sized networks.
- The two high-order bit in a class B address are always set to binary 10.
- The next 14 bits (completing the first two octets) complete the network ID. The remaining 16 bits (last two octets) represent the host ID.
- This allows for 16,384 networks and 65,534 hosts per network.



**Fig. 3.19.1.****3. Class C :**

- Class C addresses are used for small networks.
- The three high-order bits in a class C address are always set to binary 110.
- The next 21 bits (completing the first three octets) complete the network ID. The remaining 8 bits (last octet) represent the host ID.
- This allows for 2,097,152 networks and 254 hosts per network.

	From	To
Class A	0 . 0 . 0 . 0	127 . 255 . 255 . 255
	Netid Hostid	Netid Hostid
Class B	128 . 0 . 0 . 0	191 . 255 . 255 . 255
	Netid Hostid	Netid Hostid
Class C	192 . 0 . 0 . 0	223 . 255 . 255 . 255
	Netid Hostid	Netid Hostid
Class D	224 . 0 . 0 . 0	239 . 255 . 255 . 255
	Group address	Netid Hostid
Class E	240 . 0 . 0 . 0	255 . 255 . 255 . 255
	Undefined	Undefined

**Fig. 3.19.2.****4. Class D :**

- Class D addresses are reserved for IP multicast addresses.
- The four high-order bits in a class D address are always set to binary 1110.
- The remaining bits are for the address that interested hosts will recognize.
- Microsoft supports class D addresses for applications to multicast

data to multicast-capable hosts on an internetwork.

5. **Class *E*:** Class *E* addresses are experimental addresses reserved for future use. The high-order bits in a class *E* address are set to 1111.

**Que 3.20. What is InetAddress class ? Explain factory method of InetAddress.**

**Answer**

**InetAddress class :**

1. The InetAddress class is used to encapsulate both the numerical IP address and the domain name for that address.
2. The InetAddress class hides the number inside.
3. InetAddress can handle both IPv4 and IPv6 addresses.
4. The InetAddress class has no visible constructors.

**Factory methods :**

1. Factory methods are used to create an InetAddress object.
2. Factory method is a static method in a class and return an instance of that class.
3. Three commonly used InetAddress factory methods are as follows :
  - a. **The getLocalHost()** : This method simply returns the InetAddress object that represents the local host.
  - b. **The getByByName()** : This method returns an InetAddress for a host name passed to it. If this method is unable to resolve the host name then, they throw an UnknownHostException.
  - c. **The getAllByName()** : This method returns an array of InetAddresses that represent all of the addresses that a particular name resolves to. It will also throw an UnknownHostException if it cannot resolve the name to at least one address.

**Que 3.21. Write short note on instance method.**

**Answer**

1. Instance method is a method defined in a class and only accessible through the object of the class.
2. The InetAddress class has several instance methods, which can be used on the objects.
3. Following are the object returned by the methods which are as follows :
  - a. **Boolean equals(Object other)** : It returns true if this object has the same Internet address as other. Otherwise, it returns false.
  - b. **Byte[ ] getAddress()** : It returns a byte array that represents the object's Internet address in network byte order.
  - c. **String getHostAddress()** : It returns a string that represents the host address associated with the InetAddress object.

- d. **String getHostName()** : It returns a string that represents the host name associated with the InetAddress object.
- e. **String toString()** : It returns a string that lists the host name and the IP address.

## PART-5

*TCP/IP Client Sockets, URL, URL Connection,  
TCP/IP Server Socket, Datagram.*

### Questions-Answers

#### Long Answer Type and Medium Answer Type Questions

**Que 3.22. Explain TCP/IP client socket. Also, write the constructor and methods used to create a client socket.**

#### Answer

1. TCP/IP client sockets are used to implement bi-directional, point-to-point, stream-based connections between hosts on the Internet.
2. A socket can be used to connect Java I/O system to other programs that may reside either on the local machine or on any other machine on the Internet.
3. The creation of a Socket object implicitly establishes a connection between the client and server.
4. Following are the two constructors used to create client socket :
  - a. **Socket(String hostName, int port)** : Creates a socket connecting the local host to the named host, port and can throw an UnknownHostException or an IOException.
  - b. **Socket(InetAddress ipAddress, int port)** : Creates a socket using a pre-existing InetAddress object, a port and can throw an IOException.
5. Following methods are used by TCP/IP client socket :
  - a. **InetAddress getInetAddress()** : Returns the InetAddress associated with the Socket object.
  - b. **Int getPort()** : Returns the remote port to which the Socket object is connected.
  - c. **Int getLocalPort()** : Returns the local port to which the Socket object is connected.

**Que 3.23. Write short note on URL and URLConnection class in Java.**

**Answer**

1. URL is an acronym for Uniform Resource Locator.
2. It points to a resource on the World Wide Web (WWW).
3. A URL contains many information like protocol name, server name, port number and file name.
4. The URL is represented by an URL class.
5. Consider the following URL :  
http://www.quantumpage.com/aktu-paper.html
  - a. **Protocol** : In this case, http is the protocol.
  - b. **Server name or IP address** : In this case, www.quantumpage.com is the server name.
  - c. **Port number** : It is an optional attribute. If we write http://www.quantumpage.com:80/aktu-papers.html/, 80 is the port number. If port number is not mentioned in the URL, it returns - 1.
  - d. **File name or directory name** : In this case, aktu-papers.html is the file name.

**Following are the method provided by java.net.URL class :**

S. No.	Method	Description
1.	public String getProtocol()	It returns the protocol of the URL.
2.	public String getHost()	It returns the host name of the URL.
3.	public String getPort()	It returns the Port Number of the URL.
4.	public String getFile()	It returns the file name of the URL.
5.	public URLConnection openConnection()	It returns the instance of URLConnection <i>i.e.</i> , associated with this URL.

**URLConnection class :**

1. The Java URLConnection class represents a communication link between the URL and the application.
2. This class can be used to read and write data to the specified resource

referred by the URL.

3. The `openConnection()` method of `URL` class returns the object of `URLConnection` class.
4. Syntax to get the object of `URLConnection` :  
`public URLConnection openConnection() throws IOException{`
5. The `URLConnection` class use `getInputStream()` method to display all the data of a web page.
6. The `getInputStream()` method returns all the data of the specified `URL` in the stream that can be read and displayed.

**Que 3.24. Explain TCP/IP server socket.**

**Answer**

1. The `TCP/IP ServerSocket` is used to create servers that listen for either local or remote client programs to connect them on published ports.
2. `TCP/IP ServerSockets` are quite different from normal sockets.
3. When we create a `TCP/IP Server Socket`, it will register itself with the systems that have client connections.
4. The constructors for `TCP/IP ServerSocket` reflect the port number that we wish to accept connections on and, how long we want the port to be in the queue.
5. The queue length tells the system how many client connections it can leave pending before it should simply refuse connections.
6. It has constructors that create new `TCP/IP ServerSocket` objects, methods that listen for connections on a specified port, methods that configure the various `TCP/IP` server socket options, and the usual miscellaneous methods such as `toString()`.

**Que 3.25. Explain the following terms in brief :**

- i. **Socket programming**
- ii. **TCP/IP Server**

**Answer**

- i. **Socket programming :**

1. Java socket programming is used for communication between the applications running on different JRE.
2. Java socket programming can be connection-oriented or connection-less.
3. `Socket` and `ServerSocket` classes are used for connection-oriented socket programming and `DatagramSocket` and `DatagramPacket` classes are used for connection-less socket programming.

4. Sockets provide the communication mechanism between two computers using TCP.
  5. A client program creates a socket on its end of the communication and attempts to connect that socket to a server.
  6. The client in socket programming must know :
    - a. IP address of server
    - b. Port number
- ii. **TCP/IP server** : Refer Q. 3.24, Page 3-29D, Unit-3.

**Que 3.26.** What is datagram ? Give its characteristics. Also, explain datagram socket.

**Answer**

1. Datagram is a unit of transfer associated with networking.
2. Datagram is typically structured in header and payload section.
3. It provides a connectionless communication service across a packet-switched network.

**Characteristics of datagram :**

1. It is transmitted from source to destination without guarantee of delivery.
2. It provides a connectionless communication service.

**Datagram socket :**

1. It is a communication link used to send datagram between applications.
2. Datagram socket is a type of network socket which provide connectionless point for sending and receiving packets.
3. Every packet sent from a datagram socket is individually routed and delivered.
4. Java DatagramSocket and DatagramPacket classes are used for connectionless socket programming.
5. A DatagramPacket is a message that can be sent or received through DatagramSocket.
6. Commonly used constructors of DatagramSocket class are as follows :
  - a. **DatagramSocket() throws SocketException** : It creates a datagram socket and binds it with the available port number on the localhost machine.
  - b. **DatagramSocket(int port) throws SocketException** : It creates a datagram socket and binds it with the port number.
  - c. **DatagramSocket(int port, InetAddress address) throws SocketException** : It creates a datagram socket and binds it with the specified port number and host address.



7. Commonly used constructors of DatagramPacket class are as follows :
- DatagramPacket(byte[] barr, int length)** : It creates a datagram packet. This constructor is used to receive the packets.
  - DatagramPacket(byte[] barr, int length, InetAddress address, int port)** : It creates a datagram packet. This constructor is used to send the packets.

**Que 3.27. Discuss socket and server socket in Java with its package. Write a program in Java to demonstrate, how the communication is establish between client and server?**

AKTU 2019-20, Marks 07

**Answer**

**Socket** : Refer Q. 3.22, Page 3-27D, Unit-3.

**Server socket** : Refer Q. 3.24, Page 3-29D, Unit-3.

**For example :**

```
// A Java program for a Client
import java.net.*;
import java.io.*;
public class Client
{
 // initialize socket and input output streams
 private Socket socket = null;
 private DataInputStream input = null;
 private DataOutputStream out = null;
 // constructor to put ip address and port
 public Client(String address, int port)
 {
 // establish a connection
 try
 {
 socket = new Socket(address, port);
 System.out.println("Connected");
 // takes input from terminal
 input = new DataInputStream(System.in);
 // sends output to the socket
 out = new DataOutputStream(socket.getOutputStream());
 }
 }
}
```

```
catch(UnknownHostException u)
```

```
{
System.out.println(u);
}
catch(IOException i)
{
System.out.println(i);
}
// string to read message from input
String line = "";
// keep reading until "Over" is input
while (!line.equals("Over"))
{
try
{
line = input.readLine();
out.writeUTF(line);
}
catch(IOException i)
{
System.out.println(i);
}
}
// close the connection
try
{
input.close();
out.close();
socket.close();
}
catch(IOException i)
{
System.out.println(i);
}
}

public static void main(String args[])
{
Client client = new Client("127.0.0.1", 5000);
}
}
```

**VERY IMPORTANT QUESTIONS**

*Following questions are very important. These questions may be asked in your SESSIONALS as well as UNIVERSITY EXAMINATION.*

**Q. 1. What is JavaScript ? How it works ? What are the features of JavaScript ?**

**Ans.** Refer Q. 3.1.

**Q. 2. What is the difference between Java and JavaScript ? Describe the strengths and weakness of JavaScript.**

**Ans.** Refer Q. 3.2.

**Q. 3. What are scripting languages and why JavaScript is used ? Write a JavaScript function for validating form data like mandatory fields and email field.**

**Ans.** Refer Q. 3.6.

**Q. 4. Explain the role of JavaScript to develop a web page. Write a JavaScript function to check a textbox is either empty or not.**

**Ans.** Refer Q. 3.9.

**Q. 5. Explain conditional statements used in JavaScript with example.**

**Ans.** Refer Q. 3.10.

**Q. 6. Compare Java and JavaScript. Explain and demonstrate 5 different types of objects in JavaScript with example.**

**Ans.** Refer Q. 3.12.

**Q. 7. What is AJAX ? Explain its advantage and its working. Explain with example.**

**Ans.** Refer Q. 3.16.

**Q. 8. What is AJAX ? Explain the application of AJAX with the help of suitable examples.**

**Ans.** Refer Q. 3.17.



==

**Q. 9. Explain TCP/IP client socket. Also, write the constructor and methods used to create a client socket.**

==

**Ans.** Refer Q. 3.22.

**Q. 10. What is datagram ? Give its characteristics. Also, explain datagram socket.**

**Ans.** Refer Q. 3.26.

