

COMPUTER SCIENCE

Grade: XII

WEB TECHNOLOGY II



REFERENCE NOTE

NEB Important Questions for Computer Science XII

Unit 3- Web Technology II

1. What is web technology? Explain different data types used in JavaScript.
2. **Differentiate between client side scripting and Server-side scripting.**
3. What is event handling in JavaScript? Give one example.
4. **What is JQuery? Write its features and Write a program to displaying a message "Hello Class 12" using JQuery.**
5. What is JavaScript? How can you add JavaScript to an HTML page? Describe with example.
6. What is jQuery? Write its features.
7. **What is Java Script function? Explain with calling function examples.**
8. What is PHP? Write the advantages of PHP.
9. Explain the different operators used in PHP.
10. **What are the features of PHP? Write a PHP program to display the largest among three numbers.**
11. Define SQL? Write down the SQL queries to create a database, create a table in the database, insert data in the table, and query the data to display it.
12. Write down the server side script to create a database, connect with it, create a table and insert data in it.

Practical Programs

1. Write a java Script to display 1 to 10 using for loop, while loop and do while loop.
2. Create a Page with a button with value "Computer" on clicking the button your page should "Computer Science".
3. **Write a JavaScript program to display "Welcome Class-12" using onload event.**
4. Design a form with username, address, e-mail, password and submit button. Validated the form using jQuery.
5. **Design a form with username and password and submit button. Write a PHP code to get value of username and password using a) \$_POST variable and b) \$_GET variable.**
6. Write a PHP program to check if a string is a palindrome or not.
7. **Write a Java Script program to calculate the factorial of a given number.**
8. Write a java script program to input three number and find largest one using java and PHP.

Web technology

Introduction

Web Technology is the tools and techniques which enables two or more computing devices to communicate over a network i.e. Internet.

Web Technology consist of two words, the web refers to the World Wide Web generally known as World Wide Web.

WWW is the cyber space containing webpages, documents, and any other resources which are identified and located with the help of their URLs.

Technology refers to the tools and techniques that makes these resources available on the Web such as, web browsers to view content of web, Programming languages and frameworks for the development of websites, Database to store data at back end, protocols for communicating on the web, multimedia elements etc.

Web development

It is the process of designing and developing website which are hosted through internet or intranet. The process of developing web can range from developing static page to a complex such as web based application social media sites, E-commerce.

Web development includes web design, web content development, client side scripting, server side scripting, web engineering etc. Since, web development consists of several inter-related task which can be accomplish by different types of developer who focuses on different aspect of web creation.

Frontend	Backend
Client Side	Server Side
Website Design	Database
HTML	PHP
CSS	Java
JavaScript	Python
AJAX	Ruby
UI/UX	Servers
Some UI technologies	Some Backend technologies
	.NET

Scripting Language:

- JavaScript is a scripting language. A scripting language is a lightweight programming language. JavaScript code can be inserted into any HTML page, and it can be executed by all types of web browsers.
- JavaScript is used to make web pages interactive. It runs on your visitor's computer and doesn't require constant downloads from your website. JavaScript and Java are completely different language, both in concept and design.

Unlike HTML, JavaScript is case sensitive-therefore watch your capitalization closely when you write JavaScript statements, create or call variables, objects and functions.

A scripting language is a programming language designed for integrating and communicating with other programming language. It is used in server side as well as client side. Some of the most widely used scripting language are JavaScript, VBScript, PHP, Perl, Python, Ruby etc.

Server side and client side programming

Client-Side Scripting programming

Client-side scripting is performed to generate a code that can run on the client side i.e (front end) browser without needing the server-side (back end) processing. Basically, client-side scripts are placed inside an HTML document.

The client-side scripting can be used to layout the content of the web. For example, when a user makes a request through web browser for a webpage to the server, it just sent the HTML and CSS as plain text, and the browser interprets and renders the content of web in the client end (user).

Client-side scripting is designed to run as a scripting language which can be executed by web browser. Front end developer is someone who design and develop client side of a website. Generally he or she works in user interface (UI) of a website. Front end developer must be at least fluent in three different languages i.e. HTML, CSS, JavaScript whereas, there are several other libraries which can be used for front end development.

Server-Side Scripting programming

Server-side scripting also known as back-end runs on the server where the application is hosted. Server-side is used to serve content depending upon the user request. Back end helps to create dynamic web based application that allows user to interact and communicate with the application. Back end language also helps to connect front end with data base. So that, User can store and retrieve data as per the requirement. Back-end developer is responsible for server-side programming. Some of the popular server-side (back-end) scripting language are ASP, JavaScript (using SSJS (Server-side JavaScript e.g. node.js), Perl, PHP, Ruby, Python etc.

The client-side scripting emphasizes making the interface of the web application or website (UI) more appealing and functional. Whereas, server-side scripting emphasizes on data accessing methods, error handling and fast processing etc.

Note: *Full-stack developer understand both Front end and back end development process. They can accomplish entire project. Full stack developer must have expertise in client site and server site Scripting language. Moreover, he/she has a great knowledge of integrating database with the application.*

```
<html>
<head>
<title> Inline javascript Example</title>
<body>
  <form>
<center><input type="button" value="ClickMe" onclick="alert('Button Clicked: ')">
  </center></form>
</body>
</html>
```

Comparison between Client-side and Server-side Scripting

This section elaborates the fundamental differences between client-side and server-side scripts:

1. The client-side script is executed at the front-end in the client's browser while the server-side script is executed at the back end with a web server.
2. The client-side script is visible to the user of the web browser while the server-side script is hidden.
3. The client-side script is not secure while the server-side script is secure.
4. The client-side script does not need to interact with the server while the server-side script needs a web server to be processed.
5. The client-side script is executed on a local computer while the server-side script is executed on a remote computer.
6. The client-side script has a faster response time than the server-side script.
7. Client-side script is executed after the browser receives the web pages sent by the server while the server-side script cannot execute the client-side script.
8. The client-side script cannot connect with the database while the server-side script can connect with the database present on the server-side.
9. The client-side script cannot access the files while the server-side script can access and manipulate the files present at the webserver.
10. The client-side script helps create interactive web pages while the server-side script helps create web pages with dynamic data.

Internet Technology

The Internet is the global system of interconnected computer networks that uses the Internet protocol suite to communicate between networks and devices. It is a network of networks that consists of private, public, academic, business, and government networks of local to global scope, linked by a broad array of electronic, wireless, and optical networking technologies. The Internet carries a vast range of information resources and services, such as the inter-linked hypertext documents and applications of the World Wide Web, electronic mail, telephony, and file sharing.

JavaScript

JavaScript is a lightweight, interpreted programming language. It is designed for creating network-centric applications. It is complimentary to and integrated with Java. JavaScript is very easy to implement because it is integrated with HTML. It is open and cross-platform.

This tutorial has been prepared for JavaScript beginners to help them understand the basic functionality of JavaScript to build dynamic web pages and web applications.

Features

- JavaScript is a lightweight, interpreted programming language.
- Designed for creating network-centric applications.
- Complementary to and integrated with Java.
- Complementary to and integrated with HTML.
- It is case sensitive language.
- JavaScript is supportable in operating system.
- It provides good control to the users over the web browsers.

```
<script>
document.write("Hello JavaScript by JavaScript");
</script>
```

Uses of JavaScript

- Client side validation
- Dynamic drop down menus.
- Displaying date and time.
- Displaying pop-up windows and dialog boxes
- Displaying clocks.
- Event handling.
- Developing Mobile applications
- Creating web browser based games.
- Building web servers
- Adding interactivity to website.

Adding JavaScript to HTML

JavaScript, also known as JS, is one of the scripting (client-side scripting) languages, that is usually used in web development to create modern and interactive web-pages. The term "script" is used to refer to the languages that are not standalone in nature and here it refers to JavaScript which run on the client machine.

In other words, we can say that the term scripting is used for languages that require the support of another language to get executed. For example, JavaScript programs cannot get executed without the help of HTML or without integrated into HTML code.

1. Embedding code
2. Inline code
3. External file

1. Embedding code:-

To add the JavaScript code into the HTML pages, we can use the `<script>.....</script>` tag of the HTML that wrap around JavaScript code inside the HTML program. Users can also define JavaScript code in the <body> tag

(or we can say body section) or <head> tag

because it completely depends on the structure of the web page that the users use.

```
<!DOCTYPE html >
<html>
<head>
<title> page title</title>
<script>
document.write("Welcome to Javatpoint");
</script>
</head>
<body>
<p>Inthis example we saw how to add JavaScript in the head section </p>
</body>
</html>
```

We can also define the JavaScript code in the <body> tags or body section.

Let's understand through an example.

```
<!DOCTYPE html >
<html>
<head>
<title> page title</title>
</head>
<body>
<script>
document.write("Welcome to Javatpoint");
</script>
<p> In this example we saw how to add JavaScript in the body section </p>
</body>
</html>
```

2. Inline code:-

Generally, this method is used when we have to call a function in the HTML event attributes. There are many cases (or events) in which we have to add JavaScript code directly eg., OnMover event, OnClick, etc. Let's see with the help of an example, how we can add JavaScript directly in the html without using the <script>.... </script> tag.

Let's look at the example.

```
<!DOCTYPE html >
<html>
<head>
<title> page title</title>
</head>
<body>
<p>
<a href="#" onClick="alert('Welcome !');">Click Me</a>
<button onclick="alert('Hello Class 12')">Click Here</button>
</p>
<p> in this example we saw how to use inline JavaScript or directly in an HTML tag. </p>
</body>
</html>
```

3. External file:-

We can also create a separate file to hold the code of JavaScript with the (.js) extension and later incorporate/include it into our HTML document using the **src** attribute of the <script> tag. It becomes very helpful if we want to use the same code in multiple HTML documents. It also saves us from the task of writing the same code over and over again and makes it easier to maintain web pages.

In this example, we will see how we can include an external JavaScript file in an HTML document.

Let's understand through a simple example.

```
<html>
<head>
<title>Including a External JavaScript File</title>
</head>
<body>
<form>
<input type="button" value="Result" onclick="display()"/>
</form>
<script src="hello.js">
</script>
</body>
</html>
```

Now let's create separate JavaScript file

Hello.js

```
function display()
{
alert("Hello World!");
}
```

JavaScript Fundamentals

- Comments
- Variables
- Data types
- Operators
- Strict Mode
- Functions

JavaScript comments

The **JavaScript comments** are meaningful way to deliver message. It is used to add information about the code, warnings or suggestions so that end user can easily interpret the code.

The JavaScript comment is ignored by the JavaScript engine i.e. embedded in the browser.

There are two types of comments in JavaScript.

1. Single-line Comment
2. Multi-line Comment

1. Single line Comment

It is represented by double forward slashes (//). It can be used before and after the statement.

Let's see the example of single-line comment i.e. added before the statement.

```
<script>
// It is single line comment
document.write("hello javascript");
</script>
```

Let's see the example of single-line comment i.e. added after the statement.

```
<script>
var a=10;
var b=20;
var c=a+b; //It adds values of a and b variable
document.write(c); //prints sum of 10 and 20
</script>
```

2. Multi line Comment

It can be used to add single as well as multi line comments. So, it is more convenient.

It is represented by forward slash with asterisk then asterisk with forward slash. For example:

```
/* your comment here */
```

It can be used before, after and middle of the statement.

```
<script>
/* It is multi line comment.
It will not be displayed */
document.write("Example of javascript multiline comment");
</script>
```

Basic Display function in JavaScript

1. Document.write

```
<html>
<head> <title> Display functions in Javascript </title>
</head>
<body>
  <h1> Output with document.write</h1> <br>
  Hello Class 12 <br>
  <script>
    Document.write(7+5)
  </script>
  <input type="button" value="Click Me" onclick="document.write(7+5)">
</body>
</html>
```

Using Inner HTML

```
<script>
document.getElementById("demo").innerHTML = "Hello JavaScript!";
</script>
```

2. Window. Alert

```
<html>
<head> <title> Display functions in Javascript </title>
</head>
<body>
  <h1> Output Message with window alert</h1> <br>
  <script>
    alert(7+5);
  </script>
</body>
</html>
```


3. Console.log

```

<html>
<head>      <title>Display functions in Javascript </title>
</head>
<body>
  <h1> Demo of console.log</h1> <br>
  <script>
    Console.log("Gurukul College");
    Console.log("7+5");
  </script>
</body>
</html>

```

JavaScript Data Types

JavaScript provides different **data types** to hold different types of values. There are two types of data types in JavaScript.

1. Primitive data type
2. Non-primitive (reference) data type

JavaScript is a **dynamic type language**, means you don't need to specify type of the variable because it is dynamically used by JavaScript engine. You need to use **var** here to specify the data type. It can hold any type of values such as numbers, strings etc. For example:

```

var a=40;      //holding number
var b="Rahul"; //holding string

```

JavaScript primitive data types

There are five types of primitive data types in JavaScript. They are as follows:

Data Type	Description
String	represents sequence of characters e.g. "hello"
Number	represents numeric values e.g. 100
Boolean	represents boolean value either false or true
Undefined	represents undefined value
Null	represents null i.e. no value at all

JavaScript non-primitive data types

The non-primitive data types are as follows:

Data Type	Description
-----------	-------------

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

JavaScript Variable

1. JavaScript variable
2. JavaScript Local variable
3. JavaScript Global variable

A **JavaScript variable** is simply a name of storage location. There are two types of variables in JavaScript : local variable and global variable.

There are some rules while declaring a JavaScript variable (also known as identifiers).

1. Name must start with a letter (a to z or A to Z), underscore(_), or dollar(\$) sign.
2. After first letter we can use digits (0 to 9), for example value1.
3. JavaScript variables are case sensitive, for example x and X are different variables.

Correct JavaScript variables

1. `var x = 10;`
2. `var _value="Hari";`

Incorrect JavaScript variables

1. `var 123=30;`
2. `var *aa=320;`

Example of JavaScript variable

Let's see a simple example of JavaScript variable.

OOPs Concepts in Java

1. `<script>`
2. `var x = 10;`
3. `var y = 20;`
4. `var z=x+y;`
5. `document.write(z);`
6. `</script>`

JavaScript If-else

The **JavaScript if-else statement** is used to execute the code whether condition is true or false. There are three forms of if statement in JavaScript.

1. If Statement
2. If else statement
3. if else if statement

JavaScript If statement

It evaluates the content only if expression is true. The signature of JavaScript if statement is given below.

```
if(expression)
{
//content to be evaluated
}
```

Let's see the simple example of if statement in javascript.

```
<script>
var a=20;
if(a>10)
{
document.write("value of a is greater than 10");
}
</script>
```

JavaScript If...else Statement

It evaluates the content whether condition is true or false. The syntax of JavaScript if-else statement is given below.

```
if(expression)
{
//content to be evaluated if condition is true
}
Else
{
//content to be evaluated if condition is false
}
```

Let's see the example of if-else statement in JavaScript to find out the even or odd number.

```
<script>
var a=17;
if(a%2==0)
{
document.write("a is even number");
}
else
{
```

```
document.write("a is odd number");  
}  
</script>
```

3. JavaScript If...else if statement

It evaluates the content only if expression is true from several expressions. The signature of JavaScript if else if statement is given below.

```
if(expression1){  
  //content to be evaluated if expression1 is true  
}  
else if(expression2){  
  //content to be evaluated if expression2 is true  
}  
else if(expression3){  
  //content to be evaluated if expression3 is true  
}  
else{  
  //content to be evaluated if no expression is true  
}
```

Let's see the simple example of if else if statement in javascript.

```
<script>  
var a=20;  
if(a==10){  
  document.write("a is equal to 10");  
}  
else if(a==15){  
  document.write("a is equal to 15");  
}  
else if(a==20){  
  document.write("a is equal to 20");  
}  
else{  
  document.write("a is not equal to 10, 15 or 20");  
}  
</script>
```

JavaScript Switch

The **JavaScript switch statement** is used to execute one code from multiple expressions. It is just like else if statement that we have learned in previous page. But it is convenient than *if..else..if* because it can be used with numbers, characters etc.

The signature of JavaScript switch statement is given below.

```
switch(expression)
{
  case value1:
    code to be executed;
    break;
  case value2:
    code to be executed;
    break;
  .....
  default:
    code to be executed if above values are not matched;
}
```

Let's see the simple example of switch statement in javascript.

```
<script>
var grade='B';
var result;
switch(grade){
case 'A':
result="A Grade";
break;
case 'B':
result="B Grade";
break;
case 'C':
result="C Grade";
break;
default:
result="No Grade";
}
document.write(result);
</script>
```

```
<script>
var grade='B';
var result;
switch(grade){
case 'A':
result+=" A Grade";
case 'B':
result+=" B Grade";
case 'C':
result+=" C Grade";
default:
result+=" No Grade";
}
document.write(result);
</script>
```

Form Validation

It is important to validate the form submitted by the user because it can have inappropriate values. So, validation is must to authenticate user.

JavaScript provides facility to validate the form on the client-side so data processing will be faster than server-side validation. Most of the web developers prefer JavaScript form validation.

Through JavaScript, we can validate name, password, email, date, mobile numbers and more fields.

In this example, we are going to validate the name and password. The name can't be empty and password can't be less than 6 characters long.

Here, we are validating the form on form submit. The user will not be forwarded to the next page until given values are correct.

```
<html>
<body>
<script>
function validateform()
{
var name=document.myform.name.value;
var password=document.myform.password.value;

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;
}
}
</script>
<body>
<form name="myform" method="post"
action="http://www.javatpoint.com/javascriptpages/valid.jsp"
onsubmit="return validateform()" >
Name: <input type="text" name="name"><br/>
Password: <input type="password" name="password"><br/>
<input type="submit" value="register">
</form>
</body>
</html>

```

JavaScript Retype Password Validation

```

<script type="text/javascript">
function matchpass(){
var firstpassword=document.f1.password.value;
var secondpassword=document.f1.password2.value;

if(firstpassword==secondpassword){
return true;
}
else{
alert("password must be same!");
return false;
}
}
}
</script>

<form name="f1" action="register.jsp" onsubmit="return matchpass()">
Password:<input type="password" name="password" /> <br/>
Re-enter Password:<input type="password" name="password2"/> <br/>
<input type="submit">

```

</form>

Number Validation

Let's validate the textfield for numeric value only. Here, we are using isNaN() function.

```

<script>
function validate()
{
var num=document.myform.num.value;
if (isNaN(num))
{
document.getElementById("numloc").innerHTML="Enter Numeric value only";
return false;
}
Else
{
return true;
}
}
</script>
<form name="myform" onsubmit="return validate()" >
Number: <input type="text" name="num"> <span id="numloc"> </span> <br/>
<input type="submit" value="submit">
</form>

```

JavaScript validation with image

Let's see an interactive JavaScript form validation example that displays correct and incorrect image if input is correct or incorrect.

```

<script>
function validate()
{
var name=document.f1.name.value;
var password=document.f1.password.value;
var status=false;

if(name.length<1)
{
document.getElementById("nameloc").innerHTML=

```



```

" <img src='unchecked.gif' /> Please enter your name";
status=false;
}
Else
{
document.getElementById("nameloc").innerHTML=" <img src='checked.gif' />";
status=true;
}
if(password.length<6)
{
document.getElementById("passwordloc").innerHTML=
" <img src='unchecked.gif' /> Password must be at least 6 char long";
status=false;
}
Else
{
document.getElementById("passwordloc").innerHTML=" <img src='checked.gif' />";
}
return status;
}
</script>
<form name="f1" action="#" onsubmit="return validate()">
<table>
<tr><td>Enter Name:</td><td><input type="text" name="name" />
<span id="nameloc"></span></td></tr>
<tr><td>Enter Password:</td><td><input type="password" name="password"
/>
<span id="passwordloc"></span></td></tr>
<tr><td colspan="2"><input type="submit" value="register" /></td></tr>
</table>
</form>

```

IMP

JavaScript email validation

We can validate the email by the help of JavaScript.

There are many criteria that need to be follow to validate the email id such as:

- Email id must contain the @ and . character
- There must be at least one character before and after the @.
- There must be at least two characters after. (Dot).

Let's see the simple example to validate the email field.

```

<script>
function validateemail()
{
var x=document.myform.email.value;
var atposition=x.indexOf("@");
var dotposition=x.lastIndexOf(".");
if (atposition<1 || dotposition<atposition+2 || dotposition+2>=x.length)
{
  alert("Please enter a valid e-
mail address \n atpostion:"+atposition+"\n dotposition:"+dotposition);
  return false;
}
}
</script>
<body>
<form name="myform" method="post" action="#" onsubmit="return validateemail
();">
Email: <input type="text" name="email"> <br/>
  <input type="submit" value="register">
</form>

```

JavaScript Programs

1. This program adds 5 numbers and writes the answer as an HTML output.

```

<!doctype html>
<html> <body>
<script>
var i, n=5, sum=0; arr = new Array(1, 2, 3, 5, 9);
for(i=0; i<5; i++)
sum = sum + arr[i]; document.write(sum);
</script></body></html>

```

2. JavaScript program to check whether a number is a prime number or not. This progre does not take input from user.

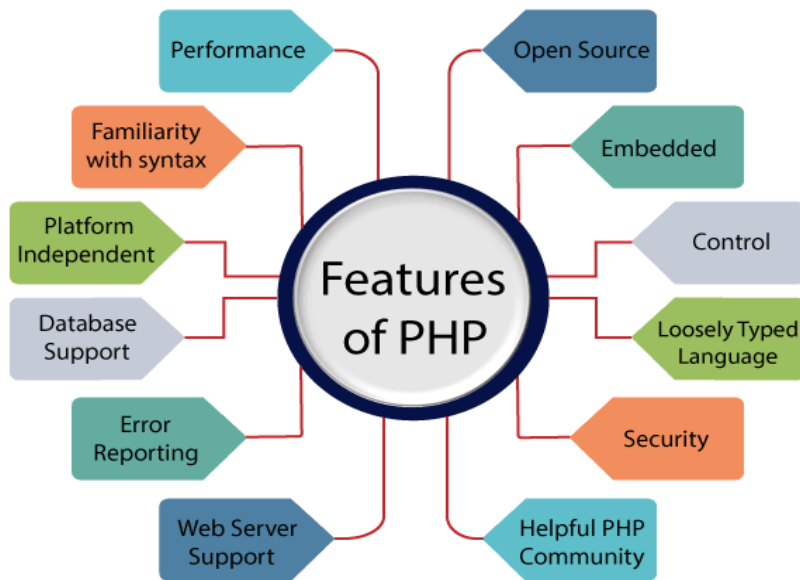
```
<!doctype html>
<html>
<body> <script>
var num, i, c=0;
num=9;
for(i=2; i<num; i++)
{
if(num%2==0)
{
C++;
break;
}
}
if(c==0)
document.write(num+" is a Prime Number");
else
document.write(num+" is not a Prime Number");
</script> </body>
</html>
```

3. JavaScript code to display number from 1 to 10 using loop

```
<!DOCTYPE html>
<html> <head>
<title>JavaScript Loops Example</title>
<script> function loop_fun1()
{
var num = "";
var i; for (i= 1; i <= 10; i++)
{
num = num + i + "<br>";
document.getElementById("loop_paral").innerHTML = nu
}
}
</script>
</head>
<body>
<p>Click on the <b>Start Loop</b> button to start the loop.</p> <button onclick="loop_fun1()">Start
Loop</button> <p id="loop_paral"></p>
</body>
</html>
```

Server Side Scripting Using PHP:

Server side script are discrete blocks of program code that execute on a web server (as opposed to a web client) computer. They are generally used to create Dynamic web pages. This means that the page displayed to the user does not exist as a document on the server in its own right, and is only brought into existence in response to a user request. Often, a server-side script provides the interface between a web-based user interface and a database that resides on a web server. PHP is a server side scripting language used to create dynamic web pages. PHP is well suited for web development. PHP stands for hypertext preprocessor. It is an interpreted language. It is embedded in HTML. It is an open source language.



Some important points need to be notice about PHP are as followed:

1. PHP stands for Hypertext Preprocessor.
2. PHP is an interpreted language, i.e., there is no need for compilation.
3. PHP is faster than other scripting languages, for example, ASP and JSP.
4. PHP is a server-side scripting language, which is used to manage the dynamic content of the website.
5. PHP can be embedded into HTML.
6. PHP is an object-oriented language.
7. PHP is an open-source scripting language.
8. PHP is simple and easy to learn language.

Features of PHP

1. Simple
2. Interpreted
3. Faster
4. Open Source
5. Platform Independent
6. Case Sensitive
7. Error Reporting
8. Real-Time Access Monitoring
9. Loosely Typed Language

Characteristics of PHP

1. Simplicity
2. Efficiency
3. Security
4. Flexibility
5. Familiarity

Advantages of PHP / what problem does it solve.

1. PHP is Free
2. PHP is Cross Platform
3. PHP is widely used
4. PHP hides its complexity
5. PHP is built for Web Programming

Introduction to PHP:

PHP is an open-source, interpreted, and object-oriented scripting language that can be executed at the server-side. PHP is well suited for web development. Therefore, it is used to develop web applications (an application that executes on the server and generates the dynamic page.).

Before learning PHP, you must have the basic knowledge of **HTML**, **CSS**, and **JavaScript**. So, learn these technologies for better implementation of PHP.

HTML - HTML is used to design static webpage.

CSS - CSS helps to make the webpage content more effective and attractive.

JavaScript - JavaScript is used to design an interactive website.

PHP is widely used in web development nowadays. PHP can develop dynamic websites easily. But you must have the basic the knowledge of following technologies for web development as well.

1. HTML
2. CSS
3. JavaScript
4. Ajax
5. XML and JSON
6. jQuery

Why use PHP / Strengths of PHP

PHP is a server-side scripting language, which is used to design the dynamic web applications with MySQL database.

1. It handles dynamic content, database as well as session tracking for the website.
2. You can create sessions in PHP.
3. It can access cookies variable and also set cookies.
4. It helps to encrypt the data and apply validation.
5. PHP supports several protocols such as HTTP (Hypertext Transfer Protocol), POP3 (Post office Protocol-3), SNMP (Simple Network Management Protocol), LDAP (Lightweight Directory Access Protocol), IMAP (Internet Message Access Protocol), and many more.
6. Using PHP language, you can control the user to access some pages of your website.
7. As PHP is easy to install and set up, this is the main reason why PHP is the best language to learn.
8. PHP can handle the forms, such as - collect the data from users using forms, save it into the database, and return useful information to the user. **For example** - Registration form.

Weaknesses of PHP

1. PHP's main strength flexibility is also its weakness. It can be a little too forgiving of errors.
2. With no strict rules, inexperienced programmers have the freedom to create some very bad solutions to simple problems.
3. Bad packages that got popular in the community are still reused when developers are trying something new or rushed for time. Some of these mistakes lead to security risks.

4. As a mature tool PHP has some legacy baggage. There are lots of internal inconsistencies, especially surrounding references and values.
5. This is mostly due to updates, which add features that clash with earlier features,
6. PHP is an interpreted language, which can reduce speed.
7. PHP 7 increased performance over previous versions by a significant amount while maintaining most language compatibility.
8. The changes didn't affect the learning curve or existing applications much while improving performance. Still, it executes more slowly than compiled languages.
9. Scaling and maintaining PHP is a complicated endeavor.
10. Context matters a great deal in dynamically typed languages, so tracking down errors gets harder the larger an application grows.
11. Experienced PHP developers can mitigate this problem by planning for scalability, but there's only so much they can do to reduce maintenance issues down the road.

Common uses of PHP

1. PHP performs system functions, i.e., from files on a system it can create, open, read, write, and close them.
2. PHP can handle forms, i.e., gather data from files, save data to a file, through email we can send data, return data to the user.
3. We add, delete, and modify elements within the database through PHP.
4. Access cookies variables and set cookies.
5. Using PHP, we can restrict users to access some pages of your website.
6. It can encrypt data.

3.15 Object Oriented Programming with Server Side Scripting

Object Oriented is an approach to software development that models application around real world objects such as employees, cars, bank accounts, etc. A class defines the properties and methods of a real world object. An object is an occurrence of a class. There are three major principles of OOP are:

- 1. Encapsulation:** This is concerned with hiding the implementation details and only exposing the methods. It used to reduce software development complexity.
- 2. Inheritance:** This is concerned with the relationship between classes. The main purpose of inheritance is the re-usability.
- 3. Polymorphism:** This is concerned with having a single form but many different implementation ways. The main purpose of polymorphism is to simplify maintaining applications and making them more extendable.

PHP in Object oriented programming PHP is an object oriented scripting language. It supports all of the above principles. The above principles are achieved via:

Encapsulation - using "get" and "set" methods.

Inheritance - using extends keyword.

Polymorphism - using implements keyword

Hardware and Software Requirements

Hardware

Software

1. Browser
2. Web server (eg. Apache)
3. PHP (interpreter)
4. MySQL Database (it is optional)

Basics of PHP:

Setting up the environment

To run PHP a web development is needed. This needs a PHP compatible web server and interpreter. Package like WAMP, LAMP, and XAMP etc. can be used which includes a web server.

Writing the code and running the script

PHP scripts are plain text. A PHP script begins with `<?php` and ends with `?>`. The PHP code is saved with extension .PHP and is saved in the root directory of web server.

The syntax of PHP is given below:

```
<?php
//your code here
?>
```

Embedding HTML and PHP

PHP codes are embedded in HTML documents using the `<?php` and `?>` tags

Comments in PHP

Comments are used to make code more readable . There are two types of comments – single line and multi-line comments . A single line comments starts with `//` while multi-line comment

Begins with `/*` and end with `*/`

Web References:

- <https://www.javatpoint.com>
- <https://www.w3schools.com>
- <https://www.tutorialspoint.com>
- <https://www.google.com>
- <https://www.wikipedia.org>