

## Topic: 1.2.3 Client- and server-side scripting

There are two main ways to customize Web pages and make them more interactive. The two are often used together because they do very different things.

### Scripts

A script is a set of instructions. For Web pages they are instructions either to the Web browser (client-side scripting) or to the server (server-side scripting). These are explained more below.

Scripts provide change to a Web page. Think of some Web pages you have visited. Any page which changes each time you visit it (or during a visit) probably uses scripting.

All log on systems, some menus, almost all photograph slideshows and many other pages use scripts. Google uses scripts to fill in your search term for you, to place advertisements, to find the thing you are searching for and so on. Amazon uses scripting to list products and record what you have bought.

### Client-side

The client is the system on which the Web browser is running. JavaScript is the main client-side scripting language for the Web. Client-side scripts are interpreted by the browser. The process with client-side scripting is:

1. the user requests a Web page from the server
2. the server finds the page and sends it to the user
3. the page is displayed on the browser with any scripts running during or after display

So client-side scripting is used to make Web pages change after they arrive at the browser. It is useful for making pages a bit more interesting and user-friendly. It can also provide useful gadgets such as calculators, clocks etc. but on the whole is used for appearance and interaction.

Client-side scripts rely on the user's computer. If that computer is slow they may run slowly. They may not run at all if the browser does not understand the scripting language. As they have to run on the user's system the code which makes up the script is there in the HTML for the user to look at (and copy or change).





## Topic: 1.2.3 Client- and server-side scripting

### Server-side

The server is where the Web page and other content lives. The server sends pages to the user/client on request. The process is:

1. the user requests a Web page from the server
2. the script in the page is interpreted by the server creating or changing the page content to suit the user and the occasion and/or passing data around
3. the page in its final form is sent to the user and then cannot be changed using server-side scripting

The use of HTML forms or clever links allow data to be sent to the server and processed. The results may come back as a second Web page.

**Server-side scripting tends to be used for allowing users to have individual accounts and providing data from databases.** It allows a level of privacy, personalization and provision of information that is very powerful. E-commerce, MMORPGs and social networking sites all rely heavily on server-side scripting.

PHP and ASP.net are the two main technologies for server-side scripting.

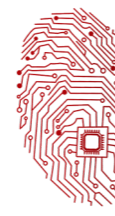
The script is interpreted by the server meaning that it will always work the same way. Server-side scripts are never seen by the user (so they can't copy your code). They run on the server and generate results which are sent to the user. Running all these scripts puts a lot of load onto a server but none on the user's system.

### The combination

A site such as Google, Amazon, Facebook or StumbleUpon will use both types of scripting:

- server-side handles logging in, personal information and preferences and provides the specific data which the user wants (and allows new data to be stored)
- client-side makes the page interactive, displaying or sorting data in different ways if the user asks for that by clicking on elements with event triggers





## Topic: 1.2.3 Client- and server-side scripting

To insert a JavaScript into an HTML page, we use the <script> tag, inside the <script> tag we use the "type" attribute to define the scripting language.

EXAMPLE :-

```
<html>
<body>
< script type="text/JavaScript">
document.write("Hello World!");
</script>
</body>
</html>
```

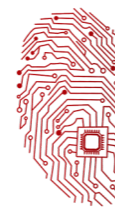
### JavaScript code and blocks

- JavaScript code is a sequence of JavaScript statements.
- Each statement is executed by the browser in the sequence they are written
- JavaScript statements are grouped together in a block.
- Blocks start with a left curly bracket and ends with a right curly bracket.
- The purpose of blocks is to make the sequence of statements execute together.

### JavaScript operators

- Arithmetic operators (+, -, \*, % MOD, ++ Increment, -- Decrement)
- Assignment operator (=)
- Comparison operators (== Equal to, === Exactly equal to, != Not equal to, <, >, <=, >=)
- Logical operators (&& AND, || OR, ! NOT)





## Topic: 1.2.3 Client- and server-side scripting

### Alert Box:

- Zak** An alert box is often used if you want to make sure information comes through to the user.
- Zak** When an alert box pops up, the user will have to click "OK" to proceed.
- Zak** Syntax: `alert("sometext");`

### Confirm Box:

- Zak** A confirm box is often used if you want the user to verify or accept something.
- Zak** When a confirm box pops up, the user will have to click either "OK" or "cancel" to proceed.
- Zak** If the user clicks "OK", the box returns true. If the user clicks "cancel", the box returns false.
- Zak** Syntax: `confirm("sometext");`

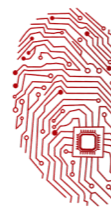
### Prompt Box:

- Zak** A prompt box is often used if you want the user to input a value before entering a page.
- Zak** When a prompt box pops up, the user will have to click either "OK" or "cancel" to proceed after entering an input value.
- Zak** If the user clicks "OK" the box returns the input value. If the user clicks "cancel" the box returns null.
- Zak** Syntax: `prompt("sometext", "defaultvalue");`

### JavaScript functions

- Zak** A function contains a code that will be executed by an event or by a call to the function.
- Zak** We may call a function from anywhere within a page
- Zak** Syntax:- `Function function name(var1, var2...varx) {Some code}`





## Topic: 1.2.3 Client- and server-side scripting

### JavaScript events

- Events are the beating heart of any JavaScript application.
- This gives us an overview of what event handling is, what its problems are and how to write proper cross-browser scripts
- Without events there are no scripts.
- Whenever a user of JavaScript takes action, he causes an event.

### Event modeling

This is example of event modeling in which we displays the date when a button is clicked:

```
<html>
<body>

<h1>My First Web Page</h1>

<p>My first paragraph.</p>

<button type="button" onclick="myFunction()">Try it</button>

<script>
function myFunction()
{
document.write(Date());
}
</script>
</body>
</html>
```

Result:

## My First Web Page

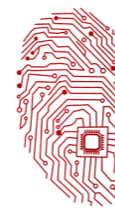
My first paragraph.

Try it

Result:

Sat Nov 08 2014 20:22:30 GMT+0500 (Pakistan Standard Time)





### Topic: 1.2.3 Client- and server-side scripting

Server side scripting as mentioned earlier is a web server technology in which a user's request is fulfilled by running a script directly on the web server to generate dynamic web pages. A common scripting language for server scripting is "PHP"

#### Uses of PHP:

- Zak Performs system functions i.e. from files on a system it can create, open, read, write and close them.
- Zak Can handle forms, i.e. gather data from files, save data to file, through email you send data, and return data to the user.
- Zak Add , delete or modify elements within your database through PHP
- Zak It can restrict users to access some pages of any website
- Zak It can encrypt data.

#### Basic Syntax for PHP:

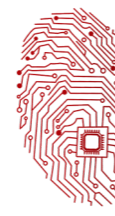
- Zak The PHP parsing engine needs a way to differentiate PHP code from other elements in the page. The mechanism for doing so is known as "escaping to PHP".
- Zak We have canonical PHP tags.
- Zak The most universally effective PHP tag style is "<?php... ?>"

#### Ways of sending information:

There are 2 ways of sending information to the web browser:

1. The **GET** Method
2. The **POST** Method





### Topic: 1.2.3 Client- and server-side scripting

#### The GET method

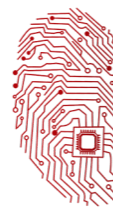
- Before the browser sends the information, it encodes it using a scheme called URL encoding.
- In this scheme, name/value pairs are joined with equal signs and different pairs are separated by the ampersand sign (&).

**Name1=value1&name2=value2&name3=value3**

- Spaces are removed and replaced with the "+" character and any other non-alphanumeric characters are replaced with a hexadecimal values.
- The get method sends the encoded user information appended to the page request.
- The page and the encoded information are separated by the "?" character.  

```
"http://www.test.com/index.htm?name1=value1&name2=value2"
```
- The get method produces a long string that appears in your server logs, in the browser's location "box".
- The GET method is restricted to send up to 1024 characters only
- Never use GET method if you have a password or other sensitive information to be sent to the user.
- It can't be used to send binary data, like images or word documents, to the server.
- The data sent by the GET method can be accessed using the QUERY\_STRING environment variable.
- The PHP provides \$\_GET associative array to access all the sent information using the GET method.





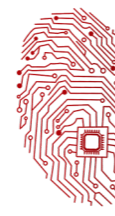
### Topic: 1.2.3 Client- and server-side scripting

Here is an example of simple PHP code using the GET method:

```
<?php
if( $_GET["name"] || $_GET["age"] )
{
echo "Welcome ". $_GET['name']. "<br />";
echo "You are ". $_GET['age']. " years old.";
exit();
}
?>
<html>
<body>
<form action="<?php $_PHP_SELF ?>" method="GET">
  Name: <input type="text" name="name" />
  Age: <input type="text" name="age" />
<input type="submit" />
</form>
</body>
</html>
```







### Topic: 1.2.3 Client- and server-side scripting

#### The POST method:

The POST method transfers information via HTTP headers. The information is encoded as described in case of GET method and put into a header called "QUERY\_STRING".

- The POST method does not have any restriction on data size to be sent.
- The POST method can be used to send ASCII as well as binary data.
- The data sent by **POST** method goes through HTTP header so the security depends on the HTTP protocol. By using Secure HTTP you can make sure that your information is secure.
- The PHP provides "\$\_POST" associative array to access all the sent information using POST method.

Here is an example of simple PHP code using the POST method:

```
<?php
if( $_POST["name"] || $_POST["age"] )
{
echo "Welcome ". $_POST['name']. "<br />";
echo "You are ". $_POST['age']. " years old.";
exit();
}
?>
<html>
<body>
<form action="<?php $_PHP_SELF ?>" method="POST">

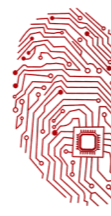
    Name: <input type="text" name="name" />
    Age: <input type="text" name="age" />

<input type="submit" />
</form>
</body>
</html>
```

#### PHP variables

- All variables in PHP are denoted with a leading dollar sign "\$".
- The value of variable is the value of its most recent assignment.
- PHP does a good job of automatically converting types from one to another when necessary.
- Variables in PHP do not have intrinsic types.  
(a variable does not know in advance whether it will be used to store a number or a string of characters.)





### Topic: 1.2.3 Client- and server-side scripting

#### Data types in PHP

- INTEGERS:** are whole numbers, without a decimal point, like "574".
- DOUBLES:** are floating point numbers, like "3.146".
- BOOLEANS:** have only two possible values either "true" or "false".
- NULL:** null is a special type that only has one value i.e. "null".
- STRINGS:** are sequences of characters like "hello friends"
- Arrays:** arrays are names and indexed collection of other values.

#### PHP control structures

##### if.....else statement :

Use this statement if you want to execute some code if a condition is true and another code if a condition is false.

*Example:*

```
if (condition)
    Code to be executed if condition is true;
else
    Code to be executed if condition is false;
```

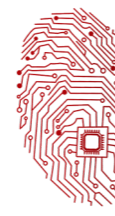
##### else if statement:

If you want to execute some code if one of the several conditions are true, use this statement.

*Example:*

```
if (condition)
    Code to be executed if condition is true;
else if (condition)
    Code to be executed if condition is true
Else
    Code to be executed if condition is false;
```





### Topic: 1.2.3 Client- and server-side scripting

#### Switch statement:

If you want to select one of many blocks of code to be executed , use the switch statement.

*Example:*

```
switch(expression)
{
case label 1
    Code to be executed if expression=label 1;
    break;
case label 2
    Code to be executed if expression= label 2;
    break;
default:
    code to be executed if expression is different from both label 1 and
    label 2;
```

