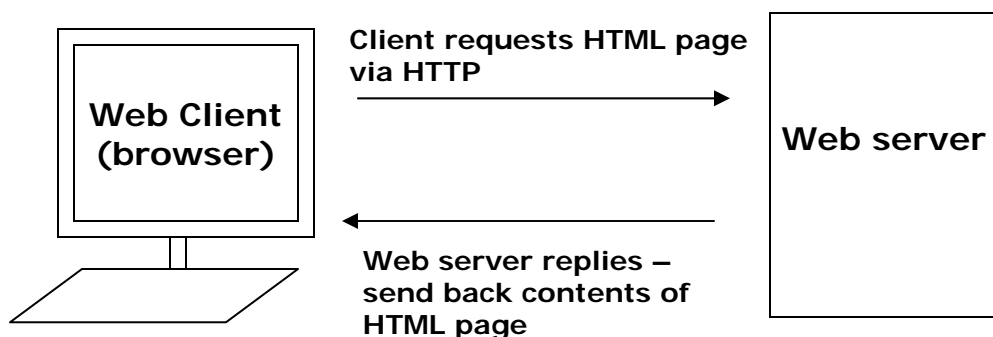


Scripting Languages - *What are they & how are they used?*

Dynamic VS Static Web content

1) Static Content: HTTP – HTML

- **HTTP** – Hypertext Transfer Protocol – used for retrieving simple web content.
- **HTML** – Hypertext Markup Language – the standard document type of the internet.



HTTP-HTML only enables users to retrieve **static content**. In other words, when you need to get another HTML page via HTTP, you need to initiate a full page reload. The HTML page being requested must exist as a static document prior to the request.

2) Data in a web page can be build dynamically –

2 technologies used: **client side** and **server side**.

Client side – content of a web page can change dynamically without submitting the page to a server.

Examples: Flash, Javascript, Java Applets.

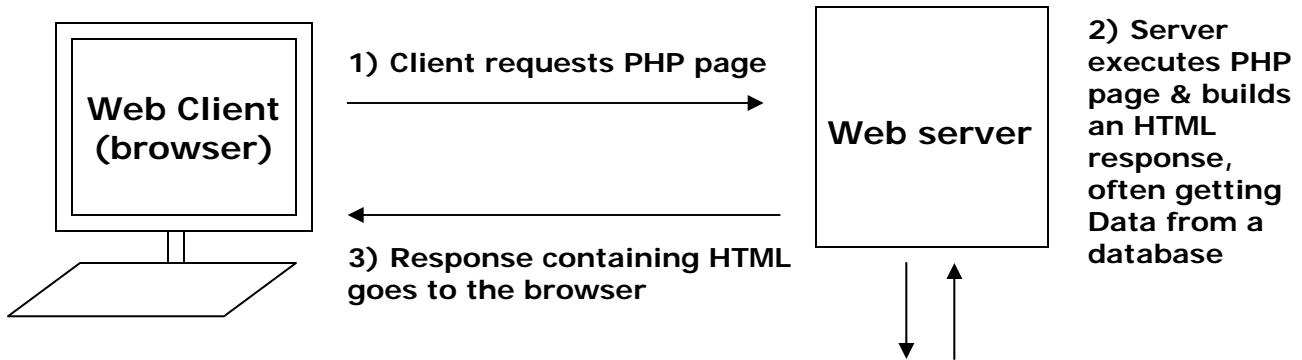
Server side – A request goes out to the server and it builds the web page on the fly, often using information from a database.

Examples: PHP, ASP (Active Server Pages), Perl/CGI, Cold Fusion, Java Server Pages (JSP), Ruby on Rails.

Some examples of server side technologies: Online catalogs and shopping carts, Content Management Systems (Drupal, Joomla, etc.), Web 2.0.

Server-side scripting

Scripting languages that are **server side** must run on a web hosting machine – i.e. with a hosting company that supports the scripting language, or you can set up your own computer to be a web host (i.e. using WAMP or MAMP).



Example 1

What does PHP code look like?

```
<?php //all PHP code goes inside these ?>
```

(Comments)

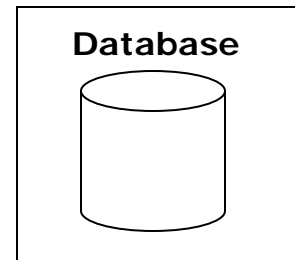
Here is an example file. Lets call it: myPage.php

```
<?php
//connect to database
$db = mysql_connect("localhost", mydb, mypasswd);
mysql_select_db("mydb", $db);

//SQL statement
$sql = "SELECT Type FROM Animals WHERE
GetsFleas = 'No'";

//code that queries the database using $sql
$result = mysql_query($sql);

?>
<!-- here is the html code -->
<html>
<body>
<p>Here are my results:</p>
<?php
while ($row = mysql_fetch_array($result)) {
    echo $row['Type'], '<br>';
}
?>
</body>
</html>
```



Database Types:

MySQL,
Postgres SQL,
SQL Server
(for ASP)

Database Server

Data gets stored in a table. A sample database table called "Animals" might look like:

Type	GetsFleas
Dog	Yes
Cat	Yes
Ferret	Yes
Fish	No
Camel	Maybe
Snake	No
Horse	Maybe

Here are my results:
Fish
Snake

PHP code lives inside the web page on the server, but the server interprets it and translates it into HTML, so when you do a “**view source**” in your browser on **myPage.php**, it looks like:

```
<html>
<body>
  <p>Here are my results: </p>
  Fish<br>
  Snake<br>
</body>
</html>
```

Not all scripting languages mix the code part with the HTML part. In Perl/CGI scripting, the script is separate from the HTML page. The HTML file gets read by the script, which will then search for certain markers within the HTML file and replace them with other values. The advantage of separating the code from the HTML is that the designers don't have to deal with it.

Example 2: A simple web form: myPage.php

When the form is submitted, the script will call itself again, and it will get the value of the radio button that was selected. It will query the database and retrieve all of the records in the table that satisfy the criteria: “maybe.” The results will be displayed when the page reloads.

Gets Fleas: Yes No Maybe

Results:

myPage.php

```
<?php

//this part will be skipped when the page is entered for the first time, because it
//will evaluate to 'false'. When the form is submitted with the radio button
//selected, it will evaluate to 'true.'

if ($_POST['myfield']) { // "myfield" is the name of the form field

    //connect to database
    $db = mysql_connect("localhost", mydb, mypasswd);
    mysql_select_db("mydb", $db);

    //build the SQL query
    $sql = "SELECT * FROM Animals WHERE GetsFleas = " . $_POST['myfield'] . "";
    $result = mysql_query($sql);
}
?>

<html>
<body>
<!-- the form will call the page again and post the form fields so the script can use
them -->
<form action="myPage.php" method="post">
Gets Fleas: <input type="radio" name="myfield" value="Yes">Yes
           <input type="radio" name="myfield" value="No">No
           <input type="radio" name="myfield" value="Maybe">Maybe
           <input type="submit" value="go">
</form><br>

Results: <br><br>
<?php
//loop through the results of the SQL query and write them out
while ($row = mysql_fetch_array($result)) {
    echo $row['Type'], '<br>';
}
?>
</body>
</html>
```

Gets Fleas: Yes No Maybe

Results:

Camel
Horse

Client-Side Scripting

Javascript – Uses the Document Object Model (DOM)

- You can change values of form fields
- You can swap images, change text, & modify CSS attributes
 - example:** `document.getElementById('myfield').value = "hey"`
sets the form field: "myfield" to the value: "hey"
- All this is done without reloading the page (i.e. without submitting it to the server)

Advantages:

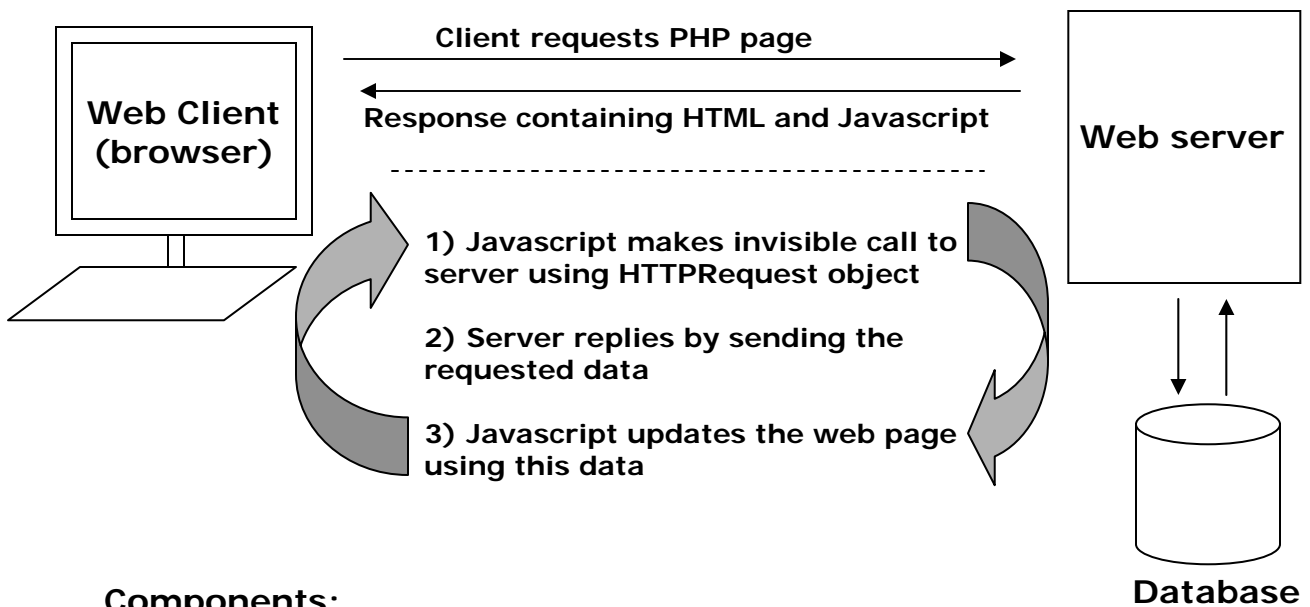
- Faster
- Don't need a hosting account that runs PHP (or other scripting language)

Disadvantages:

- Not good for handling a lot of data

Unless....

You use **AJAX** (Asynchronous Javascript and XML)



Components:

- Javascript
- XMLHttpRequest Object – enables Javascript to access server asynchronously – performs in background while user keeps working. The page does not reload like it would for a normal PHP request
- A server-side technology such as PHP or other scripting language.