

So Client (browser) Side processing (interpreting/reading code):

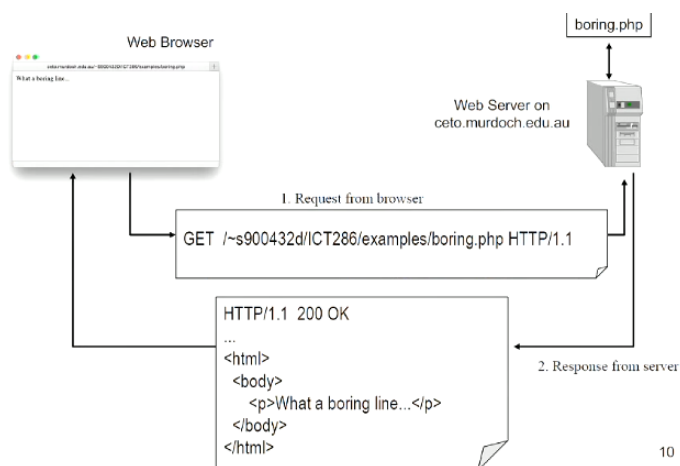
- HTML
- CSS
- Javascript

Why Server-side scripting:

- HTML + CSS only produces static web pages
- Client-side JavaScript is dynamic but only on the client side
 - It can't get access to resources on the server (eg the database)
 - Think: Siri we say something Siri will respond by information via database sadly client-side can't do that because they don't have access to server side and we can't store all the server database to Client side database
 - There is certain information which we cannot be processed on the client because security reasons eg passwords
- Servers-side technologies like PHP, JSD, ASP.NET not only allow us to give users access to their services but have the services processed by the server

Process of server-side scripting:

- Browser request URL
- Web server (apache) finds requested file php script
- Web server invokes (PHP) *interpreter* to turn the script code to HTML
- Web server sends HTML (result of interpret) back to browser
- Browser reads and displays HTML
- So output of PHP is pure HTML file



10

Process of client-side scripting: (JavaScript)

- Browser request URL
- Web server sends HTML and script code back to browser

- Browser interprets the script code and displays appropriately

PHP:

- Developed by Rasmus Lerdorf in 1994 and initially stands for personal home page now it stands for “PHP: Hypertext Preprocessor”
- PHP is a popular, general purpose scripting language that is especially suited to all kinds of dynamic web development and can be embedded into HTML

Functions of PHP:

- *Form handling/processing*
- Data exchanges between client and server (Ajax)
- File processing (access the files)
- Database access

Advantages of PHP:

- Enables the development of more dynamic web pages. PHP can verify forms, react to end-user input and access files and database
- PHP runs on the web server, so it can access files or database on or connected to the web server
- Open source and is free to use and download
- Constantly under development and is not limited to a platform
- Works well with MySQL

PHP modes:

- Copy
 - If it encounters HTML code, it copies the code to the output
- Interpret mode
 - If it encounters PHP code, it interprets the code and replaces the PHP code with its output

Primitive datatypes:

- Scalar types
 - Boolean
 - Integer
 - Double
 - String
- Compound types
 - Array
 - Object

String functions:

Function	Meaning
\$num = strlen (\$string);	\$num will contain the number of characters in \$string.
\$lower = strtolower (\$string);	\$lower will contain \$string all in lower case.
\$upper = strtoupper (\$string);	\$upper will contain \$string all in upper case.
\$new = ucfirst (\$string);	\$new will contain \$string with the first character capitalized.
\$new = ucwords (\$string);	\$new will contain \$string with the first character of every word capitalized.
\$new = trim (\$string);	\$new will contain \$string with any spaces from the beginning or end of the string removed.
\$new = chop (\$string);	\$new will contain \$string with any spaces from the end of the string removed.
\$part = substr (\$string, x, y);	\$part will contain a sub-string of \$string starting from position x, for y characters. 25

Number functions:

\$num2 = **round**(\$num1); \$num2 will contain \$num1 rounded to the nearest integer

\$num2 = **round**(\$num1, 2); \$num2 will contain \$num1 rounded to 2 decimal places

\$num2 = **ceil**(\$num1); \$num2 will contain the smallest integer greater than or equal to \$num1

28

*Form Processing: *

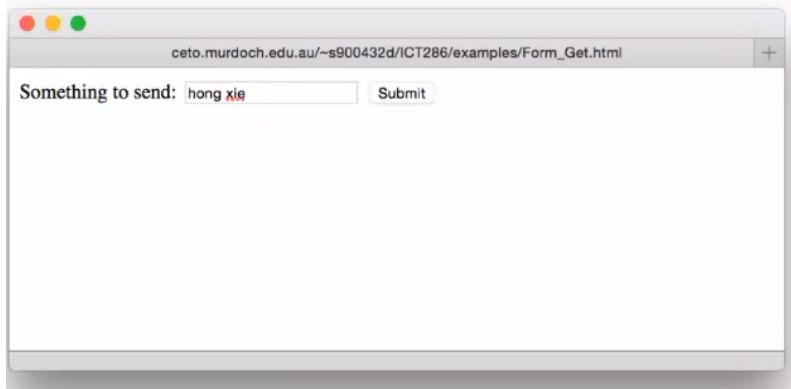
- One of the functions of PHP is form processing
- *Form tag* is used to reference PHP and it has two attributes
 - Action:
 - Specifies the URL of the script to run on the server-side when the form is submitted

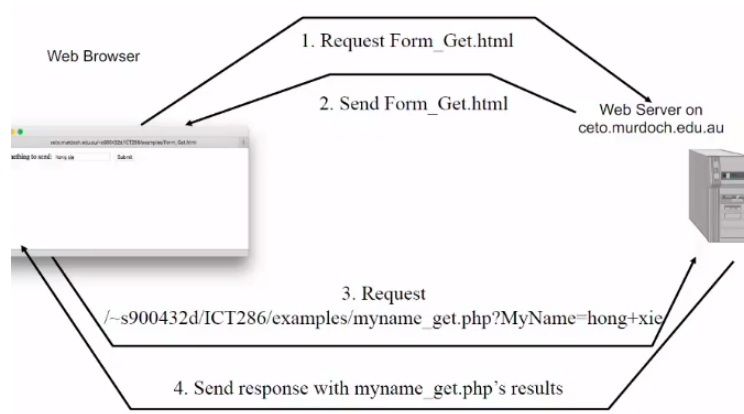
- Methods:
 - Set to either `*post*` or `*get*`; defines the argument format used to send data to the script
- HTTP `*GET*` method
 - Where the user inputs from the form, will be sent to the server script in the form of query strings
- HTTP `*POST*` method (user will not see what is sent to server-side ie URL. USE THIS)
 - Where the user inputs from the form are included in the body of the request method
- `*Name attributes*` In HTML will be available in the PHP script as a global array but the way in which you reference it in PHP depends on the method
 - HTTP `*GET*` method

```
process the form using  
$_GET[ "UserName" ]
```

- HTTP `*POST*` method

```
$_POST[ "UserName" ]
```





55

