

## HTML Basics(w3schools.com, 2013)

### What is HTML?

HTML is a language for describing web pages. HTML stands for Hyper Text Markup Language. HTML is a markup language A markup language is a set of markup tags. The tags describe document content. HTML documents contain HTML tags and plain text. HTML documents are also called web pages

### HTML Tags(elements)

HTML markup tags are usually called HTML tags. HTML tags are keywords (tag names) surrounded by angle brackets like <html>. HTML tags normally come in pairs like <b> and </b>. The first tag in a pair is the start tag, the second tag is the end tag. The end tag is written like the start tag, with a forward slash before the tag name. Start and end tags are also called opening tags and closing tags. HTML tags are not case sensitive: <P> means the same as <p>. Many web sites use uppercase HTML tags. HTML elements with no content are called empty elements. <br> is an empty element without a closing tag (the <br> tag defines a line break).

`<tagname>content</tagname>`

### HTML structure

```
<!DOCTYPE html>
<html>
<head>
<title>Title of the document</title>
</head>
<body>
    The content of the document.....
</body>
</html>
```

### The HTML <head> Element

The <head> element is a container for all the head elements. Elements inside <head> can include scripts, instruct the browser where to find style sheets, provide meta information, and more. The following tags can be added to the head section: <title>, <style>, <meta>, <link>, <script>, <noscript>, and <base>. The <title> tag defines the title of the document. The <title> element is required in all HTML/XHTML documents. The <title> element: defines a title in the browser toolbar provides a title for the page when it is added to favorites displays a title for the page in search-engine results

## Some important HTML tags

### *HTML Paragraphs*

HTML paragraphs are defined with the <p> tag.

```
<p>This is a paragraph.</p>  
<p>This is another paragraph.</p>
```

### *HTML Links*

HTML links are defined with the <a> tag.

```
<a href="http://www.w3schools.com">This is a link</a>
```

### *HTML Images*

HTML images are defined with the <img> tag.

```

```

### **HTML Attributes**

HTML elements can have attributes. Attributes provide additional information about an element. Attributes are always specified in the start tag. Attributes come in name/value pairs like: name="value". Example: HTML links are defined with the <a> tag. The link address is specified in the href attribute:

```
<a href="http://www.w3schools.com">This is a link</a>
```

Always Quote Attribute Values. Attribute values should always be enclosed in quotes. Double style quotes are the most common, but single style quotes are also allowed.

### **HTML Formatting Tags**

HTML uses tags like <b> and <i> for formatting output, like bold or italic text. These HTML tags are called formatting tags (look at table 1 for a complete reference).

| Tag      | Description  |
|----------|--|
| <b>      | Defines bold text                                    |
| <em>     | Defines emphasized text                              |
| <i>      | Defines a part of text in an alternate voice or mood |
| <small>  | Defines smaller text                                 |
| <strong> | Defines important text                               |
| <sub>    | Defines subscripted text                             |

|        |                                 |
|--------|---------------------------------|
| <sup>  | Defines superscripted text      |
| <ins>  | Defines inserted text           |
| <del>  | Defines deleted text            |
| <mark> | Defines marked/highlighted text |

**1. Table – HTML Text Formatting Tags**

## PHP Basics (php.net, 2013)

### What is PHP?

PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Example</title>
  </head>
  <body>

    <?php
      echo "Hi, I'm a PHP script!";
    ?>

  </body>
</html>
```

Instead of lots of commands to output HTML (as seen in C or Perl), PHP pages contain HTML with embedded code that does "something" (in this case, output "Hi, I'm a PHP script!"). The PHP code is enclosed in special start and end processing instructions `<?php` and `?>` that allow you to jump into and out of "PHP mode."

What distinguishes PHP from something like client-side JavaScript is that the code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of running that script, but would not know what the underlying code was. You can even configure your web server to process all your HTML files with PHP, and then there's really no way that users can tell what you have up your sleeve.

The best things in using PHP are that it is extremely simple for a newcomer, but offers many advanced features for a professional programmer. Don't be afraid reading the long list of PHP's features. You can jump in, in a short time, and start writing simple scripts in a few hours.

Although PHP's development is focused on server-side scripting, you can do much more with it.

## **What can PHP do?**

Anything. PHP is mainly focused on server-side scripting, so you can do anything any other CGI program can do, such as collect form data, generate dynamic page content, or send and receive cookies. But PHP can do much more.

There are three main areas where PHP scripts are used.

**Server-side scripting.** This is the most traditional and main target field for PHP. You need three things to make this work. The PHP parser (CGI or server module), a web server and a web browser. You need to run the web server, with a connected PHP installation. You can access the PHP program output with a web browser, viewing the PHP page through the server. All these can run on your home machine if you are just experimenting with PHP programming.

**Command line scripting.** You can make a PHP script to run it without any server or browser. You only need the PHP parser to use it this way. This type of usage is ideal for scripts regularly executed using cron (on \*nix or Linux) or Task Scheduler (on Windows). These scripts can also be used for simple text processing tasks.

**Writing desktop applications.** PHP is probably not the very best language to create a desktop application with a graphical user interface, but if you know PHP very well, and would like to use some advanced PHP features in your client-side applications you can also use PHP-GTK to write such programs. You also have the ability to write cross-platform applications this way. PHP-GTK is an extension to PHP, not available in the main distribution.

With PHP, you have the freedom of choosing an operating system and a web server. Furthermore, you also have the choice of using procedural programming or object oriented programming (OOP), or a mixture of them both.

With PHP you are not limited to output HTML. PHP's abilities includes outputting images, PDF files and even Flash movies

## **Basic syntax**

### ***PHP tags***

When PHP parses a file, it looks for opening and closing tags, which are `<?php` and `?>` which tell PHP to start and stop interpreting the code between them. Parsing in this manner allows PHP to be embedded in all sorts of different documents, as everything outside of a pair of opening and closing tags is ignored by the PHP parser. PHP also allows for short tags `<?` and `?>`