# BASIC HTML and FORMS INTRODUCTION

## 1.

## HTML

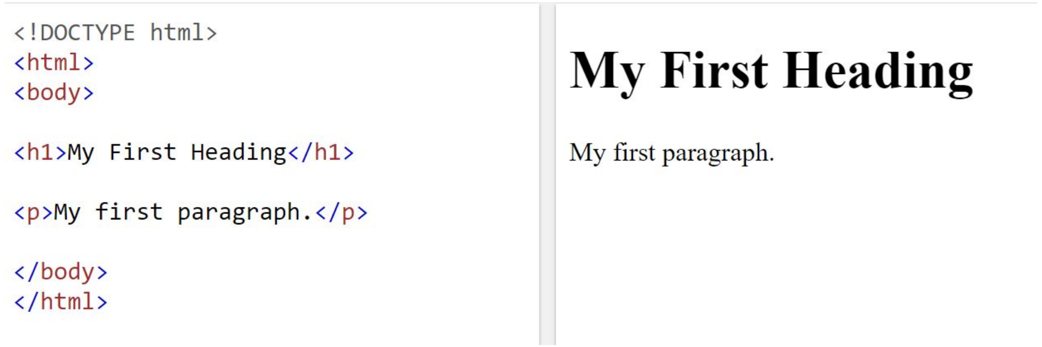
Let’s start with
the basics

# INTRODUCTION TO HTML

All HTML documents must start with a document type declaration: **<!DOCTYPE html>.**

The HTML document itself begins with **<html>** and ends with **</html>**.

The visible part of the HTML document is between **<body>** and **</body>**.



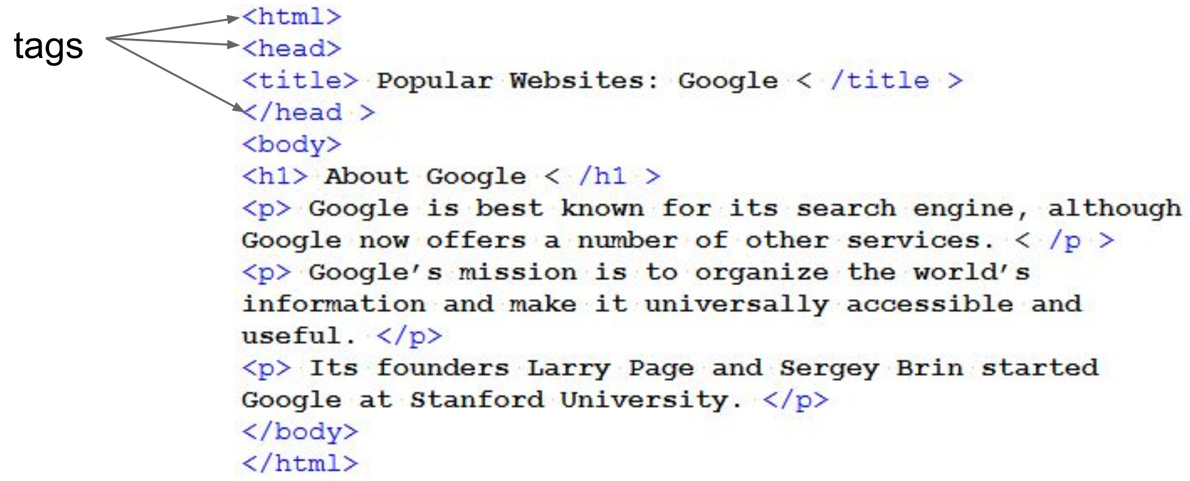
```
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>

<p>My first paragraph.</p>

</body>
</html>
```

My First Heading

My first paragraph.

## INTRODUCTION TO HTML

You don't need any special programs to write web pages — you can simply use any text editor (like Notepad++, gedit etc)



```
<html>
<head>
<title> Popular Websites: Google < /title >
</head >
<body>
<h1> About Google < /h1 >
<p> Google is best known for its search engine, although
Google now offers a number of other services. < /p >
<p> Google's mission is to organize the world's
information and make it universally accessible and
useful. </p>
<p> Its founders Larry Page and Sergey Brin started
Google at Stanford University. </p>
</body>
</html>
```

## HTML Elements

An HTML element usually consists of a **start** tag and **end** tag, with the content inserted in between:

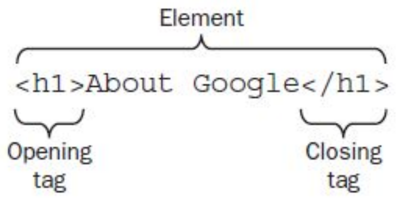
```
<tagname>Content goes here...</tagname>
```

The HTML element is everything from the start tag to the end tag:

```
<p>My first paragraph.</p>
```

## HTML tags

An example of HTML tags is shown next

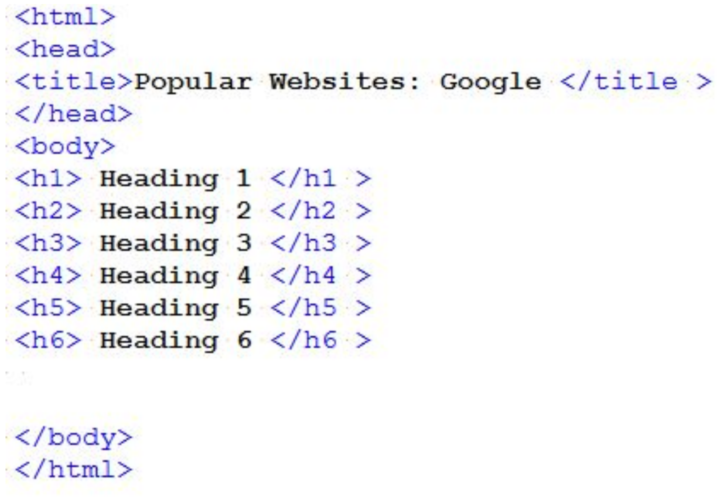


The text inside the angled brackets explains the purpose of the tag

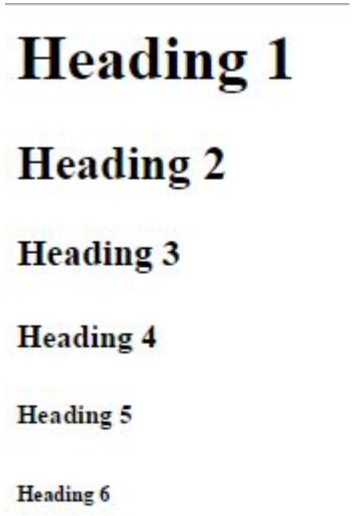
h1 indicates that it is a level 1 heading (or top - level heading)

Subheadings are also allowed ( <h2> , <h3> , <h4> , <h5> ,  and <h6> ).
Every document you create will contain some form of text <h1> , <h2> , <h3> , <h4> , <h5> , <h6> , <p> , <br/>

## Creating Headings Using <hn> Elements



```
<html>
<head>
<title>Popular Websites: Google </title >
</head>
<body>
<h1> Heading 1 </h1 >
<h2> Heading 2 </h2 >
<h3> Heading 3 </h3 >
<h4> Heading 4 </h4 >
<h5> Heading 5 </h5 >
<h6> Heading 6 </h6 >


</body>
</html>
```



Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

“

You don't need any special programs to write web pages, you can simply use any text editor (like Notepad++, gedit etc)

## HTML attributes

All HTML elements can have attributes.

Attributes provide additional information about an element.

Attributes are always specified in the start tag.

Attributes usually come in name/value pairs like: `name="value"`

## BIG CONCEPT

Attributes Tell Us About Elements

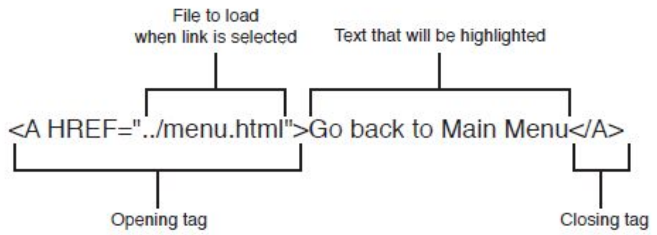
## Creating Links

To create a link in HTML, you need two things:

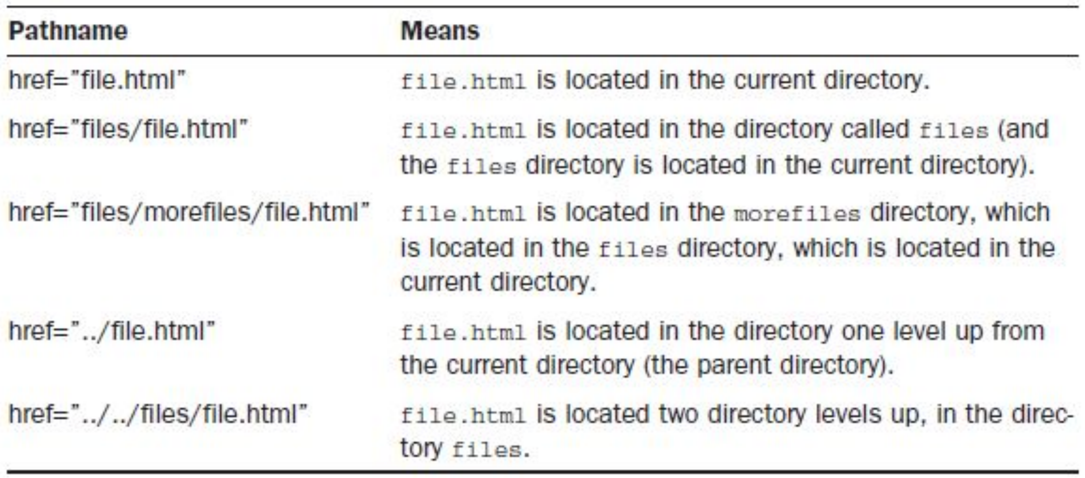
The name of the URL to which you want to link.

The text that will serve as the clickable link.

We use the HTML link tag <a>…</a>.

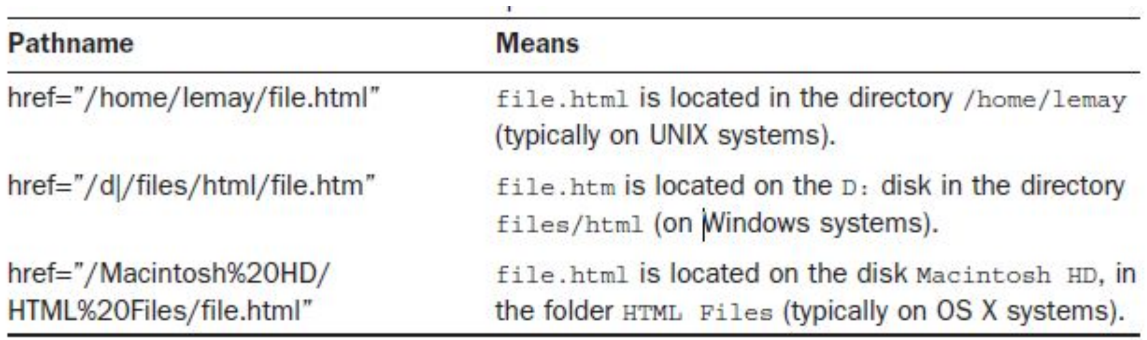

File to load when link is selected
Text that will be highlighted
<A HREF="../menu.html">Go back to Main Menu</A>
Opening tag
Closing tag

## Linking local pages using relative pathnames



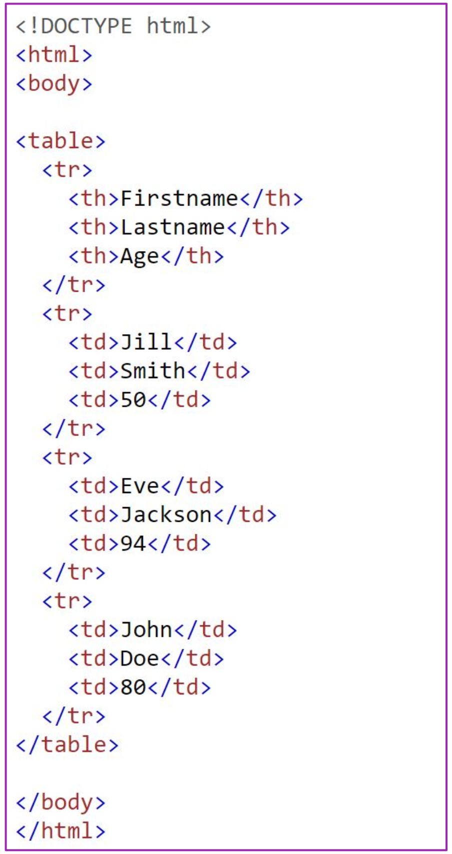
| Pathname | Means |
|---|---|
| href="file.html" | `file.html` is located in the current directory. |
| href="files/file.html" | `file.html` is located in the directory called `files` (and the `files` directory is located in the current directory). |
| href="files/morefiles/file.html" | `file.html` is located in the `morefiles` directory, which is located in the `files` directory, which is located in the current directory. |
| href="../file.html" | `file.html` is located in the directory one level up from the current directory (the parent directory). |
| href="../../files/file.html" | `file.html` is located two directory levels up, in the directory `files`. |

## Linking local pages using absolute pathnames



| Pathname | Means |
|---|---|
| href="/home/lemay/file.html" | `file.html` is located in the directory `/home/lemay` (typically on UNIX systems). |
| href="/d\|/files/html/file.htm" | `file.htm` is located on the `D:` disk in the directory `files/html` (on Windows systems). |
| href="/Macintosh%20HD/HTML%20Files/file.html" | `file.html` is located on the disk `Macintosh HD`, in the folder `HTML Files` (typically on OS X systems). |

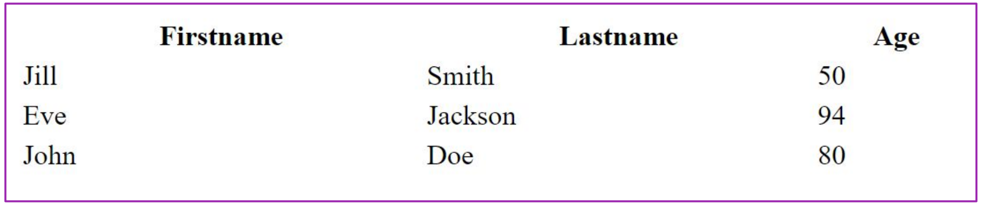
## HTML TABLES



```
<!DOCTYPE html>
<html>
<body>

<table>
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
  <tr>
    <td>John</td>
    <td>Doe</td>
    <td>80</td>
  </tr>
</table>

</body>
</html>
```

An HTML table is defined with the **<table>** tag.

Each table row is defined with the **<tr>** tag.
A table header is defined with the **<th>** tag. By default, table headings are bold and centered.
A table data/cell is defined with the **<td>** tag.



| Firstname | Lastname | Age |
|---|---|---|
| Jill | Smith | 50 |
| Eve | Jackson | 94 |
| John | Doe | 80 |

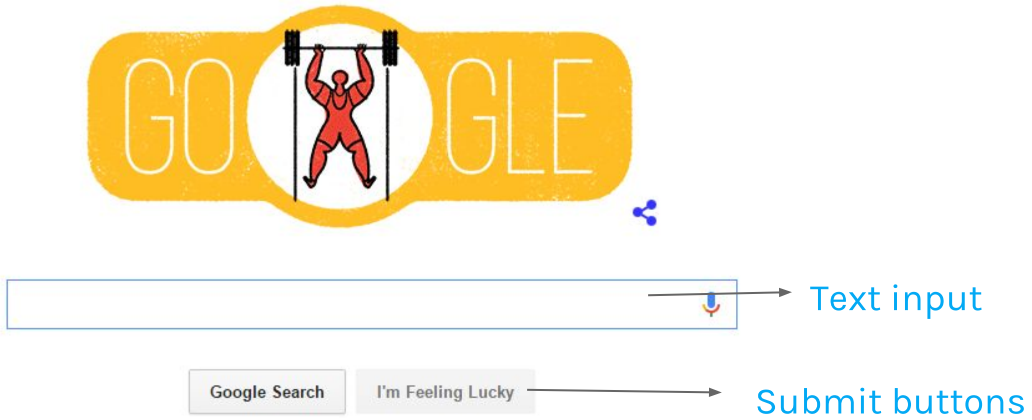
## 2.

## Forms

Let’s start with
the basics

# Introducing Forms

Here is an example of a form



## The <form> Element

The HTML **<form>** element defines a form that is used to collect user input:

```
<form>
.
form elements
.
</form>
```

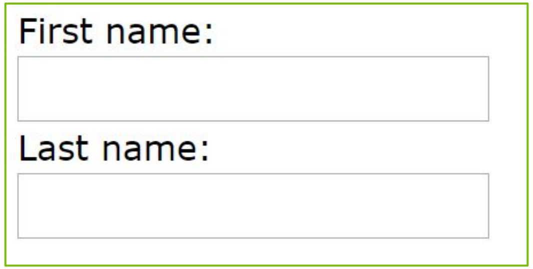
Form elements are different types of input elements, like text fields, checkboxes, radio buttons, submit buttons, and more.

## TEXT Input

**<input type="text">** defines a one-line input field for text input:

```
<form>
  First name:<br>
  <input type="text" name="firstname"><br>
  Last name:<br>
  <input type="text" name="lastname">
</form>
```

This is how it will look like in a browser:


First name:

Last name:

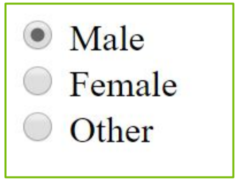
## RADIO button Input

**<input type="radio">** defines a **radio button**.

Radio buttons let a user select ONE of a limited number of choices:

```
<form>
  <input type="radio" name="gender" value="male" checked> Male<br>
  <input type="radio" name="gender" value="female"> Female<br>
  <input type="radio" name="gender" value="other"> Other
</form>
```

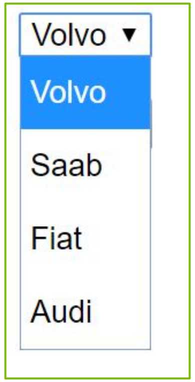
This is how it will look like in a browser:



## SELECT Input

The **<select>** element defines a **drop-down list**:

```
<select name="cars">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```



The **<option>** elements defines an option that can be selected.
By default, the first item in the drop-down list is selected.
To define a pre-selected option, add the **selected** attribute to the option:

```
<option value="fiat" selected>Fiat</option>
```

## BIG CONCEPT

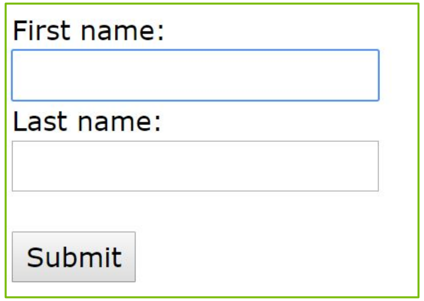
An HTML form contains **form elements**.

## SUBMIT button

**<input type="submit">** defines a button for **submitting** the form data to a form-handler.

The form-handler is typically a server page with a script for processing input data.  The form-handler is specified in the form's **action** attribute:

```
<form action="SOME_ACTION">
  First name:<br>
  <input type="text" name="firstname" value="Mickey"><br>
  Last name:<br>
  <input type="text" name="lastname" value="Mouse"><br><br>
  <input type="submit" value="Submit">
</form>
```



## Forms: buttons

| Attribute | Purpose |
|---|---|
| type | Specifies the type of button you want and takes one of the following values: submit , reset , or button . |
| name | Provides a name for the button. |
| value | Enables you to specify what the text on the button should display. |
| onclick | Used to trigger a script when the user clicks the button; the value of this attribute is the script that should be run. |

## THE ACTION ATTRIBUTE

The **action** attribute defines the action to be performed when the form is submitted.

Normally, the form data is sent to a web page on the server when the user clicks on the submit button.

If the action attribute is omitted, the action is set to the current page.

```
<form action="/some-action">
```

## THE TARGET ATTRIBUTE

The **target** attribute specifies if the submitted result will open in a new browser tab, a frame, or in the current window.

The default value is "**_self**" which means the form will be submitted in the current window.

To make the form result open in a new browser tab, use the value "**_blank**":

```
<form action="/some-action" target="_blank">
```

## THE METHOD ATTRIBUTE

The **method** attribute specifies the HTTP method (**GET** or **POST**) to be used when submitting the form data:

```
<form action="/some-action" method="get">
```

Or

```
<form action="/some-action" method="post">
```

## THE NAME ATTRIBUTE

Each input field must have a **name** attribute to be submitted.

If the name attribute is omitted, the data of that input field will not be sent at all.

This example **will only submit the "Last name"** input field:

```
<form action="/some-action">
  First name:<br>
  <input type="text" value="Mickey"><br>
  Last name:<br>
  <input type="text" name="lastname" value="Mouse"><br><br>
  <input type="submit" value="Submit">
</form>
```

## HTTP get

When you send form data to the server using the HTTP get method, the form data is appended to the URL and will be **visible in the page address field**:

```
/some-action?firstname=Mickey&lastname=Mouse
```

## HTTP post

Using the HTTP post method, the form data is sent *transparently* in what is known as the HTTP headers. (While you do not see these headers, they are not, strictly speaking, secure on their own).

If you are sending sensitive information such as credit card details, the data should be sent under a Secure Sockets Layer , or SSL , and they should be in encrypted.

If the previous form was sent using the post method, it could be represented like this in the HTTP headers:

```
User-agent: MSIE 7
Content-Type: application/x-www-form-urlencoded
Content-length: 35
...other headers go here...
FirstName=Bob & LastName=LetMeIn
```

## HTTP post or get?

You should **not use the HTTP get** method when:

- You are dealing with sensitive information, such as passwords or credit card details (because the sensitive form data would be visible as part of a URL).
- You are updating a data source such as a database or spreadsheet (because someone could make up URLs that would alter your data source).
- Your form contains a file upload control (because uploaded files cannot be passed in the URL).
- Your users might enter non - ASCII characters such as Hebrew or Cyrillic characters.

In these circumstances, you should use the HTTP **post** method.

## BIG CONCEPT

The web application needs to be able to receive input from the form and act on it.

## CREDITS

For the purpose of creating the slides for this tutorial we used the following sources

1. Beginning HTML, XHTML, CSS, and JavaScript, Jon Duckett,Wrox
2. HTML5 & CSS3 for the Real World: 2nd Edition - SitePoint Premium
3. https://www.w3schools.com

## THANKS!

Any questions?

You can find me at akolovou@di.uoa.gr