

HTML5

Alexandro Disla

HTML 5

W3School

Page 1 (starting point)

Html5 = HyperText Markup Language.-> describes what is inside a page.

HTML5 creates the foundation for all website.

CSS3 makes web pages look good. Javascript makes web page intelligent.

html tags = small drunks of html code.

index.html is the first web servers send out when you visite a site.

`<!DOCTYPE html>` : tells browser, we are using latest version of html. `<p></p>`: paragraph tag -> add a line break before and after the tags. `<body></body>`: body tag `<html></html>`: html tag -> debut et fin html tag ``: image tag pa femen `<head></head>` et `<title></title>`: n'apparait pas dans le corps; titre de la page.

Version Html:

- html 5
- html 4.01
- Xhtml 1.0

Html Rules:

1. Lowercase test-(sauf `<!DOCTYPE html>`).
2. clean tag nesting (proper tag nesting).
3. Keep the tag where they belong<<90% of the tags belong on the body section>>.
4. close your tags properly.

Html page rules:

1. html opening tag
2. head tags (division)
3. body tags (division)
4. html closing tag

Page 2 (hyperlinks)

- website are just html pages connected together.
- you must create an index.html page because the web server, by default, send out the index.html whenever a consumer visite the website.

- you connect the pages together with an html link tags (hyperlinks).

`` : It's the anchor tags. **href** is the tag attribute and this tag require a string.

`<hr>` : Horizontale rule.

Heading tags

```
<h1>InTheBody</h1> \forall i \in \{ 1,\cdots,6\}
```

mailto (key:value)

- open your default email app on your machine.

```
<a href="mailto: xxx@xxx.xx"></a>
<!-- mailto is the key -> key:value-->
```

target attributes

```
<a target="_value" href="xxx.html"><a>
```

__value or __blank hyperlink will route you on another tab in your browser. **__self** will keep you by default in the same tab.

Pro tips:

- Always specified the **__value** of the targetted page in the hyperlinks.

```
<a target="_google" href="https://google.com"><a>
```

Page 3

Paths = refer to how we point to folders and files in hyperlinks.

```
<!-- tells the browser to go up one level in the file directory.
"../ file.html"

tells the browser to go up two level in the file directory.
"../../file.html"
-->
```

client computer vs server computer

- server app turn computers into servers
- Response requested cycle: Client requested + server respond = Internet
- hosting = putting your website live on the web (payed or rent).
- web browser = called client app
- root directory = base folder of the website
- index page is, by default, the first page the web server will send
- The other pages are in the sub-directories

Page 4 (Essential Tag)

We have essential tags in html:

```
<h>
<p>
<a>
<ul>
<ol>
<li>
<div>
```

```
<!-- don't have closing tag-->
<br>
<img>
```

We have some basics tags:

```
<html>
<hi>
<p>
<title>
<!-- don't have a closing tag-->
<!DOCTYPE html>
<br>
<hr>
```


: linebreak

Smaller images = faster web page loading

we have 3 images types (jpeg,png,gif). The .jpeg format is the common picture that we all know (not very so flexible). The .png is very flexible format. The .gif is a very older version of png and it's known for his animation property.

tranparency : png>gif>jpeg

How to put css on the html?

we use the <style></style>.

Let's talk about html list:

- Ordered List logic:

```
<ol>
<li> </li>
<li> </li>
<li> </li>
</ol>
```

- Unordered List logic:

```
<ul>
<li> </li>
<li> </li>
<li> </li>
</ul>
```

we can do some tweak on the ordered list if we use the specific attribute.

```
<ol type="1">
<!-- "1" is the default value
we can modify this value.
For example:
"a", "A", "i"
-->
```

- Description list logic:

dt: data term

dd: data description

After each data term, we put the associate the data description

```
<dl>
<dt> coffee </dt>
<dd> something black we drink</dd>
<dt> Leighton Vander Esch</dt>
<dd> Linebacker of The Dallas Cowboys</dd>
</dl>
```

How to make a piece of text bold : `Bold Mentality`.

Page 5 (Block Level tag VS Incline Tag)

The `<div>` tag and the `` tag.

The division tag:

```
<div></div>
<!--allows you to divide the page into diverse section
      (Neutral container element tags)
-->
```

The span tag:

```
<span></span>
<!-- used to section off pieces of text and
don't mess with the flow of the text like the <div> do.
-->
```

css code : can turn an inline tag into a block tag.

Another example of inline tag, we have the anchor tag. we will display this tag as a block tag.

```
<style>
a {
    display:"block"
}
</style>
<a src="http://www.google.com">Google</a>
```

Page 6 (HTML forms)

Html forms are used to collect information from your page visitors. The submit button sends the information collected from the form to the web server for processing.

How to create the forms:

```
<form
name="Login"
action="name -target.ph"
method="get">

</form>
```

The Input tag in the forms:

```
<input
type="text"
name=""
```

```
size="">
<!-- type can be "password" and "submit"
```

Query String: place that the web browser can use to pass on information from one page to the next.

The name attribute is used by the web browser to move information from the form to the action page.

Example in Code:

```
<form
name="my-form"
action="my-form.php"
method="get">
<!--get is the default value for the method attribut.
post is another method and gives with the 3000 characters-->

<!-- Input: text, password, submit-->
<input type="text",name="",size="">
<!-- type can also be "password" or "submit-->

<!-- Let's have a dropdown list-->
<select name="">
<option value=""> </option>
<option value=""> </option>
<option value=""> </option>
</select>
<!-- text area-->
<textarea name="" rows=""cols="">
</textarea>
<!-- rows : height and cols: width-->

<!-- Input: checkbox, radio button-->
<input type="checkbox" name="" value="checked">
<!-- "checked" is a default value. "radio" is the other type-->
</form>
```

- The radio button can only select the option in your radio group.
- The name attribute is important for the checkbox and radio value because it creates the group.
- Size attribute set how wide your input box is and how many character can be added.

Page 7 (Html table)

Tables are designed to present tabular data.

Basics Table:

```
<table>
<!--table row and table data-->
<thead>
<tr><th></th><th></th><th></th></tr>
</thead>
<tbody>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tbody>
```

```

<tfoot>
<tr><td></td><td></td><td></td></tr>
</tfoot>
</table>
<!-- the number of td will condition the number of column.-->

```

A little css: `<style type="text/css"></style>` will be added by default. Why?

We had the `<!DOCTYPE html>` on the top of the page. And that's influe on all the default value.

`td colspan=""`: you can make the cells bigger.

`td rowspan=""`: you can make the cells bigger.

`<tr><th></th></tr>`: th is the tables heads tags inside the rows in columns.

A footer and Header:

`<thead></thead>`: To add a strict table head. It will always be display at the head of the table

`<tfoot><tfoot>`: the footer will always be display at the end of the able.

`<tbody><tbody>`: will specify the body of the table.

Page 8 (Head section And Meta)

`<html lang="en-US">` will set the language in the page. Due to parent-child inheritance, this attribute will be display everywhere in the page.

Meta tag allows you to add more information about the page.

```

<head>
<title> </title>
<meta charset="">
<meta name="" content="">
<!-- you can add your css and js codes
<style src=""> </style>
<script src=""> <script>
-->
</head>

```

It is best practice to add your javascript inside the body tag or in the footer tag.