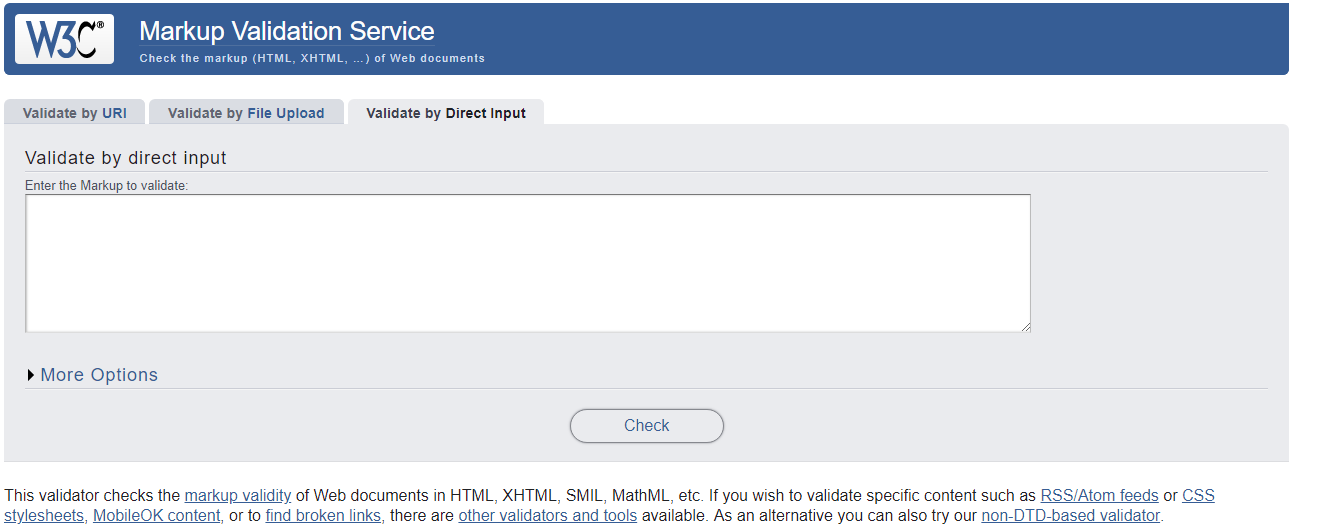
Validators

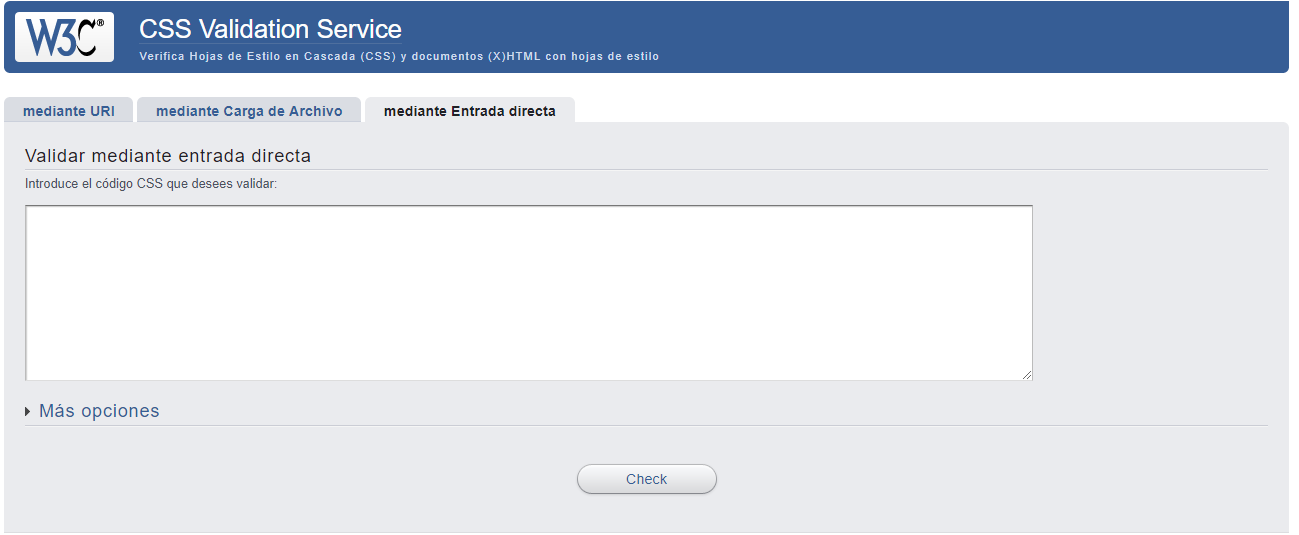
[**W3C validator**](https://validator.w3.org/)

<https://validator.w3.org/#validate_by_input>



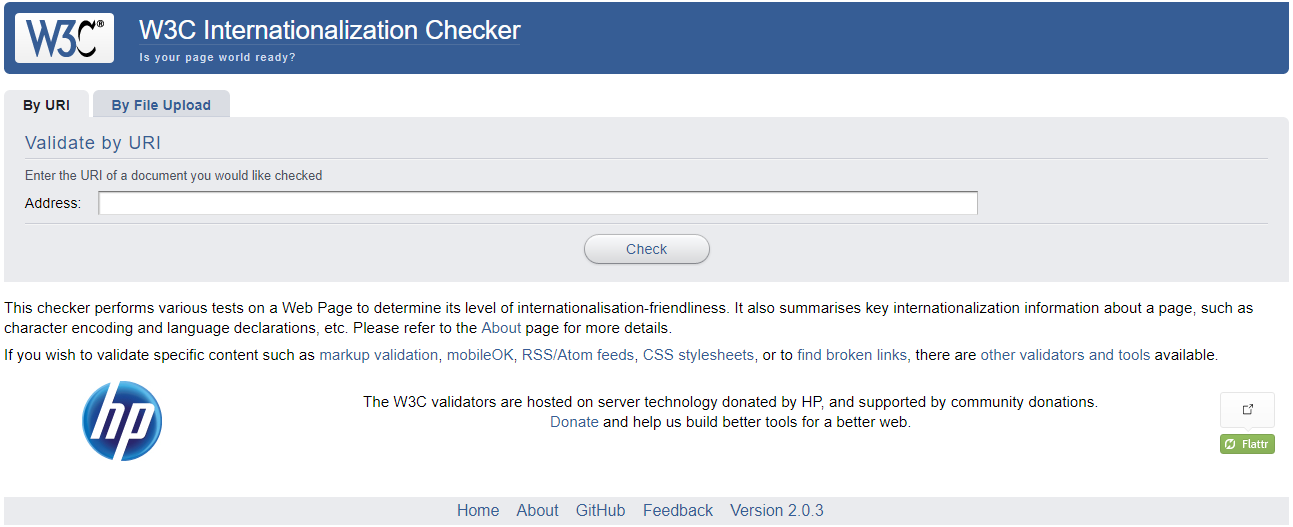
[**CSS validator**](https://jigsaw.w3.org/css-validator/)

<https://jigsaw.w3.org/css-validator/#validate_by_input>



[**W3C Internationalization Checker**](https://validator.w3.org/i18n-checker/)

<https://validator.w3.org/i18n-checker/>



[**W3C Link Checker**](http://validator.w3.org/checklink)

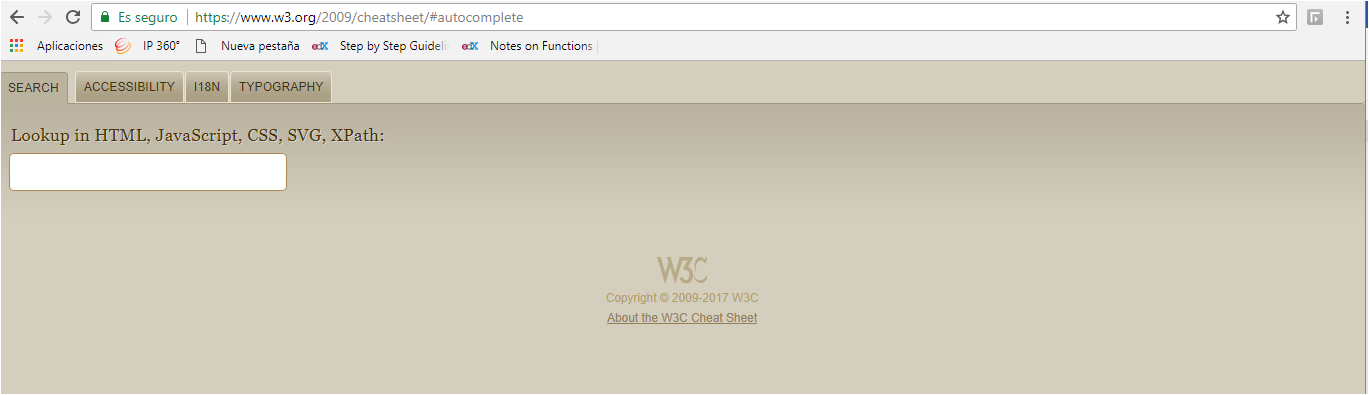
<https://validator.w3.org/checklink>



## **The W3C Cheatsheet**

The [W3C cheatsheet](http://www.w3.org/2009/cheatsheet/) provides quick access to useful information from a variety of specifications published by W3C. It aims at giving in a very compact and mobile-friendly format a compilation of useful knowledge extracted from W3C specifications, completed by summaries of guidelines developed at W3C, in particular the WCAG2 accessibility guidelines, the Mobile Web Best Practices, and a number of internationalization tips.

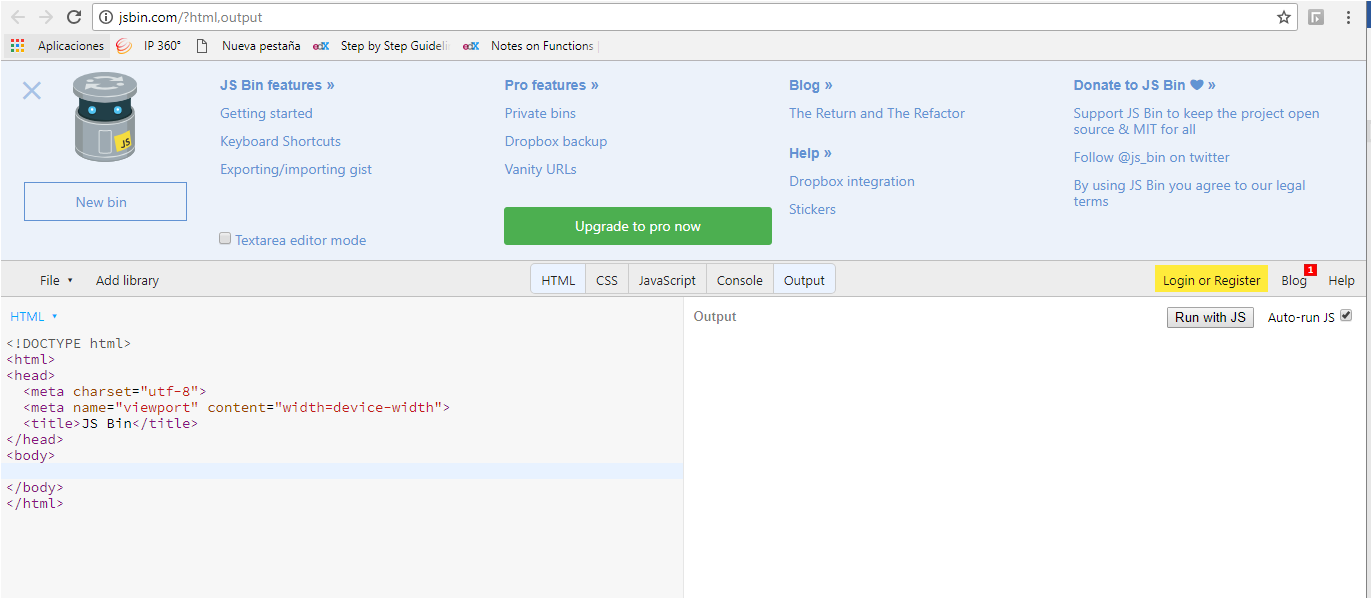
<https://www.w3.org/2009/cheatsheet/#autocomplete>



# Course tools

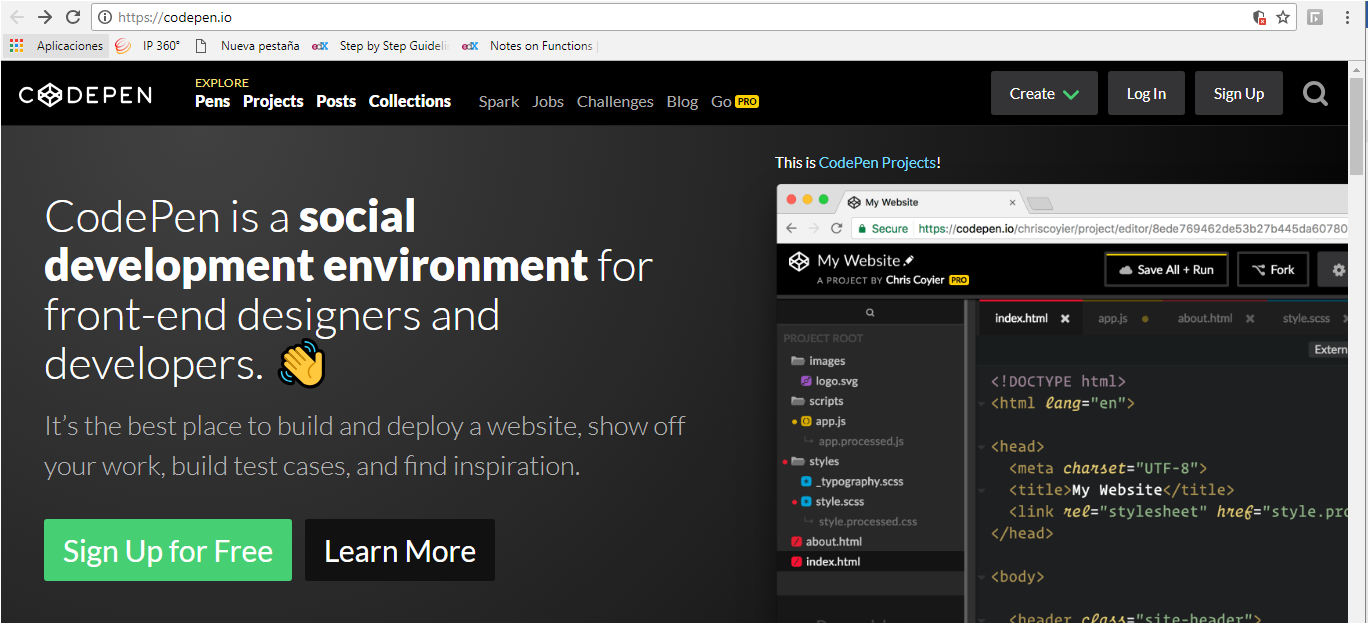
To help you practice during the whole duration of the course, you will use the following tools. Pretty much all the course's examples will actually use these tools.

* **JS Bin (**[**http://jsbin.com/**](http://jsbin.com/)**)**



JS Bin is an open source collaborative Web development debugging tool. Most of the examples you will find in this course are on jsbin. Tutorials can be found on the web (such as <http://code.tutsplus.com/tutorials/javascript-tools-of-the-trade-jsbin--net-36843>) or on YouTube. The tool is really simple, just open the link to the provided examples, look at the code, look at the result, etc. And you can modifiy the examples as you like, you can also modify / clone / save / share them. Keep in mind that it's always better to be logged in (it's free) if you do not want to loose your contributions/personnal work.

**CodePen (**[**http://codepen.io/**](http://codepen.io/)**)**



CodePen is an HTML, CSS, and JavaScript code editor that previews/showcases your code bits in your browser. It helps with cross-device testing, realtime remote pair programming and teaching.

Here's an article which will be of-interest if you use CodePen: [10 Cool Things You Can Do with CodePen and JavaScript](https://www.sitepoint.com/cool-things-codepen-javascript/), Chris Coyier, June 13, 2016.

There are many other handy tools such as [JSFiddle](http://jsfiddle.net/" \t "_blank), and [Dabblet](http://dabblet.com/" \t "_blank) (Lea Verou's tool that we will use extensively in a future CSS course). Please share your favorite tool on the discussion forum, and explain why! Share also your own code contributions, such as a nice canvas animation, a great looking HTML5 form, etc. Sharing them using JS Bin, or similar tools, would be really appreciated.

# Greater simplicity

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Page Title</title>

<link rel="stylesheet" href="style.css">

<script src="script.js"></script>

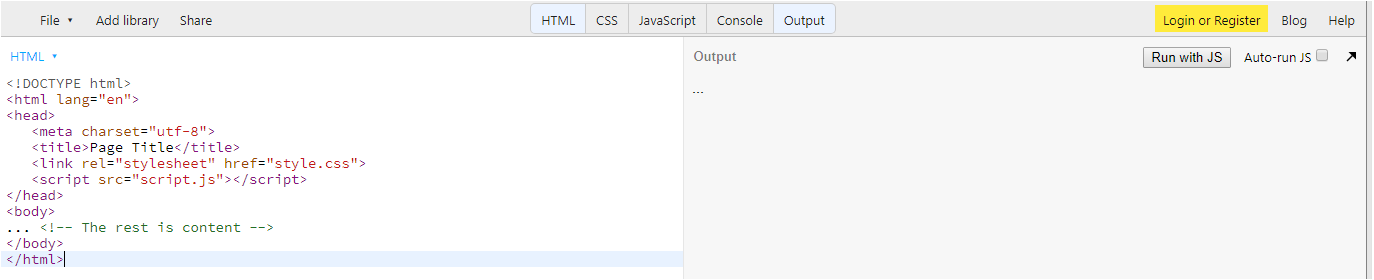
</head>

<body>

... <!-- The rest is content -->

</body>

</html>



## **Simpler Character Set Definition**

One word about the <meta charset="utf-8"> at line 4 in the HTML5 version: it is good practice to declare the character set of your document to protect against [a serious security risk](https://code.google.com/p/doctype-mirror/wiki/ArticleUtf7). For more details, please refer to the "Why Internationalization is important" section in the Course Intro chapter.

## **No more complicated DOCTYPE definitions**

The "Doctype", often called DTD (Document Type Declaration), is used by tools such as HTML validators (i.e.  [the W3C validator](http://validator.w3.org/)), and specifies the rules used by  an HTML or an XHTML page. These rules are contained in special documents called "Document Type Definitions" (also abbreviated as DTD), written in a language that may seem a bit barbaric to humans (they are intended to be read by software), and hosted by W3C.

DTDs are not used by current Web browsers to  validate the structure of an HTML page, as  they "read" the code without using the DTD to decipher it, using only "rules" contained in their own "HTML engine", but it is still preferable to indicate the doctype as modern browsers have several rendering engines that are chosen depending on the doctype.

Old HTML1 Web pages will not be rendered the same way as new HTML5 pages, since in the 90's some of them were written by hand and may contain errors, embedded HTML elements, etc.

With HTML4, doctype definitions looked like this: <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">, which was even more complicated as one had to choose between three different possibilities (doctypes could be transitional, strict, or frameset). Most of the time, the doctype definition was copied and pasted from one document to another and was nearly impossible to memorize.

With HTML5, there is only one way to indicate the doctype, and it's so simple there is no reason to forget it:

1. <!doctype html>

## **The "TYPE" attribute is optional**

With a rel="stylesheet" attribute, it is no longer necessary to indicate type="text/css" (from [the specification](http://www.w3.org/TR/html5/links.html#link-type-stylesheet): "the default type for resources given by the  [stylesheet](http://dev.w3.org/html5/spec/single-page.html#link-type-stylesheet)  keyword is text/css.")

The "type" attribute is not needed in HTML5, and even old browsers will use text/css as the default type for stylesheets today. So, either way, you can omit the "type" attribute altogether and use:

1. <link href="file.css" rel="stylesheet"/>

instead of:

1. <link href="file.css" rel="stylesheet" type="text/css"/>

We will not go into detail about the <link> element, but the fact that the type attribute is becoming optional shows the current direction taken by HTML5: towards greater simplicity.

Please see how to include a JavaScript file in our page:

1. <script src="script.js"></script>

Here again, the type attribute has been omitted. Just as a reminder, the old way to do the same thing:

1. <script type="text/javascript" src="script.js"></script>

## **More flexible syntax constraints**

If you look at the "minimal document" example, or at other examples in this course, you won't find a lot of differences compared to the same code in XHTML: attribute values are surrounded by quotes, all elements are written in lower case, etc. This is because we are used to writing this way, but HTML5 also supports a simplified syntax:

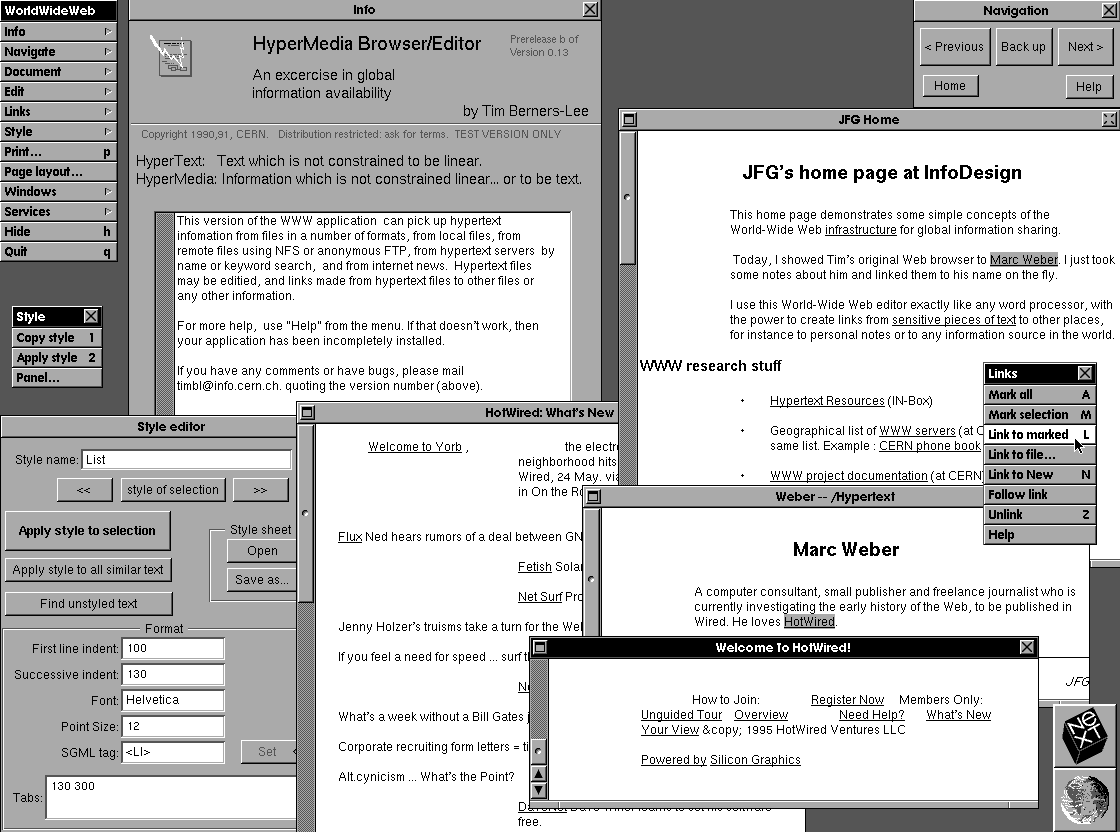
* Thanks to HTML5, you can omit quotes (not always, but most of the time) or use uppercase, lowercase or a combination of the two.
* Many elements no longer need a closing tag: </li>, </dt>, </dd>, </tr>, </th>, </td>, </thead>, </tfoot>, </tbody>, </option>, </optgroup>, </p> (in most cases), </head>, </body> and </html>. Older browsers often add closing tags automatically at render time. We recommend, however, closing tags that would naturally be closed: the ones that delimit a particular zone in the document.
* Attribute values only need to be quoted if they contain spaces or some non-alphanumeric characters, instead of writing <link rel="stylesheet" href="style.css">, we could have used <link rel=stylesheet href=style.css> instead. However, for compatibility with older browsers, it might be wiser to still use quotes...

## Week 1:

## History of HTML versions

## **HTML 1.0**

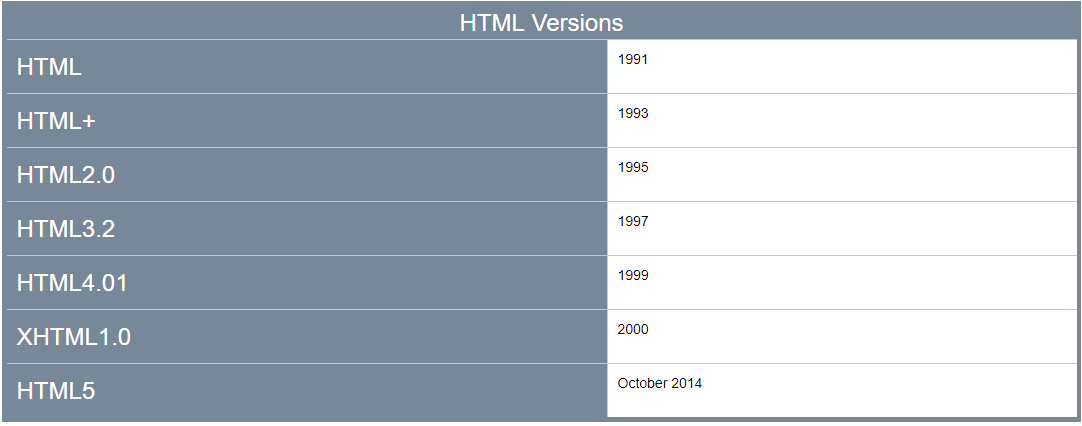
Below is a screenshot of Tim Berners-Lee's Browser Editor as developed in 1991-1992.



This was a true browser editor for the first version of HTML and ran on a NeXT workstation. Implemented in Objective-C, this very first browser in Web history made it easy to create, view and edit web documents. Hypertext Markup Language (First Version of HTML) was formally published in June 1993.

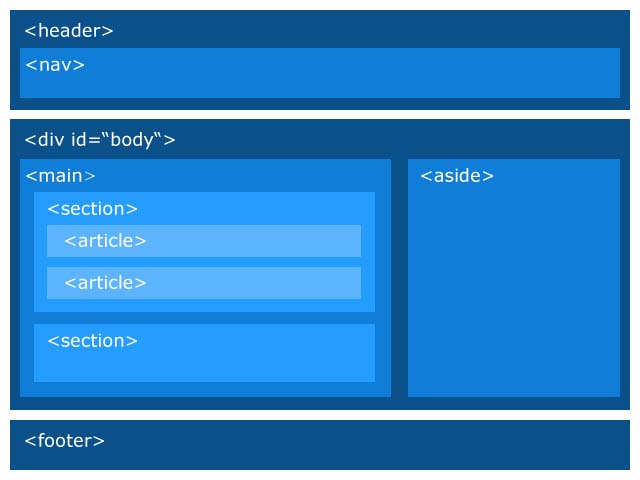
## **HTML versions**

HTML is an evolving language. For Web sites and pages created since 1991, however, it is easy to find out which HTML version they use. A Document Type Declaration, or DOCTYPE, is a piece of HTML code that states which version of HTML is being used. This declaration must appear at the very top of every Web page.  
For example: <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"> tells that the document uses the HTML4.01 version.



For those of you who are curious, the W3C published a document laying down the [HTML5 Differences from HTML4](http://www.w3.org/TR/html5-diff/) (the document was published in December 2014, shortly after the release of HTML5). Read also the [history section](http://www.w3.org/TR/2014/REC-html5-20141028/introduction.html#history-0) available in the HTML5.1 specification document.

# New structural elements



|  |  |
| --- | --- |
| **HTML5 structural elements with descriptions.** | |
| **HTML5 element** | **Description** |
| [<header>](http://www.w3.org/TR/html5/sections.html#the-header-element) | Introduction of "sectioning elements": an article, a section, the entire document (header page). Typically the header of a Web site that appears on top of each page, or a header of a long <article> or of a long <section> |
| [<footer>](http://www.w3.org/TR/html5/sections.html#the-footer-element) | Contains the footer of a site, a long <article>, or a long <section> |
| [<nav>](http://www.w3.org/TR/html5/sections.html#the-nav-element) | Section that contains the main navigation links (within the document or to other pages). |
| [<article>](http://www.w3.org/TR/html5/sections.html#the-article-element) | Independent content, which can be individually extracted from the document and syndicated (RSS or equivalent) without penalizing its understanding. Typically a blog post. |
| [<section>](http://www.w3.org/TR/html5/sections.html#the-section-element) | Generic section used to group different articles for different purposes or subjects, or to define the different sections of a single article. Generally used with a header. |
| [<time>](http://www.w3.org/TR/html5/text-level-semantics.html#the-time-element) | Used for marking up times and dates. |
| [<aside>](http://www.w3.org/TR/html5/sections.html#the-aside-element) | Section whose content is not necessarily directly related to the main content that surrounds it, but can provide additional information. |
| **<figure>** and **<figcaption>** | Used to encapsulate a figure as a single item, and contains a caption for the figure, respectively. |
| [<main>](http://www.w3.org/TR/html5/sections.html#the-aside-element) | The main element represents the main content of the body of a document or application. The main content area consists of content that is directly related to or expands upon the central topic of a document or central functionality of an application. **There can be only one <main> element in a document.** |