1. **First web page(WWW)**

The first web page went live on August 6, 1991.  It was dedicated to information on the World Wide Web project and was made by Tim Berners-Lee. It ran on a NeXT computer at the European Organization for Nuclear Research, CERN.

1. **Netscape Navigator**

**Netscape Navigator** was a [proprietary](https://en.wikipedia.org/wiki/Proprietary_software) [web browser](https://en.wikipedia.org/wiki/Web_browser), and the original browser of the [Netscape](https://en.wikipedia.org/wiki/Netscape_(web_browser)) line, from versions 1 to 4.08, and 9.x. It was the [flagship](https://en.wikipedia.org/wiki/Flagship) product of the [Netscape Communications Corp](https://en.wikipedia.org/wiki/Netscape) and was the dominant web browser in terms of [usage share](https://en.wikipedia.org/wiki/Usage_share_of_web_browsers) in the 1990s, but by 2002 its use had almost disappeared.[[2]](https://en.wikipedia.org/wiki/Netscape_Navigator#cite_note-2) This was primarily due to the increased use of [Microsoft](https://en.wikipedia.org/wiki/Microsoft)'s [Internet Explorer](https://en.wikipedia.org/wiki/Internet_Explorer) web browser software, and partly because the Netscape Corporation (later purchased by [AOL](https://en.wikipedia.org/wiki/AOL)) did not sustain Netscape Navigator's technical innovation in the late 1990s.

1. **Mosaic**

 Mosaic was also the first browser to display images inline with text instead of displaying images in a separate window.[[5]](https://en.wikipedia.org/wiki/Mosaic_(web_browser)#cite_note-faq-5) While often described as the first [graphical](https://en.wikipedia.org/wiki/Graphical_user_interface) web browser, Mosaic was preceded by [WorldWideWeb](https://en.wikipedia.org/wiki/WorldWideWeb" \o "WorldWideWeb), the lesser-known [Erwise](https://en.wikipedia.org/wiki/Erwise" \o "Erwise)[[6]](https://en.wikipedia.org/wiki/Mosaic_(web_browser)#cite_note-6) and [ViolaWWW](https://en.wikipedia.org/wiki/ViolaWWW" \o "ViolaWWW).

Mosaic was developed at the [National Center for Supercomputing Applications](https://en.wikipedia.org/wiki/National_Center_for_Supercomputing_Applications) (NCSA)[[5]](https://en.wikipedia.org/wiki/Mosaic_(web_browser)#cite_note-faq-5) at the [University of Illinois at Urbana–Champaign](https://en.wikipedia.org/wiki/University_of_Illinois_at_Urbana%E2%80%93Champaign) beginning in late 1992. NCSA released the browser in 1993,[[7]](https://en.wikipedia.org/wiki/Mosaic_(web_browser)#cite_note-mosaic_and_th_w3-7) and officially discontinued development and support on January 7, 1997.[[8]](https://en.wikipedia.org/wiki/Mosaic_(web_browser)#cite_note-8)

Starting in 1995 Mosaic lost market share to [Netscape Navigator](https://en.wikipedia.org/wiki/Netscape_Navigator), and by 1997 only had a tiny fraction of users left, by which time the project was discontinued. Microsoft licensed Mosaic to create [Internet Explorer](https://en.wikipedia.org/wiki/Internet_Explorer) in 1995.

1. **Canvas**

The HTML <canvas> element is used to draw **graphics**, on the fly, via JavaScript. The <canvas> element is only a container for **graphics**. You must use JavaScript to actually draw the **graphics**. Canvas has several methods for drawing paths, boxes, circles, text, and adding images.

1. **Difference between Scripting languages are programming languages**

Scripting languages are programming languages that don't require an explicit compilation step.

For example, in the normal case, you have to compile a C program before you can run it. But in the normal case, you don't have to compile a JavaScript program before you run it. So JavaScript is sometimes called a "scripting" language.

This line is getting more and more blurry since compilation can be so fast with modern hardware and modern compilation techniques. For instance, V8, the JavaScript engine in Google Chrome and used a lot outside of the browser as well, actually compiles the JavaScript code on the fly into machine code, rather than interpreting it. (In fact, V8's an optimizing two-phase compiler.)

Also note that whether a language is a "scripting" language or not can be more about the environment than the language. There's no reason you can't write a C interpreter and use it as a scripting language (and people have). There's also no reason you can't compile JavaScript to machine code and store that in an executable file (and people have). The language Ruby is a good example of this: The original implementation was entirely interpreted (a "scripting" language), but there are now multiple compilers for it.

Some examples of "scripting" languages (e.g., languages that are *traditionally* used without an explicit compilation step):

* Lua
* JavaScript
* VBScript and VBA
* Perl

And a small smattering of ones *traditionally* used with an explicit compilation step:

* C
* C++
* D
* Java *(but note that Java is compiled to bytecode, which is then interpreted and/or recompiled at runtime)*
* Pascal

...and then you have things like Python that sit in both camps: Python is widely used without a compilation step, but the main implementation (CPython) does that by compiling to bytecode on-the-fly and then running the bytecode in a VM, and it *can* write that bytecode out to files (.pyc, .pyo) for use without recompiling.

1. **Responsive Web Design using HTML and CSS**

Responsive Web Design is about using HTML and CSS to automatically resize, hide, shrink, or enlarge, a website, to make it look good on all devices (desktops, tablets, and phones):

When making responsive web pages, add the following <meta> element in all your web pages:

<meta name="viewport" content="width=device-width, initial-scale=1.0">

This will set the viewport of your page, which will give the browser instructions on how to control the page's dimensions and scaling.

1. **How to create links to sections on the same page in HTML**

**Using #top or #bottom**

The following examples use #top and #bottom with the [<a> tag](https://www.computerhope.com/jargon/h/html-a-tag.htm) and [href attribute](https://www.computerhope.com/jargon/h/href.htm) to create a link to that section of the page. This method is similar to using "id," but you don't have to pick a specific element. Click on "Top" or "Bottom" in the *Results* section to see what happens.

**Examples**

<a href="#top">Top</a>

<a href="#bottom">Bottom</a>

**Results**

[Top](https://www.computerhope.com/issues/ch000049.htm#top)

[Bottom](https://www.computerhope.com/issues/ch000049.htm#bottom)

**Using the id selector**

In [CSS](https://www.computerhope.com/jargon/c/css.htm), "[id](https://www.computerhope.com/jargon/i/id.htm)" is a selector that may be used to designate an area that a link should point to, similar to [anchor](https://www.computerhope.com/jargon/a/anchor.htm) in [HTML](https://www.computerhope.com/jargon/h/html.htm). The nice thing about using id is that you can create a link to any element on the page, rather than only the top or bottom. In the following sections, you'll see how to apply id to an [HTML tag](https://www.computerhope.com/jargon/t/tag.htm), and then how to link to it. This example will link to the opening paragraph at the top of this page.

**Use the #id selector**

<p id="opening">Hyperlinks are utilized by a web browser to move from one page to another...</p>

**Create a link to the selector**

<a href="#opening">Take me to the opening paragraph.</a>

**Result**

[Take me to the opening paragraph.](https://www.computerhope.com/issues/ch000049.htm#opening)