Continue

Cascading style sheet in html

style sheet language For other uses, see CSS (details page). The pseudo-element will guide you here. For more information about pseudofast symbols in chemistry, see Skeleton Formula § Pseudoelement symbols. For more information about using CSS in Wikipedia, see:Cscading Style Sheets. Cascading Style Sheets (CSS) File name extension .cssInternet media type text/cssUniform Type Identifier (UTI)public.cssInformation ByWorld Wide Web Consortium (W3C)Initial release 17.12.1996; 23 vuotta sitten (1996-12-17)Uusin julkaisuCSS 2.1 : Level 2 Revision 1 (Huhtikuu 12, 2016; 4 vuotta sitten (2016-04-12)) FormStyle-arkin kielen tyyppiContainer forStyle-säännöt HTML-elementeille (tunnisteet)Sisältää byHTML DocumentsOpen-muodon?YesWebsitewww.w3.org/TR/CS S/#css Ryöppyävät tyylisivut Tyylisivu CSS Zen Garden CSS Design Concepts -ruudun Zen-ruudun vaihto flexbox-ruudukko Filosofiat Taulukkottomat Responsiiviset Holy grail -työkalut Sass Less Stylus CSSTidy Comparisons CSS tukee Stylesheet-kieliä Casca tyyli sheetsvte HTML Dynaaminen HTML HTML5 äänipohja video XHTML Basic Mobile Profiili C-HTML HTML elementtiväli ja div HTML–määrite HTML-kehys HTML-editori Merkkikoodaus Unicode Kielikoodi AsiakirjaObjekti Malli Selaimen Objekti Malli Tyylisivut CSS Fontti perhe Web värit HTML-komentosarjat JavaScript WebGL WebCL W3C Validator WHATWG Quirks mode Web storage Rendering engine Comparisons Document markup languages HTML support XHTML 1.1 CASCADing Style Sheets (CSS) on tyylisivun kieli, jota käytetään kuvaamaan merkintäkielellä, kuten HTML-kielellä, kirjoitetun asiakirjan esitystä. [1] CSS is the cornerstone technology of the Web, along with HTML and JavaScript. [2] CSS is designed to allow you to distinguish between presentation and content, such as layout, colors, and fonts. [3] This separation can improve content availability, increase flexibility, and manage the definition of presentation properties, allow multiple web pages to share formatting by configuring the relevant CSS settings in a separate .css file that reduces the complexity and repetition of structured content, and allows the .css file to be cached to improve the page load speed between the pages that share the file and its formatting. Formatting and content separation also make it possible to display the same markup page in different styles in different rendering methods, such as screen, printed, audio (via voice-based browser or screen reader) and Braille-based touch devices. CSS also has rules on alternative formatting if content is used on a mobile device. [4] The CSS value of a name comes from the specified priority model to determine which style rule applies if more than one rule matches a specific element. This cascading priority system is predictable. CSS configuration is maintained by the World Wide Web Consortium Internet media type (MIME type) text/css is RFC 2318 (March 1998) with CSS. W3C has a free CSS verification service for CSS documents. [5] In addition to HTML, other tagging languages support CSS, including XHTML, standard XML, SVG, and XUL. The syntax CSS has a simple syntax and specifies the names of different style properties with multiple Keywords in English. A style sheet consists of a list of rules. Each rule or rule set consists of one or more selectors and definition blocks. In CSS, the switches indicate which part of the entry the style applies to by matching the tags and attributes of the entry. Switches can apply to: all types of elements, such as second-level labels h2 elements defined by the attribute, in particular: ID: unique identifier of the document category: an identifier that can mark multiple elements of document elements, based on how they are placed relative to other document trees. Categories and IDs are case sensitive, start with letters, and can contain alphanumeric characters, hyphens, and underscores. The class can be applied to any instance of elements. The id can only be used in one element. CSS switches use virtual classes to allow formatting based on data that is not included in the document tree. One example of a widely used pseudoclass is :hover, which recognizes content only when a user points to a visible element, usually by holding a mouse cursor over it. It is likened to a dial like:hover or #elementid:hover. The pseudoclass categorizes parts of a document, such as :link or :visited, while the pseudo element makes a selection that can consist of partial elements such as ::first-line or ::first-letter. [6] Selectors can be combined in many ways to achieve high specificity and flexibility. [7] Multiple switches can be mapped to the list of intermediate elements that determine the elements by location, element type, id, category, or any combination of both. The order of the electorate is important. For example, div .myClass {color: red;} applies to all myClass elements in the class that are inside the div elements, while .myClass div {color: red;} applies to all div elements that are in the myClass elements of the class. The following table summarizes the syntax of the switch that indicates the usage and the version of CSS that it has been deployed. [8] The pattern matches the element of the E 1 E:link element first defined at CsS level E is the source anchor of its hyperlink, whose item has not yet been visited (:link) or has already visited (:visited) 1 E:active a E element during certain user actions 1 E::first-line first formatted line E element 1 E::first letter The first formatted letter of E-element 1 .c all elements with class=c 1 #myid element with id 1 E.warning for an E-element whose class is a warning (the language of the document specifies category 1 E#myid an an element with an identifier equal to myid 1 .c#myid element with class=c and myid 1 E F A F element sub element from element E element 1 * any element 2 E[foo] E element with foo attribute 2 E[foo=bar] E element with foo attribute value exactly the same as bar. 2 E[foo~=bar] E-elements, whose foo attribute value is a list of a separate value from a space, one of which is exactly equal to bar 2 E[foo|=en] E element with hyphens separated by hyphens from values starting (from left) and en 2 with E:first-child a E element, mother's first subname 2 E:lang(fr) how to determine the language) 2 E ::pre-generated content before the contents of the E element after the contents created 2 E:: after the contents of the E-element after the contents of the 2 E > F element F element sub-element 2 E + F F element immediately preceded by E element 2 E[foo~=bar] E element with the foo attribute value starting exactly with the string bar. 3 E[foo$=bar] An E element with a foo attribute value ending exactly in bar 3 E[foo*=bar] in an E element with a foo attribute value containing an E element in the substring bar 3 E:root a, the main file of the document 3 E:nth-child(n) element E, the E element of the last child of the older n's last child (n), the parent's n' child , when calculating the last 3E type(n) E element, the E element of the nth sibling of type 3 N, when calculating the E element from the last 3 E:last child, the last child of its parent 3E:the first type E-element, the first sibling of type 3E: the last type E-element, only its parent's 3E:only type E-element, only s type 3E:blank E-element without children (including text nodes) 3 E :an item E-element that is the subject of the URI 3 E:enabled UI element E and is used in 3E :d UI element is disabled 3 E:revised selected ui element E (e.g. option button or check box) 3 E:not(s) E-element that does not correspond to the simple s 3 E – F element preceded by the E element 3 Notification Block A notification block consists of a list of constraint notifications. Each notification consists of a property, colons (:) and value. If there are multiple notifications in the block, the semicolon (;) shall be added to distinguish between each notification. After the last (or individual) insurance, an optional semicolon may be used. [9] The characteristics are defined in the CSS standard. Each property has a set of possible values. Some properties can affect any type of element, and others apply only to specific groups of elements. [10] [11] Values may be such as center or inherit or numeric numerical such as 200 pixels (200 pixels), 50 vw (50 percent view width) or 80% (80 percent of the width of the main element). Color values can be defined by keywords (e.g. red), hexadecimal values (e.g. #FF0000, abbreviated also by #F00), RGB values 0-255 on a scale (e.g. rgb(255, 0,0)), RGBA values that determine both colour and alpha transparency (e.g. rgba(255, 0,0,8)) or HSL or HSLA values (e.g. hsl(000, 100%, 50%), hsla(000, 100%, 50%, 80%)[12] Non-zero numeric values representing linear units of measure must contain a length unit that is either an alphabetic code or an abbreviation, such as 200 px or 50vw; or a percent sign, as in 80 percent. Some units – cm (centimeters); in (inches); mm (millimetre); pcs (pica); and pt (dot) – are absolute, which means that the rendered dimension does not depend on the structure of the page; Other – em (em); ex (ex) and px (pixel) – are relative, which means that factors such as the font size of the parent element can affect rendered measurement. These eight units were a feature of CSS 1[13] and were retained in all subsequent amendments. If the proposed CSS value and unit module level 3 is accepted as a W3C recommendation, it will offer seven other lengths: ch; K; rem; vh; vmax value; vmin; and vw. [14] Use before CSS almost all presentation attributes in HTML documents were included in the HTML entry. All font colors, background styles, element alignments, borders, and sizes had to be explicitly described, often repeatedly in HTML. CSS allows authors to transfer much of this data to another file, a style sheet, resulting in much simpler HTML code. For example, headings (h1 elements), subtitles (h2), subtitles (h3), etc. On printout and display, the font, size, color, and highlights of these elements are displayed. Before CSS, document authors who wanted to set such typographical properties, for example, for all h2 headers, had to repeat HTML presentation entries for each instance of this heading type. This made documents more complex, larger and more error-prone and harder to maintain. CSS allows you to separate a presentation from a structure. CSS can set color, font, text alignment, size, borders, spacing, layout, and many other writing style properties, and can do so independently on the screen and in printed views. CSS also defines non-visual styles, such as reading speed and emphasis for aural text readers. W3C has now dispainted the use of HTML marking for all presentation formats. [15] For example, the heading element specified in red text in the pre-CSS HTML section is written as follows: <h1>Number 1.</h1> CSS allows the same element to be encoded with style properties instead of: <h1 style=color: red;>Chapter 1.</h1> The benefits of this may not be immediately clear The power of CSS becomes clearer when style features are placed in an internal style element or, better still, in an external CSS file. For example, assume that the document contains a style element: All h1 elements in the document change to <style> h1 { color: red; } </style> automatically red without explicit code. If the author later wanted to make h1 elements blue, this could be done by changing the style element to a shape: instead of <style> h1 { color: blue; } </style> you would laboriously review the document and influence the color of each individual h1 element. Styles can also be placed in an external CSS file as described below and downloaded using syntax similar to: This detaches formatting from an HTML document and allows multiple <link href=path/to/file.css rel=stylesheet type=text/css> reshaping documents simply by editing a shared external CSS file. Sources CSS data can be provided from different sources. These sources can be a browser, user, and author. Author information can be further categorized into text, media type, precedence, selector specification, rule order, inheritance and definition of assets. Css style information can be in a separate document or embedded in an HTML document. Multiple style sheets can be imported. Different styles may be used according to the output device used; For example, the display version can be quite different from the printed version so that authors can tailor the presentation to suit each device. The style sheet with the highest priority controls the content screen. Declarations that are not defined in the primary source are passed to a lower priority source, such as user agent style. The process is called cascading. One of CSS's goals is to give users better control over the presentation. Someone who finds red headings hard to read can use a different style sheet. Depending on the browser and web site, the user can choose from different style sheets offered by designers, remove all added styles and view the site using the default browser style, or skip only the red heading style without changing the other attributes. CSS priority model (highest or minimum) Primary CSS source type Description 1 Importance !important note replaces previous priority types 2 in HTML element The inline A style sheet provided through html-style attribute 3 applies to all media types, if media-specific CSS is not configured 4 User configured Most browsers have accessibility: user-defined CSS 5 Selector specific A specific situational switch (#heading p) replaces the general definition 6 Rule order The last rule definition has a higher priority 7 Main Heritage If the property is not specified, that's it inherited from the CSS property definition of parent element 8 in the CSS rule or CSS inline style of the HTML document replaces the default browser value 9 Browser default Minimum priority: the browser default is determined by W3C initial value specifications Specificity Specificity refers to the relative weights of different rules. [16] It determines which styles apply to an element when more than one rule can apply. Based on specifications, a simple switch (e.g. H1) a specificity of 1, a class selector a specificity of 1,0 and a identification selector of 1,0,0. Since the specificity values do not run according to the decimal system, the commas are used to separate the numbers[17] (a CSS rule with 11 elements and 11 categories would be a specificity of 11,11, not 121). Thus, the following rule selectors lead to the declared specification: Selectors Specificity h1 {color: white;} 0, 0, 0,1 p em {color: green;} 0, 0, 0, 0, 2 .grape {color: red;} 0, 0, 1, 0 p.bright {color: blue;} 0, 0, 1, 1 s.bright em.dark {color: yellow;} 0, 0, 2 #id218 {color: brown;} 0, 1, 0, 0 style= 1, 0, 0, 0 DOCTYPE html><html><head><meta charset=utf-8> <style> #xyz {color: blue; } </style></head><body><p id=xyz style=color: green;>Assigning specificity</p></body></html> In the example above, the style definition overrides <style></style> on <head>, which could then be used in the style of several elements without repeating the changes. Example Based on the following single style sheet: h1 { color: pink; } For example, assume that there is an h1 element inside that contains an highlighted element (em): <h1>This illustrates inheritance</h1> If no color is specified for the em element, the highlighted word illustrate inherits the color of parent element h1. The style sheet h1 has a pink color, so the em element is also pink. The space between properties and switches is ignored. This snippet of code: body{overflow:hidden;background:#000000;background-image:url(images/bg.gif);background-repeat:no-repeat;background-position:left top;} functionally matches this: body { overflow: hidden; background color: #000000; wallpaper: url(images/bg.gif); background-repeat: no-repeat; background-position:left top; } One common way to format CSS against readability is to indent each feature and give it its own line. In addition to formatting CSS for readability, shorthand features can be used to write code faster, which is also processed faster when rendered:[19] body { overflow: hidden; background: #000 URL(images/bg.gif) no-repeat left top; } Positioning CSS 2.1 determines three positioning models: Normal flow Inline items are arranged in the same way as letters with text, one after another over available space until there is no more space, and then starts under a new line. Block items stack vertically, such as paragraphs and items in a bulleted list. A normal workflow also includes the relative position of block or text-voted items, as well as check-in screens. Floaters The floating object is taken out of the normal flow rate and, if possible, switches left or right in the available space. Other content then runs alongside the floating object. Absolute position An absolutely positioned object does not affect or affect the normal flow of other objects. It is located in its container regardless of other objects. [20] Position property property has four possible values. If the item is positioned in a way other than static, transitions and locations are set above, down, left, and right of other properties. Static Default values place an item in a normal workflow Relative Item is placed in a normal workflow, and then moves or moves from it. The following flow items have been opened as if the item had not been moved. Absolute Specifies the absolute position. The element is placed in relation to its nearest non-static pre-indies. Fixed Item is definitely placed in a fixed position on the screen, although when scrolling the rest of the document[20] Floats and The Floating Point property can have one of three values. Absolutely positioned or fixed objects cannot be floated. Other Others usually they flow around floats unless their clear property prevents them from doing so. to the left, the Object floats to the left of the line in which it would have appeared; other objects can flow around the right side of it. on the right, the Object floats to the right of the line in which it would have appeared; other objects can flow around the left side of it. clear Forces the element to appear to the left (clear:left), right (clear:right), or both sides (clear:both). [20] [21] Håkon Wium Lie, Technical Director of Opera Software and co-creator of CSS CSS network standards, first proposed Håkon Wium Lie on 10 September 2005. At the time, Lie was working with Tim Berners-Lee at CERN. [23] Several other web style sheet languages were proposed for the same reasons, and discussions on public mailing lists and within the World Wide Web Consortium led to the publication of the first W3C CSS Recommendation)[24] in 1996. In particular, Bert Bos' proposal had an impact; he became co-author of CSS1 and is considered a co-author of CSS 2. [25] Style sheets have existed in one form or another since Standard Generalized Markup Language (SGML) began in the 1980s, and CSS was developed to provide style sheets online. [26] One requirement of the web style sheet language was that style sheets come from different sources online. Therefore, existing style sheet languages, such as DSSSL and FOSI, were not appropriate. CSS, on the other hand, allows multiple style sheets to influence the style of a document by using cascading styles. [26] As HTML grew, it entinded a wider range of stylistic capabilities to meet the demands of web developers. This development gave the designer more control over the layout of the site at the price of more complex HTML. Variations of browser implementations, such as ViolaWWW and WorldWideWeb[27], made it difficult for the site to have a consistent appearance, and users were less difficult to display web content. The browser/editor developed by Tim Berners-Lee had style sheets encoded in the program with hard code. Therefore, style sheets could not be linked to web documents. [23] Cern's Robert Cailiau also wanted to separate the structure from the presentation so that different style sheets could describe different formats for printing, screen-based presentations, and editors. [27] Improving Web presentation capabilities was a source of interest for many in the online community, and nine different style sheet languages were proposed for the web-style mailing list. [26] Of these nine proposals, two had a particularly big impact on CSS: CSS: Cascading HTML style sheets[22] and stream-based style sheet suggestion (SSP). [25] [28] Two browsers operated at the test sites for the first proposals. Lie collaborated with Yves Lafon to implement CSS at Dave Raggett Arena Bert Bos implemented his own SSP proposal in the Argo browser. [25] Lie and Bos then worked together to develop the CSS standard (H was removed from the name because these style sheets could be applied not only to HTML but also to other tagging languages). [23] Lie's proposal was presented at the Mosaic and the Web conference (later www2) in Chicago, Illinois in 1994 and again with Bert Bos in 1995. [23] At this time, W3C had already been established and was interested in the development of CSS. It organised a workshop chaired by Steven Pemberton. This led to W3C adding CSS to the HTML Editorial Review Board (ERB). Lie and Bos were the main technical staff for this aspect of the project, which was also attended by other members, such as Thomas Reardon of Microsoft. In August 1996, Netscape Communication Corporation introduced an alternative style sheet language called JavaScript Style Sheets (JSSS). [23] The specification was never finalised and its dispainment is outdated. [32] By the end of 1996, CSS was ready to become official and the CSS Level 1 recommendation was published in December. The development of HTML, CSS and DOM had taken place in one group, the HTML Editorial Review Board (ERB). At the beginning of 1997, the ERB was divided into three working groups: the HTML Working Group, chaired by Dan Connolly of W3C; DOM Working Group chaired by Lauren Wood of SoftQuad; and the CSS working group chaired by Chris Lilley of the W3C. The CSS Working Group began to address issues that had not been addressed at CSS level 1, resulting in css level 2 on 4 November 1997. It was published as a recommendation of the W3C on 12 May 1998. CSS level 3, which started in 1998, is still in development in 2014. In 2005, the CSS working groups decided to implement the standards more rigorly. This meant that already published standards such as CSS 2.1, CSS 3 Selectors and CSS 3 Text were withdrawn from the standardization to working draft level. Difficulty deploying This article needs to be updated. Update this article to reflect recent events or new available information. (January 2019) The CSS 1 specification was completed in 1996. Microsoft's Internet Explorer 3[23] was released that year with little support for CSS. IE 4 and Netscape 4.x added support, but it was typically incomplete and had many flaws that prevented the useful adoption of CSS. It took more than three years for any web browser to achieve an almost complete implementation of the specification. Macintosh Internet Explorer 5.0, delivered in March 2000, was the first browser with full (better than 99 percent) CSS 1 support,[33] surpassed Opera, which had been a leader since CSS was introduced fifteen months earlier. Other browsers soon followed and many of them also implemented parts of CSS 2. [citation required] However, although later version 5 browsers began to offer a fairly complete implementation of CSS, they were still flawed in certain areas and were full of inconsistencies, bugs and other quirks. In Microsoft Internet Explorer 5.x for Windows, unlike the very different IE on the Macintosh, the implementation of the CSS tile template was incorrect compared to CSS standards. Such inconsistencies and variations in feature support made it difficult for developers to have a consistent look on browsers and platforms without using a workaround called CSS hacks and filters. IE/Windows box template errors were so severe that when Internet Explorer 6 was released, Microsoft introduced a backward-compatible CSS interpretation mode (quirks mode) alongside an alternative, corrected standard mode. Non-Microsoft browsers also offered such a mode-switch activity feature. Therefore, authors of HTML files had to make sure that they contained a particularly distinctive standard CSS-intended character to indicate that the authors intended to interpret CSS correctly according to the standards, rather than intended for the now long-obsolete IE5/Windows browser. Without this sign, browsers with quirks mode switching feature will assemble webpage objects instead of IE5/Windows version. [citation required] Problems with the fragmented deployment of CSS, together with the errata of the original specification, led W3C to change CSS 2 to CSS 2.1, which moved closer to a working snapshot of current CSS support in HTML browsers. Some CSS 2 features that no browser had failed to implement were rejected, and in some cases the specified operating diseases were changed to bring them in line with existing implementation. CSS 2.1 became a candidate recommendation on 25 February 2004, but CSS 2.1 was withdrawn to the draft work on 13 June 2005 and was not reinstated until 19 July 2007. [35] In addition to these issues, the .css extension was used by a software product used to convert PowerPoint files to Cd slideshow files[36], so some web servers served all .css[37] MIME-type applications/x-pointplus[38] instead of text/css. Supplier prefixes Individual browser suppliers sometimes introduced new parameters before standardisation and universalization. To prevent future implementations from being disrupted, suppliers assigned unique names to the parameters, such as -moz- Mozilla Firefox, a webkit named after the Apple Safari Browsing Engine, -o- Opera Browser and -ms - Microsoft Internet Explorer. Occasionally, the supplier's prefix, such as -moz-radial-gradient and weblit-linear-gradient, has: different syntax compared to their counterparties other than those in the vendor prefix. [39] Variations CSS have different levels and profiles. Each CSS layer is based on the last, usually adding new features, and is usually marked as CSS 1, CSS 2, CSS 3, and CSS 4. Profiles are typically a subset of one or more CSS levels for a specific device or user interface. There are currently profiles for mobile devices, printers and TVs. Profiles should not be confused with media types added to CSS 2. CSS 1 The first CSS specification to become an official W3C recommendation is CSS Level 1, released on December 17, 1996. Håkon Wium Lie and Bert Bos are original developers. [40] [41] Among its features is support for Font properties such as font properties, such as font and highlights Text color, backgrounds, and other elements Text attributes such as spam words, letters, and lines of text Alignment of text, images, tables, and other elements Unique identifier and general attribute group classification W3C no longer maintains the CSS 1 recommendation. [42] W3C developed CSS 2 CSS level 2 specifications, which were published as a recommendation in May 1998. The superset of CSS 1, CSS 2 includes a number of new features, including absolute, relative and fixed position of elements and z-index, concept of media types, support for aural style sheets (later replaced by CSS 3 speech modules) [43] and bidirectional text, as well as new font features such as shading. W3C no longer maintains the CSS 2 recommendation. [44] CSS 2.1 CSS Level 2 Check 1, often referred to as CSS 2.1, corrects CSS 2 errors, removes poorly supported or not fully interoperable features, and adds browser extensions that have already been implemented to the configuration. In order to comply with the W3C process of the standardisation process of the technical specifications in CSS 2.1, the draft was several years downsized. The first draft work and the recommendation for a work hanger. CSS 2.1 first became a candidate recommendation on 25 February 2004, but was returned to 13 June 2005 to the draft work for review. It returned to the recommendation of the candidate countries on 19 July 2007 and was then updated twice in 2009. However, as changes and clarifications were made, it returned to the draft work of the last call for proposals of 7 December 2010. CSS 2.1 entered into the proposal for a recommendation on 12 April 2011. [45] It had been reviewed by the W3C Advisory Committee and was finally published as a recommendation of W3C on 7 June 2011. [46] CSS 2.1 was designed as a level 2 first and final revision– but the work on the low priority for CSS 2.2 began in 2015. CSS 3 CSS3 is controlled here. For other uses, see CSS3 (details page). Unlike CSS 2, which is a large single specification that defines different characteristics, CSS 3 divided into multiple Documents called modules. Each module adds new features or extends the features defined in CSS 2 while maintaining backward compatibility. Work on CSS level 3 started with the publication of the original CSS 2 recommendation. The earliest CSS 3 drafts were published in June 1999. [47] Due to modularization, different modules have candidate recommendation dates. In the CR phase, implementations are advised not to drop vendor prefixes. [49] Summary of main module specifications[50] Module definition title Status Date css3-background[55] Backgrounds and Borders Module Level 3 Rec Candidate. Oct 2017 css3-box CSS basic box model Working Draft Jul 2018 css-cascade-3 CSS Cascading and Inheritance Level 3 Candidate Rec. May 2016 CSS Color Module CSS Color Module Level 3 Recommendation Summer 2018 css3-content CSS3 Created and replaced content Draft work Summer 2016 css fonts-3 CSS Fonts Module Level 3 Recommendation Sep 2018 css3-gcpm CSS Generated content paged media module May 2014 css3 layout template layout module work draft Note Mar 2015 css3 mediaqueries Media Recommendation Jun 2012 media polls-4 Media polls Level 4 candidate rec. Sew 2017 css3 multicolored layout module Level I Working Draft May 2018 css3-page CSS Paged Media Module Level 4 Working Draft Mar 2013 selectors-3 Selectors Level 33 Recommendation November 201 8 Selectors-4 Selectors Level 4 draft work February 2018 css3-ui CSS Basic interface module Level 3 (CSS3 UI) Recommendation Summer 2018 CSS 4 Play media Jen Simmons discusses the state of CSS in 2019, when multiple CSS 4 modules had advanced Css4 configuration because the specification is divided into many separate modules that level independently. Modules based on CSS level 2 issues started at level 3. Some of them have already reached level 4 or are already approaching level 5. Modules that define other completely new functions, such as Flexbox,[52], are set to Level 1 and some of them are approaching level 2. The CSS team sometimes publishes snapshots, a collection of complete modules, and parts of other drafts that are considered stable enough for browser developers to implement. To date, five such best practice documents have been published in Notes format in 2007,[53] 2010,[54] 2015,[55] 2017,[56] and 2018. [57] In the fact that these snapshots of the specification are mainly intended for developers, there has been an increasing demand for a similar reference document for authors, which would reflect the state of interoperable implementations such as Can I Use ... [58] and Mozilla Developer Network. [59] At the beginning of 2020, a W3C community group has been established, and define such a resource. [60] Actual versioning can also be discussed, which means that a once-produced document may not be called CSS4. Browser support For more information: Comparison of browser engines CSS functionality is not consistent between them. Because browsers do not fully parse CSS, several encoding technologies have been developed that target specific browsers with a rotary back (commonly known by CSS hacks or CSS filters). The introduction of new features in CSS can be prevented by a lack of support for large browsers. Internet Explorer, for example, was slow to increase support for many CSS 3 features, slowing down the rollout of these features and damaging the browser's reputation among developers. [61] To ensure a consistent user experience for their users, web developers often test their sites across multiple operating systems, browsers, and browser versions, increasing development time and complexity. Tools like BrowserStack are built to reduce the complexity of maintaining these environments. In addition to these testing tools, many sites maintain lists of browser support for certain CSS features, including CanIUse and Mozilla Developer Network. In addition, CSS 3 defines feature queries that provide a @supports directive that allows developers to target browsers directly to CSS with the support of specific functionality. [62] Css, which is not supported by older browsers, can sometimes also be repaired using JavaScript polyfiles, which are components of JavaScript code designed so that browsers work consistently. These workarounds — and the need to support backup functionality — can increase the complexity of development projects, which is why companies often specify a list of browser versions that they support and don't support. Since websites introduce new code standards that are not compatible with older browsers, these browsers can be cut off from many network resources (sometimes intentionally). [63] Many of the most popular sites on the Internet have not only been visually impaired by older browsers due to poor CSS support, but do not work at all, largely due to the development of JavaScript and other online technologies. Limitations The observed limitations of the current CsS features are as follows: Switches cannot rise to CSS at this time does not provide a way to select the parent or ancestors of an element that meet a specific condition. [64] Selectors Level 4, which is still in working draft mode, suggests such a switch[65], but only as part of a full selector profile, not a fast profile used for dynamic CSS formatting. [66] More advanced dial model (such as XPath) more advanced style sheets. Main reasons for the CSS working group in the past suggestions for master switches are related to browser performance and additional draft issues. [67] The reverse cannot be explicitly determined regardless of location resolution rules for properties such as z index searching for the nearest parent element with location:absolute or location:relative attribute. This strange connection has unwanted effects. For example, it is impossible to avoid defining a new scope when you need to adjust the position of an element, so that the desired scope of the parent element cannot be used. Pseudoclass dynamic behavior that cannot be managed by CSS implements virtual classes that allow a certain degree of user feedback by using alternative styles conditionally. One CSS virtual class :hover is dynamic (equivalent to a JavaScript onmouseover) and may have the potential for abuse (e.g. proximity to the cursor to implement pop-ups),[68], but CSS does not have the option that the client can disable it (a feature such as off) or limit its effects (no nochange equivalent values for each property). Rules cannot be named A CSS rule cannot be named, which would allow (for example) client-side scripts to refer to a rule even if its switch changes. The style cannot be included in another rule CSS styles are often copied into several rules to achieve the desired effect, which causes additional maintenance and requires more thorough testing. To resolve this, some new CSS features were proposed, but they were rejected afterwards. [69] [70] Certain text cannot be aligned without changing entries : in addition to the virtual element of the first letter, data cannot be applied to specific text areas without having to use positioning holder elements. In addition, previous versions of the CSS standard had several other problems, but they have been mitigated: Limitations of vertical control Although horizontal positioning of elements was always generally easy to control, vertical positioning was often intuitive, complex, or downright impossible. Simple tasks, such as centering an element vertically or placing a footer higher than the bottom of the view port, require either complex and unintuitive style rules or simple but not widely supported rules. [64] The flexible box

module greatly improved the situation, and vertical control is much simpler and supported in all modern browsers. [71] Older browsers still have these problems, but most of them (mainly Internet Explorer 9 or earlier) no longer support their providers. [72] Absence of expressions It was not standard to define property values as simple expressions (such as margin left: 10 % – 3em + 4 px;). This would be in several cases, such as calculating the size of columns that are subject to the sum of all columns. Internet Explorer versions 5 through 7 support their own expression statement[73] with similar functionality. This own expression(s) statement is not supported from Internet Explorer 8 onwards, except for compatibility modes. This decision was made for reasons of compliance with standards, browser performance, and security reasons. [73] However, CSS WG[74] has published a candidate recommendation with a calculation value to address this restriction and has since been supported by all modern browsers. [75] Although column definition is possible in the current CSS 3 (using the column calculation module),[76] layouts with multiple columns can be complex to implement in CSS 2.1. With CSS 2.1, the process is often done using loose elements that are often rendered with different browsers, different computer display formats, and different video relationships that are set to standard displays. All modern browsers support this CSS 3 feature in one format or another. [77] Benefits Of separating content from a presentation Main article: Separating presentations and content CSS makes it easier to publish content in multiple formats based on nominal parameters. Nominal parameters include clear user settings, different browsers, the type of device used to view content (desktop or mobile), the user's geographic location, and many variables. Site-wide consistency Main article: Style sheet (web development) When CSS is used effectively for inheritance and cscading, a global style sheet can be used to influence and format elements sitewide. If the formatting of the elements needs to be changed or customized, you can make the changes by modifying the rules on the general style sheet. Before CSS, such maintenance was more difficult, more expensive and more time-consuming. Bandwidth An internal or external style sheet defines a style once in the range of html elements selected by category, type, or relationship with others. This is much more effective than repeating style data for each instance of an element. An external style sheet is usually cached in the browser, so it can be used on multiple pages without reloading it, further reducing data transfer over the network. Page Reformatted Main Article: Progressive Enhancement A different style sheet can be used on a single line. This offers benefits for ease of use as well as the ability to customize a page or site for different target devices. In addition, devices that cannot understand the design still display the content. Accessibility Without CSS, web designers typically need to set their pages using technologies such as HTML tables that prevent visually impaired users from being accessible (see Untabled Web Design#Accessibility). Standardization frameworks Main article: CSS frames in the CSS framework are pre-prepared libraries designed to enable websites to be easier and formatting formatting The language of the style sheets. CSS frames include Blueprint, Bootstrap, Cascade Framework, Foundation and Materialize. Like programming and scripting language libraries, CSS frames are usually included as external .css tables referenced in HTML files <head>. They offer several ready-made options for website design and layout. While many of these frameworks have been published, some authors mostly use them for fast prototype or learning, and prefer handcrafted CSS, which is suitable for every published site without design, maintenance and download costs with many unused features in the design of the site. [78] Planning Methods As the size of CSS resources used in a project increases, the development team often has to decide on a common planning method to organise them. Goals are the ease of development, ease of collaboration during the development and performance of style sheets deployed in the browser. Popular methods include OOCSS (target-driven CSS), ACSS (atomic CSS), oCSS (organic cascade style page), SMACSS (scalable and modular architecture for CSS) and BEM (block, element, converter). [79] References ^ CSS Developer Guide. Mozilla Developer Network. Archived from the original 2015-09-25. Retrieved 2015-09-24. Flanagan and David. JavaScript - Final Guide (6 to ed.). p. 1. JavaScript is part of a triad of technologies that all Web developers need to learn: HTML to define web page content, configure CSS web page presentation, and JavaScript to define website behavior. ^What is CSS?. World Wide Web Consortium. Archived from the original 2010-11-29. Retrieved 2010-12-01. ^ Web-based mobile apps of the future with HTML 5, CSS, and JavaScript. HTMLGoodies, what are you? Archived from the original 2014-10-20. Retrieved 2014-10-16. ^ W3C CSS validation service. Archived from the original 2011-02-14. Retrieved 2012-06-30. ^ W3C CSS2.1 specification for pseudoelectors and pseudoclasses. World Wide Web Consortium. 7 June 2011. Archived from the original on April 30, 2012. Retrieved 30 April 2012. ^ See the W3C Web site archived 2006-04-23 in Wayback Machine for full configuration of switches. ^ Selectors Level 3. W3.org. Archived from the original 2015-04-30. ^ Index of CSS properties. www.w3.org. Retrieved 2020-08-09. ^ CSS color. Mozilla Developer Network. 2016-06-28. Archived from original 2016-09-21. Retrieved 2016-08-23. ^ 6.1 Length units. CSS Style Sheets, Level 1. 17 days 1996. Arkistoitu <head> <head> 14 June 2019. Retrieved 20 June 2019. ^ 5. Distance units: <length> type. Level 3 of the CSS values and units module. 6 June 2019. Archived from the original on 7 June 2019. ^ W3C HTML Working Group. HTML 5. The vocabulary of HTML and XHTML and related APIs. World Wide Web Consortium. Archived from the original on July 15, 2014. Retrieved 28 June 2014. ^ a b Meyer, Eric A. (2006). Cascading style sheets: The ultimate guide (3rd brought. O'Reilly Media, Inc. ISBN 0-596-52733-0. Archived from the original 2014-02-15. Retrieved 2014-02-16. ^ Set, overlap, and inheritance property values. Archived from the original 2014-06-11. Retrieved 2014-06-10. ^ Can a CSS class inherit one or more other categories?. StackOverflow reflow. Archived from the original 2017-10-14. Retrieved 2017-09-10. ^ Shorthand features. Tutorial. Mozilla developers. 2017-12-07. Archived from original 2018-01-30. Retrieved 2018-01-30. ^ a b c Bos, Bert; et al. (December 7, 2010). 9.3 Positioning systems. Cascading Style Sheets Level 2 Revision 1 (CSS 2.1). W3c. Archived from the original on February 18, 2011. Retrieved 16 February 2011. ^ Holzschlag, Molly E (2005). Go to HTML and CSS. Pearson Education, Inc. ISBN 0-13-185586-7. ^ a b Lie, Hakon W (October 10, 1994). CASCADING HTML Style Sheets Proposal (Proposal) (92). Cern. Archived from the original on June 4, 2014. Retrieved May 25, 2014. Cite magazine requires |journal= (help) ^ a b c d f Lie, Håkon Wium; Bos, Bert (1999). CSS Style Sheets, Web Design. Addison Wesley. ISBN 0-201-59625-3. Retrieved 23 June 2010. ^ Cascading style pages, level 1. World Wide Web Consortium. Archived from the original 2014-09. Retrieved 2014-03-07. ^ a b c Bos, Bert (14 April 1995). Simple style sheets for SGML & HTML online. World Wide Web Consortium. Archived from the original on 23 September 2009. Retrieved 20 June 2010. ^ a b c Cascading style sheets. University of Oslo. Archived from the original 2006-09-06. Retrieved September 3, 2014. ^ a b Petrie, Charles; Cailliau, Robert (Nov. Interview Robert Cailliau about the web suggestion: How it really happened. Institute of Electrical and Electronic Engineers. Archived from the original on 6 January 2011. Retrieved 18 August 2010. ^ Nielsen, Henrik Frystyk (June 7, 2002). Libwww Hackers. World Wide Web Consortium. Archived from the original on 2 December 2009. Retrieved 6 June 2010. ^Yves Lafon. World Wide Web Consortium. Archived from the original on 24 June 2010. Retrieved 17 June 2010. ^ W3C team: technology and society. World Wide Web Consortium. 18 July 2008. Archived from the original on 28 May 2010. Referenced 22.1. </length>" Lou Montulli; Brendan Eich, what are you? Scott Furman, what are you? Donna Converse; Troy Chevalier (August 22, 1996). JavaScript-based style sheets. W3c. Archived from the original on May 27, 2010. Retrieved 23 June 2010. ^ CSS software. W3.org. Archived from the original 2010-11-25. Retrieved 2011-01-15. ^ Anne van Kesteren. CSS 2.1 – Anne's weblog. Archived from the original 2012-02-16. ^ W3C News Archive in 2007. World Wide Web Consortium. Archived from the original 2011-06-28. Retrieved 2011-06-28. ^ Nitot, Tristan (18 March 2002). Invalid MIME type for CSS files. Mozilla Developer Center. Mozilla. Archived from the original 2011-05-20. Retrieved 20 June 2010. ^ McBride, Don (November 27, 2009). Types. Archived from the original on 29 October 2010. Retrieved 20 June 2010. ^ css file extension information. The file name extension database. 12 March 2010. Archived from the original on 18 July 2011. Retrieved 20 June 2010. ^ How and why do you want to use CSS provider prefixes on your site. It's a spirit tolerance. 2019-11-12. ^ Bos, / Håkon Wium Lie, Bert (1997). CSS Style Sheets: Web Design (1st Print. to ed.). Harlow, England; Reading, MA.: Addison Wesley Longman. ISBN 0-201-41998-X. ^ W3C: Cscading Style Sheets, Level 1 Archived 2014-02-09 Wayback Machine in CSS 1 ^ W3C: CSCADING Style Sheets Level 1 Configuration Archived 2011-02-11 Wayback Machine in CSS Level 1 Configuration ^ Aural Style Sheets. W3c. Archived from original 2014-10-26. Retrieved 2014-10-26. ^ W3C: Cscading Style Sheets, Level 2 Archived 2011-01-16 Wayback Machine in CSS 2 (1998 Recommendation) ^ W3C:Cscading Style Sheets, Level 2 Amendment 1 Archived 2011-11-09 In the Wayback Machine 2011-11-09 In the Wayback Machine ^ W3C: Cscading Style Sheets Standard Boasts Unprecedented Interoperability Archived 2011-06-10 at the Wayback Machine ^ Bos, Bert (18/02/2011). Descriptions of all CSS specifications. World Wide Web Consortium. Archived from the original on 31 March 2011. Retrieved 3 March 2011. ^ Bos, Bert (February 26, 2011). Current work of CSS. World Wide Web Consortium. Archived from the original on 3 March 2011. Retrieved 3 March 2011. ^ Etemad, Elika J. (December 12, 2010). Cascading Style Sheets (CSS) snapshot 2010. World Wide Web Consortium. Archived from the original on 16 March 2011. Retrieved 3 March 2011. ^ All CSS configurations. W3.org. 2014-05-22. Archived from the original 2014-05-30. ^ Atkins Jr, tab. Archived from the original on October 31, 2012. Retrieved 10 November 2018. ^ Level 1 of the CSS Flexible Box Layout Module. W3c. 19 November 2018. Archived from the original on October 19, 2012. Retrieved 18 October 2012. ^ Cascading Style Sheets (CSS) Snapshot 2007. 12 May 2011. Archived from the original 8th. 2016. Retrieved 18 July 2016. ^ Cascading Style Sheets (CSS) Snapshot 2010. 12 May 2011. Archived from the original on 16 March 2011. Retrieved 3 March 2011. ^ CSS Snapshot 2015. W3c. 13 October 2015. Archived from the original on 27 January 2017. Retrieved February 13, 2017. ^ CSS Snapshot 2017. 31 January 2017. Archived from the original on 13 February 2017. Retrieved February 15, 2017. ^ CSS Snapshot 2018. 15 November 2018. Archived from the original on 1 February 2019. Retrieved 2 January 2019. ^ Can I use... Support tables for HTML5, CSS3, etc. ^ MDN Web Docs: CSS. ^ Antti, Hiljä. OOCSS, ACSS, BEM, SMACSS: what are they? What am I supposed to wear? clubmate.fi Hiljä. Archived from the original on June 2, 2015. Retrieved 2 June 2015. Read more This section needs to be updated. Update this article to reflect recent events or new available information. (July 2012) Jeffrey Zeldman (2009): Design with network standards, New Riders, ISBN 978-0321616951 (book companion site) Dan Cederholm (2009): Web Standards Solutions, The Markup and Style Handbook, Friends of Ed, ISBN 978-1430219200 (Author's site) Meyer, Eric A. (2006). Cascading style sheets: Final guide, third edition. O'Reilly Media, Inc. ISBN 0-596-52733-0. More Eric Meyer ON CSS (2004) ISBN 0-7357-1425-8 Eric Meyer ON CSS (2002), ISBN 0-7357-1245-X Meyer, Eric A. (2001) Cascading Style Sheets 2.0 Programmer's Reference, McGraw-Hill Osborne Media, ISBN 0-07-213178-0 CSS Design Zen (2005) co-author CSS Zen Garden Owner, Dave Shea and Molly E. Holzschlag), ISBN 0-321-30347-4 Kynn Bartlett: Teach Yourself CSS in 24 Hours, 2nd Edition (2006), Sams Publishing, ISBN 978-0672329067 Cascading Style Sheets by Designing for the Web (2005) by Håkon Wium Lie and Bert Bos, ISBN 0-321-19312-1 Cascading Style Sheets, PhD Håkon Wium Lie's dissertation provides an authoritable historical reference CS From Keith Schengili-Roberts (2003): Core CSS, 2nd Edition, Prentice Hall, ISBN 0-13-009278-9 On the Analysis of Cascading Style Sheets, Pierre Geneves, Nabil Layaida and Vincent Quint, Proceedings of the 21st International Conference on World Wide Web (Www'12) , p. 809–818, 2012. External Links CSSat Wikipedia Sister ProjectsWiktionaryMedia Definitions wikimedia CommonsTextbooks from WikibooksResources WikiversityData WikidataDiscussion meta-WikiDocumentation from MediaWiki's official website CSS Curlie Retrieved 2Version 2 Hypertext Transfer, used by The World Wide Web HTTP/2 International StandardRFC 7540Created IETF IntroducedMay 14, 2015; 5 years ago (2015-05-14)Http/3 HTTP/2 (originally called HTTP/2.0) corrected http/3 HTTP/2 is a major check of the HTTP network protocol used by the World Wide Web. It is derived from a previous experimental SPDY protocol originally developed by Google. [1] [2] HTTP/2 was developed by the HTTP working group (also called the httpbis working group, where bis means other)[lending required] Internet Engineering Task Force (IETF). [3] [4] [5] HTTP/2 is the first new HTTP version since HTTP/1.1, standardized in RFC 2068 in 1997. Work presented HTTP/2 to the Internet Engineering Steering Group (IESG) for consideration as a proposed standard in December 2014[6][7] and approved by IESG for publication in accordance with the proposed standard on 17 February 2015 (and updated in February 2020 for TLS 1.3). [8] [9] HTTP/2 specification published as RFC 7540 on 14 May 2015. [10] Standardization work was supported by Chrome, Opera, Firefox,[11] Internet Explorer 11, Safari, Amazon Silk, and Edge browsers. [12] Most major browsers had added HTTP/2 support by the end of 2015. [13] About 98% of the webs used have a feature[14], while according to W3Techs, as December 2020[update], 50% of the top 10 million websites supported HTTP/2. [15] Its successor is HTTP/3, which is based on the concepts of HTTP/2. [16] [2] HTTP/3 support was added to Chrome in September 2019 and Safari 14, macOS in Big Sur, was the first browser where HTTP/3 was enabled by default, in 2020, otherwise it is not yet enabled by default in any browser, while stable versions of Chrome and Firefox and HTTP/3 do not have default support. [17] [18] [19] Objectives HTTP/2 The Working Group Charter explained by Daniel Stenberg mentions a number of objectives and concerns:[4] Establish a consultation mechanism to enable customers and servers to choose the use of HTTP/1.1, 2.0 or possibly non-HTTP protocols. Maintain high-level compatibility with HTTP/1.1 (for example, methods, status codes, URLs, and most header fields). Reduce latency to improve page load speed in browsers by considering: compressing HTTP header data from HTTP/2 Server Push pipelining requests, To fix http 1.x terminal problem multiple requests through a single TCP connection Support http in common use cases, such as desktop browsers, mobile browsers, web APIs, web servers of different scales, proxy servers, reverse proxy servers, firewalls, and content distribution networks. Differences to HTTP/1.1 The proposed changes do not require changes to the functionality of existing web applications, but new applications can take advantage of new features to increase speed. [20] HTTP/2 leaves all http/1.1 high-level semantics information, such as methods, status codes, header fields, and URNs, at the same level. What is new is how data is framed and transported between the client and the server. [20] Powerful websites minimize the number of requests needed to render an entire page by reducing (reducing the amount of code and compressing smaller pieces of code into bundles without reducing its ability to function), resources such as images and scripts. However, mining may not be convenient or efficient, and the page and mixed resources can still separate HTTP connections. HTTP/2 allows a server to push content so it can respond more queries than the customer requested. This allows the server to provide information that it knows the browser needs to render the web page without waiting for the browser to examine the first response and without overheads for the additional request cycle. [21] The first draft of Http/2 (which was a copy of the SPDY) has made HTTP/2 performance of the multiplication of requests and responses to avoid the problem of http 1 (including HTTP pipelining), header compression, and prioritization of requests. [22] However, since HTTP/2 works in addition to one Internet protocols, such as smoke. Since HTTP/2 only exploits http.s blocked migration encoding mechanism because it provides its own, earlier efficient mechanisms for streaming data. [24] Genesis SPDY SPDY and its subsequent differences (pronounced like speedy) were the previous HTTP replacement protocol developed by a Google-led research project. [25] SPDY uses the same TCP tube, but different protocols to perform its reduction. The basic changes to HTTP/1 to create the HTTP included: actual pipelining of requests without FIFO restrictions, a message framing mechanism to simplify client and server development, mandatory compression (including headers), priority timing, and even two-way communication. [26] The HTTP team considered Google's SPDY protocol, Microsoft's HTTP Speed+Mobility proposal (SPDY-based)[25] and a network-friendly HTTP update. [27] In July 2012, Facebook provided feedback on each proposal and recommended http2 be based on SPDY. [28] The original draft of HTTP/2 was published in November 2012 and was based on a direct copy of the SPDY. [29] The main difference between HTTP/1.1 and SPDY was that each SPDY user is assigned a power id, which means that there is one TCP channel that connects the user to the server. The SPDY divides requests into either a control or data by using a simple structure binary protocol with two types of frames. [26] [30] The SPDY result improved markedly compared to HTTP, with load speeds on the new page ranging from 11,81 % to 47,7 %. [31] SPDY was used as a jumping point to develop HTTP/2. Among the many detailed differences between protocols, the most notable is that HTTP/2 uses a fixed Huffman code-based header compression algorithm instead of dynamic power-based compression of SPDY. This will help reduce the chances of Oracle attacks, such as the CRIME attack. [30] On February 9, 2015, Google announced plans to remove SPDY support from Chrome in favor of HTTP/2 support. [32] It took effect, starting with Chrome 51. [33] [34] Encryption HTTP/2 is configured for both HTTP-UUR TSEs (i.e. without encryption) and HTTPS-UUR-TSEs (via TLS). ALPN extension[35] requiring TLS 1.2 or new). [36] Although the standard itself does not require the use of encryption,[37] all major customer evidence (Firefox,[38] Chrome, Safari, Opera, IE, Edge) has declared its support for HTTP/2 over HTTPS only, which effectively makes encryption mandatory. [39] The http/2 development committee and the Protocol itself have been criticised. FreeBSD and Lacquer developer Poul-Henning Kamp claims that the standard was prepared on an unrealistically short schedule, excluding all the foundations of the new HTTP/2 SPDY protocol except the SPDY protocol should have missed opportunities for improvement. [40] Kamp himself criticises the protocol for inconsistency and unnecessary, overwhelming complexity. [40] He also states that the protocol violates the protocol layering principle[40], for example by copying the flow control that is part of the transport layer (TCP). However, most of the concerns have been related to encryption issues. Encryption Required encryption commendation cost and access to certificates Initially, some team members[who?] tried to enable the encryption requirement in the protocol. This was criticised. Critics noted that encryption does not have low computing costs and that many HTTP applications do not actually need encryption and their providers do not want to use additional resources for it. Proponents of encryption have said that this encryption overhead is virtually void. [41] Poul-Henning Kamp has criticised the IETF for pursuing a specific political programme with HTTP/2. [40] [42] [43] Criticism of the mandatory encryption agenda for the current certification framework is not new and is not unique to creation of the open source community – a Cisco employee stated in 2013 that the current certification model is not compatible with small devices such as routers, as the current model requires not only annual registration and cancellation of non-nerivial fees for each certificate, but must be repeated continuously on an annual basis. [44] In the end, the working group could not agree on mandatory encryption[37], although most customer transactions require implementation, which effectively makes encryption a requirement. The lack of opportunistic encryption HTTP/2 was also criticized for not supporting opportunistic encryption, which is similar to the STARTTLS mechanism, which has long been used on SMTP. Critics have stated that the HTTP/2 proposal violates the IETF's own [rfc:7258 RFC7258 Pervasive Monitoring is a Attack] also status as Best Current Practice 188. [45] According to RFC7258/BCP188, passive monitoring should be considered an attack and protocols designed by the IETF should take steps to protect against passive monitoring opportunistic encryption]. Several specifications have been submitted for the opportunistic encryption of HTTP/2[46][47][48], a draft of which was accepted as the official work of the working group, which led to the publication of RFC 8164 in May 2017. "TCP head-of-line blocking Although HTTP2 planning effectively addresses the HTTP/2 event-level head-of-line blocking issue by allowing multiple concurrent HTTP transactions, all these events are duplicated through a single TCP connection. This means that the TCP stream packet-level head-of-line blocker simultaneously prevents all transactions from being adopted through this connection. This head-of-line blocking by HTTP/2 is now widely considered a design flaw, and much of the effort behind QUIC and HTTP/3 is dedicated to reducing head-of-line blocking issues. [49] [50] Milestones for development Date Milestone[4] December 20, 2007[51][52] First HTTP/1.1 Revision Internet draft 23. 2008[53] First HTTP Security Properties Internet Draft Early 2012[54] Call for proposals HTTP 2.0.10.–25.11.2012[55][56] Last http/1.1 review request of the working group 28.11.2012. 2012[57][58] First WG draft of HTTP 2.0 based on draft mbelshe-httpbis-spdy-00 Held/Eliminated Working Group Last Call for HTTP Security Properties September 2013[59][60] First HTTP/1.1 Review to IESG as proposed standard 12. 2014[61] IESG hyväksyi HTTP/1.1 Version julkaistavaksi ehdotettuna standardina 6.6.2014[51][62] Julkaise HTTP/1.1 Tarkistus nimellä RFC 7230, 7231, 7232, 7233 , 7234, 7235 elokuuta 1, 2014 – syyskuu 1, 2014[7][63] Työryhmä Viimeinen kehotus HTTP/2:lle 16.12.2014[6] Toimita HTTP/2 IESG:lle harkittavaksi ehdotettuna standardina 31.12.2014–14.1.2014, 2015[64] HTTP/2 review request of the working group 16.6.2014[65] HTTP/2 proposed version 3 (HTTP/2) n julkaistavaksi Ehdotettu standardi 14.5.2015[66] Julkaise HTTP/2 nimellä RFC 7540 February 2020 RFC 8740: HTTP/2 ja TLS 1.3 Palvelinpuolen tuki Pääartikkeli: Web-palvelinhjelmiston vertailu Palvelinohjelmisto Apache 2.4.12 tukee HTTP/2:ta moduulin mod_h2,[67], vaikka se tällä sitä tukea kyseistä moduulia. From Apache 2.4.17 all patches are included in the main Apache spring tree, although the module itself was renamed mod_http2. [68] Old versions of spdy were supported through mod_spdy[69], but the development of mod_spdy has stopped. [70] Apache Tomcat supports HTTP/2 with version 8.5 and newer with configuration change. [71] Apache Traffic Server supports HTTP/2. [72] Caddy supports HTTP/2. [73] Charles Proxy supports HTTP/2 from Version Charles 4. [74] Citrix NetScaler 11.x supports HTTP/2. [75] Sucuri supports HTTP/2. [76] Barracuda Networks WAF (Web Application Firewall) supports HTTP/2. [78] h2o was built from the outset against HTTP/2 support. [79] HAProxy 1.8 supports HTTP/2. [80] Platform 9.3 supports HTTP/2. [81] lighttpd 1.4.56 supports HTTP/2. [82] LiteSpeed Web Server 5.0 supports HTTP/2. [83] Microsoft IIS supports HTTP/2 on Windows 10,[84] on Windows Server 2016 and Windows Server 2019. Netty 4.1 supports HTTP/2. [85] nginx 1.9.5 supports HTTP/2,[86] was released on September 22, 2015 using module ngx_http_v2_module and HTTP/2 Server Push and support version 1.13.9 on February 20, 2018. [87] Node .js 8.13.0. [88] (5.0 supports HTTP/2 with module[89] and node 8.4 deployed experimental built-in support for HTTP/2. [90] .NET Core 2.2.0-preview 1 adds HTTP/2 support. [91] OpenLiteSpeed 1.3.11 and 1.4.8 support HTTP/2. [92] Proxygen supports HTTP/2. Pulse Secure Virtual Traffic Manager 9.9 supports HTTP/2. [93] Radware Alteon NG supports HTTP/2. [94] ShimmerCat supports HTTP/2. [95] Vert.x 3.3 supports HTTP/2. Warp (a Haskell web server that is used by default on Yesod) supports HTTP/2. Wildfly 9 supports HTTP/2. Content distribution networks Akamai was the first major CDN to support HTTP/2 and HTTP/2 Server Push. Microsoft Azure supports HTTP/2. PageCDN supports HTTP/2 off-screen and provides a user interface for installing HTTP/2 Server Push on the CDN dashboard. [96] CDN77 supports HTTP/2 using nginx (August 20, 2015). Cloudflare supports HTTP/2, which uses nginx with SPDY as a backup browser without support while retaining all security and performance services. [97] Cloudflare was the first major CDN to support HTTP/2 Server Push. [98] AWS CloudFront supports HTTP/2[99] as of September 7, 2016. Quickly supports HTTP/2, including Server Push. [100] Imperva Incapsula CDN supports HTTP/2. [101] Implementation also includes support for waf and DDoS mitigating features. KeyCDN supports HTTP/2 using nginx (October 6, 2015). The HTTP/2 test is a test page that checks whether the server supports HTTP/2. Voxility supports HTTP/2 using nginx as of July 2016. Implementation supports Cloud DDoS mitigation services. [102] StackPath supports HTTP/2. Implementations Other implementations are collected on the GitHub HTTP/2 wiki. See also gRPC HTTP pipelining HTTP request and response messages HTTP/3 QUIC SPDY WebSocket Web browser comparison § Protocol Support References ^ Bright, Peter (February 18, 2015). HTTP/2 ready, coming to the browses in a few weeks. Ars Technica. A b Cimpanu, Catalin. HTTP-over-QUIC renamed HTTP/3 | Zdnet. Zdnet. Retrieved November 19, 2018. ^ Thomson, M. (tol.), Belshe M. and R. Peon. Hypertext transfer protocol version 2: draft-ietf-httpbis-http2-16. ietf.org HTTPis working group. Retrieved February 11, 2015. ^ a b c Hypertext transfer protocol Bis (httpbis). Internet Design Task 2012. ^ IETF HTTP Working Group. IETF HTTP Working Group. Retrieved January 3, 2015. 2014-12-16 IESG mode changed to Requested ^ a b Raymor, Brian (August 6, 2014). Wait for it – HTTP/2 starts the last call in the workgroup! Microsoft Open Technologies. Archived from the original on October 6, 2014. Retrieved 17 October 2018. ^ a b IESG (17 February 2015). Protocol operation: Hypertext transfer protocol version 2 to the proposed standard (draft-ietf-httpbis-http2-17.txt). httpbis (mailing list). Retrieved February 18, 2015. ^ Mark Nottingham (18 February 2015). HTTP/2. Approved. ietf.org. Internet Design Working Group. Retrieved March 8, 2015. ^ RFC 7540 - Hypertext Transfer Protocol Version 2 (HTTP/2). Ietf. may 2015. Retrieved May 14, 2015. ^ See what's new in Firefox!. www.mozilla.org the Mozilla Foundation. February2015. ^ Can the rise of spdy threaten HTTP?. blog.restlet.com. Restlet, Inc. October 2011. Archived from the original on 6 January 2014. Retrieved 23 July 2012. ^ HTTP2 browser support. Retrieved March 9, 2017. ^ Can I use... Html5, CSS3, etc. caniuse.com. Retrieved 8 August 2020. ^ Use of HTTP/2 on websites. World Wide Web Technology Surveys. W3Techs and W3Techs technologies. Retrieved 12 October 2020. ^ Bishop, Mike (July 9, 2019). Hypertext transfer protocol version 3 (HTTP/3). tools.ietf.org. ^ Can I use... Html5, CSS3, etc. caniuse.com. Retrieved 19 January 2020. Daniel Stenberg, Daniel Stenberg announces HTTP/3 support on Firefox Nightly. Twitter. Retrieved November 5, 2019. ^ Cimpanu, Catalin (26.9.2019). Cloudflare, Google Chrome, and Firefox add HTTP/3 support. Zdnet. Retrieved 27 September 2019. A b Hya Grigorik. Chapter 12: HTTP 2.0. High-performance browser network. O'Reilly Media, Inc. HTTP/2 does not modify the semantics of HTTP's application in any way ^ Pratt, Michael. Apiux. apiux.com. Retrieved 14 March 2014. ^ Do Synodinos (November 2012). HTTP 2.0 First draft published. InfoQ.com. C4Media Inc. ^ Javier Garza (Oct. How does HTTP/2 resolve the Head of Line blocking (HOL) issue. ^ Belshe, Mike; Thomson and Martin. Peon, Roberto (May 2015). Hypertext transfer protocol version 2 (HTTP/2). tools.ietf.org. Retrieved 17.11.2017. HTTP/2 uses DATA frames to send data loads on messages. The blocked transfer mechanism is set to section 4.1 of the [RFC7230] label shall NOT be used in HTTP/2 ^ a b Sebastian Anthony (24th March 2012). S&M vs. SPDY: Microsoft and Google are fighting for the future of HTTP 2.0. Extremetech. A b Grigorik, Ilya. Life outside http 1.1: Google SPDY. ^ Willy Tarreau; With Amos Jeffries. Adrien de Croy, 2000-2008; Poul-Henning Kamp (March 29, 2012). Proposal to be web-friendly for HTTP update. Network working group. ^ Belshe, Bayeer (July 15, 2012). HTTP/2 expression of interest (mailing list). W3c. ^ Dio Synodinos (November 30, 2012). HTTP/2 First draft published. InfoQ. ^ a b Ilya, Grigorik (2015). HTTP/2: an extract from high-performance browser networking (May 2015, first plot). Sebastopol, Calif., 211–224. ISBN 9781491932483. OCLC 1039459460. ^SPDY: An experimental protocol for a faster network. Chrome projects. ^ Chris Bentzel; Bence Béky (February 9, 2015). Hello HTTP/2, goodbye SPDY. Chrome blog. Update: To better align support with chrome's release cycle, SPDY and NPN support will be removed when Chrome 51 is released. ^ API deductions and deletions in Chrome 51. Chrome Platform Status. Retrieved 9 March 2015. ^ Shadrin, Nick (June 7, 2016). Supporting the new SPDY/3.1 support can be removed ^ Nginx. Nginx. Retrieved July 10, 2017. ^ RFC 7301 - Transport Layer Security (TLS) application layer protocol conferences extension. ietf. July 2014. ^ Belshe, M.; Peon, R.; Thomson, M. Hypertext Transfer Protocol Version 2, use of TLS features. Retrieved February 10, 2015. ^ a b HTTP/2 FAQ. IETF HTTP Working Group. Retrieved September 8, 2014. ^ Network/http2. MozillaWiki, what are you? Retrieved September 7, 2014. ^ HTTP/2 implementation status. MNT blog. ^ a b c d Kamp, Poul-Henning (January 6, 2015). HTTP/2.0 – The IETF calls it in (bad protocol, bad policy). ACM queue. The Cite journal requires |journal= (help) ^ Grigorik, Ilya 9. TLS still fast?. Retrieved December 30, 2015. ^ Kamp, P. H. (2015). Http2.0. ACM communications. 58 (3): 40. doi:10.1145/2717515. ^ Kamp, Poul-Henning (January 7, 2015). Re: Last Call: <draft-ietf-httpbis-http2-16.txt> (Hypertext Transfer Protocol version 2) to Proposed Standard. ietf-http-wg@w3.org (Mailing List). Retrieved January 12, 2015. ^ Lear, Eliot (August 25, 2013). Mandatory encryption is* theater. ietf-http-wg@w3.org (Mailing List). ^ Murenin, Constantine A. (January 9, 2015). Re: Last Call: <draft-ietf-httpbis-http2-16.txt> (Hypertext Transfer Protocol version 2) to Proposed Standard. ietf-http-wg@w3.org (Mailing List). Retrieved January 26, 2015. ^ Mark Nottingham; Martin Thomson. Opportunistic encryption for HTTP-UURI: draft-nottingham-http2-encryption-03. Internet Technology Task Force. ^ Mark Nottingham; Martin Thomson. Opportunistic protection HTTP: draft-ietf-httpbis-http2-encryption-01. Internet Technology Task Force. ^ Huston, Geoff (March 4, 2019). A quick look at QUIC. www.circleid.com. Retrieved 2.8.2019. ^ Gal, Shauli (June 22, 2017). Full picture in HTTP/2 and HOL blocking. Good. <draft-ietf-httpbis-http2-16.txt></draft-ietf-httpbis-http2-16.txt>August 3, 2019. ^ a b Nottingham, Mark (7 June 2014). RFC2616 is dead. Retrieved September 20, 2014. ^ Nottingham, Mark (January 1, 2012). Re-ingestion of HTTPbis. Retrieved September 20, 2014. ^ Nottingham, Mark (14 October 2012). Last call of the working group HTTP/1.1 sst and p2. Retrieved September 20, 2014. ^ Nottingham, Mark (23 October 2012). Second working group last call HTTP/1.1 p4-p7. Retrieved September 20, 2014. ^ Nottingham, Mark (November 30, 2012). First draft of HTTP2. Retrieved September 20, 2014. ^ Nottingham, Mark (2011). Hypertext transfer protocol (HTTP/1.1): Message syntax and routing. Archived from the original on August 13, 2014. Retrieved September 20, 2014. ^ Last call: <draft-ietf-httpbis-p1-messaging-24.txt>(Hypertext transfer protocol (HTTP/1.1): Message syntax and routing) to the proposed standard. IESG. October 21, 2013. Retrieved September 20, 2014. ^ Protocol operation: Hypertext Transfer Protocol (HTTP/1.1): Message syntax and routing to the proposed standard (draft-ietf-httpbis-p1-messaging-26.txt). ietf-announce. IESG. February 12, 2014. Retrieved January 18, 2015. ^ RFC's team of journalists (6 June 2014). RFC 7230 hypertext transfer protocol (HTTP/1.1): Message syntax and routing. ietf-announce. Retrieved January 18, 2015. ^ RFC's team of journalists (August 1, 2014). Last call for proposals from the working group: draft-ietf-httpbis-http2-14 and draft-ietf-httpbis-header-compression-09. HTTP Task Force. Retrieved September 20, 2014. ^ Last call: <draft-ietf-httpbis-http2-16.txt>(Hypertext Transfer Protocol version 2) to the proposed standard from IESG 2014-12-31. Internet Technology Task Force. 2014. Retrieved 1 January 2015. ^ IESG agenda: 2015-01-22. Ietf. Archived from the original on January 15, 2015. Retrieved January 15, 2015. ^ RFC supplier team (14 May 2015). RFC 7540 In hypertext transfer protocol version 2 (HTTP/2). ietf-announce. ^ http/2 module apache httpd. Retrieved July 28, 2015. ^ Apache 2.4.17 release change log. Retrieved August 12, 2019. mod_spdy is now an Apache Developers blog. ^ Log /httpd/mod_spdy. svn.apache.org. Retrieved 3.2.2017. ^ Apache Tomcat Migration. Retrieved July 29, 2016. ^ Apache traffic server downloads. trafficserver.apache.org Retrieved July 10, 2017. ^ Server, Caddy Web (March 23, 2016). Caddy 2 - Ultimate Server with automatic HTTPS. caddyserver.com. Referenced 8.<draft-ietf-httpbis-http2-16.txt><draft-ietf-httpbis-http2-16.txt>. A public artifact. August 2, 2016. Retrieved 27 October 2020. ^ 3 simple steps to bring HTTP/2 performance to old Web applications. September 22, 2015. ^ Sucuri «» HTTP/2 – Notification of HTTP/2 support. The suturi. Retrieved December 5, 2015. ^ Robert Haynes. Goodbye SPDY, hello HTTP/2. F5 networks. Retrieved September 18, 2015. ^ Risov Chakraborty. What's new in barracuda's web application firewall. Barracuda networks. ^ H2O - optimized HTTP/2 server. h2o.example.ie.net. ^ New versions of HaProxy 1.8. haproxy.com. Retrieved 28 September 2018. ^ New in HTTP/2, ESI, and LiteMage Cache Support. April 17, 2015. ^ Rob Trace; David Walp (October 8, 2013). HTTP/2: Long-awaited sequel. MSDN IEBlog site. With Microsoft Corporation. ^ Netty.news: Netty 4.1.0.Final published. netty.io. Retrieved 1 June 2016. ^ nginx change log. www.nginx.com September 22, 2015. ^ Changes nginx 1.14.2. nginx.org. 4.12.2018. Retrieved 27 September 2019. ^ Foundation, Node js. Node v8.13.0 (LTS). Node.js. www.github.com July 26, 2016. ^ Node v8.4.0 (current). nodejs.org August 15, 2017. ^ ASP.NET Core 2.2.0 preview1 is now available. Retrieved 22 August 2018. ^ OpenLiteSpeed 1.4.5 Update Log. LiteSpeed Technologies, Inc., February 26, 2015. Retrieved February 26, 2015. ^ Pulse Virtual Traffic Manager. August 22, 2017. ^ Radware combines an integrated HTTP/2 gateway with its leading Fastview technology to increase the acceleration of web server environments. July 20, 2015. ^ www.shimmercat.com. March 23, 2016. ^ Why PageCDN, and what problem does it solve?. PageCDN. Retrieved 11 January 2020. ^HTTP/2 is here! Goodbye, SPDY? Not just yet. CloudFlare, what are you? Retrieved December 5, 2015. ^ Krasnov, Vlad (April 28, 2016). HTTP/2 Server Push support notification. CloudFlare, what are you? Retrieved May 18, 2016. ^ Amazon CloudFront now supports HTTP/2. Amazon Web Services, Inc. Retrieved September 8, 2016. ^ Notification of limited availability of HTTP/2. Retrieved August 22, 2017. ^ HTTP/2 is here: What you need to know. Retrieved November 1, 2015. ^ HTTP/2 more at risk of cyberattacks?. The age of knowledge. August 3, 2016. Retrieved 4 February 2019. External links Official website RFC 7540 - Hypertext Transfer Protocol version 2 (HTTP/2) RFC 7541 – HPACK: Header pack for HTTP/2 SPDY (draft-mbelshe-httpbis-spdy -00) HTTP Speed+Mobility (Draft Montenegro-httpbis-speed-mobility-01) Proposal for an online friendly HTTP update (draft-tarreau-httpbis-network-friendly-00) HTTP/2 Website Online Tester Retrieved