

- **News**
- **SEO Parramatta**
- **Web Design Parramatta**
- **Local SEO Parramatta**
- **Parramatta SEO services**

- **More**

Parramatta web design agency
Search Engine Optimisation Parramatta
Affordable SEO Parramatta
Custom web design Parramatta
eCommerce web design Parramatta
Parramatta digital marketing
Best SEO agency Parramatta
SEO expert Parramatta
Responsive web design Parramatta
Small business SEO Parramatta
Web development Parramatta
SEO consultant Parramatta
Website designers Parramatta
SEO company Parramatta
Web design company Parramatta
SEO audit Parramatta

- **About Us**

- **Contact Us**



Parramatta SEO performance —

- [Creative agencies Parramatta](#)
- [Parramatta SEO performance](#)
- [Parramatta business websites](#)
- [SEO website redesign Parramatta](#)
- [Web design solutions Parramatta](#)
- [Search engine ranking Parramatta](#)
- [Parramatta SEO tools](#)

Maximise your business potential with Parramatta conversion optimisation We deliver impactful strategies designed to boost your brand awareness, improve online visibility, and generate a steady flow of qualified leads in Parramatta

Take your digital presence further with Parramatta marketing funnel We develop custom strategies aimed at increasing your online visibility, improving search engine rankings, and achieving sustainable growth for your Parramatta-based business

Experience outstanding online performance through Optimised website content Parramatta Our expert team specialises in delivering solutions that improve rankings, drive engagement, and generate valuable leads for consistent business growth in Parramatta

[Local SEO](#) .

Parramatta business websites

Experience outstanding online performance through Top web developers Parramatta Our expert team specialises in delivering solutions that improve rankings, drive engagement, and generate valuable leads for consistent business growth in Parramatta

Maximise your business potential with Digital agency Parramatta We deliver impactful strategies designed to boost your brand awareness, improve online visibility, and generate a steady flow of qualified leads in Parramatta

Choose excellence in digital marketing with Search visibility Parramatta Our proven approaches drive website traffic, enhance customer engagement, and significantly improve conversion rates, supporting long-term business success in Parramatta

Best [SEO Packages Parramatta](#) Parramatta.



SEO website redesign Parramatta

Maximise your business potential with Web and SEO consulting Parramatta We deliver impactful strategies designed to boost your brand awareness, improve online visibility, and generate a steady flow of qualified leads in Parramatta

Experience outstanding online performance through Parramatta design and development Our expert team specialises in delivering solutions that improve rankings, drive engagement, and generate valuable leads for consistent business growth in Parramatta

Choose excellence in digital marketing with Results-driven SEO Parramatta Our proven approaches drive website traffic, enhance customer engagement, and significantly improve conversion rates, supporting long-term business success in Parramatta

Web design solutions Parramatta

Transform your business growth with SEO Parramatta Our strategies enhance visibility, attract targeted traffic, and maximise conversions for sustained success Partner with us for measurable digital marketing outcomes today

Take your digital presence further with Web Design Parramatta We develop custom strategies aimed at increasing your online visibility, improving search engine rankings, and achieving sustainable growth for your Parramatta-based business

Experience outstanding online performance through Local SEO Parramatta Our expert team specialises in delivering solutions that improve rankings, drive engagement, and generate valuable leads for consistent business growth in Parramatta

Responsive web design Parramatta - Parramatta SEO Reporting

1. On-page SEO Parramatta
2. SEO Competitor Analysis Parramatta
3. Parramatta SEO Consultants

KEY ADVANTAGES LOCAL SEO



SYDNEY WEBSITE DESIGN AGENCY
SUITE 87, LEVEL 33, AUSTRALIA SQUARE,
265 GEORGE ST, SYDNEY NSW 2000
PHONE: 1300 684 339

CONTENT MARKETING
TYPES FOR SMALL BUSINESS
AND BRAND BUILDING

Search engine ranking Parramatta

Choose excellence in digital marketing with Parramatta SEO services Our proven approaches drive website traffic, enhance customer engagement, and significantly improve conversion rates, supporting long-term business success in Parramatta

Take your digital presence further with Parramatta web design agency We develop custom strategies aimed at increasing your online visibility, improving search engine rankings, and achieving sustainable growth for your Parramatta-based business

Experience outstanding online performance through Search Engine Optimisation Parramatta Our expert team specialises in delivering solutions that improve rankings, drive engagement, and generate valuable leads for consistent business growth in Parramatta

Responsive web design Parramatta - Backlink Building Parramatta

1. Parramatta SEO Reporting
2. Backlink Building Parramatta
3. Mobile SEO Parramatta

Parramatta SEO tools

Maximise your business potential with Affordable SEO Parramatta We deliver impactful strategies designed to boost your brand awareness, improve online visibility, and generate a steady flow of qualified leads in Parramatta

Take your digital presence further with Custom web design Parramatta We develop custom strategies aimed at increasing your online visibility, improving search engine rankings, and achieving sustainable growth for your Parramatta-based business

Experience outstanding online performance through eCommerce web design Parramatta Our expert team specialises in delivering solutions that improve rankings, drive engagement, and generate valuable leads for consistent business growth in Parramatta

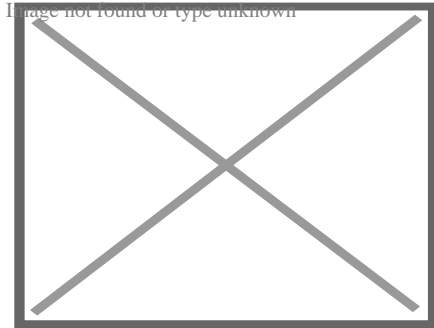
Responsive web design Parramatta - Parramatta SEO Reporting

1. Parramatta SEO Consultants
2. Local Business SEO Parramatta
3. SEO Consulting Parramatta



About Web crawler

This article is about the internet bot. For the search engine, see [WebCrawler](#). "Web spider" redirects here; not to be confused with [Spider web](#). "Spiderbot" redirects here. For the video game, see [Arac \(video game\)](#).



Architecture of a Web crawler

A **Web crawler**, sometimes called a **spider** or **spiderbot** and often shortened to **crawler**, is an [Internet bot](#) that systematically browses the [World Wide Web](#) and that is typically operated by search engines for the purpose of [Web indexing](#) (*web spidering*).^[1]

Web [search engines](#) and some other [websites](#) use Web crawling or spidering [software](#) to update their [web content](#) or indices of other sites' web content. Web crawlers copy pages for processing by a search engine, which [indexes](#) the downloaded pages so that users can search more efficiently.

Crawlers consume resources on visited systems and often visit sites unprompted. Issues of schedule, load, and "politeness" come into play when large collections of pages are accessed. Mechanisms exist for public sites not wishing to be crawled to make this known to the crawling agent. For example, including a [robots.txt](#) file can request [bots](#) to index only parts of a website, or nothing at all.

The number of Internet pages is extremely large; even the largest crawlers fall short of making a complete index. For this reason, search engines struggled to give relevant search results in the early years of the World Wide Web, before 2000. Today, relevant results are given almost instantly.

Crawlers can validate [hyperlinks](#) and [HTML](#) code. They can also be used for [web scraping](#) and [data-driven programming](#).

Nomenclature

[\[edit\]](#)

A web crawler is also known as a *spider*,^[2] an *ant*, an *automatic indexer*,^[3] or (in the FOAF software context) a *Web scutter*.^[4]

Overview

[\[edit\]](#)

A Web crawler starts with a list of **URLs** to visit. Those first URLs are called the *seeds*. As the crawler visits these URLs, by communicating with **web servers** that respond to those URLs, it identifies all the **hyperlinks** in the retrieved web pages and adds them to the list of URLs to visit, called the *crawl frontier*. URLs from the frontier are **recursively** visited according to a set of policies. If the crawler is performing archiving of **websites** (or **web archiving**), it copies and saves the information as it goes. The archives are usually stored in such a way they can be viewed, read and navigated as if they were on the live web, but are preserved as 'snapshots'.^[5]

The archive is known as the *repository* and is designed to store and manage the collection of **web pages**. The **repository** only stores **HTML** pages and these pages are stored as distinct files. A repository is similar to any other system that stores data, like a modern-day database. The only difference is that a repository does not need all the functionality offered by a database system. The repository stores the most recent version of the web page retrieved by the crawler.^[citation needed]

The large volume implies the crawler can only download a limited number of the Web pages within a given time, so it needs to prioritize its downloads. The high rate of change can imply the pages might have already been updated or even deleted.

The number of possible URLs crawled being generated by server-side software has also made it difficult for web crawlers to avoid retrieving **duplicate content**. Endless combinations of **HTTP GET** (URL-based) parameters exist, of which only a small selection will actually return unique content. For example, a simple online photo gallery may offer three options to users, as specified through HTTP GET parameters in the URL. If there exist four ways to sort images, three choices of **thumbnail** size, two file formats, and an option to disable user-provided content, then the same set of content can be accessed with 48 different URLs, all of which may be linked on the site. This **mathematical combination** creates a problem for crawlers, as they must sort through endless combinations of relatively minor scripted changes in order to retrieve unique content.

As Edwards *et al.* noted, "Given that the **bandwidth** for conducting crawls is neither infinite nor free, it is becoming essential to crawl the Web in not only a scalable, but efficient way, if some reasonable measure of quality or freshness is to be maintained."^[6] A crawler must carefully choose at each step which pages to visit next.

Crawling policy

[\[edit\]](#)

The behavior of a Web crawler is the outcome of a combination of policies:^[7]

- a *selection policy* which states the pages to download,
- a *re-visit policy* which states when to check for changes to the pages,
- a *politeness policy* that states how to avoid overloading [websites](#).
- a *parallelization policy* that states how to coordinate distributed web crawlers.

Selection policy

[\[edit\]](#)

Given the current size of the Web, even large search engines cover only a portion of the publicly available part. A 2009 study showed even large-scale [search engines](#) index no more than 40–70% of the indexable Web;[\[8\]](#) a previous study by [Steve Lawrence](#) and [Lee Giles](#) showed that no [search engine indexed](#) more than 16% of the Web in 1999.[\[9\]](#) As a crawler always downloads just a fraction of the [Web pages](#), it is highly desirable for the downloaded fraction to contain the most relevant pages and not just a random sample of the Web.

This requires a metric of importance for prioritizing Web pages. The importance of a page is a function of its [intrinsic](#) quality, its popularity in terms of links or visits, and even of its URL (the latter is the case of [vertical search engines](#) restricted to a single [top-level domain](#), or search engines restricted to a fixed Web site). Designing a good selection policy has an added difficulty: it must work with partial information, as the complete set of Web pages is not known during crawling.

Junghoo Cho *et al.* made the first study on policies for crawling scheduling. Their data set was a 180,000-pages crawl from the stanford.edu domain, in which a crawling simulation was done with different strategies.[\[10\]](#) The ordering metrics tested were [breadth-first](#), [backlink](#) count and partial [PageRank](#) calculations. One of the conclusions was that if the crawler wants to download pages with high Pagerank early during the crawling process, then the partial Pagerank strategy is the better, followed by breadth-first and backlink-count. However, these results are for just a single domain. Cho also wrote his PhD dissertation at Stanford on web crawling.[\[11\]](#)

Najork and Wiener performed an actual crawl on 328 million pages, using breadth-first ordering.[\[12\]](#) They found that a breadth-first crawl captures pages with high Pagerank early in the crawl (but they did not compare this strategy against other strategies). The explanation given by the authors for this result is that "the most important pages have many links to them from numerous hosts, and those links will be found early, regardless of on which host or page the crawl originates."

Abiteboul designed a crawling strategy based on an [algorithm](#) called OPIC (On-line Page Importance Computation).[\[13\]](#) In OPIC, each page is given an initial sum of "cash" that is distributed equally among the pages it points to. It is similar to a PageRank computation, but it is faster and is only done in one step. An OPIC-driven crawler downloads first the pages in the crawling frontier with higher amounts of "cash". Experiments were carried in a 100,000-pages synthetic graph with a power-law distribution of in-links. However, there was no comparison with other strategies nor experiments in the real Web.

Boldi *et al.* used simulation on subsets of the Web of 40 million pages from the .it domain and 100 million pages from the WebBase crawl, testing breadth-first against depth-first, random ordering and an omniscient strategy. The comparison was based on how well PageRank computed on a partial crawl approximates the true PageRank value. Some visits that accumulate PageRank very quickly (most notably, breadth-first and the omniscient visit) provide very poor progressive approximations.^{[14][15]}

Baeza-Yates *et al.* used simulation on two subsets of the Web of 3 million pages from the .gr and .cl domain, testing several crawling strategies.^[16] They showed that both the OPIC strategy and a strategy that uses the length of the per-site queues are better than **breadth-first** crawling, and that it is also very effective to use a previous crawl, when it is available, to guide the current one.

Daneshpajouh *et al.* designed a community based algorithm for discovering good seeds.^[17] Their method crawls web pages with high PageRank from different communities in less iteration in comparison with crawl starting from random seeds. One can extract good seed from a previously-crawled-Web graph using this new method. Using these seeds, a new crawl can be very effective.

Restricting followed links

^[edit]

A crawler may only want to seek out HTML pages and avoid all other **MIME types**. In order to request only HTML resources, a crawler may make an HTTP HEAD request to determine a Web resource's MIME type before requesting the entire resource with a GET request. To avoid making numerous HEAD requests, a crawler may examine the URL and only request a resource if the URL ends with certain characters such as .html, .htm, .asp, .aspx, .php, .jsp, .jspx or a slash. This strategy may cause numerous HTML Web resources to be unintentionally skipped.

Some crawlers may also avoid requesting any resources that have a "?" in them (are dynamically produced) in order to avoid **spider traps** that may cause the crawler to download an infinite number of URLs from a Web site. This strategy is unreliable if the site uses **URL rewriting** to simplify its URLs.

URL normalization

^[edit]

Main article: **URL normalization**

Crawlers usually perform some type of **URL normalization** in order to avoid crawling the same resource more than once. The term *URL normalization*, also called *URL canonicalization*, refers to the process of modifying and standardizing a URL in a consistent manner. There are several types of normalization that may be performed including conversion of URLs to lowercase, removal of "." and ".." segments, and adding trailing slashes to the non-empty path component.^[18]

Path-ascending crawling

[[edit](#)]

Some crawlers intend to download/upload as many resources as possible from a particular web site. So *path-ascending crawler* was introduced that would ascend to every path in each URL that it intends to crawl.[19] For example, when given a seed URL of `http://llama.org/hamster/monkey/page.html`, it will attempt to crawl `/hamster/monkey/`, `/hamster/`, and `/`. Cothey found that a path-ascending crawler was very effective in finding isolated resources, or resources for which no inbound link would have been found in regular crawling.

Focused crawling

[[edit](#)]

Main article: [Focused crawler](#)

The importance of a page for a crawler can also be expressed as a function of the similarity of a page to a given query. Web crawlers that attempt to download pages that are similar to each other are called **focused crawler** or **topical crawlers**. The concepts of topical and focused crawling were first introduced by [Filippo Menczer](#)[20][21] and by Soumen Chakrabarti *et al.*[22]

The main problem in focused crawling is that in the context of a Web crawler, we would like to be able to predict the similarity of the text of a given page to the query before actually downloading the page. A possible predictor is the anchor text of links; this was the approach taken by Pinkerton [23] in the first web crawler of the early days of the Web. Diligenti *et al.*[24] propose using the complete content of the pages already visited to infer the similarity between the driving query and the pages that have not been visited yet. The performance of a focused crawling depends mostly on the richness of links in the specific topic being searched, and a focused crawling usually relies on a general Web search engine for providing starting points.

Academic focused crawler

[[edit](#)]

An example of the [focused crawlers](#) are academic crawlers, which crawls free-access academic related documents, such as the *citeseerxbot*, which is the crawler of [CiteSeer](#)^X search engine. Other academic search engines are [Google Scholar](#) and [Microsoft Academic Search](#) etc. Because most academic papers are published in [PDF](#) formats, such kind of crawler is particularly interested in crawling [PDF](#), [PostScript](#) files, [Microsoft Word](#) including their [zipped](#) formats. Because of this, general open-source crawlers, such as [Heritrix](#), must be customized to filter out other [MIME types](#), or a [middleware](#) is used to extract these documents out and import them to the focused crawl database and repository.[25] Identifying whether these documents are academic or

not is challenging and can add a significant overhead to the crawling process, so this is performed as a post crawling process using [machine learning](#) or [regular expression](#) algorithms. These academic documents are usually obtained from home pages of faculties and students or from publication page of research institutes. Because academic documents make up only a small fraction of all web pages, a good seed selection is important in boosting the efficiencies of these web crawlers.[\[26\]](#) Other academic crawlers may download plain text and [HTML](#) files, that contains [metadata](#) of academic papers, such as titles, papers, and abstracts. This increases the overall number of papers, but a significant fraction may not provide free PDF downloads.

Semantic focused crawler

[\[edit\]](#)

Another type of focused crawlers is semantic focused crawler, which makes use of [domain ontologies](#) to represent topical maps and link Web pages with relevant ontological concepts for the selection and categorization purposes.[\[27\]](#) In addition, ontologies can be automatically updated in the crawling process. Dong et al.[\[28\]](#) introduced such an ontology-learning-based crawler using a [support-vector machine](#) to update the content of ontological concepts when crawling Web pages.

Re-visit policy

[\[edit\]](#)

The Web has a very dynamic nature, and crawling a fraction of the Web can take weeks or months. By the time a Web crawler has finished its crawl, many events could have happened, including creations, updates, and deletions.

From the search engine's point of view, there is a cost associated with not detecting an event, and thus having an outdated copy of a resource. The most-used cost functions are freshness and age.[\[29\]](#)

Freshness: This is a binary measure that indicates whether the local copy is accurate or not. The freshness of a page p in the repository at time t is defined as:

$$F_{\{p\}}(t) = \begin{cases} 1 & \text{if } p \text{ is equal to the local copy at time } t \\ 0 & \text{otherwise} \end{cases}$$

Image not found or type unknown

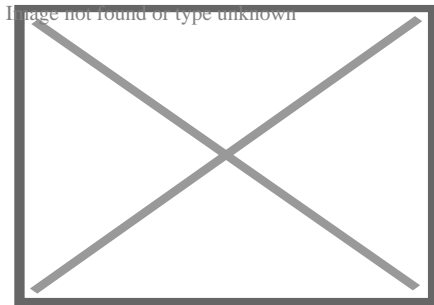
Age: This is a measure that indicates how outdated the local copy is. The age of a page p in the repository, at time t is defined as:

$$A_{p(t)} = \begin{cases} 0 & \text{if } p \text{ is not modified at time } t \\ \text{mod} \end{cases}$$

Image not found or type unknown

Coffman *et al.* worked with a definition of the objective of a Web crawler that is equivalent to freshness, but use a different wording: they propose that a crawler must minimize the fraction of time pages remain outdated. They also noted that the problem of Web crawling can be modeled as a multiple-queue, single-server polling system, on which the Web crawler is the server and the Web sites are the queues. Page modifications are the arrival of the customers, and switch-over times are the interval between page accesses to a single Web site. Under this model, mean waiting time for a customer in the polling system is equivalent to the average age for the Web crawler.[30]

The objective of the crawler is to keep the average freshness of pages in its collection as high as possible, or to keep the average age of pages as low as possible. These objectives are not equivalent: in the first case, the crawler is just concerned with how many pages are outdated, while in the second case, the crawler is concerned with how old the local copies of pages are.



Evolution of Freshness and Age in a web crawler

Two simple re-visiting policies were studied by Cho and Garcia-Molina:[31]

- Uniform policy: This involves re-visiting all pages in the collection with the same frequency, regardless of their rates of change.
- Proportional policy: This involves re-visiting more often the pages that change more frequently. The visiting frequency is directly proportional to the (estimated) change frequency.

In both cases, the repeated crawling order of pages can be done either in a random or a fixed order.

Cho and Garcia-Molina proved the surprising result that, in terms of average freshness, the uniform policy outperforms the proportional policy in both a simulated Web and a real Web crawl. Intuitively, the reasoning is that, as web crawlers have a limit to how many pages they can crawl in a given time frame, (1) they will allocate too many new crawls to rapidly changing pages at the expense of less frequently updating pages, and (2) the freshness of rapidly changing pages lasts for shorter period than that of less frequently changing pages. In other words, a proportional policy allocates more resources to crawling frequently updating pages, but experiences less overall freshness time from them.

To improve freshness, the crawler should penalize the elements that change too often. [32] The optimal re-visiting policy is neither the uniform policy nor the proportional policy. The optimal method for keeping average freshness high includes ignoring the pages that change too often, and the optimal for keeping average age low is to use access frequencies that monotonically (and sub-linearly) increase with the rate of change of each page. In both cases, the optimal is closer to the uniform policy than to the proportional policy: as [Coffman et al.](#) note, "in order to minimize the expected obsolescence time, the accesses to any particular page should be kept as evenly spaced as possible". [30] Explicit formulas for the re-visit policy are not attainable in general, but they are obtained numerically, as they depend on the distribution of page changes. Cho and Garcia-Molina show that the exponential distribution is a good fit for describing page changes, [32] while [Ipeirotis et al.](#) show how to use statistical tools to discover parameters that affect this distribution. [33] The re-visiting policies considered here regard all pages as homogeneous in terms of quality ("all pages on the Web are worth the same"), something that is not a realistic scenario, so further information about the Web page quality should be included to achieve a better crawling policy.

Politeness policy

[edit]

Crawlers can retrieve data much quicker and in greater depth than human searchers, so they can have a crippling impact on the performance of a site. If a single crawler is performing multiple requests per second and/or downloading large files, a server can have a hard time keeping up with requests from multiple crawlers.

As noted by Koster, the use of Web crawlers is useful for a number of tasks, but comes with a price for the general community. [34] The costs of using Web crawlers include:

- network resources, as crawlers require considerable bandwidth and operate with a high degree of parallelism during a long period of time;
- server overload, especially if the frequency of accesses to a given server is too high;
- poorly written crawlers, which can crash servers or routers, or which download pages they cannot handle; and
- personal crawlers that, if deployed by too many users, can disrupt networks and Web servers.

A partial solution to these problems is the [robots exclusion protocol](#), also known as the robots.txt protocol that is a standard for administrators to indicate which parts of their Web servers should not be accessed by crawlers. [35] This standard does not include a suggestion for the interval of visits to the same server, even though this interval is the most effective way of avoiding server overload. Recently commercial search engines like [Google](#), [Ask Jeeves](#), [MSN](#) and [Yahoo! Search](#) are able to use an extra "Crawl-delay:" parameter in the [robots.txt](#) file to indicate the number of seconds to delay between requests.

The first proposed interval between successive pageloads was 60 seconds.^[36] However, if pages were downloaded at this rate from a website with more than 100,000 pages over a perfect connection with zero latency and infinite bandwidth, it would take more than 2 months to download only that entire Web site; also, only a fraction of the resources from that Web server would be used.

Cho uses 10 seconds as an interval for accesses,^[31] and the WIRE crawler uses 15 seconds as the default.^[37] The MercatorWeb crawler follows an adaptive politeness policy: if it took t seconds to download a document from a given server, the crawler waits for $10t$ seconds before downloading the next page.^[38] Dill *et al.* use 1 second.^[39]

For those using Web crawlers for research purposes, a more detailed cost-benefit analysis is needed and ethical considerations should be taken into account when deciding where to crawl and how fast to crawl.^[40]

Anecdotal evidence from access logs shows that access intervals from known crawlers vary between 20 seconds and 3–4 minutes. It is worth noticing that even when being very polite, and taking all the safeguards to avoid overloading Web servers, some complaints from Web server administrators are received. Sergey Brin and Larry Page noted in 1998, "... running a crawler which connects to more than half a million servers ... generates a fair amount of e-mail and phone calls. Because of the vast number of people coming on line, there are always those who do not know what a crawler is, because this is the first one they have seen."^[41]

Parallelization policy

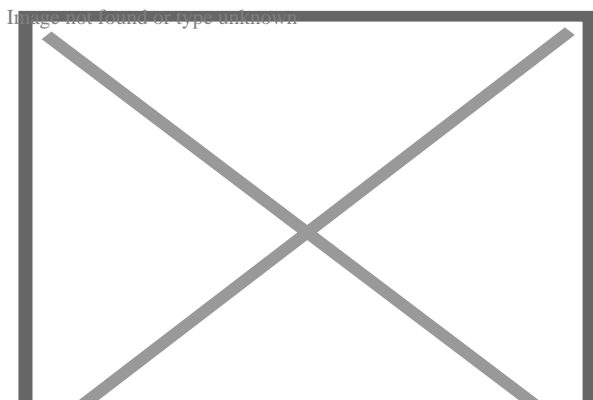
[\[edit\]](#)

Main article: [Distributed web crawling](#)

A **parallel** crawler is a crawler that runs multiple processes in parallel. The goal is to maximize the download rate while minimizing the overhead from parallelization and to avoid repeated downloads of the same page. To avoid downloading the same page more than once, the crawling system requires a policy for assigning the new URLs discovered during the crawling process, as the same URL can be found by two different crawling processes.

Architectures

[\[edit\]](#)



High-level architecture of a standard Web crawler

A crawler must not only have a good crawling strategy, as noted in the previous sections, but it should also have a highly optimized architecture.

Shkapenyuk and Suel noted that:[\[42\]](#)

While it is fairly easy to build a slow crawler that downloads a few pages per second for a short period of time, building a high-performance system that can download hundreds of millions of pages over several weeks presents a number of challenges in system design, I/O and network efficiency, and robustness and manageability.

Web crawlers are a central part of search engines, and details on their algorithms and architecture are kept as business secrets. When crawler designs are published, there is often an important lack of detail that prevents others from reproducing the work. There are also emerging concerns about "[search engine spamming](#)", which prevent major search engines from publishing their ranking algorithms.

Security

[\[edit\]](#)

While most of the website owners are keen to have their pages indexed as broadly as possible to have strong presence in [search engines](#), web crawling can also have [unintended consequences](#) and lead to a [compromise](#) or [data breach](#) if a search engine indexes resources that should not be publicly available, or pages revealing potentially vulnerable versions of software.

Main article: [Google hacking](#)

Apart from standard [web application security](#) recommendations website owners can reduce their exposure to opportunistic hacking by only allowing search engines to index the public parts of their websites (with [robots.txt](#)) and explicitly blocking them from indexing transactional parts (login pages, private pages, etc.).

Crawler identification

[\[edit\]](#)

Web crawlers typically identify themselves to a Web server by using the [User-agent](#) field of an [HTTP](#) request. Web site administrators typically examine their [Web servers'](#) log and use the user agent field to determine which crawlers have visited the web server and how often. The user agent field may include a [URL](#) where the Web site administrator may find out more information about the crawler. Examining Web server log is tedious task, and therefore some administrators use tools to identify, track and verify Web crawlers. [Spambots](#) and other malicious Web crawlers are unlikely to place identifying information in the user agent field, or they may mask their identity

as a browser or other well-known crawler.

Web site administrators prefer Web crawlers to identify themselves so that they can contact the owner if needed. In some cases, crawlers may be accidentally trapped in a **crawler trap** or they may be overloading a Web server with requests, and the owner needs to stop the crawler. Identification is also useful for administrators that are interested in knowing when they may expect their Web pages to be indexed by a particular **search engine**.

Crawling the deep web

[[edit](#)]

A vast amount of web pages lie in the **deep or invisible web**.^[43] These pages are typically only accessible by submitting queries to a database, and regular crawlers are unable to find these pages if there are no links that point to them. Google's **Sitemaps** protocol and **mod oai**^[44] are intended to allow discovery of these **deep-Web** resources.

Deep web crawling also multiplies the number of web links to be crawled. Some crawlers only take some of the URLs in `` form. In some cases, such as the **Googlebot**, Web crawling is done on all text contained inside the hypertext content, tags, or text.

Strategic approaches may be taken to target deep Web content. With a technique called **screen scraping**, specialized software may be customized to automatically and repeatedly query a given Web form with the intention of aggregating the resulting data. Such software can be used to span multiple Web forms across multiple Websites. Data extracted from the results of one Web form submission can be taken and applied as input to another Web form thus establishing continuity across the Deep Web in a way not possible with traditional web crawlers.^[45]

Pages built on **AJAX** are among those causing problems to web crawlers. **Google** has proposed a format of AJAX calls that their bot can recognize and index.^[46]

Visual vs programmatic crawlers

[[edit](#)]

There are a number of "visual web scraper/crawler" products available on the web which will crawl pages and structure data into columns and rows based on the users requirements. One of the main difference between a classic and a visual crawler is the level of programming ability required to set up a crawler. The latest generation of "visual scrapers" remove the majority of the programming skill needed to be able to program and start a crawl to scrape web data.

The visual scraping/crawling method relies on the user "teaching" a piece of crawler technology, which then follows patterns in semi-structured data sources. The dominant method for teaching a visual crawler is by highlighting data in a browser and training columns and rows. While the technology is not new, for example it was the basis of Needlebase which has been bought by Google (as part of a larger acquisition of ITA Labs^[47]), there is continued growth and investment

in this area by investors and end-users.^[*citation needed*]

List of web crawlers

[*edit*]

Further information: [List of search engine software](#)

The following is a list of published crawler architectures for general-purpose crawlers (excluding focused web crawlers), with a brief description that includes the names given to the different components and outstanding features:

Historical web crawlers

[*edit*]

- **WolfBot** was a massively multi threaded crawler built in 2001 by Mani Singh a Civil Engineering graduate from the University of California at Davis.
- **World Wide Web Worm** was a crawler used to build a simple index of document titles and URLs. The index could be searched by using the **grep Unix** command.
- Yahoo! Slurp was the name of the **Yahoo!** Search crawler until Yahoo! contracted with **Microsoft** to use **Bingbot** instead.

In-house web crawlers

[*edit*]

- Applebot is **Apple's** web crawler. It supports **Siri** and other products.^[48]
- **Bingbot** is the name of Microsoft's **Bing** webcrawler. It replaced **Msnbot**.
- Baiduspider is **Baidu's** web crawler.
- DuckDuckBot is **DuckDuckGo's** web crawler.
- **Googlebot** is described in some detail, but the reference is only about an early version of its architecture, which was written in C++ and **Python**. The crawler was integrated with the indexing process, because text parsing was done for full-text indexing and also for URL extraction. There is a URL server that sends lists of URLs to be fetched by several crawling processes. During parsing, the URLs found were passed to a URL server that checked if the URL have been previously seen. If not, the URL was added to the queue of the URL server.
- **WebCrawler** was used to build the first publicly available full-text index of a subset of the Web. It was based on **lib-WWW** to download pages, and another program to parse and order URLs for breadth-first exploration of the Web graph. It also included a real-time crawler that followed links based on the similarity of the anchor text with the provided query.

- [WebFountain](#) is a distributed, modular crawler similar to Mercator but written in C++.
- [Xenon](#) is a web crawler used by government tax authorities to detect fraud.[\[49\]](#)[\[50\]](#)

Commercial web crawlers

[\[edit\]](#)

The following web crawlers are available, for a price::

- [Diffbot](#) - programmatic general web crawler, available as an [API](#)
- [SortSite](#) - crawler for analyzing websites, available for [Windows](#) and [Mac OS](#)
- [Swiftbot](#) - [Swifttype](#)'s web crawler, available as [software as a service](#)
- [Aleph Search](#) - web crawler allowing massive collection with high scalability

Open-source crawlers

[\[edit\]](#)

- [Apache Nutch](#) is a highly extensible and scalable web crawler written in Java and released under an [Apache License](#). It is based on [Apache Hadoop](#) and can be used with [Apache Solr](#) or [Elasticsearch](#).
- [Grub](#) was an open source distributed search crawler that [Wikia Search](#) used to crawl the web.
- [Heritrix](#) is the [Internet Archive](#)'s archival-quality crawler, designed for archiving periodic snapshots of a large portion of the Web. It was written in [Java](#).
- [ht://Dig](#) includes a Web crawler in its indexing engine.
- [HTTrack](#) uses a Web crawler to create a mirror of a web site for off-line viewing. It is written in [C](#) and released under the [GPL](#).
- [Norconex Web Crawler](#) is a highly extensible Web Crawler written in [Java](#) and released under an [Apache License](#). It can be used with many repositories such as [Apache Solr](#), [Elasticsearch](#), [Microsoft Azure Cognitive Search](#), [Amazon CloudSearch](#) and more.
- [mnoGoSearch](#) is a crawler, indexer and a search engine written in C and licensed under the [GPL](#) (*NIX machines only)
- [Open Search Server](#) is a search engine and web crawler software release under the [GPL](#).
- [Scrapy](#), an open source webcrawler framework, written in python (licensed under [BSD](#)).
- [Seeks](#), a free distributed search engine (licensed under [AGPL](#)).
- [StormCrawler](#), a collection of resources for building low-latency, scalable web crawlers on [Apache Storm](#) (Apache License).
- [tkWWW Robot](#), a crawler based on the [tkWWW](#) web browser (licensed under [GPL](#)).
- [GNU Wget](#) is a [command-line](#)-operated crawler written in [C](#) and released under the [GPL](#). It is typically used to mirror Web and FTP sites.

- [YaCy](#), a free distributed search engine, built on principles of peer-to-peer networks (licensed under GPL).

See also

[\[edit\]](#)

- [Automatic indexing](#)
- [Gnutella crawler](#)
- [Web archiving](#)
- [Webgraph](#)
- [Website mirroring software](#)
- [Search Engine Scraping](#)
- [Web scraping](#)

References

[\[edit\]](#)

- [^] ["Web Crawlers: Browsing the Web"](#). Archived from *the original* on 6 December 2021.
- [^] Spetka, Scott. *"The TkWWW Robot: Beyond Browsing"*. NCSA. Archived from *the original* on 3 September 2004. Retrieved 21 November 2010.
- [^] Kobayashi, M. & Takeda, K. (2000). "Information retrieval on the web". *ACM Computing Surveys*. **32** (2): 144–173. *CiteSeerX* 10.1.1.126.6094. doi:10.1145/358923.358934. *S2CID* 3710903.
- [^] See [definition of scutter on FOAF Project's wiki](#) Archived 13 December 2009 at the [Wayback Machine](#)
- [^] Masanès, Julien (15 February 2007). *Web Archiving*. Springer. p. 1. ISBN 978-3-54046332-0. Retrieved 24 April 2014.
- [^] Edwards, J.; McCurley, K. S.; and Tomlin, J. A. (2001). "An adaptive model for optimizing performance of an incremental web crawler". *Proceedings of the 10th international conference on World Wide Web*. pp. 106–113. *CiteSeerX* 10.1.1.1018.1506. doi:10.1145/371920.371960. ISBN 978-1581133486. *S2CID* 10316730. Archived from *the original* on 25 June 2014. Retrieved 25 January 2007.{{cite book}}: CS1 maint: multiple names: authors list (link)
- [^] Castillo, Carlos (2004). *Effective Web Crawling* (PhD thesis). University of Chile. Retrieved 3 August 2010.
- [^] Gulls, A.; A. Signori (2005). "The indexable web is more than 11.5 billion pages". *Special interest tracks and posters of the 14th international conference on World Wide Web*. ACM Press. pp. 902–903. doi:10.1145/1062745.1062789.
- [^] Lawrence, Steve; C. Lee Giles (8 July 1999). "Accessibility of information on the web". *Nature*. **400** (6740): 107–9. *Bibcode*:1999Natur.400..107L. doi:10.1038/21987. *PMID* 10428673. *S2CID* 4347646.
- [^] Cho, J.; Garcia-Molina, H.; Page, L. (April 1998). "Efficient Crawling Through URL Ordering". *Seventh International World-Wide Web Conference*. Brisbane, Australia. doi:

- 10.1142/3725. ISBN 978-981-02-3400-3. Retrieved 23 March 2009.
11. ^ Cho, Junghoo, "Crawling the Web: Discovery and Maintenance of a Large-Scale Web Data", PhD dissertation, Department of Computer Science, Stanford University, November 2001.
 12. ^ Najork, Marc and Janet L. Wiener. "Breadth-first crawling yields high-quality pages". Archived 24 December 2017 at the Wayback Machine In: *Proceedings of the Tenth Conference on World Wide Web*, pages 114–118, Hong Kong, May 2001. Elsevier Science.
 13. ^ Abiteboul, Serge; Mihai Preda; Gregory Cobena (2003). "Adaptive on-line page importance computation". *Proceedings of the 12th international conference on World Wide Web*. Budapest, Hungary: ACM. pp. 280–290. doi:10.1145/775152.775192. ISBN 1-58113-680-3. Retrieved 22 March 2009.
 14. ^ Boldi, Paolo; Bruno Codenotti; Massimo Santini; Sebastiano Vigna (2004). "UbiCrawler: a scalable fully distributed Web crawler" (PDF). *Software: Practice and Experience*. **34** (8): 711–726. CiteSeerX 10.1.1.2.5538. doi:10.1002/spe.587. S2CID 325714. Archived from the original (PDF) on 20 March 2009. Retrieved 23 March 2009.
 15. ^ Boldi, Paolo; Massimo Santini; Sebastiano Vigna (2004). "Do Your Worst to Make the Best: Paradoxical Effects in PageRank Incremental Computations" (PDF). *Algorithms and Models for the Web-Graph. Lecture Notes in Computer Science*. Vol. 3243. pp. 168–180. doi:10.1007/978-3-540-30216-2_14. ISBN 978-3-540-23427-2. Archived from the original (PDF) on 1 October 2005. Retrieved 23 March 2009.
 16. ^ Baeza-Yates, R.; Castillo, C.; Marin, M. and Rodriguez, A. (2005). "Crawling a Country: Better Strategies than Breadth-First for Web Page Ordering." In: *Proceedings of the Industrial and Practical Experience track of the 14th conference on World Wide Web*, pages 864–872, Chiba, Japan. ACM Press.
 17. ^ Shervin Daneshpajouh, Mojtaba Mohammadi Nasiri, Mohammad Ghodsi, A Fast Community Based Algorithm for Generating Crawler Seeds Set. In: *Proceedings of 4th International Conference on Web Information Systems and Technologies (Webist-2008)*, Funchal, Portugal, May 2008.
 18. ^ Pant, Gautam; Srinivasan, Padmini; Menczer, Filippo (2004). "Crawling the Web" (PDF). In Levene, Mark; Poulouvassilis, Alexandra (eds.). *Web Dynamics: Adapting to Change in Content, Size, Topology and Use*. Springer. pp. 153–178. ISBN 978-3-540-40676-1. Archived from the original (PDF) on 20 March 2009. Retrieved 9 May 2006.
 19. ^ Cothey, Viv (2004). "Web-crawling reliability" (PDF). *Journal of the American Society for Information Science and Technology*. **55** (14): 1228–1238. CiteSeerX 10.1.1.117.185. doi:10.1002/asi.20078.
 20. ^ Menczer, F. (1997). ARACHNID: Adaptive Retrieval Agents Choosing Heuristic Neighborhoods for Information Discovery Archived 21 December 2012 at the Wayback Machine. In D. Fisher, ed., *Machine Learning: Proceedings of the 14th International Conference (ICML97)*. Morgan Kaufmann
 21. ^ Menczer, F. and Belew, R.K. (1998). Adaptive Information Agents in Distributed Textual Environments Archived 21 December 2012 at the Wayback Machine. In K. Sycara and M. Wooldridge (eds.) *Proc. 2nd Intl. Conf. on Autonomous Agents (Agents '98)*. ACM Press
 22. ^ Chakrabarti, Soumen; Van Den Berg, Martin; Dom, Byron (1999). "Focused crawling: A new approach to topic-specific Web resource discovery" (PDF). *Computer Networks*. **31** (11–16): 1623–1640. doi:10.1016/s1389-1286(99)00052-3. Archived from the original (PDF)

on 17 March 2004.

23. ^ Pinkerton, B. (1994). [Finding what people want: Experiences with the WebCrawler](#). In Proceedings of the First World Wide Web Conference, Geneva, Switzerland.
24. ^ Diligenti, M., Coetzee, F., Lawrence, S., Giles, C. L., and Gori, M. (2000). [Focused crawling using context graphs](#). In Proceedings of 26th International Conference on Very Large Databases (VLDB), pages 527-534, Cairo, Egypt.
25. ^ Wu, Jian; Teregowda, Pradeep; Khabisa, Madian; Carman, Stephen; Jordan, Douglas; San Pedro Wandelper, Jose; Lu, Xin; Mitra, Prasenjit; Giles, C. Lee (2012). "Web crawler middleware for search engine digital libraries". *Proceedings of the twelfth international workshop on Web information and data management - WIDM '12*. p. 57. doi: [10.1145/2389936.2389949](#). ISBN 9781450317207. S2CID 18513666.
26. ^ Wu, Jian; Teregowda, Pradeep; Ramírez, Juan Pablo Fernández; Mitra, Prasenjit; Zheng, Shuyi; Giles, C. Lee (2012). "The evolution of a crawling strategy for an academic document search engine". *Proceedings of the 3rd Annual ACM Web Science Conference on - Web Sci '12*. pp. 340–343. doi: [10.1145/2380718.2380762](#). ISBN 9781450312288. S2CID 16718130.
27. ^ Dong, Hai; Hussain, Farookh Khadeer; Chang, Elizabeth (2009). ["State of the Art in Semantic Focused Crawlers"](#). *Computational Science and Its Applications – ICCSA 2009. Lecture Notes in Computer Science*. Vol. 5593. pp. 910–924. doi: [10.1007/978-3-642-02457-3_74](#). hdl:20.500.11937/48288. ISBN 978-3-642-02456-6.
28. ^ Dong, Hai; Hussain, Farookh Khadeer (2013). ["SOF: A semi-supervised ontology-learning-based focused crawler"](#). *Concurrency and Computation: Practice and Experience*. **25** (12): 1755–1770. doi: [10.1002/cpe.2980](#). S2CID 205690364.
29. ^ Junghoo Cho; Hector Garcia-Molina (2000). ["Synchronizing a database to improve freshness"](#) (PDF). *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. Dallas, Texas, United States: ACM. pp. 117–128. doi: [10.1145/342009.335391](#). ISBN 1-58113-217-4. Retrieved 23 March 2009.
30. ^ **a b** E. G. Coffman Jr; Zhen Liu; Richard R. Weber (1998). "Optimal robot scheduling for Web search engines". *Journal of Scheduling*. **1** (1): 15–29. CiteSeerX [10.1.1.36.6087](#). doi: [10.1002/\(SICI\)1099-1425\(199806\)1:1<15::AID-JOS3>3.0.CO;2-K](#).
31. ^ **a b** Cho, Junghoo; Garcia-Molina, Hector (2003). "Effective page refresh policies for Web crawlers". *ACM Transactions on Database Systems*. **28** (4): 390–426. doi: [10.1145/958942.958945](#). S2CID 147958.
32. ^ **a b** Junghoo Cho; Hector Garcia-Molina (2003). "Estimating frequency of change". *ACM Transactions on Internet Technology*. **3** (3): 256–290. CiteSeerX [10.1.1.59.5877](#). doi: [10.1145/857166.857170](#). S2CID 9362566.
33. ^ Ipeirotis, P., Ntoulas, A., Cho, J., Gravano, L. (2005) [Modeling and managing content changes in text databases Archived 5 September 2005 at the Wayback Machine](#). In Proceedings of the 21st IEEE International Conference on Data Engineering, pages 606-617, April 2005, Tokyo.
34. ^ Koster, M. (1995). Robots in the web: threat or treat? *ConneXions*, 9(4).
35. ^ Koster, M. (1996). [A standard for robot exclusion Archived 7 November 2007 at the Wayback Machine](#).
36. ^ Koster, M. (1993). [Guidelines for robots writers Archived 22 April 2005 at the Wayback Machine](#).

37. ^ Baeza-Yates, R. and Castillo, C. (2002). [Balancing volume, quality and freshness in Web crawling](#). In *Soft Computing Systems – Design, Management and Applications*, pages 565–572, Santiago, Chile. IOS Press Amsterdam.
38. ^ Heydon, Allan; Najork, Marc (26 June 1999). ["Mercator: A Scalable, Extensible Web Crawler"](#) (PDF). Archived from [the original](#) (PDF) on 19 February 2006. Retrieved 22 March 2009. {{cite journal}}: Cite journal requires |journal= (help)
39. ^ Dill, S.; Kumar, R.; Mccurley, K. S.; Rajagopalan, S.; Sivakumar, D.; Tomkins, A. (2002). ["Self-similarity in the web"](#) (PDF). *ACM Transactions on Internet Technology*. **2** (3): 205–223. doi:10.1145/572326.572328. S2CID 6416041.
40. ^ M. Thelwall; D. Stuart (2006). ["Web crawling ethics revisited: Cost, privacy and denial of service"](#). *Journal of the American Society for Information Science and Technology*. **57** (13): 1771–1779. doi:10.1002/asi.20388.
41. ^ Brin, Sergey; Page, Lawrence (1998). ["The anatomy of a large-scale hypertextual Web search engine"](#). *Computer Networks and ISDN Systems*. **30** (1–7): 107–117. doi:10.1016/s0169-7552(98)00110-x. S2CID 7587743.
42. ^ Shkapenyuk, V. and Suel, T. (2002). [Design and implementation of a high performance distributed web crawler](#). In *Proceedings of the 18th International Conference on Data Engineering (ICDE)*, pages 357–368, San Jose, California. IEEE CS Press.
43. ^ Shestakov, Denis (2008). [Search Interfaces on the Web: Querying and Characterizing Archived](#) 6 July 2014 at the [Wayback Machine](#). TUCS Doctoral Dissertations 104, University of Turku
44. ^ Michael L Nelson; Herbert Van de Sompel; Xiaoming Liu; Terry L Harrison; Nathan McFarland (24 March 2005). ["mod_oai: An Apache Module for Metadata Harvesting"](#): cs/0503069. [arXiv:cs/0503069](#). Bibcode:2005cs.....3069N. {{cite journal}}: Cite journal requires |journal= (help)
45. ^ Shestakov, Denis; Bhowmick, Sourav S.; Lim, Ee-Peng (2005). ["DEQUE: Querying the Deep Web"](#) (PDF). *Data & Knowledge Engineering*. **52** (3): 273–311. doi:10.1016/s0169-023x(04)00107-7.
46. ^ ["AJAX crawling: Guide for webmasters and developers"](#). Retrieved 17 March 2013.
47. ^ ITA Labs ["ITA Labs Acquisition"](#) Archived 18 March 2014 at the [Wayback Machine](#) 20 April 2011 1:28 AM
48. ^ ["About Applebot"](#). Apple Inc. Retrieved 18 October 2021.
49. ^ Norton, Quinn (25 January 2007). ["Tax takers send in the spiders"](#). Business. *Wired*. Archived from the original on 22 December 2016. Retrieved 13 October 2017.
50. ^ ["Xenon web crawling initiative: privacy impact assessment \(PIA\) summary"](#). Ottawa: Government of Canada. 11 April 2017. Archived from the original on 25 September 2017. Retrieved 13 October 2017.

Further reading

[edit]

- Cho, Junghoo, ["Web Crawling Project"](#), UCLA Computer Science Department.
- [A History of Search Engines](#), from Wiley

- **WIVET** is a benchmarking project by **OWASP**, which aims to measure if a web crawler can identify all the hyperlinks in a target website.
- Shestakov, Denis, "**Current Challenges in Web Crawling**" and "**Intelligent Web Crawling**", slides for tutorials given at ICWE'13 and WI-IAT'13.

- **v**
- **t**
- **e**

Internet search

Types

- Web search engine (List)
- Metasearch engine
- Multimedia search
- Collaborative search engine
- Cross-language search
- Local search
- Vertical search
- Social search
- Image search
- Audio search
- Video search engine
- Enterprise search
- Semantic search
- Natural language search engine
- Voice search

Tools

- Cross-language information retrieval
- Search by sound
- Search engine marketing
- Search engine optimization
- Evaluation measures
- Search oriented architecture
- Selection-based search
- Document retrieval
- Text mining
- Web crawler
- Multisearch
- Federated search
- Search aggregator
- Index/Web indexing
- Focused crawler
- Spider trap
- Robots exclusion standard
- Distributed web crawling
- Web archiving
- Website mirroring software
- Web query
- Web query classification

Protocols and standards

- Z39.50
- Search/Retrieve Web Service
- Search/Retrieve via URL
- OpenSearch
- Representational State Transfer
- Wide area information server

See also

- Search engine
- Desktop search
- Online search

- **v**
- **t**
- **e**

Web crawlers

Internet bots designed for Web crawling and Web indexing

Active

- 80legs
- bingbot
- Crawljax
- Fetcher
- Googlebot
- Heritrix
- HTTrack
- PowerMapper
- Wget

Discontinued

- FAST Crawler
- msnbot
- RBSE
- TkWWW robot
- Twiceler

Types

- Distributed web crawler
- Focused crawler

Authority control databases: **National**  **Google** 
Edit this at Wikidata

About MediaWiki

Not to be confused with **Wikimedia**.

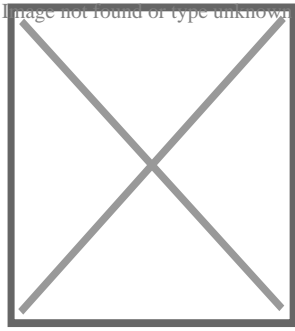


This article **relies excessively on references to primary sources**. Please improve this article by adding **secondary or tertiary sources**.

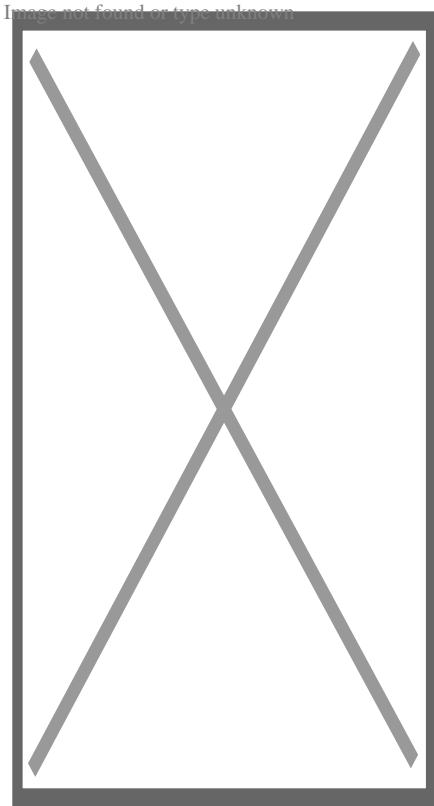
Find sources: "MediaWiki" – news ▪ newspapers ▪ books ▪ scholar ▪ JSTOR (January 2025) *(**Learn how and when to remove this message**)*

- gerrit.wikimedia.org/g/mediawiki/core/  **Edit this at Wikidata**

MediaWiki



Screenshot



The **Main Page** of the **English Wikipedia**
running an alpha version of MediaWiki
1.40

Original author(s)	○ Magnus Manske
	○ Lee Daniel Crocker
Developer(s)	Wikimedia Foundation
Initial release	January 25, 2002; 23 years ago

Stable release	1.43.0 ^[1]  Edit this on Wikidata December 2024; 2 months ago
Repository	
Written in	PHP ^[2]
Operating system	Windows, macOS, Linux, FreeBSD, OpenBSD, Solaris
Size	79.05 MiB (compressed)
Available in	459 ^[3] languages
Type	Wiki software
License	GPLv2+ ^[4]
Website	mediawiki.org  Edit this at Wikidata

MediaWiki is [free and open-source wiki software](#) originally developed by [Magnus Manske](#) for use on [Wikipedia](#) on [January 25, 2002](#), and further improved by [Lee Daniel Crocker](#),^{[5][6]} after which development has been coordinated by the [Wikimedia Foundation](#). It powers several wiki hosting websites across the Internet, as well as most websites hosted by the Wikimedia Foundation including Wikipedia, [Wiktionary](#), [Wikimedia Commons](#), [Wikiquote](#), Meta-Wiki and [Wikidata](#), which define a large part of the set requirements for the software.^[7] Besides its usage on Wikimedia sites, MediaWiki has been used as a [knowledge management](#) and [content management system](#) on websites such as [Fandom](#), [wikiHow](#) and major internal installations like [Intellipedia](#) and [Diplopedia](#).

MediaWiki is written in the [PHP programming language](#) and stores all text content into a [database](#). The software is optimized to efficiently handle large projects, which can have terabytes of content and hundreds of thousands of [views](#) per second.^{[7][8]} Because Wikipedia is one of the world's largest and most visited websites, achieving scalability through multiple layers of [caching](#) and [database replication](#) has been a major concern for developers. Another major aspect of MediaWiki is its internationalization; its interface is available in more than 400 languages.^[9] The software has hundreds of configuration settings^[10] and more than 1,000 [extensions](#) available for enabling various features to be added or changed.^[11]

Key features

[\[edit\]](#)

MediaWiki provides a rich core feature set and a mechanism to attach [extensions](#) to provide additional functionality.

Internationalization and localisation

[\[edit\]](#)

Niklas Laxström explains the features that allowed [translatewiki.net](#) to provide MediaWiki with more than 400 locales.

Due to the strong emphasis on multilingualism in the Wikimedia projects, [internationalization and localization](#) has received significant attention by developers. The user interface has been fully or partially translated into more than 400 languages on [translatewiki.net](#),^[9] and can be further customized by site administrators (the entire interface is editable through the wiki).

Several extensions, most notably those collected in the MediaWiki Language Extension Bundle, are designed to further enhance the multilingualism and internationalization of MediaWiki.

Installation and configuration

[\[edit\]](#)

Installation of MediaWiki requires that the user have [administrative privileges](#) on a server running both PHP and a compatible type of SQL [database](#). Some users find that setting up a [virtual host](#) is helpful if the majority of one's site runs under a framework (such as [Zope](#) or [Ruby on Rails](#)) that is largely incompatible with MediaWiki.^[12] [Cloud hosting](#) can eliminate the need to deploy a new server.^[13]

An installation PHP script is accessed via a [web browser](#) to initialize the wiki's settings. It prompts the user for a minimal set of required parameters, leaving further changes, such as enabling uploads,^[14] adding a site logo,^[15] and installing extensions, to be made by modifying configuration settings contained in a file called LocalSettings.php.^[16] Some aspects of MediaWiki can be configured through special pages or by editing certain pages; for instance, abuse filters can be configured through a special page,^[17] and certain gadgets can be added by creating [JavaScript](#) pages in the MediaWiki namespace.^[18] The MediaWiki community publishes a comprehensive installation guide.^[19]

Markup

[\[edit\]](#)

One of the earliest differences between MediaWiki (and its predecessor, [UseModWiki](#)) and other wiki engines was the use of "[free links](#)" instead of [CamelCase](#). When MediaWiki was created, it was typical for wikis to require text like "WorldWideWeb" to create a link to a page about the [World Wide Web](#); links in MediaWiki, on the other hand, are created by surrounding words with double square brackets, and any spaces between them are left intact, e.g. [[World Wide Web]]. This change was logical for the purpose of creating an encyclopedia, where accuracy in titles is important.

MediaWiki uses an extensible[\[20\]](#) [lightweight wiki markup](#) designed to be easier to use and learn than [HTML](#). Tools exist for converting content such as [tables](#) between MediaWiki markup and HTML.[\[21\]](#) Efforts have been made to create a MediaWiki markup spec, but a consensus seems to have been reached that Wikicode requires [context-sensitive grammar](#) rules.[\[22\]](#)[\[23\]](#) The following side-by-side comparison illustrates the differences between wiki markup and HTML:

MediaWiki syntax
(the "behind the scene"
used to add formatting)

====A dialogue====

"Take some more [[tea]]," the March Hare said to Alice, very earnestly.

"I've had nothing yet," Alice replied in an offended tone: "so I can't take more."

"You mean you can't take "less", " said the Hatter: "it's ""very"" easy to take "more" than nothing."

(Quotation above from [Alice's Adventures in Wonderland](#) by [Lewis Carroll](#))

Editing interface

[\[edit\]](#)

See also: [VisualEditor](#)

Image not found or type unknown



Editing interface of MediaWiki 1.44.0-wmf.4 with **syntax highlighting**, showing the edit toolbar of 2017 wikitext editor and some examples of wiki syntax

MediaWiki's default page-editing tools have been described as somewhat challenging to learn.^[24] A survey of students assigned to use a MediaWiki-based wiki found that when they were asked an **open question** about main problems with the wiki, 24% cited technical problems with formatting, e.g. "Couldn't figure out how to get an image in. Can't figure out how to show a link with words; it inserts a number."^[25]

To make editing long pages easier, MediaWiki allows the editing of a subsection of a page (as identified by its header). A registered user can also indicate whether or not an edit is minor. Correcting spelling, grammar or punctuation are examples of minor edits, whereas adding paragraphs of new text is an example of a non-minor edit.

Sometimes while one user is editing, a second user saves an edit to the same part of the page. Then, when the first user attempts to save the page, an **edit conflict** occurs. The second user is then given an opportunity to merge their content into the page as it now exists following the first user's page save.

MediaWiki's user interface has been localized in many different languages. A language for the wiki content itself can also be set, to be sent in the "Content-Language" HTTP header and "lang" **HTML attribute**.

VisualEditor has its own integrated wikitext editing interface known as 2017 wikitext editor, the older editing interface is known as 2010 wikitext editor.

Application programming interface

[edit]

MediaWiki has an extensible **web API** (**application programming interface**) that provides direct, high-level access to the data contained in the MediaWiki databases. Client programs can use the API to log in, get data, and post changes. The API supports thin web-based JavaScript clients and end-user applications (such as vandal-fighting tools). The API can be accessed by the **backend** of another web site.^[26] An extensive **Python bot** library, **Pywikibot**,^[27] and a popular semi-automated tool called **AutoWikiBrowser**, also interface with the API.^[28] The API is accessed via URLs such as `https://en.wikipedia.org/w/api.php?action=query&list=recentchanges`. In this case, the query would be asking Wikipedia for information relating to the last 10 edits to the site. One of the perceived advantages of the API is its language independence; it listens for **HTTP** connections from clients and can send a response in a variety of formats, such as **XML**, serialized PHP, or **JSON**.^[29] **Client code** has been developed to provide layers of **abstraction** to the API.^[30]

Tracking edits

[[edit](#)]

Among the features of MediaWiki to assist in tracking edits is a Recent Changes feature that provides a list of recent edits to the wiki. This list contains basic information about those edits such as the editing user, the edit summary, the page edited, as well as any tags (e.g. "possible [vandalism](#)")[\[31\]](#) added by customizable abuse filters and other extensions to aid in combating unhelpful edits.[\[32\]](#) On more active wikis, so many edits occur that it is hard to track Recent Changes manually. Anti-vandal software, including user-assisted tools,[\[33\]](#) is sometimes employed on such wikis to process Recent Changes items. Server load can be reduced by sending a continuous feed of Recent Changes to an [IRC channel](#) that these tools can monitor, eliminating their need to send requests for a refreshed Recent Changes feed to the API.[\[34\]](#)[\[35\]](#)

Another important tool is watchlisting. Each logged-in user has a watchlist to which the user can add whatever pages he or she wishes. When an edit is made to one of those pages, a summary of that edit appears on the watchlist the next time it is refreshed.[\[36\]](#) As with the recent changes page, recent edits that appear on the watchlist contain clickable links for easy review of the article history and specific changes made.

There is also the capability to review all edits made by any particular user. In this way, if an edit is identified as problematic, it is possible to check the user's other edits for issues.

MediaWiki allows one to link to specific versions of articles. This has been useful to the scientific community, in that expert peer reviewers could analyse articles, improve them and provide links to the trusted version of that article.[\[37\]](#)

Navigation

[[edit](#)]

Wikilinks

[[edit](#)]

Navigation through the wiki is largely through internal wikilinks. MediaWiki's wikilinks implement page existence detection, in which a link is colored blue if the target page exists on the local wiki and red if it does not. If a user clicks on a red link, they are prompted to create an article with that title. Page existence detection makes it practical for users to create "wikified" articles—that is, articles containing links to other pertinent subjects—without those other articles being yet in existence.

Interwiki links

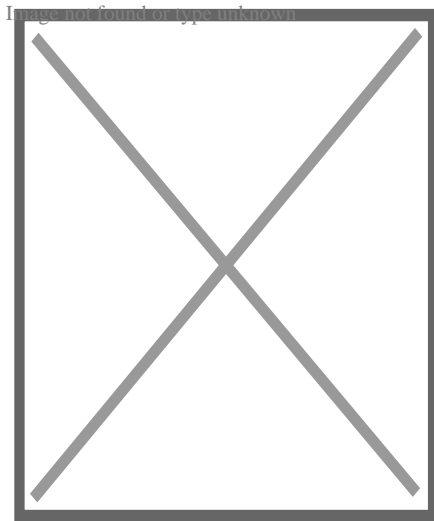
[[edit](#)]

"Inter-wiki link" redirects here. For help with interwiki linking on Wikipedia, see [Help:Interwiki linking](#).

Interwiki links function much the same way as namespaces. A set of interwiki prefixes can be configured to cause, for instance, a page title of [wikiquote:Jimbo Wales](#) to direct the user to the Jimbo Wales article on [Wikiquote](#).^[38] Unlike internal wikilinks, interwiki links lack page existence detection functionality, and accordingly there is no way to tell whether a blue interwiki link is broken or not.

Interlanguage links

[[edit](#)]



An example of interlanguage links

Interlanguage links are the small navigation links that show up in the sidebar in most MediaWiki skins that connect an article with related articles in other languages within the same Wiki family. This can provide language-specific communities connected by a larger context, with all wikis on the same server or each on its own server.^[39]

Previously, Wikipedia used interlanguage links to link an article to other articles on the same topic in other editions of Wikipedia. This was superseded by the launch of Wikidata.^[40]

Content organization

[[edit](#)]

Page tabs and associated pages

[\[edit\]](#)



MediaWiki page tabs, using the "Vector 2010" skin. The red coloration of the "discussion" tab indicates that the article does not yet have a talk page. As with any other red wikilink, clicking on it prompts the user to create the page.

Page tabs are displayed at the top of pages. These tabs allow users to perform actions or view pages that are related to the current page. The available default actions include viewing, editing, and discussing the current page. The specific tabs displayed depend on whether the user is logged into the wiki and whether the user has sysop privileges on the wiki. For instance, the ability to move a page or add it to one's watchlist is usually restricted to logged-in users. The site administrator can add or remove tabs by using JavaScript or installing extensions.^[41]

Each page has an associated history page from which the user can access every version of the page that has ever existed and generate **diffs** between two versions of his choice. Users' contributions are displayed not only here, but also via a "user contributions" option on a sidebar. In a 2004 article, Carl Challborn and Teresa Reimann noted that "While this feature may be a slight deviation from the collaborative, 'ego-less' spirit of wiki purists, it can be very useful for educators who need to assess the contribution and participation of individual student users."^[42]

Namespaces

[\[edit\]](#)

"Talk page" redirects here. For talk pages on Wikipedia, see [Help:Talk pages](#).

MediaWiki provides many features beyond **hyperlinks** for structuring content. One of the earliest such features is **namespaces**. One of Wikipedia's earliest problems had been the separation of encyclopedic content from pages pertaining to maintenance and communal discussion, as well as personal pages about encyclopedia editors. Namespaces are prefixes before a page title (such as "User:" or "Talk:") that serve as descriptors for the page's purpose and allow multiple pages with different functions to exist under the same title. For instance, a page titled "[[The Terminator]]", in the default namespace, could describe **the 1984 movie** starring **Arnold Schwarzenegger**, while a page titled "[[User:The Terminator]]" could be a profile describing a user who chooses this name as a pseudonym. More commonly, each namespace has an associated "Talk:" namespace, which can be used to discuss its contents, such as "User talk:" or "Template talk:". The purpose of having discussion pages is to allow content to be separated from discussion surrounding the content.^{[43][44]}

Namespaces can be viewed as **folders** that separate different basic types of information or functionality. Custom namespaces can be added by the site administrators. There are 16 namespaces by default for content, with 2 "pseudo-namespaces" used for dynamically generated "Special:" pages and links to media files. Each namespace on MediaWiki is numbered: content page namespaces have even numbers and their associated talk page namespaces have odd numbers.^[45]

Category tags

^[edit]

Users can create new categories and add pages and files to those categories by appending one or more category tags to the content text. Adding these tags creates links at the bottom of the page that take the reader to the list of all pages in that category, making it easy to browse related articles.^[46] The use of categorization to organize content has been described as a combination of:

- **Collaborative tagging systems** like **del.icio.us** and
- **Hierarchical classifications** like the **Dewey Decimal Classification**.^[47]

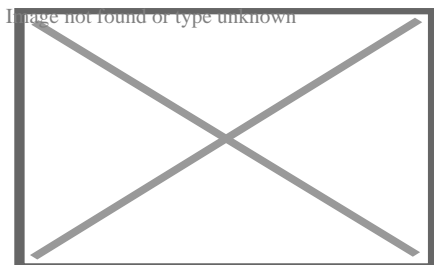
Subpages

^[edit]

In addition to namespaces, content can be ordered using *subpages*. This simple feature provides automatic **breadcrumbs** of the pattern `[[Page title/Subpage title]]` from the page after the slash (in this case, "Subpage title") to the page before the slash (in this case, "Page title").

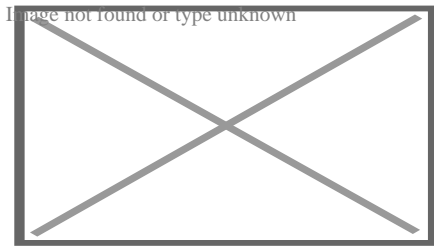
Customization

^[edit]



Users can configure custom **JavaScript** that is executed on every pageview. This has led to JavaScript tools that users can "install", the "navigation popups" tool shown here displays a small preview of an article when hovering over a link title.

If the feature is enabled, users can customize their stylesheets and configure **client-side JavaScript** to be executed with every pageview. On Wikipedia, this has led to a large number of additional tools and helpers developed through the wiki and shared among users. For instance, *navigation popups* is a custom JavaScript tool that shows previews of articles when the user hovers over links and also provides shortcuts for common maintenance tasks.^[48]



A **screenshot** of a wiki using MediaWiki with a customized skin

The entire MediaWiki user interface can be edited through the wiki itself by users with the necessary permissions (typically called "administrators"). This is done through a special namespace with the prefix "MediaWiki:", where each page title identifies a particular user interface message. Using an extension,^[49] it is also possible for a user to create personal scripts, and to choose whether certain sitewide scripts should apply to them by toggling the appropriate options in the user preferences page.

Templates

[edit]

The "MediaWiki:" namespace was originally also used for creating custom text blocks that could then be dynamically loaded into other pages using a special syntax. This content was later moved into its own namespace, "Template:".

Templates are text blocks that can be dynamically loaded inside another page whenever that page is requested. The template is a special link in double **curly brackets** (for example "date=October 2018"), which calls the template (in this case located at **Template:Disputed**) to load in place of the template call.

Templates are **structured documents** containing **attribute–value pairs**. They are defined with **parameters**, to which are assigned **values** when **transcluded** on an article page. The name of the parameter is **delimited** from the value by an **equals sign**. A class of templates known as **infoboxes** is used on Wikipedia to collect and present a subset of information about its subject, usually on the top (mobile view) or top right-hand corner (desktop view) of the document.

Pages in other namespaces can also be transcluded as templates. In particular, a page in the main namespace can be transcluded by prefixing its title with a colon; for example, **:MediaWiki** transcludes the article "MediaWiki" from the main namespace. Also, it is possible to mark the portions of a page that should be transcluded in several ways, the most basic of which are:^[50]

- `<noinclude>...</noinclude>`, which marks content that is not to be transcluded;
- `<includeonly>...</includeonly>`, which marks content that is not rendered unless it is transcluded;
- `<onlyinclude>...</onlyinclude>`, which marks content that is to be the *only* content transcluded.

A related method, called template *substitution* (called by adding `subst:` at the beginning of a template link) inserts the contents of the template into the target page (like a **copy and paste** operation), instead of loading the template contents dynamically whenever the page is loaded. This can lead to inconsistency when using templates, but may be useful in certain cases, and in most cases requires fewer **server** resources (the actual amount of savings can vary depending on wiki configuration and the complexity of the template).

Templates have found many different uses. Templates enable users to create complex table layouts that are used consistently across multiple pages, and where only the content of the tables gets inserted using template parameters. Templates are frequently used to identify problems with a Wikipedia article by putting a template in the article. This template then outputs a graphical box stating that the article content is disputed or in need of some other attention, and also categorize it so that articles of this nature can be located. Templates are also used on user pages to send users standard messages welcoming them to the site,[51] giving them awards for outstanding contributions,[52][53] warning them when their behavior is considered inappropriate,[54] notifying them when they are blocked from editing,[55] and so on.

Groups and restriction of access

[**edit**]

MediaWiki offers flexibility in creating and defining user groups. For instance, it would be possible to create an arbitrary "ninja" group that can block users and delete pages, and whose edits are hidden by default in the recent changes log. It is also possible to set up a group of "autoconfirmed" users that one becomes a member of after making a certain number of edits and waiting a certain number of days.[56] Some groups that are enabled by default are bureaucrats and sysops. Bureaucrats have the power to change other users' rights. Sysops have power over page protection and **deletion** and the blocking of users from editing. MediaWiki's available controls on editing rights have been deemed sufficient for publishing and maintaining important documents such as a manual of **standard operating procedures** in a hospital.[57]

MediaWiki comes with a basic set of features related to restricting access, but its original and ongoing design is driven by functions that largely relate to content, not content segregation. As a result, with minimal exceptions (related to specific tools and their related "Special" pages), page access control has never been a high priority in core development and developers have stated that users requiring secure user access and authorization controls should not rely on MediaWiki, since it was never designed for these kinds of situations. For instance, it is extremely difficult to

create a wiki where only certain users can read and access some pages.[58] Here, wiki engines like Foswiki, MoinMoin and Confluence provide more flexibility by supporting advanced security mechanisms like access control lists.

Extensibility

[edit]

The MediaWiki codebase contains various hooks using callback functions to add additional PHP code in an extensible way. This allows developers to write extensions without necessarily needing to modify the core or having to submit their code for review. Installing an extension typically consists of adding a line to the configuration file, though in some cases additional changes such as database updates or core patches are required.

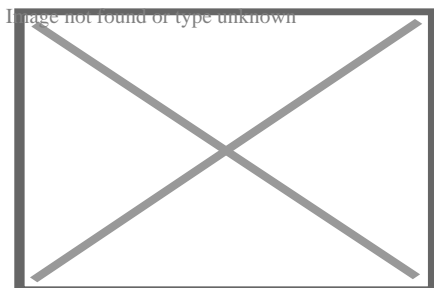
Five main extension points were created to allow developers to add features and functionalities to MediaWiki. Hooks are run every time a certain event happens; for instance, the ArticleSaveComplete hook occurs after a save article request has been processed.[59] This can be used, for example, by an extension that notifies selected users whenever a page edit occurs on the wiki from new or anonymous users.[60] New tags can be created to process data with opening and closing tags (<newtag>...</newtag>).[61] Parser functions can be used to create a new command (...).[62] New special pages can be created to perform a specific function. These pages are dynamically generated. For example, a special page might show all pages that have one or more links to an external site or it might create a form providing user submitted feedback.[63] Skins allow users to customize the look and feel of MediaWiki.[64] A minor extension point allows the use of Amazon S3 to host image files.[65]

Extensions

[edit]

Text manipulation

[edit]



Among the most popular extensions is a parser function extension, ParserFunctions, which allows different content to be rendered based on the result of [conditional statements](#).[\[66\]](#) These conditional statements can perform functions such as evaluating whether a parameter is empty, comparing strings, evaluating mathematical expressions, and returning one of two values depending on whether a page exists. It was designed as a replacement for a notoriously inefficient template called Qif.[\[67\]](#) Schindler recounts the history of the ParserFunctions extension as follows:[\[68\]](#)

In 2006 some Wikipedians discovered that through an intricate and complicated interplay of templating features and CSS they could create conditional wiki text, i.e. text that was displayed if a template parameter had a specific value. This included repeated calls of templates within templates, which bogged down the performance of the whole system. The developers faced the choice of either disallowing the spreading of an obviously desired feature by detecting such usage and explicitly disallowing it within the software or offering an efficient alternative. The latter was done by Tim Starling, who announced the introduction of parser functions, wiki text that calls functions implemented in the underlying software. At first, only conditional text and the computation of simple mathematical expressions were implemented, but this already increased the possibilities for wiki editors enormously. With time further parser functions were introduced, finally leading to a framework that allowed the simple writing of extension functions to add arbitrary functionalities, like e.g. geo-coding services or widgets. This time the developers were clearly reacting to the demand of the community, being forced either to fight the solution of the issue that the community had (i.e. conditional text), or offer an improved technical implementation to replace the previous practice and achieve an overall better performance.

Another parser functions extension, StringFunctions, was developed to allow evaluation of string length, string position, and so on. Wikimedia communities, having created awkward workarounds to accomplish the same functionality,[\[69\]](#) clamored for it to be enabled on their projects.[\[70\]](#) Much of its functionality was eventually integrated into the ParserFunctions extension,[\[71\]](#) albeit disabled by default and accompanied by a warning from Tim Starling that enabling string functions would allow users "to implement their own parsers in the ugliest, most inefficient programming language known to man: MediaWiki wikitext with ParserFunctions."[\[72\]](#)

Since 2012 an extension, Scribunto, has existed that allows for the creation of "modules"—wiki pages written in the scripting language [Lua](#)—which can then be run within templates and standard wiki pages. Scribunto has been installed on Wikipedia and other Wikimedia sites since 2013 and is used heavily on those sites. Scribunto code runs significantly faster than corresponding wikitext code using ParserFunctions.[\[73\]](#)

For footnotes and academic-related display

[\[edit\]](#)

Another very popular extension is a citation extension that enables footnotes to be added to pages using inline references.^[74] This extension has, however, been criticized for being difficult to use and requiring the user to memorize complex syntax. A gadget called **RefToolbar** attempts to make it easier to create citations using common templates. MediaWiki has some extensions that are well-suited for academia, such as mathematics extensions^[75] and an extension that allows molecules to be rendered in **3D**.^[76]

Integration

[\[edit\]](#)

A generic Widgets extension exists that allows MediaWiki to integrate with virtually anything. Other examples of extensions that could improve a wiki are category suggestion extensions^[77] and extensions for inclusion of **Flash Videos**,^[78] YouTube videos,^[79] and **RSS feeds**.^[80] **Metavid**, a site that archives video footage of the **U.S. Senate** and **House** floor proceedings, was created using code extending MediaWiki into the domain of collaborative video authoring.^[81]

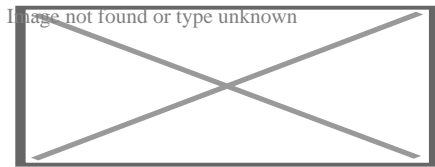
Combating linkspam

[\[edit\]](#)

There are many **spambots** that search the web for MediaWiki installations and add **linkspam** to them, despite the fact that MediaWiki uses the **nofollow** attribute to discourage such attempts at **search engine optimization**.^[82] Part of the problem is that third party republishers, such as **mirrors**, may not independently implement the nofollow tag on their websites, so marketers can still get **PageRank** benefit by inserting links into pages when those entries appear on third party websites.^[83] **Anti-spam** extensions have been developed to combat the problem by introducing **CAPTCHAs**,^[84] **blacklisting** certain URLs,^[85] and allowing bulk deletion of pages recently added by a particular user.^[86]

Searches and queries

[[edit](#)]



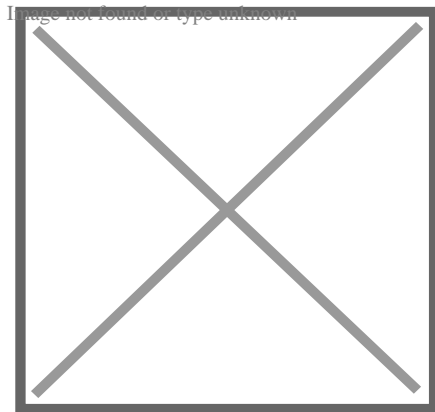
A search box showing a [drop-down list](#)

MediaWiki comes pre-installed with a standard text-based search. Extensions exist to let MediaWiki use more sophisticated third-party search engines, including [Elasticsearch](#) (which since 2014 has been in use on Wikipedia), [Lucene](#)^[87] and [Sphinx](#).^[88]

Various MediaWiki extensions have also been created to allow for more complex, [faceted search](#), on both data entered within the wiki and on [metadata](#) such as pages' revision history.^{[89][90]} [Semantic MediaWiki](#) is one such extension.^{[91][92]}

Rich content

[[edit](#)]



[Images](#) can be arranged in galleries, a feature that is used extensively for Wikimedia's media archive, [Wikimedia Commons](#).

Various extensions to MediaWiki support [rich content](#) generated through specialized syntax. These include mathematical formulas using [LaTeX](#), graphical timelines over mathematical [plotting](#), [musical scores](#) and [Egyptian hieroglyphs](#).

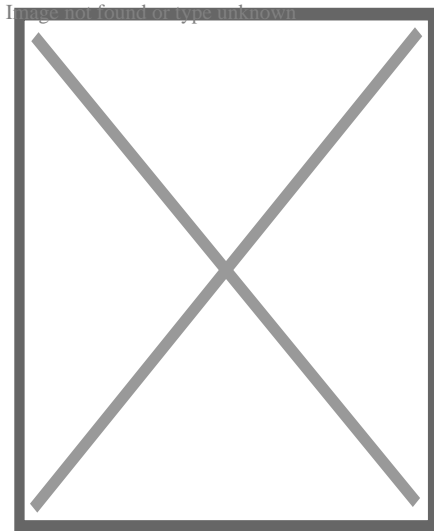
The software supports a wide variety of uploaded media files, and allows image galleries and thumbnails to be generated with relative ease. There is also support for [Exif metadata](#). MediaWiki

operates the [Wikimedia Commons](#), one of the largest [free content](#) media archives.

For WYSIWYG editing, [VisualEditor](#) is available to use in MediaWiki which simplifying editing process for editors and has been bundled since MediaWiki 1.35.^[93] Other extensions exist for handling WYSIWYG editing to different degrees.^[94]

Database

[\[edit\]](#)



A schematic of the MediaWiki database structure

MediaWiki can use either the [MySQL/MariaDB](#), [PostgreSQL](#) or [SQLite relational database management system](#). Support for [Oracle Database](#) and [Microsoft SQL Server](#) has been dropped since MediaWiki 1.34.^[95] A MediaWiki database contains several dozen [tables](#), including a page table that contains page titles, page ids, and other metadata;^[96] and a revision table to which is added a new row every time an edit is made, containing the page id, a brief textual summary of the change performed, the user name of the article editor (or its IP address the case of an unregistered user) and a timestamp.^{[97][98]}

In a 4½ year period prior to 2008, the MediaWiki database had 170 [schema](#) versions.^[99] Possibly the largest schema change was done in 2005 with MediaWiki 1.5, when the storage of metadata was separated from that of content, to improve performance flexibility. When this upgrade was applied to Wikipedia, the site was locked for editing, and the schema was converted to the new version in about 22 hours. Some software enhancement proposals, such as a proposal to allow sections of articles to be watched via watchlist, have been rejected because the necessary schema changes would have required excessive Wikipedia downtime.^[100]

Performance and storage

[\[edit\]](#)

Because it is used to run one of the highest-traffic sites on the Web, Wikipedia, MediaWiki's performance and **scalability** have been highly optimized.[101] MediaWiki supports **Squid**, **load-balanced** database replication, client-side caching, **memcached** or table-based caching for frequently accessed processing of query results, a simple static file cache, feature-reduced operation, revision compression, and a job queue for database operations. MediaWiki developers have attempted to optimize the software by avoiding expensive algorithms, database queries, etc., caching every result that is expensive and has temporal locality of reference, and focusing on the hot spots in the code through **profiling**.[102]

MediaWiki code is designed to allow for data to be written to a read-write database and read from read-only databases, although the read-write database can be used for some read operations if the read-only databases are not yet up to date. **Metadata**, such as article revision history, article relations (links, categories etc.), user accounts and settings can be stored in core databases and cached; the actual revision text, being more rarely used, can be stored as append-only **blobs** in external storage. The software is suitable for the operation of large-scale **wiki farms** such as **Wikimedia**, which had about 800 wikis as of August 2011. However, MediaWiki comes with no built-in GUI to manage such installations.

Empirical evidence shows most revisions in MediaWiki databases tend to differ only slightly from previous revisions. Therefore, subsequent revisions of an article can be concatenated and then compressed, achieving very high **data compression ratios** of up to 100×.[102]

For more information on the architecture, such as how it stores wikitext and assembles a page, see **External links**.

Limitations

[**edit**]

The parser serves as the *de facto* standard for the MediaWiki syntax, as no formal syntax has been defined. Due to this lack of a formal definition, it has been difficult to create **WYSIWYG** editors for MediaWiki, although several WYSIWYG extensions do exist, including the popular **VisualEditor**.

MediaWiki is not designed to be a suitable replacement for dedicated **online forum** or blogging software,[103] although extensions do exist to allow for both of these.[104][105]

It is common for new MediaWiki users to make certain mistakes, such as forgetting to sign posts with four tildes (~~~~),[106] or manually entering a plaintext signature,[107] due to unfamiliarity with the idiosyncratic particulars involved in communication on MediaWiki discussion pages. On the other hand, the format of these discussion pages has been cited as a strength by one educator, who stated that it provides more fine-grain capabilities for discussion than traditional threaded discussion forums. For example, instead of 'replying' to an entire message, the participant in a discussion can create a hyperlink to a new wiki page on any word from the original page. Discussions are easier to follow since the content is available via hyperlinked wiki page, rather than a series of reply messages on a traditional threaded discussion forum. However,

except in few cases, students were not using this capability, possibly because of their familiarity with the traditional linear discussion style and a lack of guidance on how to make the content more 'link-rich'.^[108]

MediaWiki by default has little support for the creation of dynamically assembled documents, or pages that aggregate data from other pages. Some research has been done on enabling such features directly within MediaWiki.^[109] The **Semantic MediaWiki** extension provides these features. It is not in use on Wikipedia, but in more than 1,600 other MediaWiki installations.^[110] The Wikibase Repository and Wikibase Repository client are however implemented in **Wikidata** and **Wikipedia** respectively, and to some extent provides **semantic web** features, and linking of centrally stored data to infoboxes in various Wikipedia articles.

Upgrading MediaWiki is usually fully automated, requiring no changes to the site content or template programming. Historically troubles have been encountered when upgrading from significantly older versions.^[111]

Security

^[edit]

MediaWiki developers have enacted security standards, both for core code and extensions.^[112] **SQL queries** and HTML output are usually done through wrapper functions that handle validation, escaping, filtering for prevention of **cross-site scripting** and **SQL injection**.^[113] Many security issues have had to be patched after a MediaWiki version release,^[114] and accordingly MediaWiki.org states, "The most important security step you can take is to keep your software up to date" by subscribing to the announcement **mailing list** and installing security updates that are announced.^[115]

Support

^[edit]

Support for MediaWiki users consists of:

- MediaWiki.org, including the Support Desk.
- An official mailing list, Mediawiki-l.
- Several books have been written about MediaWiki administration,^[116] including some free online books.^{[117][118]}

License

^[edit]

MediaWiki is free and open-source and is distributed under the terms of the **GNU General Public License** version 2 or any later version. Its documentation, located at its official website at www.mediawiki.org, is released under the **Creative Commons BY-SA 4.0** license, with a set of

help pages intended to be freely copied into fresh wiki installations and/or distributed with MediaWiki software in the **public domain** instead to eliminate legal issues for wikis with other licenses.[119][120] MediaWiki's development has generally favored the use of **open-source media formats**.[121]

Development

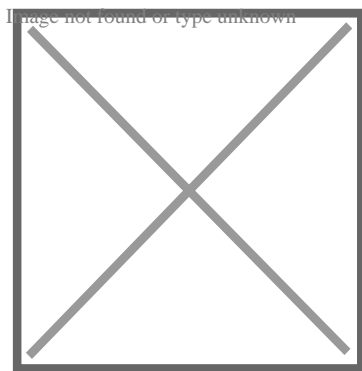
[**edit**]

MediaWiki has an active volunteer community for development and maintenance. MediaWiki developers are spread around the world, though with a majority in the United States and Europe. Face-to-face meetings and programming sessions for MediaWiki developers have been held once or several times a year since 2004.[122]

Anyone can submit **patches** to the project's **Git/Gerrit repository**.[123] There are also paid programmers who primarily develop projects for the **Wikimedia Foundation**. MediaWiki developers participate in the **Google Summer of Code** by facilitating the assignment of mentors to students wishing to work on MediaWiki core and extension projects.[124] During the year prior to November 2012, there were about two hundred developers who had committed changes to the MediaWiki core or extensions.[125] Major MediaWiki releases are generated approximately every six months by taking snapshots of the development branch, which is kept continuously in a runnable state;[126] **minor releases**, or **point releases**, are issued as needed to correct **bugs** (especially security problems). MediaWiki is developed on a **continuous integration** development model, in which software changes are pushed live to Wikimedia sites on regular basis.[126] MediaWiki also has a public **bug** tracker, *phabricator.wikimedia.org*, which runs **Phabricator**. The site is also used for **feature** and **enhancement** requests.

History

[**edit**]



Magnus Manske in 2012

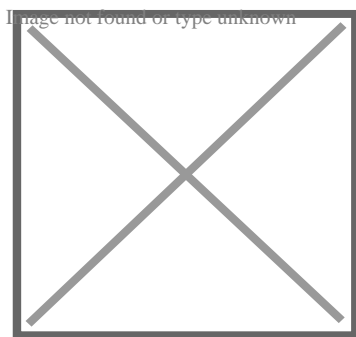
When Wikipedia was launched in January 2001, it ran on an existing **wiki software** system, **UseModWiki**. UseModWiki is written in the **Perl** programming language, and stores all wiki pages in text (**.txt**) files. This software soon proved to be limiting, in both functionality and performance.

In mid-2001, **Magnus Manske**—a developer and student at the **University of Cologne**, as well as a **Wikipedia editor**—began working on new software that would replace UseModWiki, specifically designed for use by Wikipedia. This software was written in the **PHP** scripting language, and stored all of its information in a **MySQL** database. The new software was largely developed by August 24, 2001, and a test wiki for it was established shortly thereafter.

The first full implementation of this software was the new **Meta Wikipedia** on November 9, 2001. There was a desire to have it implemented immediately on the English-language Wikipedia.^[127] However, Manske was apprehensive about any potential **bugs** harming the nascent website during the period of the final exams he had to complete immediately prior to Christmas;^[128] this led to the launch on the English-language Wikipedia being delayed until January 25, 2002. The software was then, gradually, deployed on all the Wikipedia language sites of that time. This software was referred to as "the PHP script" and as "phase II", with the name "phase I", retroactively given to the use of UseModWiki.

Increasing usage soon caused load problems to arise again, and soon after, another rewrite of the software began; this time being done by **Lee Daniel Crocker**, which became known as "phase III". This new software was also written in PHP, with a MySQL backend, and kept the basic interface of the phase II software, but with the added functionality of a wider **scalability**. The "phase III" software went live on Wikipedia in July 2002.

The **Wikimedia Foundation** was announced on June 20, 2003. In July, Wikipedia contributor Daniel Mayer suggested the name "MediaWiki" for the software, as a play on "Wikimedia".^[129] The MediaWiki name was gradually phased in, beginning in August 2003. The name has frequently caused confusion due to its (intentional) similarity to the "Wikimedia" name (which itself is similar to "Wikipedia").^[130] The first version of MediaWiki, 1.1, was released in December 2003.



MediaWiki logo until April 1, 2021

The old **product logo** was created by **Erik Möller**, using a flower photograph taken by **Florence Nibart-Devouard**, and was originally submitted to the logo contest for a new **Wikipedia logo**, held from July 20 to August 27, 2003.^{[131][132]} The logo came in third place, and was chosen to represent MediaWiki rather than Wikipedia, with the second place logo being used for the Wikimedia Foundation.^[133] The double square brackets (**[[**) symbolize the **syntax** MediaWiki uses for creating **hyperlinks** to other wiki pages; while the **sunflower** represents the diversity of content on Wikipedia, its constant growth, and the wilderness.^[134]

Later, Brooke Vibber, the **chief technical officer** of the **Wikimedia Foundation**,^[135] took up the role of **release manager**.^{[136][101]}

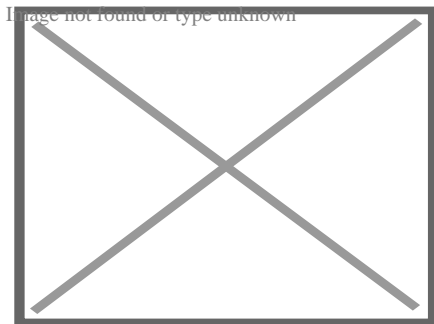
Major milestones in MediaWiki's development have included: the **categorization system** (2004); **parser** functions, (2006); **Flagged Revisions**, (2008);^[68] the "*ResourceLoader*", a delivery system for **CSS** and JavaScript (2011);^[137] and the **VisualEditor**, a "what you see is what you get" (**WYSIWYG**) editing platform (2013).^[138]

The contest of designing a new logo was initiated on June 22, 2020, as the old logo was a bitmap image and had "high details", leading to problems when rendering at high and low resolutions, respectively. After two rounds of voting, the new and current MediaWiki logo designed by **Serhio Magpie** was selected on October 24, 2020, and officially adopted on April 1, 2021.^[139]

Sites using MediaWiki

^[edit]

See also: **Category:MediaWiki websites**



Fandom also makes use of MediaWiki.

MediaWiki's most famous use has been in **Wikipedia** and, to a lesser degree, the Wikimedia Foundation's other projects. **Fandom**, a **wiki hosting service** formerly known as Wikia, runs on MediaWiki. Other public wikis that run on MediaWiki include **wikiHow** and **SNPedia**. **WikiLeaks** began as a MediaWiki-based site, but is no longer a wiki.

A number of alternative wiki encyclopedias to Wikipedia run on MediaWiki, including **Citizendium**, **Metapedia**, **Scholarpedia** and **Conservapedia**. MediaWiki is also used internally by a large number of companies, including **Novell** and **Intel**.^{[140][141]}

Notable usages of MediaWiki within governments include **Intellipedia**, used by the **United States Intelligence Community**, **Diplopedia**, used by the **United States Department of State**, and milWiki, a part of **milSuite** used by the **United States Department of Defense**. **United Nations agencies** such as the **United Nations Development Programme** and **INSTRAW** chose to implement their wikis using MediaWiki, because "this software runs Wikipedia and is therefore guaranteed to be thoroughly tested, will continue to be developed well into the future, and future technicians on these wikis will be more likely to have exposure to MediaWiki than any other wiki software."^[142]

The **Free Software Foundation** uses MediaWiki to implement the **LibrePlanet** site.^[143]

Comparison to other online collaboration software

[[edit](#)]

Main article: [Comparison of wiki software](#)

Users of online [collaboration software](#) are familiar with MediaWiki's functions and layout due to its noted use on Wikipedia. A 2006 overview of social software in academia observed that "Compared to other wikis, MediaWiki is also fairly aesthetically pleasing, though simple, and has an easily customized side menu and [stylesheet](#)."^[144] However, in one assessment in 2006, [Confluence](#) was deemed to be a superior product due to its very usable API and ability to better support multiple wikis.^[76]


A 2009 study at the [University of Hong Kong](#) compared [TWiki](#) to MediaWiki. The authors noted that TWiki has been considered as a collaborative tool for the development of educational papers and technical projects, whereas MediaWiki's most noted use is on Wikipedia. Although both platforms allow discussion and tracking of progress, TWiki has a "Report" part that MediaWiki lacks. Students perceived MediaWiki as being easier to use and more enjoyable than TWiki. When asked whether they recommended using MediaWiki for [knowledge management](#) course group project, 15 out of 16 respondents expressed their preference for MediaWiki giving answers of great certainty, such as "of course", "for sure".^[145] TWiki and MediaWiki both have flexible plug-in architecture.^[146]

A 2009 study that compared students' experience with MediaWiki to that with [Google Docs](#) found that students gave the latter a much higher rating on user-friendly layout.^[147]

A 2021 study conducted by the [Brazilian Nuclear Engineering Institute](#) compared a MediaWiki-based [knowledge management system](#) against two others that were based on [DSpace](#) and [Open Journal Systems](#), respectively.^[148] It highlighted ease of use as an advantage of the MediaWiki-based system, noting that because the Wikimedia Foundation had been developing MediaWiki for a site aimed at the general public (Wikipedia), "its user interface was designed to be more user-friendly from start, and has received large user feedback over a long time", in contrast to DSpace's and OJS's focus on niche audiences.^[148]

See also

[[edit](#)]

-  [Free and open-source software portal](#)
- [List of content management systems](#)
- [List of wiki software](#)
- [BlueSpice](#)
- [Semantic MediaWiki](#)
- [XOWA](#) – for viewing Wikipedia and other wikis offline
- [PHP](#) – a programming language that powers MediaWiki

References

[[edit](#)]

1. ^ ["Announcing MediaWiki 1.43.0"](#). December 21, 2024.
2. ^ Reed, Sam (December 19, 2019). ["Announcing MediaWiki 1.34.0"](#). mediawiki-announce (Mailing list). [Archived](#) from the original on December 19, 2019. Retrieved December 19, 2019.
3. ^ ["Names.php • mediawiki"](#). github.com. April 8, 2021. [Archived](#) from the original on July 15, 2021. Retrieved May 19, 2021.
4. ^ ["Copyright"](#). mediawiki.org. [Archived](#) from the original on September 19, 2015. Retrieved September 7, 2015.
5. ^ Magnus Manske's announcement of ["PHP Wikipedia"](#), wikipedia-l, August 24, 2001
6. ^ [Barrett, Daniel J.](#) (October 2008). [MediaWiki](#). O'Reilly Media. ISBN 978-0-596-51979-7. Retrieved April 23, 2010.
7. ^ [a b](#) ["What is MediaWiki?"](#). MediaWiki. January 9, 2021. [Archived](#) from the original on July 22, 2018. Retrieved March 27, 2021.
8. ^ ["Wikipedia:Statistics – Wikipedia, the free encyclopedia"](#). Wikipedia. [Archived](#) from the original on August 28, 2021. Retrieved May 30, 2010.
9. ^ [a b](#) ["Message group statistics: MediaWiki core"](#). translatewiki.net. August 20, 2023. [Archived](#) from the original on August 20, 2023. Retrieved August 20, 2023. "488 languages (not including languages that are supported but have no translations)"
10. ^ ["Category:MediaWiki configuration settings"](#). MediaWiki. September 11, 2016. [Archived](#) from the original on November 10, 2014. Retrieved September 11, 2016.
11. ^ ["Extension Matrix"](#). MediaWiki. [Archived](#) from the original on September 11, 2016. Retrieved September 6, 2017.
12. ^ Lerner, Reuven M. (February 23, 2006), [Installing and Customizing MediaWiki](#), Linux Journal, archived from [the original](#) on April 6, 2010, retrieved April 23, 2010
13. ^ Petrazickis, Leons (2009), [Deploying PHP applications on IBM DB2 in the cloud: MediaWiki as a case study](#), Proceedings of the 2009 Conference of the Center for Advanced Studies on Collaborative Research, pp. 304–305, doi:10.1145/1723028.1723069, S2CID 27463043
14. ^ ["Manual:\\$wgEnableUploads"](#). MediaWiki. [Archived](#) from the original on June 25, 2010. Retrieved May 30, 2010.
15. ^ ["Manual:\\$wgLogo"](#). MediaWiki. December 12, 2009. [Archived](#) from the original on June 25, 2010. Retrieved May 30, 2010.
16. ^ ["Manual:LocalSettings.php"](#). MediaWiki. March 29, 2007. [Archived](#) from the original on June 25, 2010. Retrieved May 30, 2010.
17. ^ ["Extension:AbuseFilter"](#). MediaWiki. [Archived](#) from the original on June 25, 2010. Retrieved May 30, 2010.
18. ^ Cacycle. ["wikEd"](#). [Archived](#) from the original on November 23, 2007.
19. ^ ["Manual:Installation guide"](#). MediaWiki. [Archived](#) from the original on June 25, 2010. Retrieved May 30, 2010.
20. ^ ["Manual:Extending wiki markup"](#). MediaWiki. [Archived](#) from the original on May 1, 2011. Retrieved May 30, 2010.

21. ^ ["HTML to Wiki Converter – tables"](#). WMF Labs. March 29, 2008. [Archived](#) from the original on July 13, 2014. Retrieved June 12, 2014.
22. ^ ["Markup spec"](#). MediaWiki. [Archived](#) from the original on December 19, 2007. Retrieved May 30, 2010.
23. ^ ["Extricating Meaning from Wikimedia Article Archives"](#) (PDF). [Archived](#) (PDF) from the original on March 10, 2011. Retrieved January 2, 2011.
24. ^ Jakes, David (August 15, 2006), [Wild about Wikis](#), Tech & Learning, archived from [the original](#) on May 2, 2010, retrieved April 23, 2010
25. ^ Foley, Brian & Chang, Tae (2008), [Wiki as a professional development tool](#) (PDF), Technology and Teacher Education, [archived](#) (PDF) from the original on April 30, 2011, retrieved April 23, 2010
26. ^ ["API"](#). MediaWiki. May 17, 2010. [Archived](#) from the original on May 27, 2010. Retrieved May 30, 2010.
27. ^ ["Pywikibot – MediaWiki"](#). mediawiki.org. [Archived](#) from the original on March 9, 2018. Retrieved March 16, 2018.
28. ^ ["Wikipedia:AutoWikiBrowser – Wikipedia, the free encyclopedia"](#). En.wikipedia.org. [Archived](#) from the original on April 20, 2010. Retrieved May 30, 2010.
29. ^ Bartolo, Laura M.; Lowe, Cathy S.; Songar, Poonam; Tandy, Robert J. (May 20, 2009), [Facilitating Wiki/Repository Communication with Metadata](#), Georgia Institute of Technology, archived from [the original](#) on January 9, 2011, retrieved April 23, 2010
30. ^ ["API:Client code"](#). MediaWiki. May 24, 2010. [Archived](#) from the original on June 26, 2010. Retrieved May 30, 2010.
31. ^ ["Tags – Wikipedia, the free encyclopedia"](#). En.wikipedia.org. [Archived](#) from the original on March 4, 2010. Retrieved May 30, 2010.
32. ^ ["Manual:Tags"](#). MediaWiki. August 31, 2009. [Archived](#) from the original on June 25, 2010. Retrieved May 30, 2010.
33. ^ ["Wikipedia:Huggle – Wikipedia, the free encyclopedia"](#). En.wikipedia.org. [Archived](#) from the original on March 31, 2011. Retrieved May 30, 2010.
34. ^ ["IRC/Channels"](#). Meta-Wiki. [Archived](#) from the original on March 23, 2010. Retrieved May 30, 2010.
35. ^ Daniel Nasaw (July 25, 2012). ["Meet the 'bots' that edit Wikipedia"](#). BBC News. [Archived](#) from the original on July 28, 2012. Retrieved July 30, 2012.
36. ^ ["Manual:Watchlist"](#). MediaWiki. November 24, 2009. [Archived](#) from the original on May 1, 2011. Retrieved May 30, 2010.
37. ^ Kevin Yager (March 16, 2006), "Wiki ware could harness the Internet for science", *Nature*, **440** (7082): 278, [Bibcode:2006Natur.440..278Y](#), [doi:10.1038/440278a](#), [PMID 16541049](#)
38. ^ ["Manual:Interwiki"](#). MediaWiki. [Archived](#) from the original on December 3, 2010. Retrieved May 30, 2010.
39. ^ ["Interlanguage links"](#). MediaWiki. [Archived](#) from the original on March 12, 2021. Retrieved March 17, 2021.
40. ^ Pintscher, Lydia (September 23, 2013). ["Wikidata is Here!"](#). [Commons:Village pump](#). [Archived](#) from the original on December 6, 2021. Retrieved March 17, 2021.

41. ^ ["Help:Navigation"](#). MediaWiki. May 21, 2010. *Archived* from the original on May 27, 2010. Retrieved May 30, 2010.
42. ^ Carl Challborn & Teresa Reimann (December 2004), [Wiki products: a comparison](#) (PDF), Athabasca University, *archived* (PDF) from the original on December 23, 2010, retrieved April 23, 2010
43. ^ Newman, Aaron; Steinberg, Adam; Thomas, Jeremy (2008). *Enterprise 2.0 Implementation*. McGraw-Hill Professional. p. 185. ISBN 978-0-07-159160-7.
44. ^ Malcolm, Jeremy (2008). *Multi-Stakeholder Governance and the Internet Governance Forum*. Terminus Press. pp. 188, 280. ISBN 978-0-9805084-0-6.
45. ^ Ebersbach, Anja; Glaser, Markus; Heigl, Richard; Dueck, Gunter (2006). *Wiki*. Springer. pp. 55, 80–82, 109, 120–121, 156. ISBN 978-3-540-25995-4.
46. ^ ["Help:Categories"](#). MediaWiki. *Archived* from the original on June 25, 2010. Retrieved May 30, 2010.
47. ^ Jakob Voss (April 27, 2006). "Collaborative thesaurus tagging the Wikipedia way". *arXiv:cs.IR/0604036*.
48. ^ Lupin. ["Wikipedia:Tools/Navigation popups"](#). *Archived* from the original on July 18, 2006.
49. ^ ["Extension:Gadgets"](#). MediaWiki. March 30, 2010. *Archived* from the original on June 25, 2010. Retrieved May 30, 2010.
50. ^ Anderson, Mark; Carr, Leslie; Millard, David E. (July 4, 2017). *There and Here: Patterns of Content Transclusion in Wikipedia*. 28th ACM Conference on Hypertext and Social Media. Prague, Czech Republic: ACM. pp. 115–124. doi:10.1145/3078714.3078726. ISBN 978-1-4503-4708-2.
51. ^ [ÅfÆ'Å†â€™Åfâ€ Åçâ,-â,,çÅfÆ'Åçâ,-Å ÅfÅçÅçâ€šÅ-Åçâ€žÅçÅfÆ'Å†â€™ÅfÅçÅçâ€šÅ-Å,Å](#) (May 16, 2010). ["Template:Welcome – Wikipedia, the free encyclopedia"](#). En.wikipedia.org. *Archived* from the original on May 6, 2011. Retrieved May 30, 2010.
52. ^ T Kriplean; I Beschastnikh; et al. (2008), "Articulations of wikiwork: Uncovering valued work in wikipedia through barnstars", *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, Proceedings of the ACM, pp. 47–56, doi: 10.1145/1460563.1460573, ISBN 9781605580074, S2CID 7164949
53. ^ [ÅfÆ'Å†â€™Åfâ€ Åçâ,-â,,çÅfÆ'Åçâ,-Å ÅfÅçÅçâ€šÅ-Åçâ€žÅçÅfÆ'Å†â€™ÅfÅçÅçâ€šÅ-Å,Å](#) ["Wikipedia:Barnstars – Wikipedia, the free encyclopedia"](#). En.wikipedia.org. *Archived* from the original on June 24, 2009. Retrieved May 30, 2010.
54. ^ [ÅfÆ'Å†â€™Åfâ€ Åçâ,-â,,çÅfÆ'Åçâ,-Å ÅfÅçÅçâ€šÅ-Åçâ€žÅçÅfÆ'Å†â€™ÅfÅçÅçâ€šÅ-Å,Å](#) ["Template:Test – Wikipedia, the free encyclopedia"](#). En.wikipedia.org. *Archived* from the original on November 11, 2009. Retrieved May 30, 2010.
55. ^ ["Template:Test5 – Wikipedia, the free encyclopedia"](#). En.wikipedia.org. June 19, 2008. *Archived* from the original on April 24, 2009. Retrieved May 30, 2010.
56. ^ ["Manual:User rights management"](#). MediaWiki. *Archived* from the original on June 25, 2010. Retrieved May 30, 2010.
57. ^ H Zielke; W Boemke; M Kastrup; C Melzer (November 21, 2007), *Operating Procedures in Clinical Practice* (PDF), Royal College of Anaesthetists, *archived* (PDF) from the original on May 15, 2011, retrieved April 25, 2010

58. ^ ["Security issues with authorization extensions"](#). MediaWiki. [Archived](#) from the original on June 26, 2010. Retrieved May 30, 2010.
59. ^ ["Manual:Hooks/ArticleSaveComplete"](#). MediaWiki. May 26, 2010. [Archived](#) from the original on November 10, 2014. Retrieved May 30, 2010.
60. ^ ["Extension:Recent Activity Notify"](#). MediaWiki. [Archived](#) from the original on September 27, 2014. Retrieved May 30, 2010.
61. ^ ["Manual:Tag extensions"](#). MediaWiki. May 21, 2010. [Archived](#) from the original on October 12, 2014. Retrieved May 30, 2010.
62. ^ ["Manual:Parser functions"](#). MediaWiki. March 22, 2010. [Archived](#) from the original on October 18, 2014. Retrieved May 30, 2010.
63. ^ ["Manual:Special pages"](#). MediaWiki. [Archived](#) from the original on November 10, 2014. Retrieved May 30, 2010.
64. ^ ["Manual:Skins"](#). MediaWiki. May 14, 2010. [Archived](#) from the original on November 25, 2014. Retrieved May 30, 2010.
65. ^ ["Manual:Integration with S3"](#). MediaWiki. March 22, 2010. [Archived](#) from the original on November 27, 2014. Retrieved May 30, 2010.
66. ^ ["Extension:ParserFunctions"](#). MediaWiki. December 25, 2009. [Archived](#) from the original on June 25, 2010. Retrieved May 30, 2010.
67. ^ ["Wikipedia:Miscellany for deletion/Template:Qif – Wikipedia, the free encyclopedia"](#). En.wikipedia.org. [Archived](#) from the original on February 25, 2015. Retrieved May 30, 2010.
68. ^ [a b](#) M Schindler; D Vrandecic (2009), [Introducing new features to Wikipedia](#), Proceedings of WebSci, [archived](#) from the original on June 24, 2018, retrieved June 24, 2018
69. ^ ["Category:String manipulation templates – Wikipedia, the free encyclopedia"](#). En.wikipedia.org. May 15, 2010. [Archived](#) from the original on May 6, 2011. Retrieved May 30, 2010.
70. ^ ["Bug 6455 – Enable StringFunctions on WMF wikis"](#). bugzilla.wikimedia.org. [Archived](#) from the original on January 22, 2012. Retrieved October 9, 2010.
71. ^ ["Extension:StringFunctions"](#). MediaWiki. [Archived](#) from the original on June 25, 2010. Retrieved May 30, 2010.
72. ^ ["r51497 – Code Review"](#). MediaWiki. [Archived](#) from the original on November 27, 2014. Retrieved May 30, 2010.
73. ^ ["Lua performance"](#). [Archived](#) from the original on August 24, 2018. Retrieved December 27, 2018.
74. ^ ["Extension:Cite"](#). MediaWiki. May 3, 2010. [Archived](#) from the original on October 20, 2014. Retrieved May 30, 2010.
75. ^ ["Category:Math extensions"](#). MediaWiki. December 26, 2009. [Archived](#) from the original on May 1, 2011. Retrieved May 30, 2010.
76. ^ [a b](#) Marieke Guy (January 2007), [Wikido: Exploiting the Potential of Wikis](#), Ariadne, [archived](#) from the original on April 7, 2010, retrieved April 23, 2010
77. ^ ["Extension:CategorySuggest"](#). MediaWiki. [Archived](#) from the original on September 26, 2014. Retrieved May 30, 2010.
78. ^ ["Category:Flash Video extensions"](#). MediaWiki. [Archived](#) from the original on September 15, 2014. Retrieved May 30, 2010.
79. ^ ["Category:YouTube extensions"](#). MediaWiki. September 16, 2008. [Archived](#) from the original on May 1, 2011. Retrieved May 30, 2010.

80. ^ *"Category:RSS extensions"*. MediaWiki. *Archived* from the original on December 3, 2010. Retrieved May 30, 2010.
81. ^ M Dale; A Stern; M Deckert; W Sack (2009), *System demonstration: Metavid.org: a social website and open archive of congressional video*, Proceedings of the 10th Annual International Conference on Digital Government Research: Social Networks: Making Connections between Citizens, Data and Government, pp. 309–310, ISBN 978-1-60558-535-2
82. ^ *"Wiki spam"*. Meta-Wiki. *Archived* from the original on November 7, 2014. Retrieved May 30, 2010.
83. ^ Goldman, Eric, Wikipedia's Labor Squeeze and its Consequences, vol. 8, *Journal on Telecommunications and High Technology Law*
84. ^ *"Extension:ConfirmEdit"*. MediaWiki. May 5, 2010. *Archived* from the original on October 20, 2014. Retrieved May 30, 2010.
85. ^ *"Extension:SpamBlacklist"*. MediaWiki. March 24, 2010. *Archived* from the original on October 20, 2014. Retrieved May 30, 2010.
86. ^ *"Extension:Nuke"*. MediaWiki. May 19, 2010. *Archived* from the original on October 20, 2014. Retrieved May 30, 2010.
87. ^ *Lucene-search MediaWiki extension Archived* June 2, 2012, at the *Wayback Machine*, mediawiki.org
88. ^ *SphinxSearch MediaWiki extension Archived* October 22, 2014, at the *Wayback Machine*, mediawiki.org
89. ^ Masanori Arita & Kazuhiro Suwa (September 17, 2008), "Search extension transforms Wiki into a relational system: A case for flavonoid metabolite database", *BioData Min*, **1** (1), *BioData Mining*: 7, doi:10.1186/1756-0381-1-7, PMC 2556319, PMID 18822113
90. ^ Finn Arup Nielsen (October 15, 2009). "Lost in localization: a solution with neuroinformatics 2.0?". *NeuroImage*. **48** (1): 11–3. doi:10.1016/J.NEUROIMAGE.2009.05.073. ISSN 1053-8119. PMID 19497377. Wikidata Q21011200.
91. ^ Eric Ras; Jörg Rech; Sebastian Weber (August 1, 2008), *Collaborative Authoring of Learning Elements for Adaptive Learning Spaces* (PDF), Fifth International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, *archived* (PDF) from the original on May 3, 2011, retrieved April 23, 2010
92. ^ Hartung, Michael; et al. "A Platform for Collaborative Management of Semantic Grid Metadata". *Intelligent distributed computing, systems and applications*. p. 123.
93. ^ *"Extension:VisualEditor"*. MediaWiki. *Archived* from the original on February 21, 2021. Retrieved March 15, 2021.
94. ^ *"Category:WYSIWYG extensions"*. MediaWiki. April 10, 2008. *Archived* from the original on May 1, 2011. Retrieved May 30, 2010.
95. ^ *"Manual:Installation requirements"*. MediaWiki. *Archived* from the original on March 8, 2021. Retrieved March 14, 2021.
96. ^ *"Manual:Page table"*. MediaWiki. May 15, 2010. *Archived* from the original on November 25, 2014. Retrieved May 30, 2010.
97. ^ *"Manual:Revision table"*. MediaWiki. *Archived* from the original on November 24, 2014. Retrieved May 30, 2010.

98. ^ Ortega, Felipe; González-Barahona, Jesus M.; Robles, Gregorio (2007), *The Top-Ten Wikipedias: A Quantitative Analysis Using WikiXRay*, [CiteSeerX 10.1.1.107.1424](#)
99. ^ Curino, Carlo A.; Tanca, Letizia; Zaniolo, Carlo (2008), *Information Systems Integration and Evolution: Ontologies at Rescue* (PDF), Workshop on Semantic, [archived](#) (PDF) from the original on December 22, 2009, retrieved April 23, 2010
100. ^ T Dumitras; P Narasimhan (2009), *No downtime for data conversions: Rethinking hot upgrades* (PDF), [archived from the original](#) (PDF) on June 16, 2010, retrieved April 29, 2010
101. ^ **a b** "Wikipedia and MediaWiki". Presentation MediaWiki development (video). April 28, 2006. [Archived](#) from the original on April 14, 2011. Retrieved September 23, 2009.
102. ^ **a b** Bergsma, Mark, *Wikimedia Architecture* (PDF), [archived](#) (PDF) from the original on March 5, 2016, retrieved October 21, 2015
103. ^ "Manual:What is". MediaWiki. [Archived](#) from the original on July 22, 2018. Retrieved May 30, 2010.
104. ^ "Extension:StructuredDiscussions". MediaWiki. [Archived](#) from the original on December 27, 2018. Retrieved December 27, 2018.
105. ^ "Extension:Wikilog". MediaWiki. November 27, 2009. [Archived](#) from the original on September 22, 2014. Retrieved May 30, 2010.
106. ^ "Help:Signatures". MediaWiki. [Archived](#) from the original on November 15, 2008. Retrieved May 30, 2010.
107. ^ N Augar; R Raitman; W Zhou (2004), *Teaching and learning online with wikis, Beyond the comfort zone*, pp. 95–104, [CiteSeerX 10.1.1.133.1456](#)
108. ^ Cubric, Marija (2007), *Analysis of the use of Wiki-based collaborations in enhancing student learning*, University of Hertfordshire, p. 11, [archived from the original](#) on May 15, 2011, retrieved April 24, 2010
109. ^ Albertsen, Johannes & Bouvin, Niels Olof (2008), *User defined structural searches in mediawiki*, *Proceedings of the nineteenth ACM conference on Hypertext and hypermedia*, ISBN 978-1-59593-985-2
110. ^ "Extension:Semantic MediaWiki – WikiApiary". [Archived](#) from the original on October 26, 2018. Retrieved October 12, 2019.
111. ^ T Dumitras; P Narasimhan (2009), *Toward upgrades-as-a-service in distributed systems*, *Proceedings of the 10th ACM/IFIP/USENIX International Conference on Middleware*, pp. 1–2
112. ^ "Security for developers". MediaWiki. [Archived](#) from the original on November 25, 2014. Retrieved May 30, 2010.
113. ^ Perrin, Chad (April 30, 2008), *Five security tips from MediaWiki's lead developer*, Tech Republic[[permanent dead link](#)]
114. ^ "News". MediaWiki. [Archived](#) from the original on October 6, 2014. Retrieved May 30, 2010.
115. ^ "Manual:Security". MediaWiki. March 22, 2010. [Archived](#) from the original on November 10, 2014. Retrieved May 30, 2010.
116. ^ [Books about MediaWiki Archived](#) December 27, 2018, at the [Wayback Machine](#), mediawiki.org
117. ^ *MediaWiki Administrator's Handbook*. Wikibooks. [Archived](#) from the original on October 20, 2014. Retrieved October 20, 2014.

118. ^ *MediaWiki User Guide*, Wikibooks, *archived* from the original on October 20, 2014, retrieved October 20, 2014
119. ^ *"MediaWiki.org Project:Copyrights"*. *Archived* from the original on August 23, 2023. Retrieved August 23, 2023.
120. ^ *"Project:PD help"*. MediaWiki. *Archived* from the original on October 29, 2020. Retrieved May 30, 2010.
121. ^ Rafe Needleman (November 19, 2008), *Wikipedia gears up for flood of video and photo files*, C-Net, *archived* from the original on August 6, 2009, retrieved April 23, 2010
122. ^ *"Events"*. Mediawiki.org. *Archived* from the original on December 27, 2018. Retrieved December 27, 2018.
123. ^ *"Development policy"*. MediaWiki. July 19, 2013. *Archived* from the original on May 10, 2017. Retrieved August 4, 2013.
124. ^ *"Summer of Code"*. MediaWiki. March 26, 2013. *Archived* from the original on May 10, 2017. Retrieved August 4, 2013.
125. ^ *"Wikimedia"*. Open Hub. *Archived* from the original on September 14, 2017. Retrieved November 15, 2012. Approximate counts (not deduplicated) as of November 4, 2012: 139 for core, 155 for extensions supported by WMF, 190 and 42 for extensions only hosted on WMF's Git and SVN repositories respectively.
126. ^ ***a b** "Version lifecycle"*. MediaWiki. September 5, 2018. *Archived* from the original on June 17, 2020. Retrieved October 21, 2018.
127. ^ Bartlett, Manning (November 14, 2001). *"Magnus's new script..."* Wikimedia Lists. Wikimedia Foundation. *Archived* from the original on October 2, 2019. Retrieved October 2, 2019.
128. ^ Manske, Magnus (November 14, 2001). *"Magnus's new script..."* Wikimedia Lists. Wikimedia Foundation. *Archived* from the original on October 2, 2019. Retrieved October 2, 2019.
129. ^ Mayer, Daniel (July 19, 2003). *"Phase IV, Wikibooks.org/.com and WikimediaFoundation.org/.com (was Wikis and uniformity)"*. Wikipedia-L mailing list. *Archived* from the original on July 12, 2017. Retrieved January 18, 2015.
130. ^ *"Differences between Wikipedia, Wikimedia, MediaWiki, and wiki"*. MediaWiki. *Archived* from the original on July 1, 2009. Retrieved May 30, 2010.
131. ^ *"International logo contest"*. Meta-Wiki. *Archived* from the original on May 9, 2020. Retrieved April 9, 2020.
132. ^ *"International logo contest/results"*. Meta-wiki. Wikimedia Foundation. January 10, 2007. *Archived* from the original on November 4, 2015. Retrieved March 14, 2007.
133. ^ *"Historical/Logo history"*. Meta-wiki. Wikimedia Foundation. January 17, 2007. *Archived* from the original on November 4, 2015. Retrieved March 14, 2007.
134. ^ Erik Möller (July 26, 2003). *"File talk:EloquenceSunflowerNew-Small.png – Meta"*. Meta-wiki. Wikimedia Foundation. *Archived* from the original on January 30, 2016. Retrieved February 3, 2013.
135. ^ David Weinberger (2007). *Everything Is Miscellaneous: The Power of the New Digital Disorder*. Times Books. p. 99. ISBN 978-0-8050-8043-8.
136. ^ *"MediaWiki history"*. MediaWiki website. *Archived* from the original on October 27, 2020. Retrieved August 4, 2013.








137. ^ ["MediaWiki ResourceLoader"](#). Mediawiki.org. *Archived* from the original on March 8, 2013. Retrieved July 6, 2013.
138. ^ ["VisualEditor – MediaWiki"](#). MediaWiki. *Archived* from the original on September 27, 2013. Retrieved September 15, 2013.
139. ^ Sarabadani, Amir (March 31, 2021). ["Logo of MediaWiki has changed"](#). Wikimedia Foundation. *Archived* from the original on April 2, 2021. Retrieved April 2, 2021.
140. ^ [MediaWiki testimonials Archived](#) January 11, 2012, at the [Wayback Machine](#), mediawiki.org
141. ^ ["The story of Intelpedia: A model corporate wiki"](#). Socialmedia.biz. *Archived* from the original on September 16, 2013. Retrieved August 16, 2013.
142. ^ A. Maron; M. Maron (2007). ["A stealth transformation: introducing wikis to the UN"](#). Knowledge Management for Development Journal. *Archived from the original* on May 4, 2011. Retrieved October 9, 2010.
143. ^ ["LibrePlanet Homepage"](#). *Archived* from the original on March 18, 2011. Retrieved December 10, 2018.
144. ^ Bryant, Todd (2006), [Social Software in Academia](#) (PDF), Educause Quarterly, archived from [the original](#) (PDF) on December 22, 2009, retrieved April 23, 2010
145. ^ Liang, M.; Chu, S.; Siu, F.; Zhou, A. (December 3–4, 2009), [Comparing User Experiences in Using Twiki & Mediawiki to Facilitate Collaborative Learning](#) (PDF), Proceedings of the 2009 International Conference on Knowledge Management, archived from [the original](#) (PDF) on May 14, 2011
146. ^ Schulz, Judith (2009), [Company-Wiki as a knowledge transfer instrument for reducing the shortage of skilled workers](#) (PDF), Institute of Technology and Education, *archived* (PDF) from the original on March 4, 2016, retrieved April 25, 2010
147. ^ Chu, S.; Kennedy, D.; Mak, M. (December 3–4, 2009), [MediaWiki and Google Docs as online collaboration tools for group project co-construction](#) (PDF), Proceedings of the 2009 International Conference on Knowledge Management, archived from [the original](#) (PDF) on May 14, 2011, retrieved April 23, 2010
148. ^ **a b** Grecco, Claudio Henrique dos Santos; Augusto, Silas Cordeiro; Souza, Jaqueline Tavares Viana de; Carvalho, Paulo Victor Rodrigues; Davila, Adriana Loureiro (July 25, 2021). ["A Method for the evaluation of knowledge management systems"](#). Brazilian Journal of Radiation Sciences. **9** (2B). doi:10.15392/bjrs.v9i2B.1250. ISSN 2319-0612. S2CID 237733021. *Archived* from the original on November 12, 2021. Retrieved November 12, 2021.

External links

[[edit](#)]

MediaWiki at Wikipedia's [sister projects](#)

-  not found or type unknown [Definitions](#) from Wiktionary
-  not found or type unknown [Media](#) from Commons

-  **Quotations** from Wikiquote
-  **Textbooks** from Wikibooks
-  **Resources** from Wikiversity
-  **Data** from Wikidata
-  **Discussions** from Meta-Wiki
-  **Documentation** from MediaWiki
- **MediaWiki homepage**  **Edit this at Wikidata**

- **v**
- **t**
- **e**

Wikipedia

Overview (outline)

- Biases
 - gender
 - geographical
 - ideological
 - racial
- Censorship
- Conflict-of-interest editing
 - political editing incidents
- Criticism
- Deletion of articles
 - deletionism and inclusionism
 - notability
- Disputes
- "Ignore all rules"
- MediaWiki
- Plagiarism
- Predictions of the project's end
- Reliability
 - Fact-checking
 - *Citation needed*
- Vandalism

- Administrators
- AfroCrowd
- Arbitration Committee
- Art+Feminism
- Bots
 - Lsjbot
- Edit count
- List of Wikipedias
- *The Signpost*
- Wikimedian of the Year
- Wikipedian in residence
- WikiProject
- Women in Red

Events

- Edit-a-thon
- WikiConference India
- Wiki Indaba
- WikiConference North America
- Wikimania

Wiki Loves

Community (Wikipedians)

- Earth
- Folklore
- Monuments
- Pride
- Science

People (list)

- Esra'a Al Shafei
- Lee Daniel Crocker
- Florence Devouard
- Sue Gardner
- David Gerard
- James Heilman
- Maryana Iskander
- Dariusz Jemielniak
- Rebecca MacKinnon
- Katherine Maher
- Magnus Manske
- Erik Möller
- Jason Moore
- Raju Narisetti
- Steven Pruitt
- Annie Rauwerda
- Larry Sanger
- María Sefidari
- Lisa Seitz-Gruwell
- Rosie Stephenson-Goodknight

- Bomis
 - Nupedia
- First edit
- Logo
- Internet Watch Foundation
- Scientology
- Hillsborough disaster Wikipedia posts
- VisualEditor
- #1Lib1Ref
- Wikimedia Foundation actions
 - on the Chinese Wikipedia
 - 2021
 - against MENA Wikimedians
 - 2022

History

Controversies

- Alan MacMasters hoax
- Carlos Bandeirense Mirandópolis hoax
- Edit wars
- Essay controversy
- Henryk Batuta hoax
- Jar'Edo Wens hoax
- Seigenthaler biography incident
- *Star Trek Into Darkness* debate
- United States congressional staff edits
- Weintraub controversy
- Zhemao hoaxes

Coverage

- American politics
 - Donald Trump
- COVID-19 pandemic
- Death
- Israeli–Palestinian conflict
- Russian invasion of Ukraine

Honors

- 274301 Wikipedia
- *Viola angustifolia*
- *Wikipedia Monument*

References and analysis

- Academic studies
- Bibliography
- Cultural
- Films
- Listen to Wikipedia
- Wikipediocracy

Mobile

- Apps
- QRpedia
- Wapedia
- Wikipedia Zero
- WikiReader
- Wikiwand

Content use

- DBpedia
- Depths of Wikipedia
- Google and Wikipedia
- Health information
- Kiwix
- Science information
- Wikipedia-based education

Related

- *The Iraq War: A Historiography of Wikipedia Changelogs*
- LGBTQ and Wikipedia
- *Magna Carta (An Embroidery)*
- People imprisoned for editing Wikipedia
- Print Wikipedia
- Wiki rabbit hole
- Wikimedia Foundation
- Wikimedia movement
- Wikipedia for World Heritage
- Wikipedia in India
- Wikiracing
- List of online encyclopedias
- List of wikis

-  **List**
-  **Category**

- **V**

- **t**
- **e**

Wiki software

.NET

- FlexWiki
- SharePoint

Java

- Confluence
- Jive
- Traction TeamPage
- XWiki

JavaScript

- TiddlyWiki
- Wiki.js

Perl

- Foswiki
- ikiwiki
- TWiki
- UseModWiki
- WikiBase

PHP

- BlueSpice
- BookStack
- DokuWiki
- MediaWiki
- PhpWiki
- Phriction
- PmWiki
- PukiWiki
- Tiki

Python



- Allura
- Bloodhound
- Kallithea
- Kuma
- MoinMoin
- Trac
- Zim

Ruby

- Gollum
- Redmine

Other languages

- CLiki
 - Common Lisp
- Federated Wiki
 - CoffeeScript
 - Haskell

-  Comparison
-  List
-  Category

- v
- t
- e

Wikimedia Foundation

Projects ○ Wikipedia community (Wikipedians)

Current

- Maryana Iskander
- Lisa Seitz-Gruwell
- Dariusz Jemielniak
- Rebecca MacKinnon
- Raju Narisetti
- Rosie Stephenson-Goodknight
- Esra'a Al Shafei
- Jimmy Wales

People

- Hampton Lintorn-Catlin
- Danese Cooper
- Bishakha Datta
- Florence Devouard
- Oscar van Dillen
- Sue Gardner
- Arnon Geshuri
- Mike Godwin
- Aaron Halfaker
- James Heilman
- Guy Kawasaki
- Patricio Lorente
- Katherine Maher
- Erik Möller
- Larry Sanger
- María Sefidari
- Lila Tretikov
- Luis Villa

- Projects**
 - Wikipedia
 - history
 - List of Wikipedias
 - Censorship of Wikipedia
 - Wiktionary
 - Wikimedia Commons
 - Wikidata
 - Wikiquote
 - Wikibooks
 - Wikisource
 - Wikispecies
 - Wikinews
 - Wikiversity
 - Wikivoyage
 - Wikifunctions
 - Abstract Wikipedia
- Other**
 - Wikimedia movement
 - List of Wikimedia chapters
 - Bangladesh
 - Deutschland
 - Israel
 - Polska
 - UK
 - Ukraine
 - Wikimania
 - Wiki Indaba
 - WikiConference India
 - WikiConference North America
 - MediaWiki
 - Litigation
 - Monkey selfie copyright dispute
 - *Wikimedia Foundation v. NSA*
 - Knowledge Engine
- Related**
 - *The Signpost*
 - *Wikipedia Monument*
 - Wikimedian of the Year
 - Tides Foundation
 - Artificial intelligence in Wikimedia projects
 - Google and Wikipedia
 - Wikipedia for World Heritage

Authority control databases Image not found or type unknown [Edit this at Wikidata](#)

International	<ul style="list-style-type: none">VIAFFAST
National	<ul style="list-style-type: none">GermanyUnited StatesFranceBnF dataIsrael
Other	<ul style="list-style-type: none">IdRef

Check our other pages :

- [Affordable SEO Parramatta](#)
- [Website designers Parramatta](#)
- [SEO Parramatta](#)
- [eCommerce web design Parramatta](#)
- [Best SEO agency Parramatta](#)
- [SEO expert Parramatta](#)
- [Web development Parramatta](#)

Responsive web design Parramatta

SEO Parramatta

Phone : 1300 684 339

City : Sydney

State : NSW

Zip : 2000

[Google Business Profile](#)

[Google Business Website](#)

Company Website : <https://sydney.website/seo-sydney/local-seo/seo-parramatta/>

USEFUL LINKS

[SEO Website](#)

[SEO Services Sydney](#)

[Local SEO Sydney](#)

[SEO Ranking](#)

[SEO optimisation](#)

LATEST BLOGPOSTS

[SEO community](#)

[SEO Buzz](#)

[WordPress SEO](#)

[SEO Audit](#)

[Sitemap](#)

[Privacy Policy](#)

[About Us](#)

[SEO Castle Hill](#) | [SEO Fairfield](#) | [SEO Hornsby](#) | [SEO Liverpool](#) | [SEO North Sydney](#) | [SEO Norwest](#) | [SEO Parramatta](#) | [SEO Penrith](#) | [SEO Strathfield](#) | [SEO Wetherill Park](#)

Follow us