

Security-isolated JS widgets

Safety first, flexibility too
(long version)

Wikimedia Hackathon - Jerusalem
2016-04-03

```
▼ <div id="plugin  
placeholder"> e
```

```
▼ <iframe style  
none;" src="h  
/brion/mw-js-
```

**What is the state of user-supplied
JavaScript code in MediaWiki
today?**

JavaScript from gadgets and user scripts is run directly in the web page's security context.

Good: has access to everything you do!

- Can **read and write data** with your credentials easily
- Can **hook into more or less anything** in the user interface
- Lots of **helper libraries** available to you already

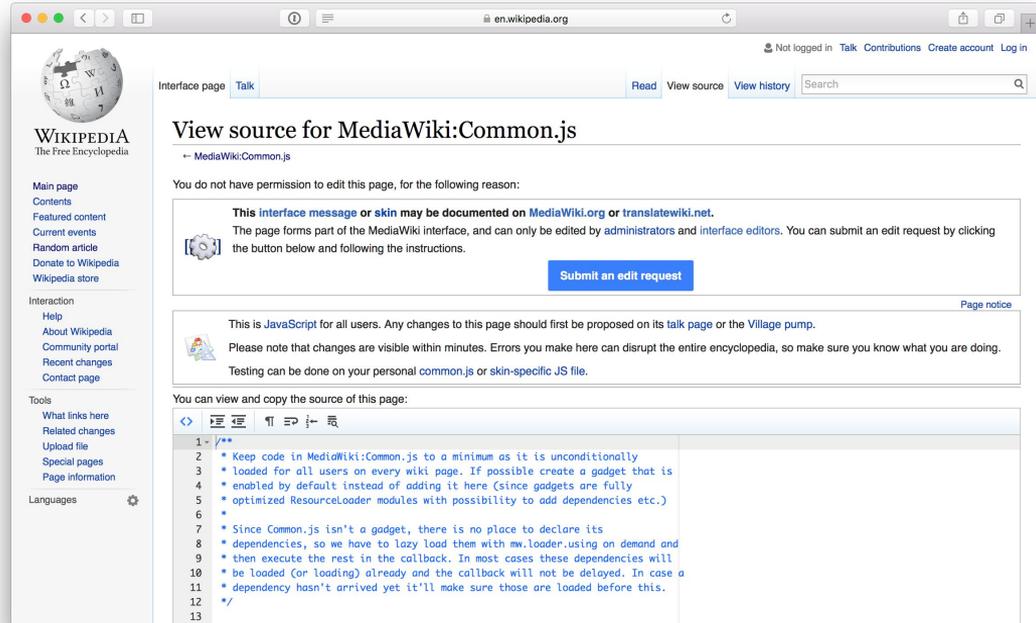
Bad: has access to everything you do!

- Can **read private data** on-wiki or **modify things** you didn't expect
- Can pull in external data that **risks your privacy** by leaking IP
- Can pull in external code that further risks security through **cross-site scripting**
- Hooking directly into user interface means code is exposed to **unexpected dependency breakage**

What do we do about that?

Site JS and Gadgets live in restricted-access “MediaWiki:” namespace

Only a few thousand people can introduce code, instead of billions



The screenshot shows a web browser window displaying the source code for MediaWiki:Common.js on the English Wikipedia. The browser's address bar shows "en.wikipedia.org". The page title is "View source for MediaWiki:Common.js". Below the title, there is a navigation bar with "Interface page" and "Talk" tabs, and buttons for "Read", "View source", "View history", and a search box. The main content area features a prominent message: "You do not have permission to edit this page, for the following reason: This interface message or skin may be documented on MediaWiki.org or translatewiki.net." Below this message is a blue button labeled "Submit an edit request". Further down, there is a notice about JavaScript for all users, stating that changes should be proposed on the talk page and that errors can be disruptive. At the bottom, there is a section for viewing and copying the source code, which shows a list of 13 lines of JavaScript code. The code includes comments about keeping code in MediaWiki:Common.js to a minimum, loading it for all users, and using mw.loader for lazy loading dependencies.

en.wikipedia.org

Not logged in [Talk](#) [Contributions](#) [Create account](#) [Log in](#)

Interface page [Talk](#) [Read](#) [View source](#) [View history](#)

View source for MediaWiki:Common.js

← MediaWiki:Common.js

You do not have permission to edit this page, for the following reason:

This interface message or skin may be documented on [MediaWiki.org](#) or [translatewiki.net](#).

The page forms part of the MediaWiki interface, and can only be edited by administrators and interface editors. You can submit an edit request by clicking the button below and following the instructions.

[Submit an edit request](#)

Page notice

This is **JavaScript** for all users. Any changes to this page should first be proposed on its [talk page](#) or the [Village pump](#).

Please note that changes are visible within minutes. Errors you make here can disrupt the entire encyclopedia, so make sure you know what you are doing.

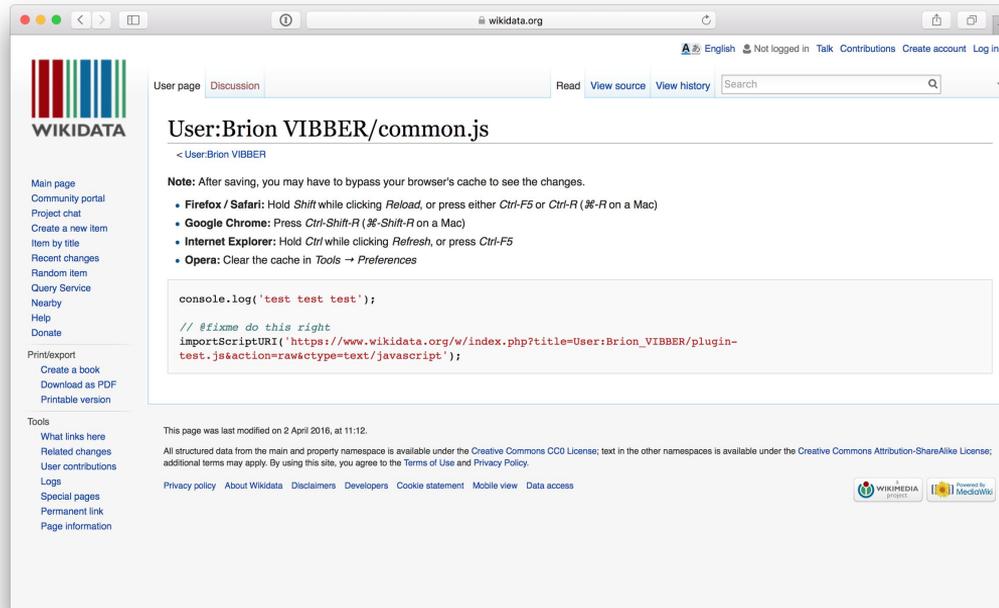
Testing can be done on your personal [common.js](#) or [skin-specific JS file](#).

You can view and copy the source of this page:

```
1 - /*
2   * Keep code in MediaWiki:Common.js to a minimum as it is unconditionally
3   * loaded for all users on every wiki page. If possible create a gadget that is
4   * enabled by default instead of adding it here (since gadgets are fully
5   * optimized ResourceLoader modules with possibility to add dependencies etc.)
6   *
7   * Since Common.js isn't a gadget, there is no place to declare its
8   * dependencies, so we have to lazy load them with mw.loader.using on demand and
9   * then execute the rest in the callback. In most cases these dependencies will
10  * be loaded (or loading) already and the callback will not be delayed. In case a
11  * dependency hasn't arrived yet it'll make sure those are loaded before this.
12  */
13
```

User JS only runs for yourself, and using other people's code requires going to unusual places and cut-n-pasting obscure commands

Ease-of-use barrier prevents most people from running insecure code! ;)



Security isolation through <iframe> separation

WIKIDATA

MediaWiki (Q83)

wiki software
No aliases defined
+ In more languages

Statements

operating system	cross-platform	edit
	+ 1 reference	+ add
bug tracking system	https://phabricator.wikimedia.org/	edit
	+ 0 references	+ add reference
		+ add
source code repository	https://github.com/wikimedia/mediawiki/	edit
	+ 0 references	+ add reference
	https://phabricator.wikimedia.org/diffusion/MW/	edit
	+ 0 references	+ add reference
		+ add

Example iframe-based widget plugin

- Runs on Wikidata via user-script shim
- The payload has its own HTML and JS libraries
- All that is isolated in an iframe, currently loaded from another site
 - Frame code has no way to access DOM or JS objects from the wiki!

English | [Bron VIBBER](#) | [Task](#) | [Preferences](#) | [Beta](#) | [Help](#) | [History](#) | [Contributors](#) | [Log out](#)

MediaWiki (Q83) iframe: 460 x 270

Wikidata

Main page
Community portal
Project chat
Create a new item
Item by title
Recent changes
Random item
Query Service
Nearby
Help
Donate

Print/export
Create a book
Download as PDF
Printable version

Tools
What links here
Related changes
Special pages
Permanent link
Page information
Concept URI
Cite this page

Statements

operating system cross-platform edit
+ 1 reference add

bug tracking system https://phabricator.wikimedia.org/ edit
+ 0 references add reference add

source code repository https://github.com/wikimedia/mediawiki edit
+ 0 references add reference

https://phabricator.wikimedia.org/id/fusion/MW/ edit
+ 0 references add reference add

Ins... | Co... | Deb... | Style ... | Perfor... | Ne... | Settings | Search

faceholder | **iframe** | html | body | div#mynetwork | div.vis-network | canvas

```

<h1 id="firstHeading" class="firstHeading" lang="en"></h1>
<div id="bodyContent" class="mw-body-content">
  <div id="siteSub">From Wikidata</div>
  <div id="contentSub"></div>
  <div id="jump-to-nav" class="mw-jump"></div>
  <div id="mw-content-text" class="mw-content-ltr" dir="ltr" lang="en">
    <div id="wb-item-083" class="wikibase-entityview wb-item" dir="ltr" lang="en">
      <div id="plugin-example-placeholder">
        <iframe style="border: medium none;" src="https://rawgit.com/brion/mw-js-plugin/master/example-data-plugin/index.html" frameborder="0" height="540" width="920">
          #document
          <!DOCTYPE html>
          <html>
            <head>
              <script type="text/javascript" src="index.js"></script>
            <div id="mynetwork">
              <div class="vis-network" tabindex="900" style="position: relative; overflow: hidden; -moz-user-select: none; width: 100%; height: 100%;">
                <canvas height="536" width="906" style="position: relative; -moz-user-select: none; width: 100%; height: 100%;">
                  <div class="vis-network-tooltip" style="left: 539px; top: 73px; visibility: hidden;">species of plant</div>
                </div>
              </div>
            </body>
          </html>
        </div>
      </div>
    </div>
  </div>

```

Rules | Computed | Fonts | Box Model | Animations

Filter Styles

```

element {
  border: medium none;
}

```

Inherited from div#mw-content-text

```

.mw-content-ltr {
  direction: ltr;
}

```

Inherited from load.php:1 @screen

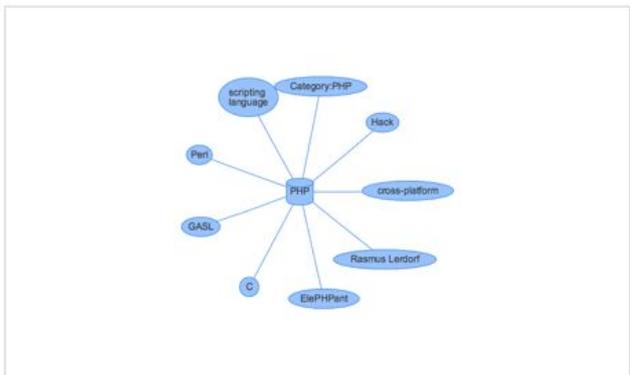
```

.mw-content-ltr {
  direction: ltr;
}

```

Inherited from div#bodyContent

PHP (Q59)



programming language edit

Hypertext Preprocessor | PHP: Hypertext Preprocessor | Personal Home Page

In more languages

Statements

Stack Exchange tag	<ul style="list-style-type: none"> http://stackoverflow.com/tags/php.g 0 references 	<ul style="list-style-type: none"> edit add reference add
--------------------	---	--

file extension	<ul style="list-style-type: none"> php 0 references 	<ul style="list-style-type: none"> edit add reference
	<ul style="list-style-type: none"> php.html 0 references 	<ul style="list-style-type: none"> edit add reference
	<ul style="list-style-type: none"> php4 0 references 	<ul style="list-style-type: none"> edit add reference
	<ul style="list-style-type: none"> php3 0 references 	<ul style="list-style-type: none"> edit add reference

Interactivity through limited API

Example exposes a very limited API to the widget:

- Host wiki sends the URL and title of the page down
- Iframe code can request navigation to another wiki page

Clicking on a related node in the graph navigates to that node, which draws a new graph.

Next steps: richer APIs

But still limited for your safety!

Would be great to allow widgets to plug in to the user interface via **clearly defined stable APIs**.

These can be provided consistently on both **desktop and mobile** interfaces, on different and **custom skins**, or even in **native mobile apps** through a special web view.

Adding custom tabs, toolbar+dialog plugins to the source and visual editors, in-browser image editors, etc should all be possible within the wiki's UI this way.

Next steps: privacy

Solving the “web bug” problem

HTML or JS in an iframe can still cause network access through loading images, scripts, etc from third-party sites. This would **expose a user's IP address**, risking privacy exposure equivalent to opening an external link.

Thus the iframe isolation alone **isn't safe enough** to allow completely unreviewed code to **autoload** in the content area.

Sane ideas:

opt-in click-to-play for unreviewed content widgets, with ability to review & version-lock to allow autoplay in content.

A nice review queue and tools to help review. Yeah, that'd be cool.

Crazy ideas:

JavaScript recompilation and restricted DOM proxies such as using the [caja framework](#).

Scrapping JavaScript and running lua with custom non-DOM user interface is probably too crazy, and kills ability to reuse open-source JS libraries.

Next steps: merging disparate projects?

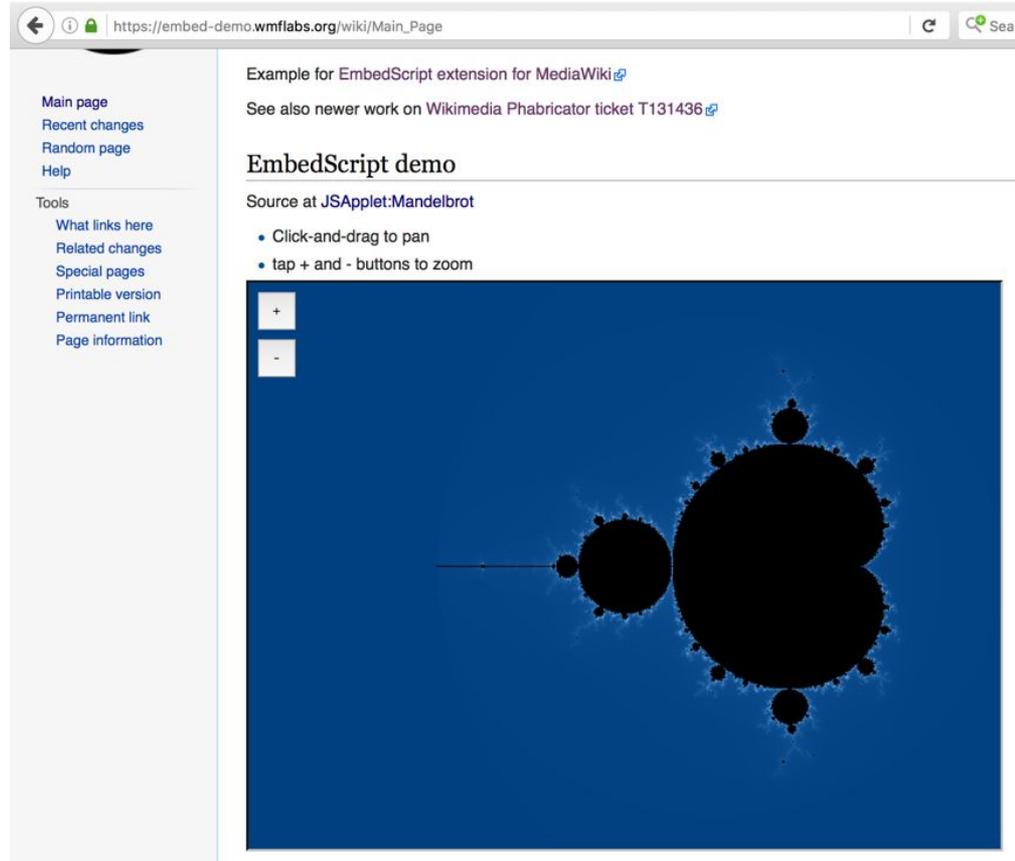
Gadgets, user JS, Widgets, WikiWidgets, oh my!

Modernize old EmbedScript extension's transport

Older experiment that injects code hosted on the wiki into the iframe.

Modern iframe sandbox options can remove need to use an external site to host the frame.

Integrate old loader shim with the newer message posting.
Port the examples!



The screenshot shows a web browser window with the URL https://embed-demo.wmflabs.org/wiki/Main_Page. The page content includes:

- Example for EmbedScript extension for MediaWiki
- See also newer work on Wikimedia Phabricator ticket T131436
- EmbedScript demo**
- Source at JSApplet:Mandelbrot
- Click-and-drag to pan
- tap + and - buttons to zoom

The main content area displays a fractal image (Mandelbrot set) on a dark blue background. In the top-left corner of the image area, there are two white buttons: a '+' button for zooming in and a '-' button for zooming out.

Review popular Gadgets & user scripts

See how much work it would take to create APIs rich enough for them to be workable!

Some will be a poor fit, but many should be doable.

User profile Appearance Editing Recent changes Watchlist Notifications **Gadgets** Beta features

[Definitions](#) · [Descriptions](#)

Below is a list of custom features ("gadgets") you may enable for your account. Most of them require JavaScript to be enabled in your browser. Please note that these tools are not part of the MediaWiki software, and are usually developed and maintained by users on Wikipedia. The numbers of users for each gadget on this wiki are listed at [Gadget usage statistics](#).

 Be advised that **you take full responsibility for any action performed** using these features. For more information see our [policies and guidelines](#).

Browsing

- After rolling back a user's edit, automatically open their contributions page
- Require confirmation before performing rollback on mobile devices ([documentation](#))
- Disable [access keys](#)
- Focus the cursor in the search bar on loading the Main Page
- [GoogleTrans](#): open a translation popup for the selected text or the word under the cursor when pushing the shift button
- [ImageAnnotator](#): view image notes and comments on file description pages
- Redirect image links to Commons for files hosted there
- [Navigation popups](#): article previews and editing functions pop up when hovering over links
- Open external links in a new tab or window
- Open search results in a new tab or window when holding down the Ctrl key
- [Print options](#): control how pages are printed (for example, remove images or backgrounds)
- [revisionjumper](#): quickly navigate between page revisions
- [Twinkle](#): automate common tasks such as reporting vandalism, warning vandals, requesting deletion, welcoming users, and tagging articles ([preferences](#))
- Suppress display of fundraiser banners
- Suppress display of [CentralNotices](#)
- Enable the [Teahouse](#) "Ask a question" feature
- [Reference Tooltips](#): hover over inline citations to see reference information without moving away from the article text
- [FormWizard](#): a [wizard](#) for creating and expanding project pages

A few 'WikiWidgets' are in use on es.wikipedia.org

Can be ported to use iframe isolation for safety; transport parameters from the placeholder div via a limited API.



WIKIPEDIA
La enciclopedia libre

Portada
Portal de la comunidad
Actualidad
Cambios recientes
Páginas nuevas
Página aleatoria
Ayuda
Donaciones
Notificar un error

Imprimir/exportar
Crear un libro
Descargar como PDF
Versión para imprimir

En otros proyectos
Wikimedia Commons

Herramientas
Lo que enlaza aquí
Cambios en enlazadas
Subir archivo
Páginas especiales
Enlace permanente
Información de la página
Elemento de Wikidata
Citar esta página

Brion VIBBER  Discusión Taller Preferencias Beta Lista de seguimiento Contribuciones Salir

Artículo [Discusión](#)

Leer [Editar código](#) [Editar](#) [Ver historial](#) 

Juego de la vida

El **juego de la vida** es un [autómata celular](#) diseñado por el [matemático británico John Horton Conway](#) en 1970.

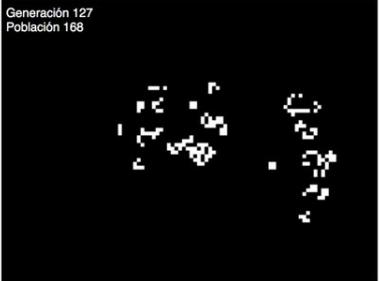
Hizo su primera aparición pública en el número de octubre de 1970 de la revista *Scientific American*, en la columna de [juegos matemáticos](#) de [Martin Gardner](#). Desde un punto de vista teórico, es interesante porque es equivalente a una [máquina universal de Turing](#), es decir, todo lo que se puede computar [algorítmicamente](#) se puede computar en el juego de la vida.

Desde su publicación, ha atraído mucho interés debido a la gran variabilidad de la evolución de los patrones. Se considera que la vida es un buen ejemplo de [emergencia](#) y [autoorganización](#). Es interesante para los [científicos](#), [matemáticos](#), [economistas](#) y otros observar cómo patrones complejos pueden provenir de la implementación de reglas muy sencillas.

La vida tiene una variedad de patrones reconocidos que provienen de determinadas posiciones iniciales. Poco después de la publicación, se descubrieron el [pentaminó R](#), el planeador o caminador (en inglés glider, conjunto de células que se desplazan) y el explosionador (células que parecen formar la onda expansiva de una explosión), lo que atrajo un mayor interés hacia el juego. Contribuyó a su popularidad el hecho de que se publicó justo cuando se estaba lanzando al mercado una nueva generación de [miniordenadores](#) baratos, lo que significaba que se podía jugar durante horas en máquinas que, por otro lado, no se utilizarían por la noche.

Para muchos aficionados, el juego de la vida sólo era un desafío de programación y una manera divertida de usar ciclos de la CPU. Para otros, sin embargo, el juego adquirió más connotaciones filosóficas. Desarrolló un seguimiento casi fanático a lo largo de los [años 1970](#) hasta mediados de los [80](#).

El juego de la vida es en realidad un [juego de cero jugadores](#), lo que quiere decir que su evolución está determinada por el estado inicial y no necesita ninguna entrada de datos posterior. El "[tablero de juego](#)" es una malla formada por cuadrados ("células") que se extiende por el infinito en todas las direcciones. Cada célula tiene 8 células vecinas, que son las que están próximas a ella, incluidas las diagonales. Las células tienen dos estados: están "vivas" o "muertas" (o "encendidas" y "apagadas"). El estado de la malla evoluciona a lo largo de unidades de tiempo discretas (se podría decir que por [turnos](#)). El estado de todas las células se tiene en cuenta para calcular el estado de las mismas al turno siguiente. Todas las células se actualizan simultáneamente.



Generación 127
Población 168



Widgets extension could benefit from iframes



The screenshot shows the MediaWiki page for the 'Widgets' extension. At the top, there is a navigation bar with 'Extension' and 'Discussion' tabs, and a search box. Below the navigation, the page title is 'Extension:Widgets'. A yellow box contains a lightbulb icon and the text: 'Please also consider contributing widgets you created. Thank you for sharing!'. The main content area starts with a paragraph: 'The **Widgets** extension allows the creation of raw HTML pages that can be embedded (similarly to templates) in normal wiki pages. You do this by creating pages in the *Widget* namespace. They avoid the security problems of raw HTML in editable wiki pages because the privilege to edit in the Widget namespace is managed. Many pre-written Widgets are available. This extension is maintained by Yaron Koren.' Below this is a 'Contents' table of contents with links to sections like 'Downloading', 'Installing', 'Usage', and 'Widget page syntax'. On the right side, there is a 'MediaWiki extensions manual' section for 'Widgets' with a green header and a 'Release status: stable' badge. This section contains a table with the following data:

Implementation	Parser function
Description	Allows adding free-type widgets to the wiki by editing pages in Widget namespace
Author(s)	Sergey Chernyshev, Yaron Koren
Latest version	1.2.1 (November 2015)
MediaWiki	1.20+
Database changes	No
License	GNU General Public License 2.0 or later
Download	Download snapshot (Git master) Git [?]: <ul style="list-style-type: none">repo summarybrowse file tree

Content widgets should usually not depend much on the parent page.

Could be ported to use iframe isolation for additional safety; transport parameters from the placeholder div via a limited API.

the end

<https://phabricator.wikimedia.org/T131436>