



# CityJS Wikimedia Hackathon 2022

Writing user scripts & gadgets



# Outline of this workshop

1. Basics of User scripts and Gadgets
2. Introduction to JavaScript and jQuery
3. [Hands-on activity] Modifying a wiki site elements via DOM manipulation using JQuery & Javascript.
4. [Hands-on activity] MediaWiki Configuration and Javascript modules

# What are User scripts?

**User scripts** are programs written in JavaScript for use on wikis by users. It provides dynamic functionality like manipulating the webpage elements, making web request without changing the page content etc.

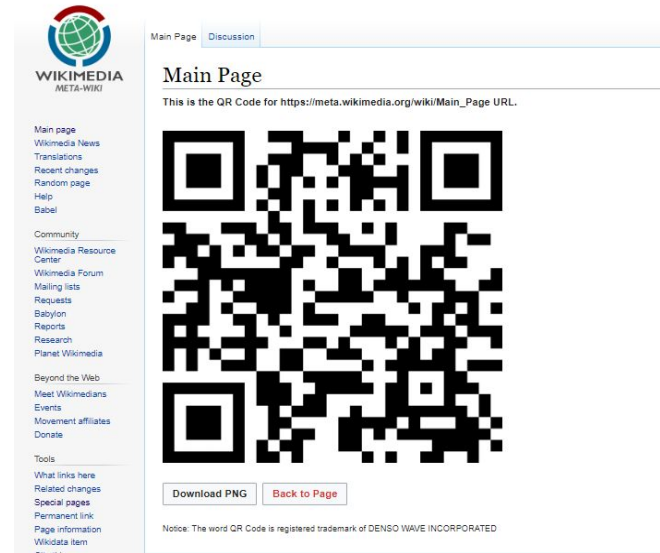
# Difference b/w Gadget and User script

Gadget is nothing but a User script. But it is installed on the wiki for wider uses. Wiki users can enable Gadget through Special:Preferences without any code copy-paste.

Only Interface admin on the wiki can install user scripts as a gadget.

# QR-Code generator

This script is good example of a **User script**. It adds a "Generate QR code" option in the sidebar under toolbar section. This option helps generate QR code of the current page URL on wiki. User can also download QR code in PNG format. Since this script is not useful for all users, it is not installed as a Gadget.



The screenshot shows the Wikimedia Meta-Wiki Main Page. The page title is "Main Page" and the URL is "https://meta.wikimedia.org/wiki/Main\_Page\_URL". A large QR code is displayed in the center of the page. Below the QR code, there are two buttons: "Download PNG" and "Back to Page". A notice at the bottom of the page states: "Notice: The word QR Code is registered trademark of DENSO WAVE INCORPORATED".

WIKIMEDIA  
META-WIKI

Main page  
Wikimedia News  
Translations  
Recent changes  
Random page  
Help  
Babel

Community  
Wikimedia Resource Center  
Wikimedia Forum  
Mailing lists  
Requests  
Babylon  
Reports  
Research  
Planet Wikimedia

Beyond the Web  
Meet Wikimedians  
Events  
Movement affiliates  
Donate

Tools  
What links here  
Related changes  
Special pages  
Permanent link  
Page information  
Wikidata item

Main Page Discussion

Main Page

This is the QR Code for [https://meta.wikimedia.org/wiki/Main\\_Page\\_URL](https://meta.wikimedia.org/wiki/Main_Page_URL).

Download PNG Back to Page

Notice: The word QR Code is registered trademark of DENSO WAVE INCORPORATED

# HotCat

HotCat is also a User script. It augments the category bar with quick links to remove, change, or add categories. But it is very useful for editor to do categorization. So many wikis have adopted it as a Gadget. It can be activated in the "Gadgets" tab of Special:Preferences.



# Navigation Popups

It offers easy access to article previews and several Wikipedia functions in popup windows which appear when you hover the cursor over links. Like HotCat, it is very useful. So many wikis have adopted it as a Gadget. It can be activated in the "Gadgets" tab of Special:Preferences.



The screenshot shows a Wikipedia page with a navigation popup window. The page header includes "Main Page" and "Talk" tabs. The main content area displays a "Welcome to Wikipedia" message. A popup window is overlaid on the page, showing a list of actions for the "Encyclopedia" article, including "actions" and "popups". The popup also displays a small image of a book and a list of actions for the article, including "1 From to". The popup text describes an encyclopedia as a type of reference work or compendium holding a comprehensive summary of information from either all branches of knowledge or a particular branch of knowledge. It also mentions that encyclopedias are divided into articles or entries, which are usually accessed alphabetically by article name. The popup also shows a small image of Hilary Swank and a list of actions for the article, including "1 From to".

# Basics of JavaScript

JavaScript is a programming language of HTML and the Web. It is mostly used for DOM manipulation. It is also used for backend nowadays.

We will only cover its basic functions in this session.



# Basics of JavaScript .....

- **Variables** - JavaScript variables are containers for storing data values.

```
var pi = 3.14;
```

```
var person = "John Doe";
```

```
var answer = 'Yes I am!';
```

# Basics of JavaScript .....

- **Function** - A JavaScript function is a block of code designed to perform a particular task.

```
function myFunction(p1, p2) {  
    return p1 * p2; // Return product of p1 and p2  
}
```

# JavaScript functions

1. **console.log('Message')** - To log the message in developer console. Mostly used for debugging.
2. **alert('Message')** - This method displays an alert box with a specified message and an OK button.
3. **window.open('URL')** - This method opens a new browser window, or a new tab
4. **setInterval( function, millisecond)** - This method calls a function or evaluates an expression at every specified intervals.

# Basic of jQuery

**jQuery** is an open source JavaScript library. It greatly simplifies JavaScript programming. And it is easy to learn. jQuery is widely famous with its philosophy of “**Write less, do more.**” It is represented by **\$**.

Example-

**JavaScript:-** `document.getElementById("element")`

**jQuery:-** `$("#element")`

This syntax format for jQuery is **\$(selector).action()**

# jQuery - Selector

Although there are many types of selector like tag selector, parent of a child or child of parent selector, etc. But broadly we use two selectors most.

1. **ID Selector**:- It uses the id attribute of an HTML tag to find the specific element. We use # (hash) notation for it. Ex. `$("#ele")`
2. **Class selector**:- This selector finds elements with a specific class. We use . (dot) notation for it. Ex. `$(".ele")`

# jQuery - Basic structure

```
$( document ).ready( function() {  
  
    // Code here  
  
});
```

Whenever you will write code, you will have to wrap your code into this basic structure. It allows to load web page first before making any changes to DOM.

# Hurray, time now for a hands-on activity

- First select the element
- Right click on the element and click on “Inspect”.
- Now developer console will open and will point to that element. Copy the element’s id or class name.
- Paste following code

```
$('#elementID').hide()
```

Or

```
$('.className').hide()
```

# Some jQuery action

- **hide()** and **show()** - You can hide and show HTML elements
- **fadeIn()** and **fadeOut()** - used to fade in/out a hidden/visible element
- **slideDown()** and **slideUp()** - slide methods slide elements up and down
- **animate()** - used to create custom animations. Take css object as arg.
- **css()** - Sets or returns style properties for the selected elements.
- **text()** - Sets or returns the text content of selected elements
- **append()** and **prepend()** - Inserts content at the end/beginning of the selected element

We will not cover all the actions in this session. You can visit the official documentation to explore other actions.



# jQuery events

An event represents the precise moment when something happens. Ex-mov-ing a mouse over an element and clicking on an element.

- Function will fire once when someone clicks on an element.

```
$("#elemenID").click(function(){  
  
    console.log('Clicked on the element!');  
  
});
```

Apart from click() event, there are many events like dblclick(), mouseenter(), hover()

# Where to put code?

Developer console is a sandbox. The code will no longer run after refreshing the page. There are three ways to run the code permanently:

Personal **common.js** or **global.js** - You can put the code in local wiki's **Special:MyPage/common.js** or Meta's **Special:MyPage/global.js**. This code will only run for you..

# MediaWiki configurations

It plays very important role in writing the user scripts. To facilitate JavaScript programming, the MediaWiki software exposes a number of configuration values in an object called `mw.config`. Most of these variables have a **wg** prefix. By this configuration, we can get data like Page name, Database name of wiki, and skin etc.

Example:-

```
mw.config.get("wgPageName");
```

```
mw.config.get("wgDBname");
```

# MediaWiki Configurations

Try these configuration variables:-

**wgContentLanguage, wgSiteName, wgServer, wgCanonicalNamespace**

**wgUserEditCount, wgUserName, and wgUserGroups.**

Get full list here:-

<https://www.mediawiki.org/wiki/Manual:Interface/JavaScript>

# MediaWiki module

Apart from MediaWiki configuration, MediaWiki exposes different kind of module for specific task. Most of these modules are not enabled by default. You have to add them as dependency, then only you can use them.

Example:-

For notification purpose, we use **mw.notify**.

For web storage, we use **mediawiki.storage** module.

# Notification module

The **mw.notify** is used to create the bubble notification. From MediaWiki 1.35, it is enabled by default. Let's learn by doing.

```
mw.notify( 'This is a notification.' );
```

```
mw.notify( 'Test', { autoHide: false } );
```

```
mw.notify( "Error", { type : "error" } );
```

# mediawiki.util module

This module contains function such as **parseImageUrl()**, **validateEmail()**, **escapeRegExp()** and so on. But one of the most used function is **addPortletLink()**. It adds a link to a portlet menu on the page. To use a MediaWiki module, you will have to load it as a dependency.

```
$.when( mw.loader.using( 'mediawiki.util' ), $.ready ).then( function () {
```

```
    // Code here
```

```
});
```

It will wait to load the module first.

# mediawiki.util - addPortletLink

```
$.when( mw.loader.using( 'mediawiki.util' ), $.ready ).then( function () {  
    mw.util.addPortletLink( portletId, href, text );  
});
```

1. **portletId** :- It can be p-cactions, p-personal, p-navigation, p-tb.
2. **href**:- Link of the page
3. **text**:- Text to show in Portel.



# Follow-up

There are many Mediawiki modules that you can explore. Some of the modules are very essential for writing advanced UserScripts.

1. **OOUI** (Object-Oriented User Interface) :- This module provides rich and stylish button, form, and dialog etc. Have you ever seen VisualEditor? It is written using this module.
2. **mediawiki.api** :- This module provides api calls. With this module, you can get the data from wikis, edit or move the page, etc. It allows many operations to manipulate the data.

# Task 1

Write a UserScript a show the edit count of the user after their username in top navigation link.

## Hints:-

1. Use jQuery ready wrapper so that code run only after web content is loaded.
2. Inspect the username to get element ID.
3. Use `.append()` to add edit counts in the end of element.
4. Use `wgUserEditCount` to get Users edit count.

## Task 2

Write a UserScript to make page title in red if page is not editable by user.

### Hints:-

1. Use jQuery ready wrapper so that code runs only after web content is loaded.
2. Inspect the page title to get element ID.
3. Use `.css()` to change the color of element.
4. Use `wglsProbablyEditable` to check if page is editable or not, it returns true or false. So use if condition.

# Thank you

You can reach me via/on:

**Email:-** 0freerunning@gmail.com

**Wikipage:-** <https://meta.wikimedia.org/wiki/User:Jayprakash12345>

**Linkedin:-** <https://www.linkedin.com/in/jayprakash12345/>