

Chrome Extensions



History of Mediation Web

- [Shredder](#)
- [Riot](#)
- [The Web Stalker](#)
- [Ambulator](#)
- [List of Others](#) (that are broken)



Extension Art

- [girlsvsgit](#)
- [Misspelling Generator](#)
- [MAICgregator](#)
- [Facebook Demetricator](#)
- [Shaved Bieber](#)
- [Google Alarm](#)
- [IMMA LET YOU FINISH BUT...](#)
- [KANYE VISION](#)
- [Add Art & Addendum](#)
- [Tourettes Machine](#)
- [HappyTube](#)

Extensions vs Alternate Browsers

- Technical barriers lowered
- Extensions are shareable
- Mico-interventions on individual sites

Making Extensions



Extensions

- Extensions are bundles of HTML/CSS/JS and assets
- [Chrome-specific features](#)
 - Storage, notifications, history, bookmarks, etc.
- Architecture of an extension:
 - Actions (UI)
 - Background pages (logic)
 - Content scripts (manipulations)
- [Extensions Overview](#)



Manifest File

- Mandatory file!
- Important information about your extension
- JSON format ([What is JSON](#), [JSON in 10 Minutes](#), [JSON Codecademy](#))
- [Chrome documentation](#)

```
{  
  "manifest_version": 2,  
  "name": "Your extension name",  
  "description": "Your extension description",  
  "version": "1.0",  
  "icons": {  
    "16": "icon.png"  
  }  
}
```

Content Scripts

- Inject files into the specified website or websites
 - CSS file(s)
 - JS file(s)
 - Image, video, audio
- Put these files in the same directory as manifest.json


```
{
  "manifest_version": 2,
  "name": "Your extension name here.",
  "description": "Your extension description here",
  "version": "1.0",
  "icons": {
    "16": "icon.png"
  },

  "content_scripts": [
    {
      "matches": ["<all_urls>"],
      "css": ["css/main.css"],
      "js": ["js/helper-code.js", "js/main.js"],
      "run_at": "document_end"
    }
  ]
}
```

Loading an Extension

1. Open Chrome
2. Type in "chrome://extensions/" in the URL bar
3. Check the box for developer mode
4. Click the button for "Load unpacked extension"

Making Changes

- Anytime you make a change to your extension code:
 1. Go to "chrome://extensions/" and refresh the page
 - Alternative: click "Update extensions now"
 2. Open up a new page (or refresh an old page) to run your new extension code

CSS Extensions Tips

- You can make your extension site-specific using the manifest.json
- You are injecting your CSS into an existing page, so specificity will be a problem. This is a case where using “!important” is very helpful.
- Helpful CSS:
 - * selector – selects all elements
 - “display: none” – removes element from the page
 - “visibility: hidden” – hides element (but leaves space for it)

Local Files in CSS

- Your CSS is being injected into an external site
- Requires permissions in manifest.json
 - [web_accessible_resources](#)
- Special syntax for referring to local assets:
 - "chrome-extension://__MSG_@@extension_id__/" refers to your extension's directory
- More [information](#)

manifest.json

```
{
  "manifest_version": 2,
  "name": "Borked CSS",
  "description": "This extension borks the CSS",
  "version": "1.0",
  "icons": {
    "16": "icon.png"
  },
  "content_scripts": [
    {
      "matches": ["<all_urls>"],
      "css": ["css/css-injection.css"],
      "run_at": "document_end"
    }
  ],
  // We have to declare what local assets we want to be able to use. This
  // will give us access to anything in the images folder.
  "web_accessible_resources": [
    "images/*"
  ]
}
```

css-injection.css

```
body {
  background: url("chrome-extension://__MSG_@@extension_id__/images/clinton.png");
}
```

Debugging Content Scripts

- Open up the Chrome developer tools
 - For CSS: inspect elements
 - For JS: use the console

Publishing

- [Chrome Reference](#)
 - Go to [Chrome Developer Dashboard](#)
 - Pay developer signup fee (\$5)
 - Upload your .zip
 - Add store content
 - Icons
 - Detailed description
 - 1 or more screenshots/YouTube videos
 - Etc.
 - Publish