girls who code

# Chrome Extensions Workshop

October 29, 2020

# PARTS OF AN EXTENSION

01

### Manifest

Provides necessary info about your extension.

02

### Background Script

Interacts with the background (i.e. windows, tabs, etc.)

03

### Content Script

Interacts with the web pages themselves.

04

### Popup

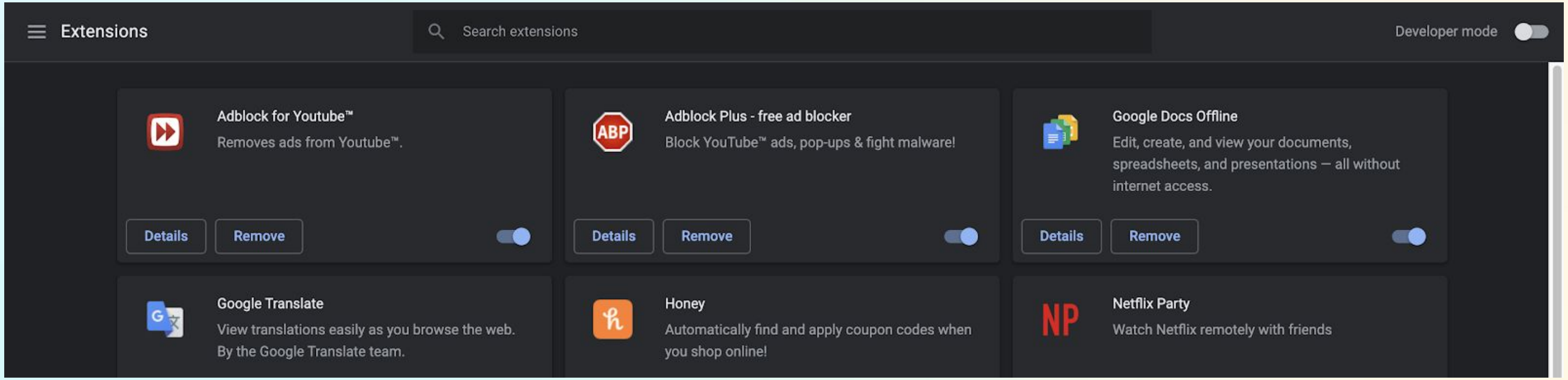
What appears when you click on the extension icon.

## LET’S MAKE SOMETHING

Too often, I have dozens of tabs open in Chrome. They slow down my computer, are hard to manage, and make my browser super cluttered. I want to keep my tabbing in check with a tab counter that notifies me when I have too many open.

## GETTING STARTED

- ❑ Go to *chrome://extensions*
- ❑ Turn Developer Mode on (see top left corner for toggle)
- ❑ You should see three buttons: Load unpacked, Pack extension, and Update. Once you've created your directory, click Load unpacked and select your the folder containing your files.



Adblock for Youtube™
Removes ads from Youtube™.

Adblock Plus - free ad blocker
Block YouTube™ ads, pop-ups & fight malware!

Google Docs Offline
Edit, create, and view your documents, spreadsheets, and presentations — all without internet access.

Google Translate
View translations easily as you browse the web. By the Google Translate team.

Honey
Automatically find and apply coupon codes when you shop online!

Netflix Party
Watch Netflix remotely with friends

## CREATING THE PROJECT

➔ ***Open a text editor of your choice, and create a directory with these four files:***

- manifest.json
- popup.html
- popup.js
- background.js

**What are these files for?**

- The manifest provides important information about your extension (name, permissions, scripts, etc.)
- The popup files affect the appearance of the extension's window in the browser
- The background script manages and monitors browser events like navigating to a new page, removing a bookmark, or closing a tab
- (There is no content script here because we won't need any information about the pages themselves)

## STEPS (follow along)

### Populating the manifest

1. Basics: name, version, description
2. “manifest_version” is 2 (most recent)
3. Permissions allow you to use specific Chrome APIs
   > For this project, we need “tabs” and “notifications”
   > https://developer.chrome.com/extensions/declare_permissions
4. The “background” section specifies background scripts to use and whether the script should be persistent (we will set it to false, as a non-persistent background script is inactive until an event occurs)
5. Use the “browser action” section to specify popup details

## STEPS (follow along)

### Creating the counter

1. Create basic counters in background.js
2. Set up listeners in background.js
   > Does it work? Use console.log() to check!
3. Consider the initial tab count before creation/deletion in background.js
4. Add basic HTML to popup.html
5. Create a port in popup.js
6. Create a function in background.js that connects to the port when necessary and sends (posts) a message to the port in the popup script
7. Set up a listener in popup.js that will detect post requests
8. Update the count and change the HTML of popup.html

## A note about console.log(): there are a lot of consoles to check

### BACKGROUND

> Go to chrome://extensions
> Find your extension
> Background page
> Console

### POPUP

> Click extension icon
> Right click in the popup box
> Inspect
> Console

### CONTENT

> Go to any content page
> Right click anywhere
> Inspect
> Console

## STEPS (follow along)

### Displaying the count in the popup

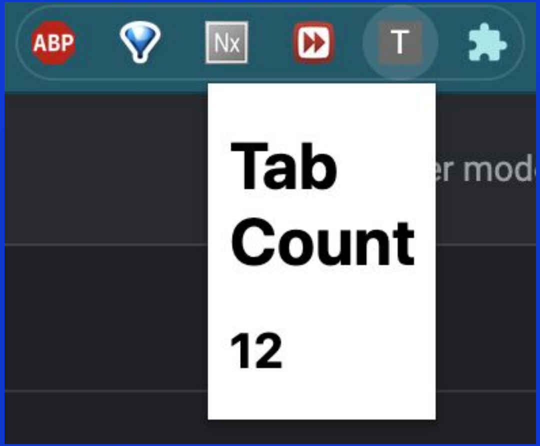
1. Add basic HTML to popup.html
2. Create a port in popup.js
3. Create a function in background.js that connects to the port when necessary and sends (posts) a message to the port in the popup script
4. Set up a listener in popup.js that will detect post requests
5. Update the count and change the inner HTML in popup.html

## STEPS (follow along)

### Setting up notifications

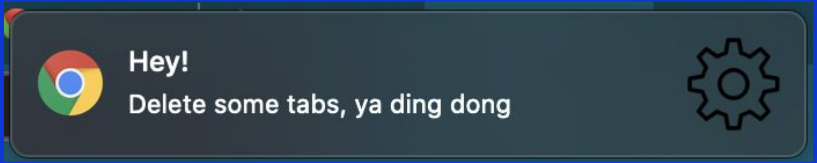
1. To create a notification, you need to specify an options object
   > Specify icon image, notification type, title, message, and priority
2. When a tab is created, compare the count to a specified limit
3. Create notification
   > Most Chrome API methods are asynchronous and return immediately without waiting for the operation to finish. If an extension needs to know the outcome of an asynchronous operation it can pass a callback function into the method. The callback is executed after the method returns
   > Tip: Add “console.log("Last error:", chrome.runtime.lastError)” to callback functions to print errors

## That’s it!



Tab Count

12



Hey!
Delete some tabs, ya ding dong

Check out SamanthaLLee/ChromeExtensionTutorial over on Github for the code & this slideshow :)

girls who code

## Thank you for joining us!

Follow us on IG @rutgersgwc

@GIRLSWHOCODE