

Documentación de Owait

Índice

Introducción.....	2
Manual de uso	
Manual de instalación.....	3
Manual de uso.....	6
Funcionamiento interno.....	10

Introducción

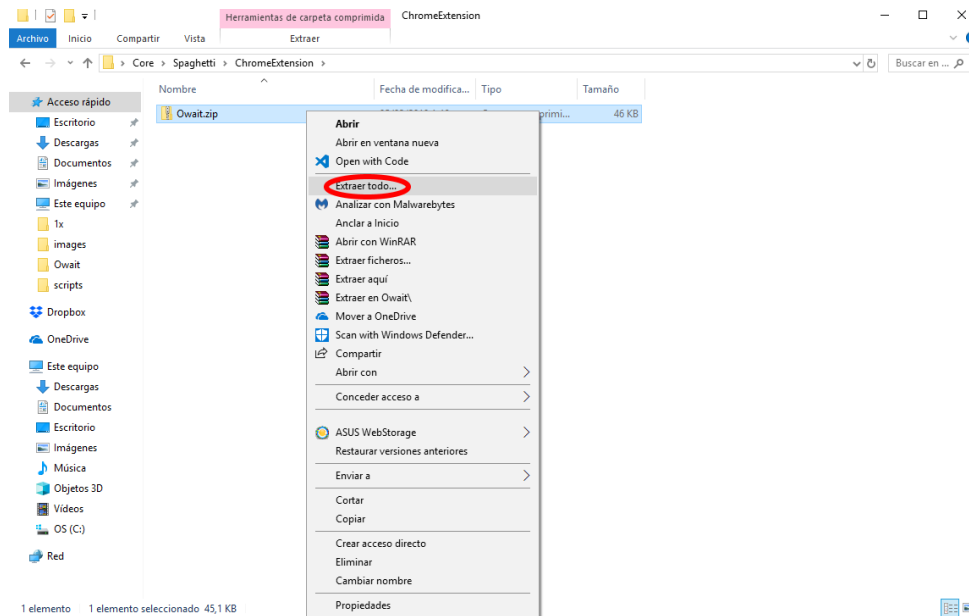
Owait es una extensión de Google Chrome que transforma caracteres, palabras o frases en cualquier web a aquellas indicadas por el usuario. Principalmente transforma todas las instancias de la letra o en OwO, pero esto es ampliamente configurable y se pueden añadir y eliminar todas las transformaciones. El programa también transforma todas las imágenes a OwO escrito en diferentes fuentes (Todas las imágenes usadas son creadas por mi). El fin de la extensión es crear divertidos resultados al aplicar todas las transformaciones.



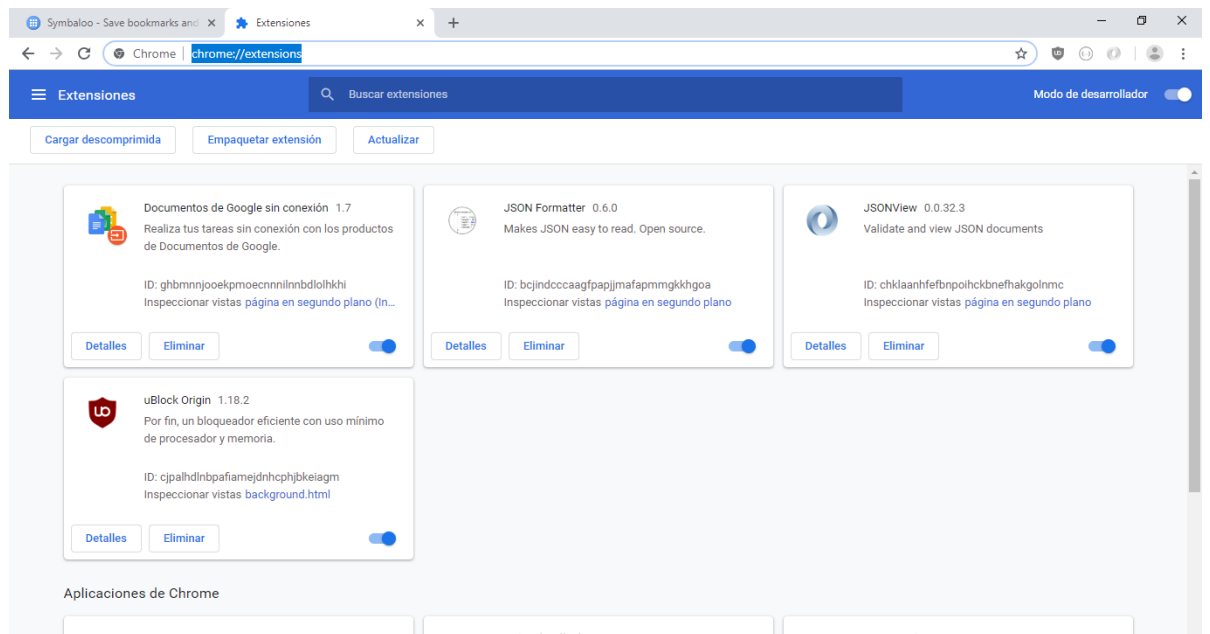
Manual de uso e instalación

Manual de instalación

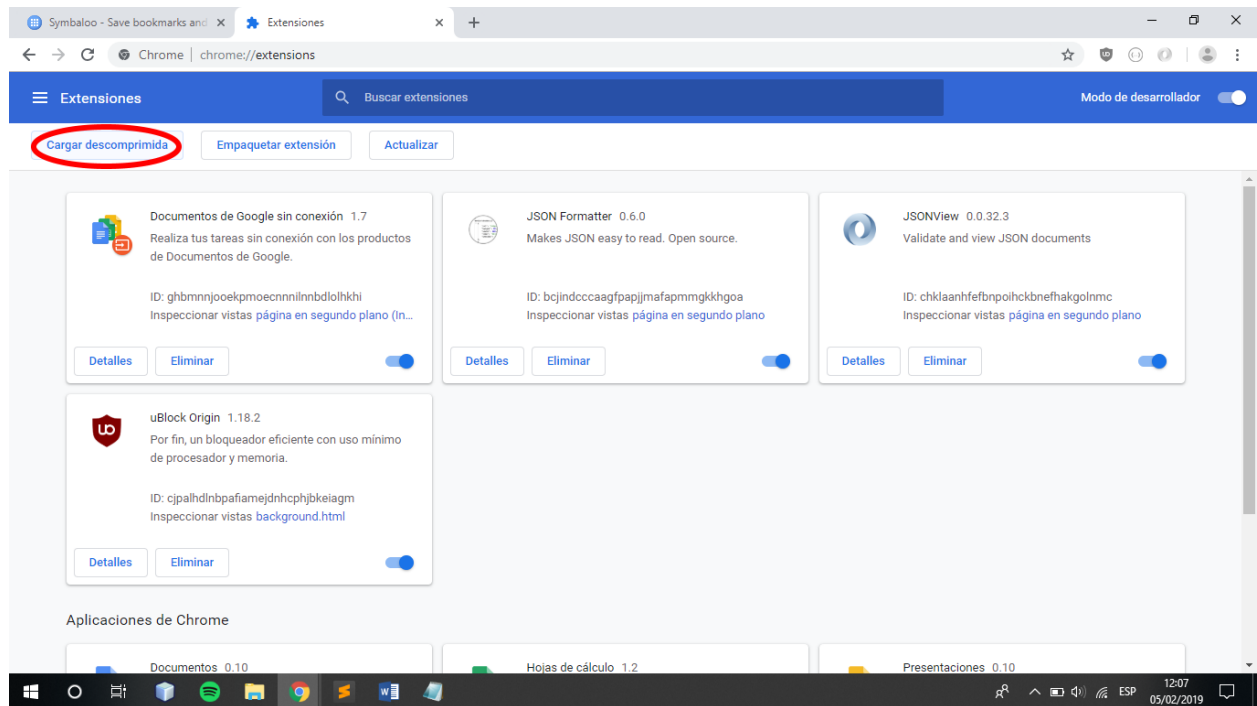
1. Descargar el zip Owait y descomprimirlo (Click derecho, extraer todo)



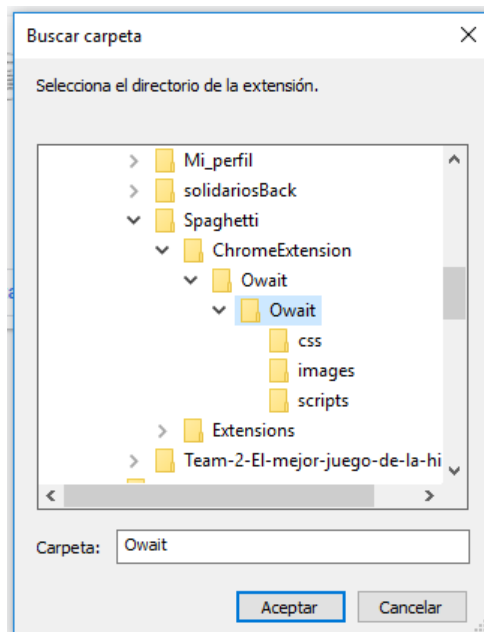
2. Introducir en la barra de búsqueda de Google Chrome `chrome://extensions`



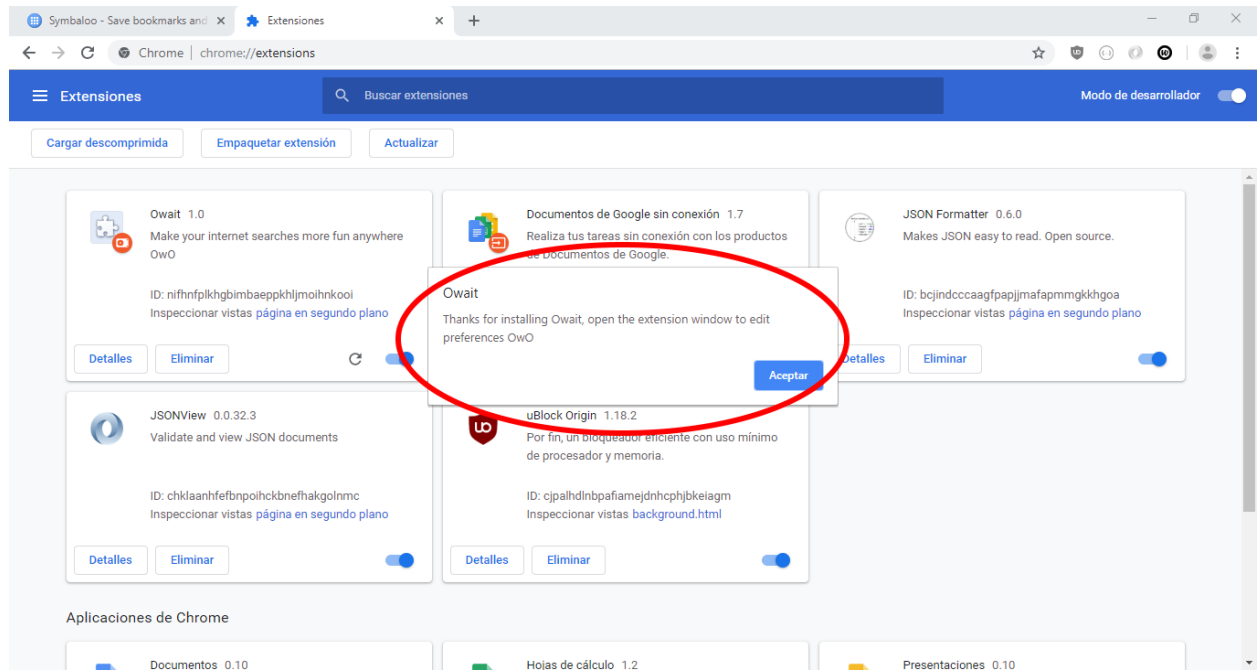
3. Seleccionar en la barra superior "Cargar descomprimida" (O load unpacked según tu idioma de Chrome)



4. Navegar hasta el archivo descomprimido Owait en la ventana emergente y seleccionarla.

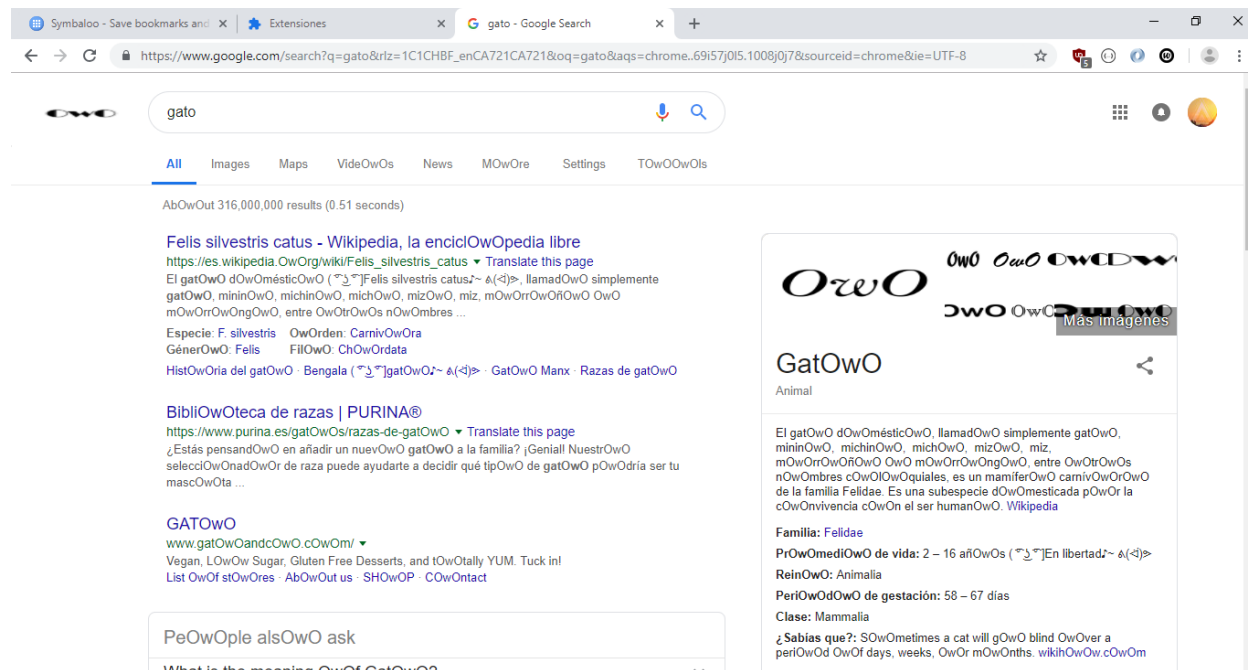


5. Una alerta de chrome te avisará de que la extensión ha sido instalada correctamente

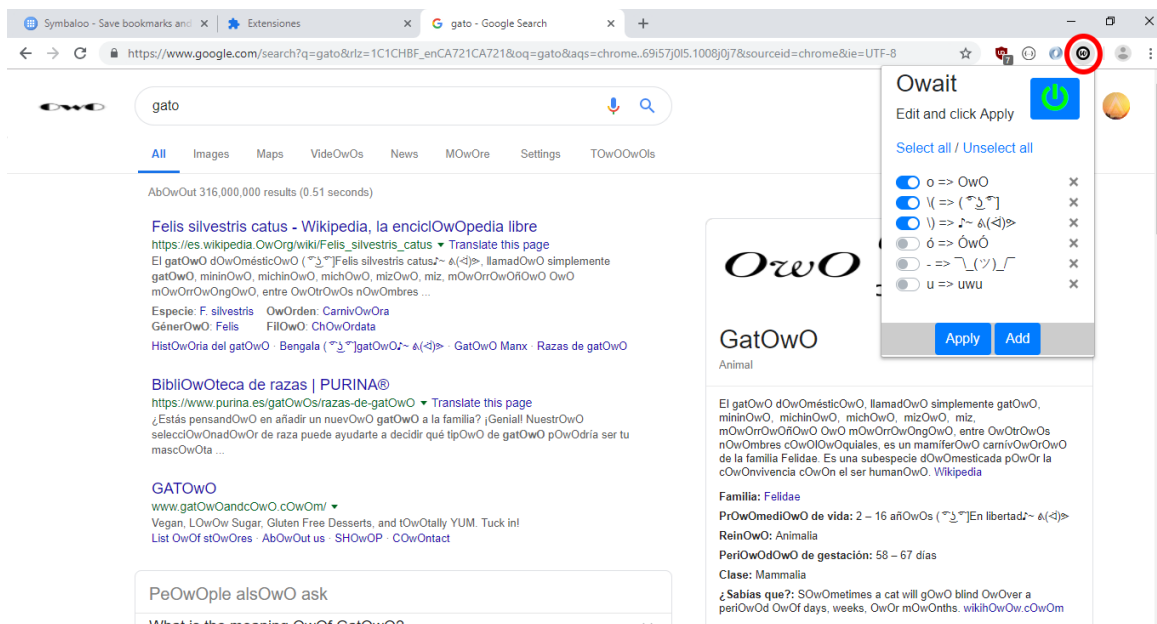


Manual de uso

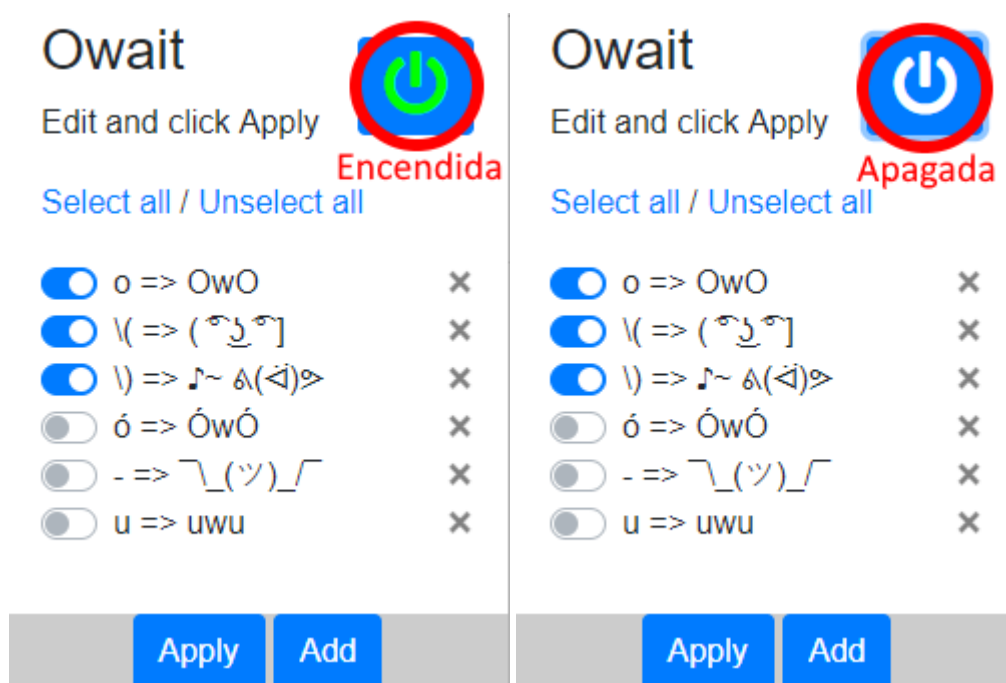
Después de la instalación la extensión ya debería funcionar, para comprobarlo puedes hacer cualquier búsqueda en Chrome y comprobar que todas las imágenes y el texto son transformados por la extensión.



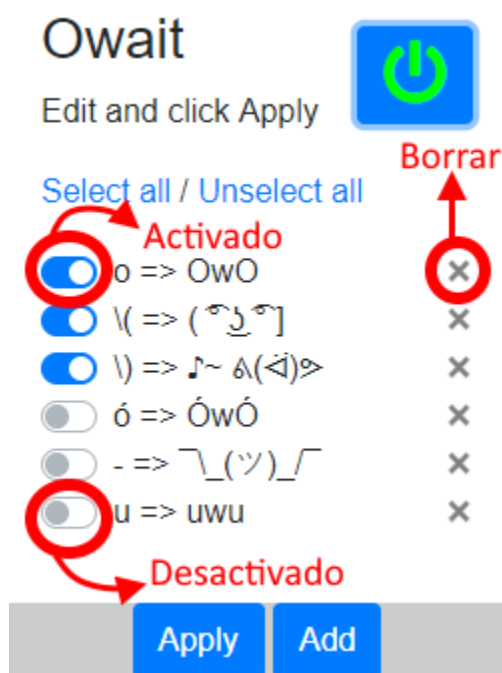
A la derecha de la barra de búsqueda de chrome encontrarás el icono de Owait (Un círculo con una w dentro). Si haces click en él se abrirá un cuadro de diálogo en el que puedes configurar las transformaciones de texto que se realizan.



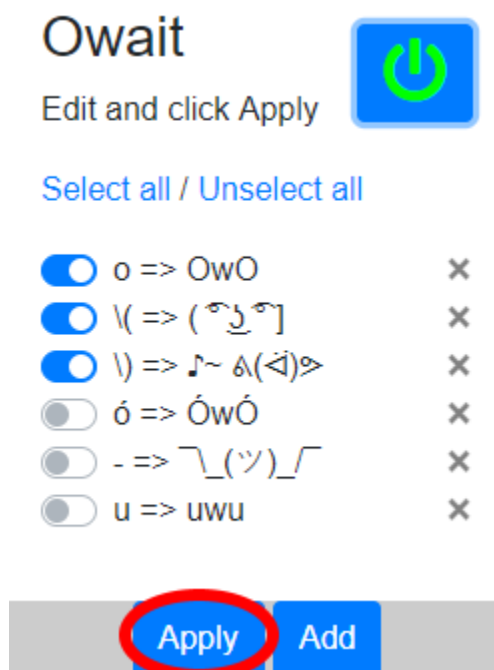
Puedes detener la extensión o hacer que vuelva a actuar con el botón de apagado/encendido.



El diálogo te muestra las transformaciones configuradas con un switch para activarlas o desactivarlas y una cruz a la derecha para eliminar las transformaciones.

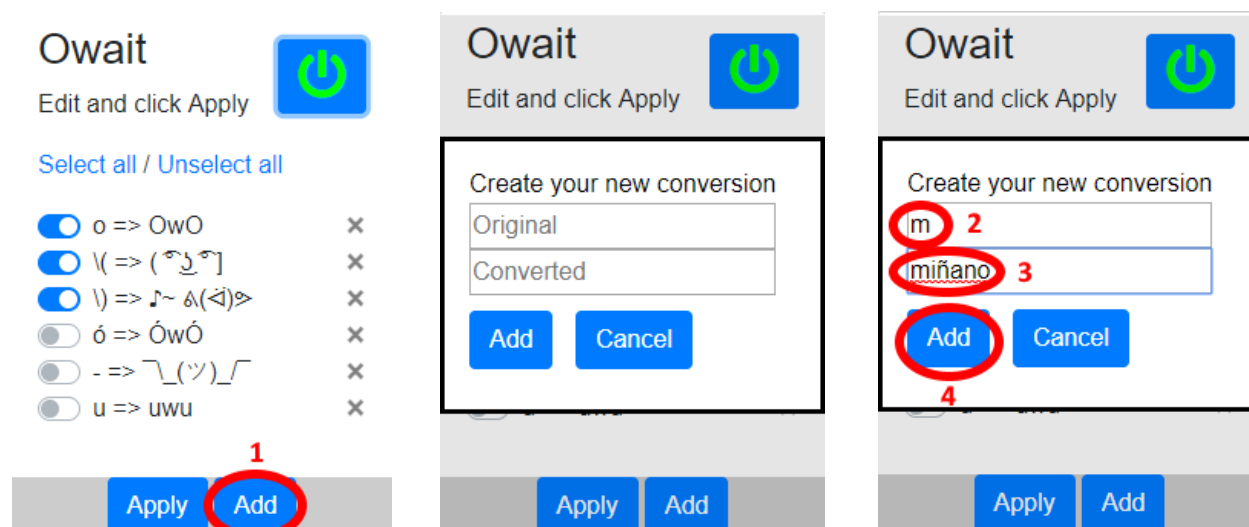


Después de activar o desactivar una transformación deberás hacer click en apply para que se guarden tus cambios.



Añadir nuevas transformaciones se puede hacer en tan solo 4 pasos

1. Hacer click en el botón inferior de 'Add'
2. En el campo 'original' introducir la palabra/letra/frase a transformar. Por ejemplo: m
3. En el campo 'converted' introducir la palabra/letra/frase que la sustituirá. Por ejemplo: Miñano
4. Hacer click en el botón Add



[illegible]

Funcionamiento interno

manifest.json

Como toda extensión de Chrome, Owait tiene un archivo manifest.json en el cual se especifican el nombre y la descripción de la extensión y su versión. También se especifican el icono que utilizará, el script de background, los permisos y demás scripts que serán utilizados. En mi caso llamaré en este apartado a un script llamado content.js para que actúe en todos los dominios cuando DOM haya acabado de procesar la página (document_end).

popup.js

Script asociado a popup.html que crea los switches con las transformaciones y los botones para realizar modificaciones. Añade EventListeners a los botones y switches para hacer las diferentes acciones y guardar los datos.

background.js

Este archivo se usa tradicionalmente para escuchar a un evento. En mi caso, detecta cuando la extensión ha sido instalada. Cuando este evento se produzca. El script cargará los datos predeterminados de transformaciones de data.json. Si localStorage ya contiene datos de la extensión, no se modificarán. En otro caso se guardarán los datos predeterminados. La extensión se activa y muestra una alerta avisando al usuario de que la extensión ha sido instalada y dónde puede modificar las transformaciones.

content.js

Este script es el encargado de realizar las transformaciones deseadas. Se activa cuando el DOM de una nueva página ha sido cargado y itera por todos los nodos de texto de los elementos html indicados realizando las transformaciones adecuadas y añadiendo la clase owo. Todas las imágenes que tienen un src distinto al array de links en el script (Con imágenes creadas por mí y subidas a imgur) son asignadas un src aleatorio de entre el array de links. Como muchas páginas hoy en día son dinámicas, este script funcionaría originalmente sólo con páginas estáticas y por lo tanto sólo afectaría por ejemplo al primer post en Facebook una vez cargado. Debido a esto usé un **MutationObserver**, un objeto de javascript que observa cambios en el DOM de una página. Dados los parámetros 'subtree: true, childList: true, attributes: true', el observador se activará cuando detecte una modificación en el número de nodos o en los atributos de algún elemento recursivamente. Gracias a este elemento, el script afecta dinámicamente a todos los elementos de la página, ya que una vez el observador detecte un cambio, ejecutará una función en la que volverá a comprobar los imágenes y realizar transformaciones de texto en los elementos sin la clase owo.