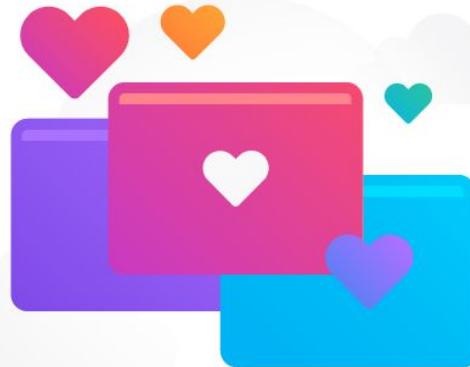




Les extensions navigateurs

Découverte et prise en main





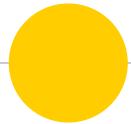
Objectifs et prérequis

“



Sommaire

- ◉ Anatomie d'une extension
- ◉ Les permissions
- ◉ La sécurité
- ◉ Comment développer
- ◉ La compatibilité
- ◉ Publier son extension



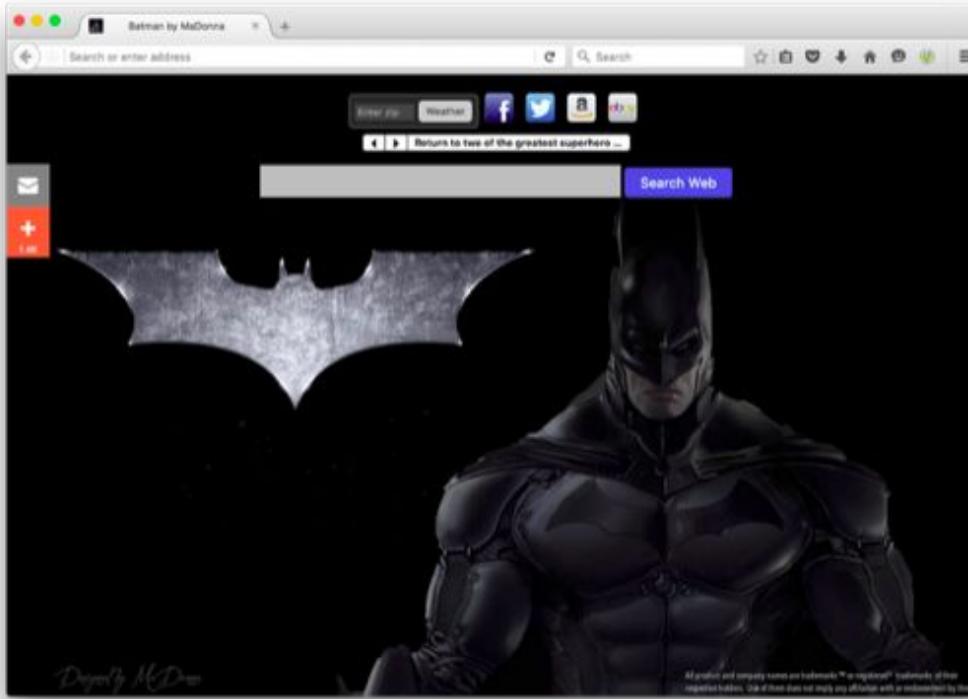
Exemples

et idées ?

A screenshot of a web browser window comparing product prices. The top bar shows the Amazon logo and the text "Compare : Beats by Dr. Dre Mix Headphones \$173". A yellow circle highlights the price "\$173". Below the bar, the text "Beats by Dr. Dre Mixr Headphones" is displayed, followed by a 5-star rating icon and the price "\$179". To the left, a large heading "Product Comparison" is shown with the subtext "Keep an eye out for product comparisons that can save time and money." To the right, there is an image of the headphones and five small thumbnail images representing different product types: headphones, earbuds, a single earbud, a neckband, and a clip-on device.

[Amazon Assistant for Firefox](#), [OneNote Web Clipper](#), and [Grammarly for Firefox](#)

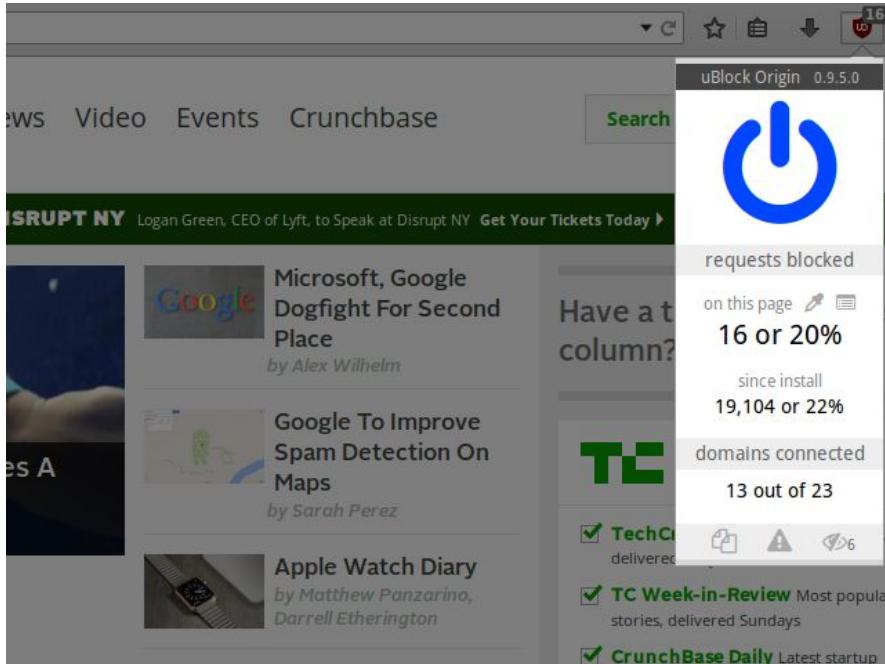
Enrichir l'expérience Utilisateur



Examples: [MyWeb New Tab](#), [Tabliss](#), and [VivaldiFox](#)

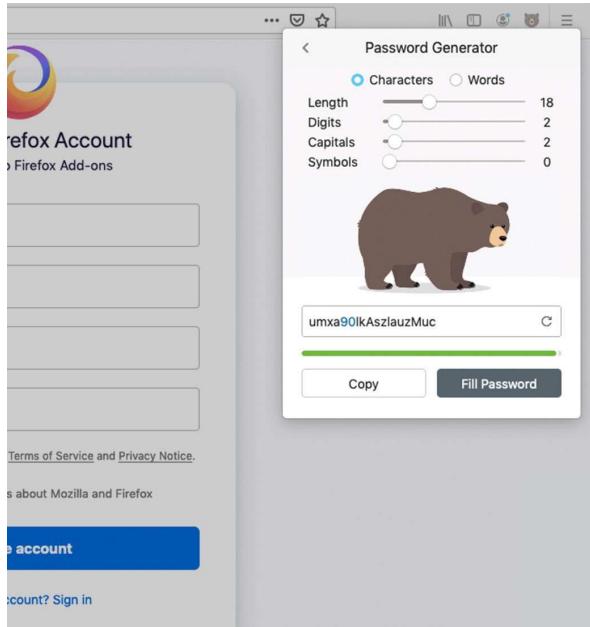
Personnaliser selon ses envies





Examples: [uBlock Origin](#), [Reader](#), and [Toolbox for Google Play Store™](#)

Ajouter/supprimer du contenu



Examples: [RememBear](#), or [random password generator](#)

Ajouter des outils

The screenshot shows the aXe developer tools interface. At the top, there are tabs for Inspector, Console, Debugger, Style Editor, Performance, Memory, Network, Storage, and aXe. The aXe tab is selected. Below the tabs, a navigation bar includes icons for back, forward, search, and refresh, along with a status message "1 of 14".

The main area displays a summary of violations:

Violations found	27
Needs review	1
Elements must have sufficient color contrast	14
Images must have alternate text	8
Form elements must have labels	1
Links must have discernible text	5

A specific violation is highlighted: "Elements must have sufficient color contrast". The details for this violation are as follows:

Issue description: Ensures the contrast between foreground and background colors meets WCAG 2 AA contrast ratio thresholds.

Impact: serious

Element location: `#header > .fl_left > p`

Element source: `<p>Gefälscht CompuTech</p>`

To solve this violation, you need to:

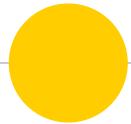
Fix the following:
Element has insufficient color contrast of 4.47 (foreground color: #777777, background color: #ffffff, font size: 9.0pt, font weight: normal)

Related node: `</p>`

Issue tags: category:color wcag2aa wcag143

Examples: [Web Developer](#), [Web React Developer Tools](#), and [aXe Developer Tools](#)

Outils de développement



Anatomie d'une extension

Comment cela fonctionne ?

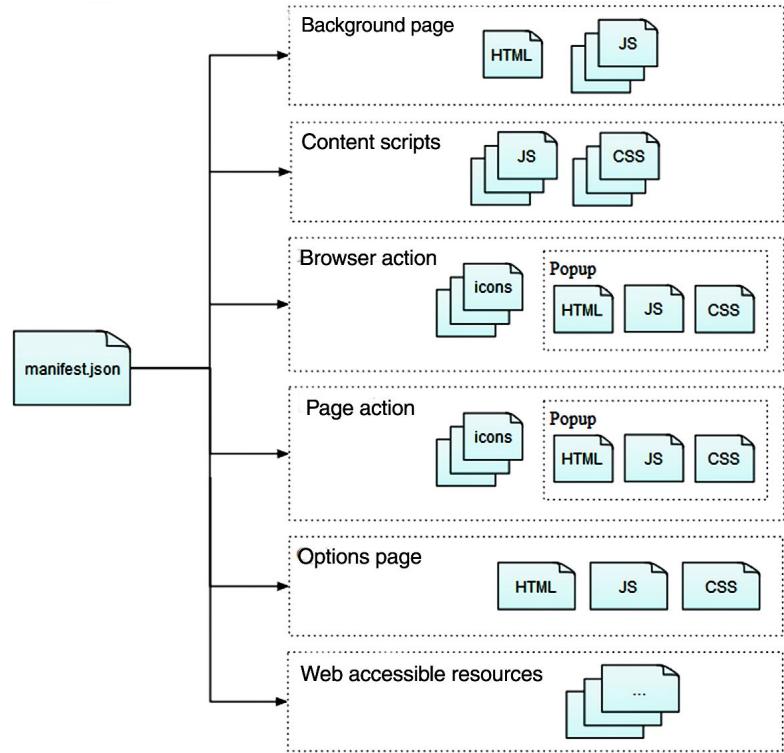


Les différents scripts et fichiers



Manifest.json

- obligatoire
- scripts
- permissions
- ressources
- infos



```
{  
  
  "description": "Adds a solid red border to all webpages matching mozilla.org. See  
  https://developer.mozilla.org/en-US/Add-ons/WebExtensions/Examples#borderify",  
  "manifest_version": 2,  
  "name": "Borderify",  
  "version": "1.0",  
  "homepage_url": "https://github.com/mdn/webextensions-examples/tree/master/borderify",  
  "icons": {  
    "48": "icons/border-48.png"  
  },  
  
  "content_scripts": [  
    {  
      "matches": ["*://*.mozilla.org/*"],  
      "js": ["borderify.js"]  
    }  
  ]  
}
```

exemple de manifest.json



Les 3 types de scripts

- Background Scripts
- Content Scripts
- User Interface scripts



Background Scripts

- peut utiliser les WebExtension API + les API privilégiés (via permissions du manifest.json)
- exécution indépendante de l'onglet
- dans toutes les instances du navigateur
- peut écouter des évènements
- n'a pas accès au contenu de la page



Content Scripts

- inséré depuis le manifest ou depuis le background script
- limité aux API WebExtension (pas privilégiés)
- peut interagir avec le DOM
- peut communiquer avec les background scripts



User Interface scripts

- exécutés en même temps que les pages qui les contiennent (ex: popup, sidebar, panel)
- s'arrêtent à la fermeture de ces dernières



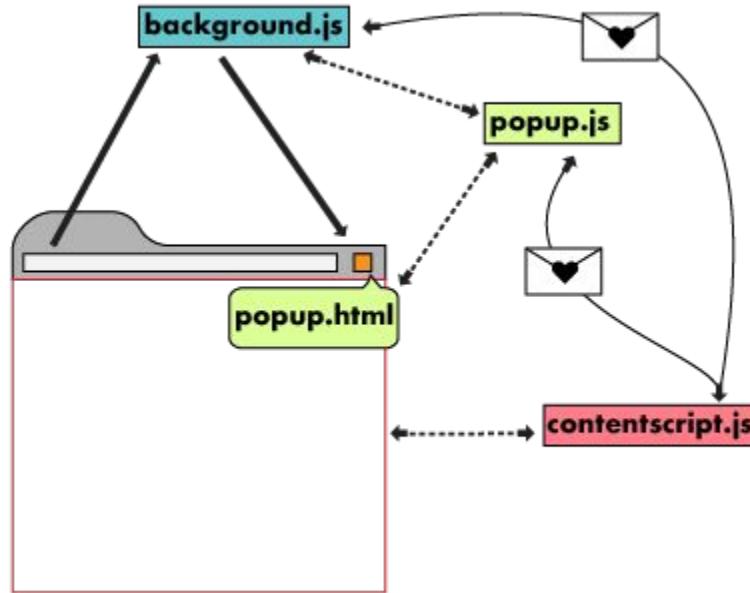
Exemples d'UI

- Browser Action (toolbar)
- Page Action (in URL)
- Sidebar
- DevTools
- Notification
- Address bar suggestion



API messaging

- Permet de communiquer entre les scripts
les “content script” peuvent accéder à des api
“privilégiées” en déléguant au script “background”



Workflow

```
// Sends a message to all privileged scripts (can be used by Content Scripts, but can't be used to
// send to Content Scripts)
browser.runtime.sendMessage(optionalID, messageData);

// Sends a message to a Content Script (from a privileged script)
browser.tabs.sendMessage(tabId, optionalObject);

// Receive messages sent via methods above
browser.runtime.onMessage(function);
```

Message API





Attention aux scopes

- you can not communicate between background scripts because they all share the same scope
- `runtime.sendMessage()` from a Content Script, then all privileged scripts (such as Background Scripts) will receive this message

```
// contentScript.js

// Send a message. In this case, we send an object
browser.runtime.sendMessage({
    type: "notification", // Include a 'type' property so the receiving scripts know what to do
    data: { url: "example.com", numbers: [1, 2, 3, 4, 5] } // Some data you want to send
});
```

```
//background.js

// Listen for messages
browser.runtime.onMessage(handleMessage);

// Handle received messages
function handleMessage( message ) {
    console.log( message.type )           // "notification"
    console.log( message.data.url )       // "example.com"
    console.log( message.data.numbers )    // Array[1, 2, 3, 4, 5]
}
```

Exemples d'échanges entre content and background scripts



Les permissions

Autorisations implicites ou non



Mozilla Foundation (US) | <https://addons.mozilla.org/en-US/>



Add Trello Super Powers?

It requires your permission to:

- Access your data for trello.com
- Access browser tabs
- Display notifications to you

[Cancel](#)

[Add](#)

Demande d'autorisation

```
{  
    "manifest_version": 2,  
    "name": "Super Awesome Extension",  
    "version": "1",  
  
    "description": "Super awesome Add-on for super awesome people.",  
    "icons": {  
        "16": "assets/logos/logo_small.svg",  
        "32": "assets/logos/logo_small.svg",  
        "48": "assets/logos/logo_small.svg",  
        "64": "assets/logos/logo.svg",  
        "128": "assets/logos/logo.svg"  
    },  
  
    "permissions": [  
        "tabs", "storage", "https://awesome.com/*/", "notifications", "downloads"  
    ]  
}
```

exemple de manifest.json avec demande d'autorisation



Autorisations implicites

- alarms
- contextMenus
- contextualIdentities
- cookies
- identity
- idle
- menus
- storage
- theme
- webRequest
- webRequestBlocking

[Voir la liste complète](#)



Permissions explicites

- bookmarks
- browserSettings
- geolocation
- history
- nativeMessaging
- notifications
- sessions
- tabs
- topSites
- webNavigation

[Voir la liste complète](#)



Sécurité

Rester vigilant



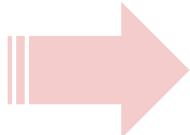
**“Personnalise ton
navigateur avec cette
extension !”**



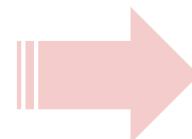
Super extension Ponify !



Install me !



**Informations sensible
(cookie, session, etc)**



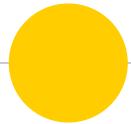
Thanks !

Navigue bien partout !





On ne clique pas sans regarder !



Comment développer ?

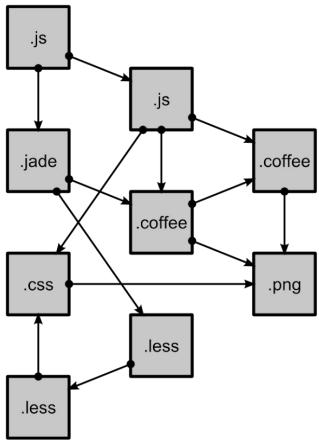
les outils à connaître



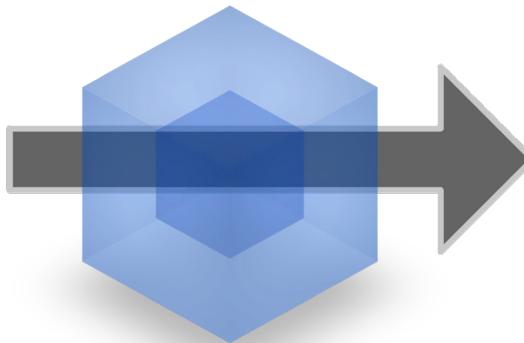
**Vanilla JS is most power
and lightweight
cross-platform framework**

HTML + CSS + JS

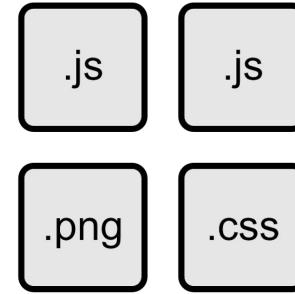




modules
with dependencies



webpack
MODULE BUNDLER



static
assets

Angular, Vue, React with Webpack, Parcel, etc



manifest.json



script.js

Minimum requis





Avec Firefox (recommandé)

- about:debugging > charger extension
- node-based application with hot reload :
[“web-ext run”](#) (recommandé)



Avec Chrome

- chrome://extensions
- “Developer mode” checkbox is checked.
- Click the “Load unpacked extension” button
and then select the Chrome App’s folder.



Compatibilité

Le respect des standards...



Les Web Extensions

- standard : W3 Browser Extensions
- éviter de mettre à jour à chaque version de navigateur



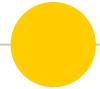
Des implémentations différentes

- API namespace
- API asynchronous event handling
- API function coverage
- Manifest keys
- Extension packaging
- Publishing



Focus : Namespaces

- `browser.*` : the proposed standard for the extensions API, used by Firefox and Edge.
- `chrome.*` used by Chrome and Opera.



Focus : API Async

- promises, the proposed standard for the extensions API, used by Firefox.
- callbacks, used by Chrome, Edge, and Opera.



Focus : API Compatibility List

- ◉ Tous les navigateurs ne supportent pas toutes les API Web Extensions

[Voir la Liste de compatibilité par navigateur](#)



Comment faire ?

“

```
{  
  // ...  
  
  "background": {  
    "scripts": [  
      "browser-polyfill.js",  
      "background.js"  
    ]  
  },  
  
  "content_scripts": [{  
    // ...  
    "js": [  
      "browser-polyfill.js",  
      "content.js"  
    ]  
  }]  
}
```

Charger le script [polyfill](#)





Comment publier ?

Extend the world



Publier sur Firefox

- Package extension
- Connexion avec compte Firefox sur Mozilla Addon
- Chargement du paquet
- Signature du paquet
- Diffusion sur la liste des extensions (si souhaité)



Publier sur Chrome

- Package extension
- Connexion avec compte Google Developer
- Chargement du paquet
- Paiement des charges (5\$)
- Publication



We are done !

Un petit récapitulatif



Demo Time !

“



Pour aller plus loin

- Paramétriser son extension
- Différents sections du fichier manifest.json
 - (update, commands, etc)
- Internationalization



MERCI !

“