

Extensions Chrome & HTML5

gildas.lormeau@softeam.fr



Extensions Chrome

- Introduction
- Intégration dans l'interface du navigateur
- Architecture
- Exemple d'extension pas à pas
- Utilisation de HTML5

Introduction

Historique

- 1999 : bookmarklets
- 2002 : extensions firefox
- 2004 : greasemonkey (userscripts)
- 2009 : projet mozilla jetpack
- 2010 : extensions chrome
- 2011 : extensions opéra et safari

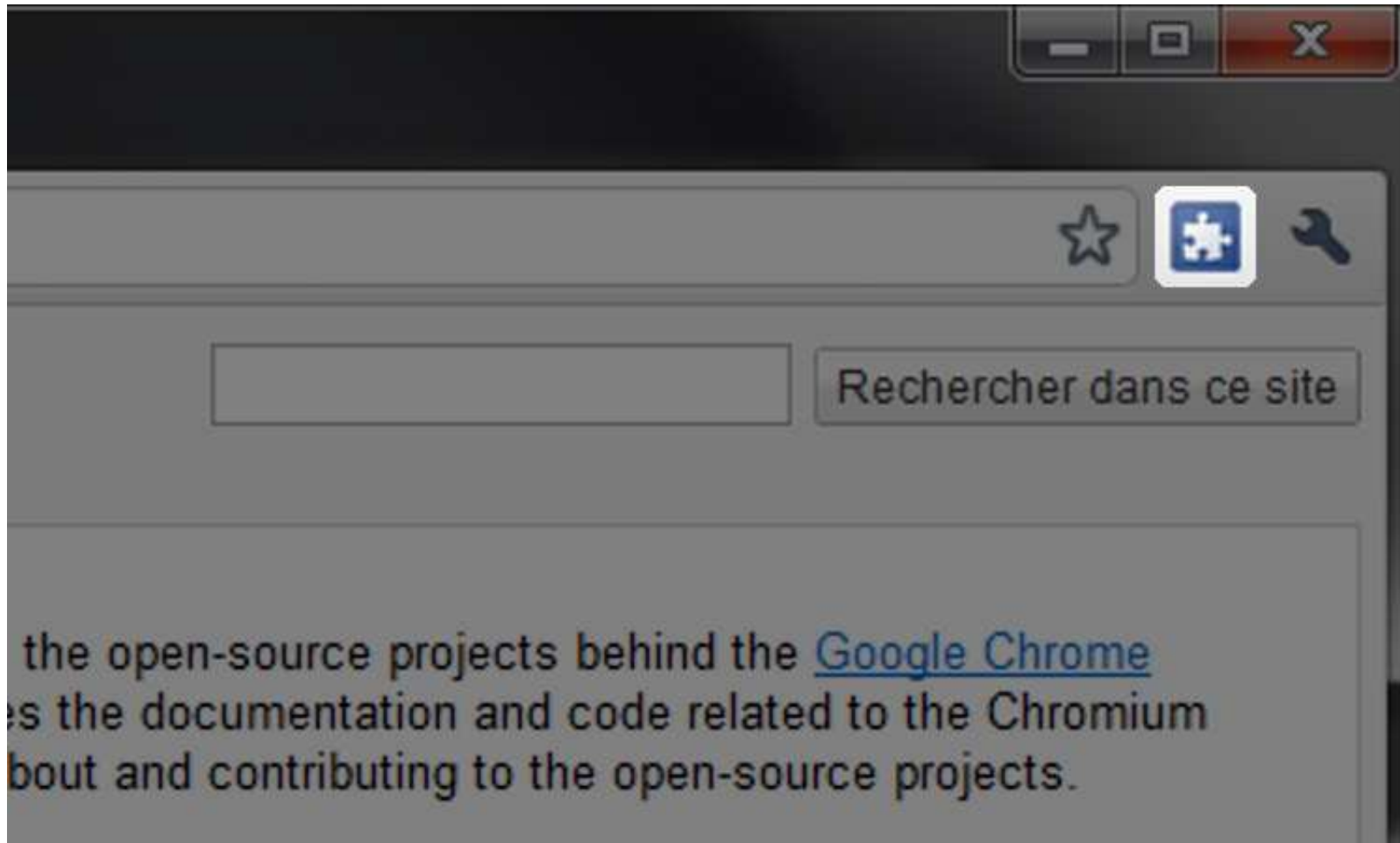
Extensions Chrome

- Objectif :
 - étendre les capacités du navigateur
 - étendre les fonctionnalités d'un site web
- Caractéristiques :
 - simplicité : intégration dans le navigateur
 - stabilité : processus séparé
 - sécurité : permissions
- Délivrée sous forme de fichier zip (extension .crx)

Intégration dans le navigateur

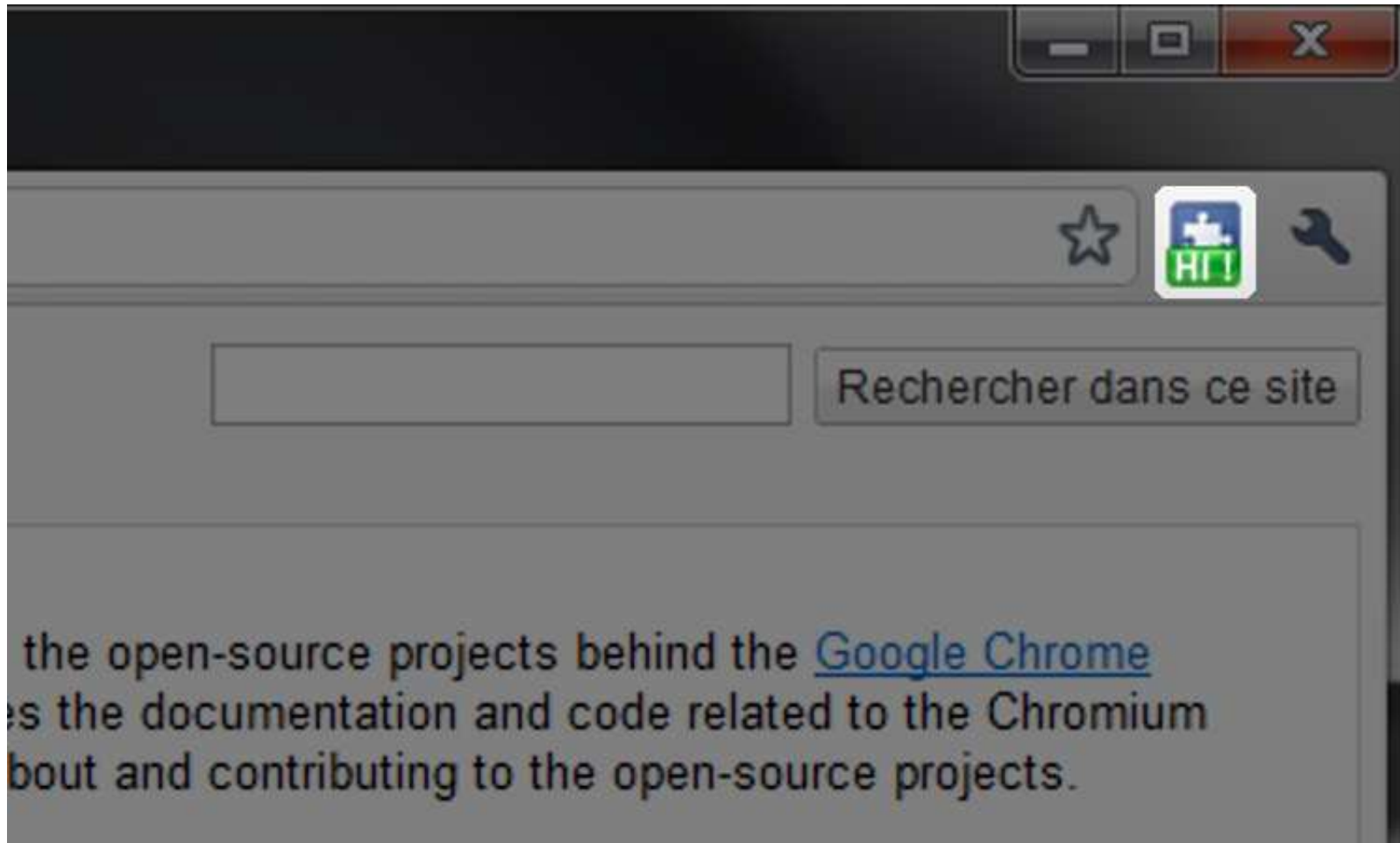
Intégration dans le navigateur

Bouton principal : icône



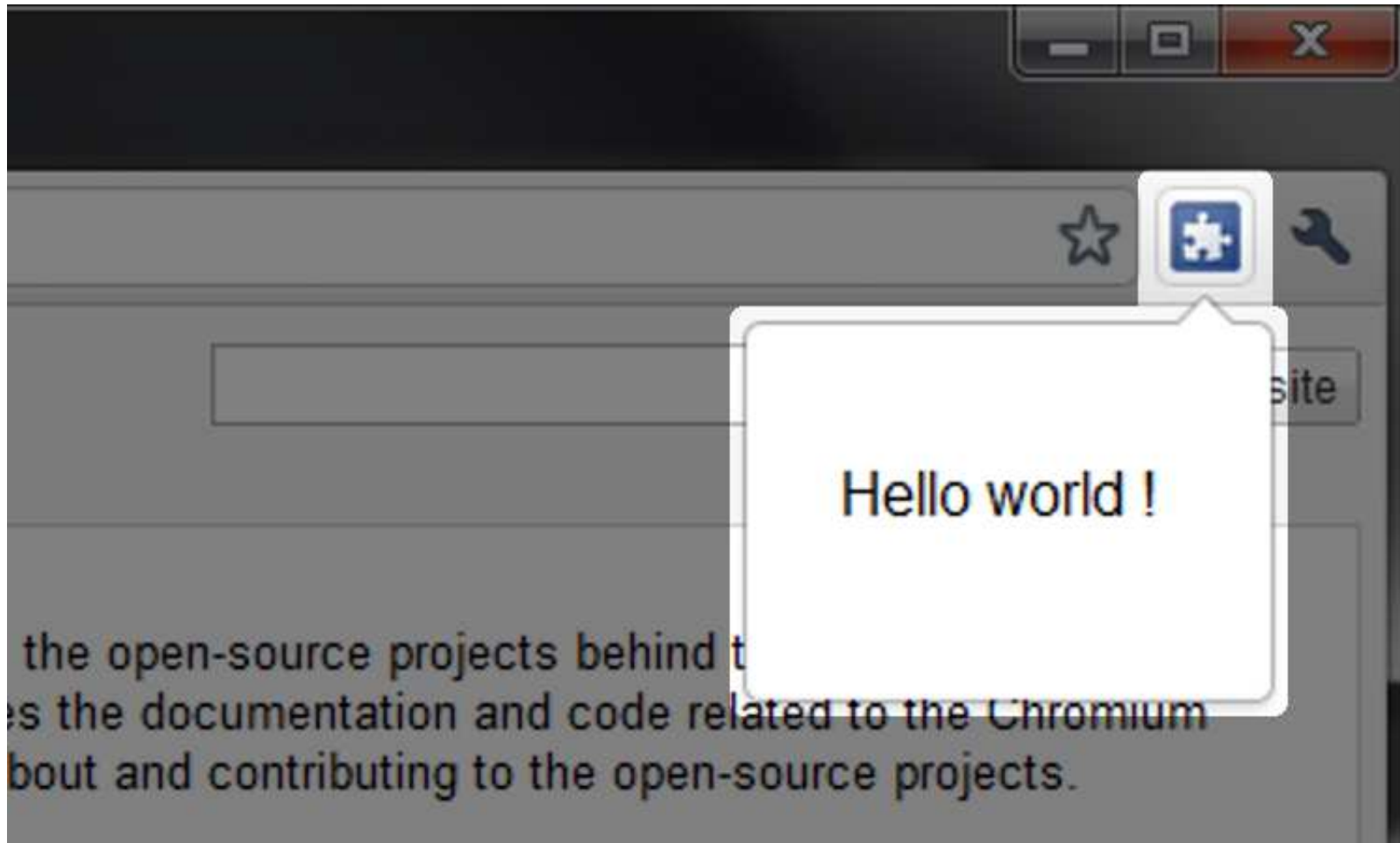
Intégration dans le navigateur

Bouton principal : badge



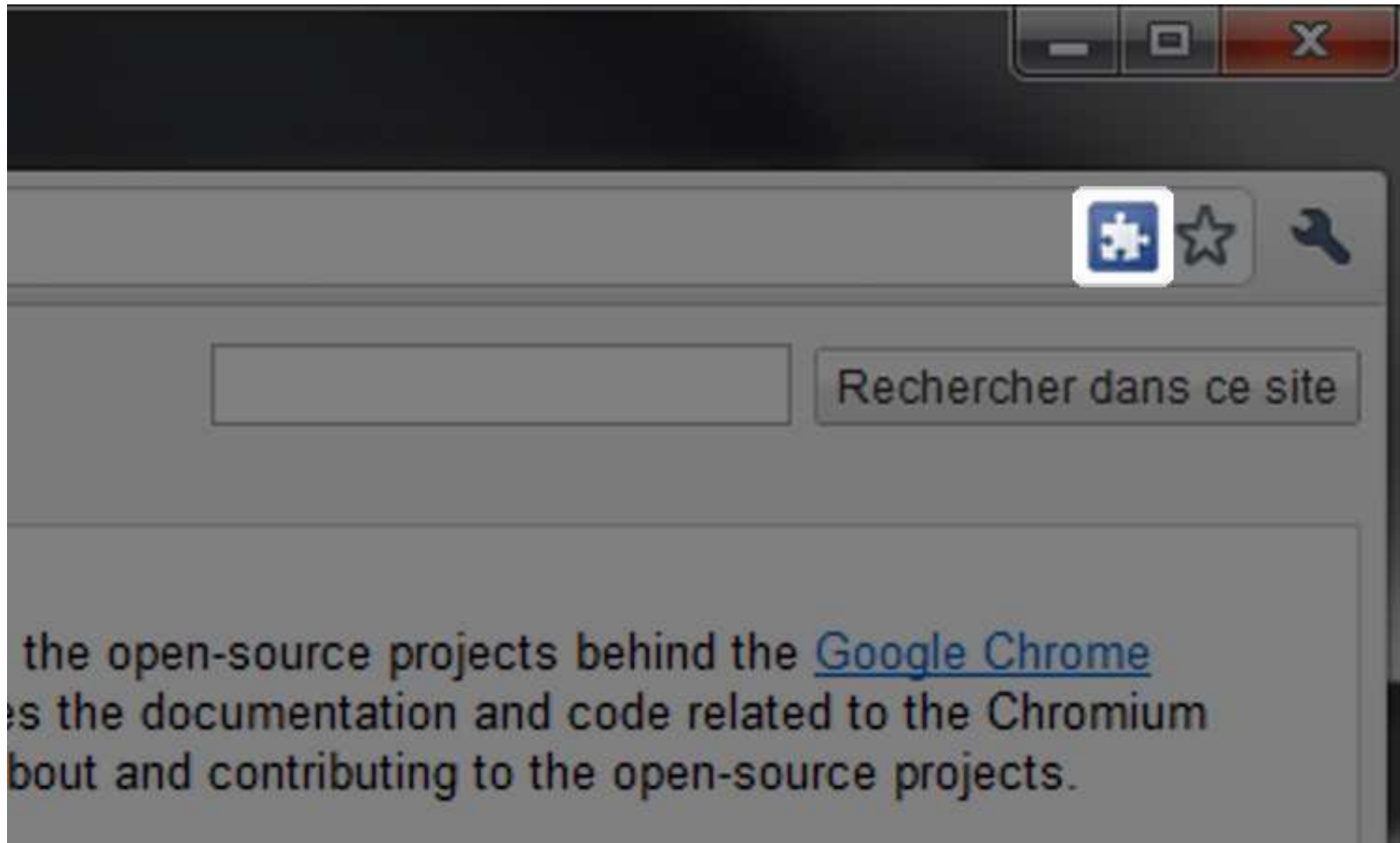
Intégration dans le navigateur

Bouton principal : popup



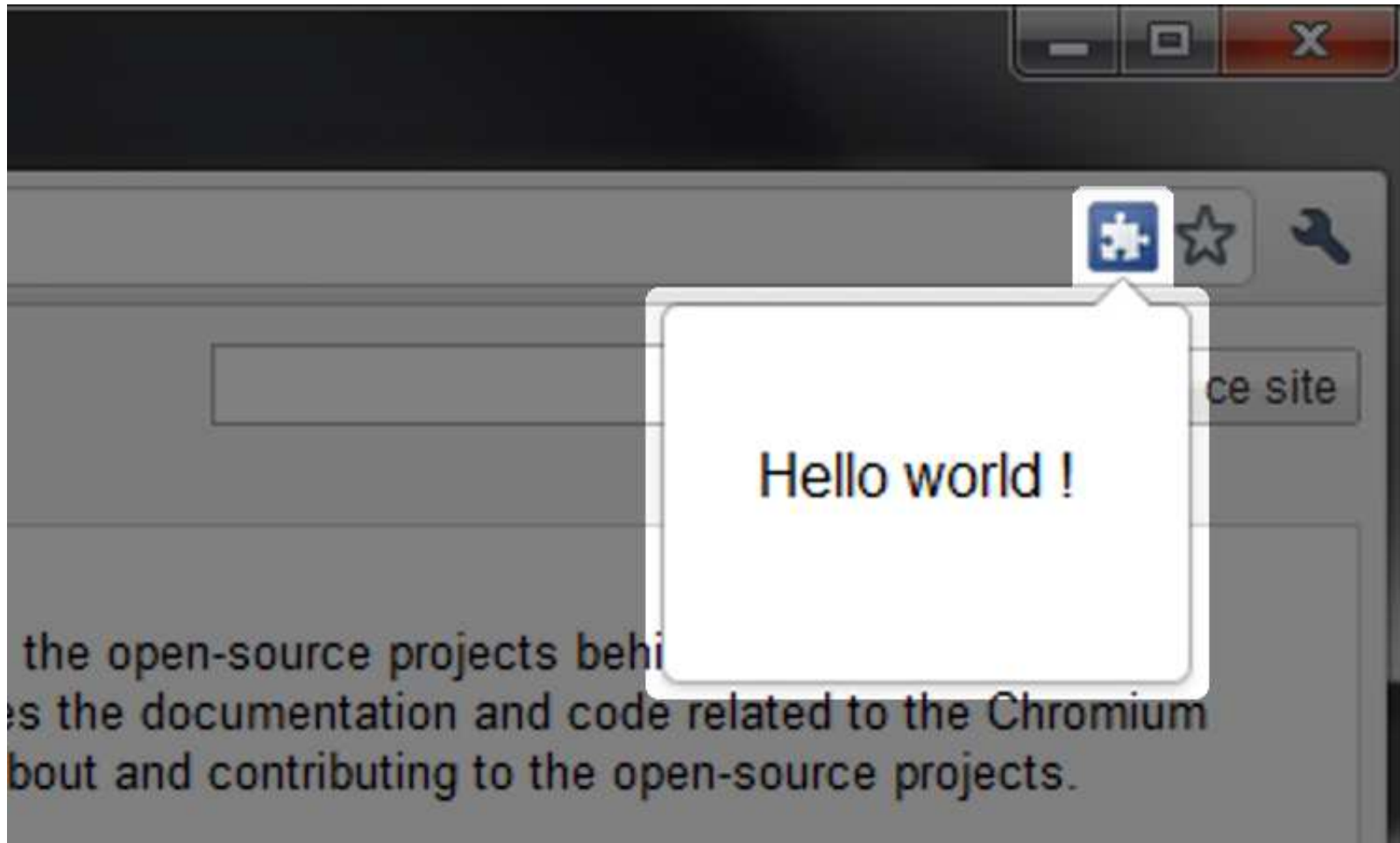
Intégration dans le navigateur

Icône contextuelle



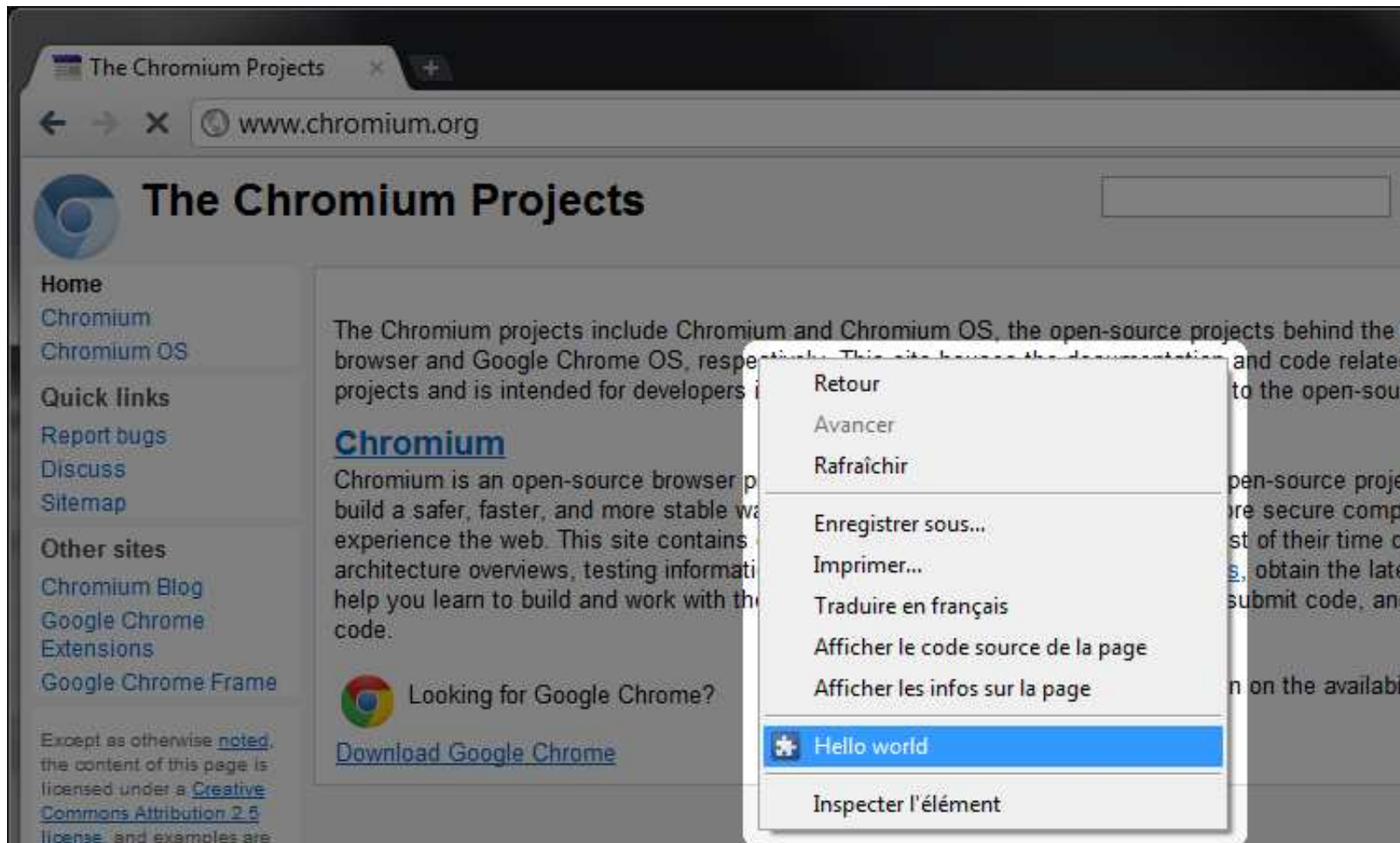
Intégration dans le navigateur

Icône contextuelle : popup



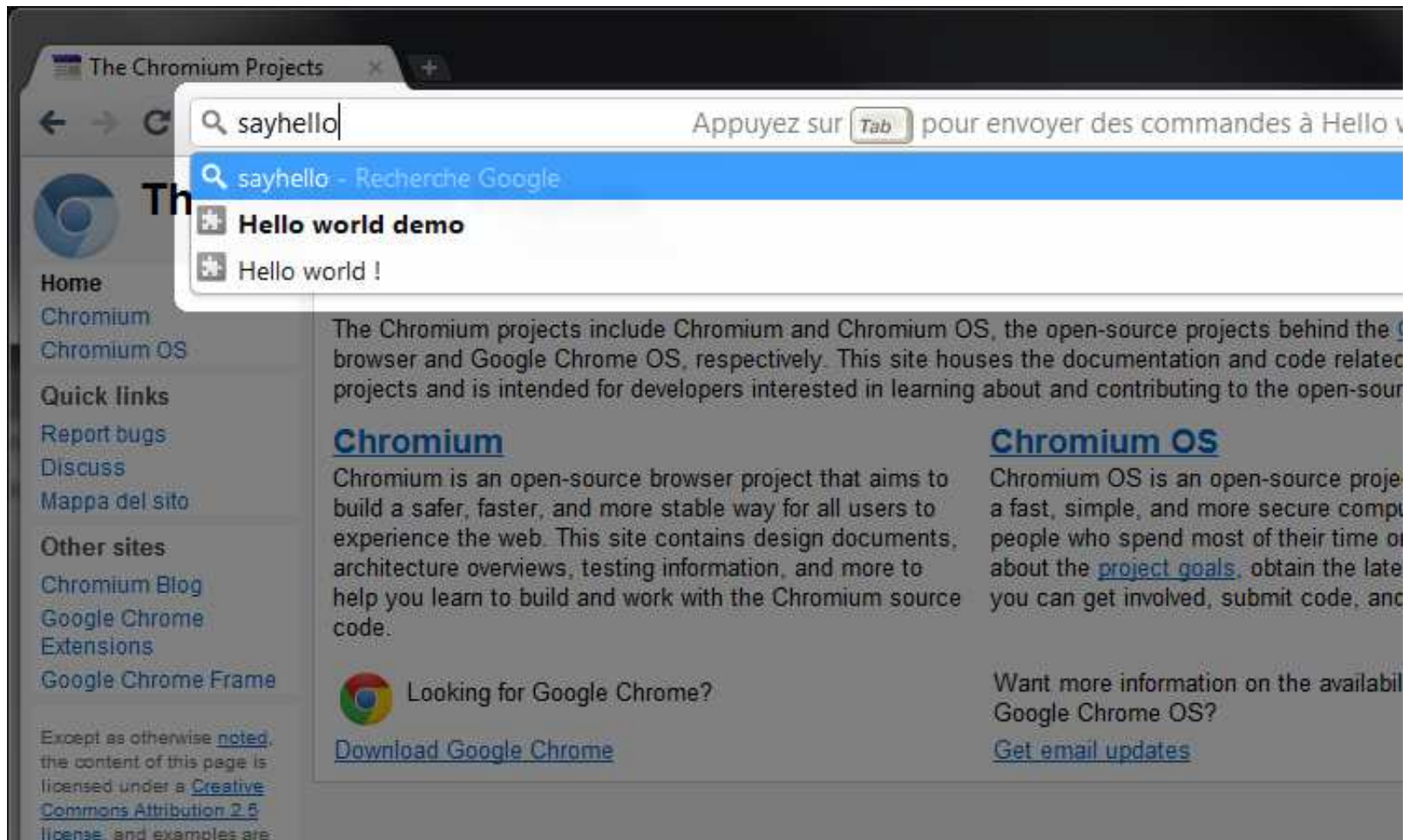
Intégration dans le navigateur

Menu contextuel



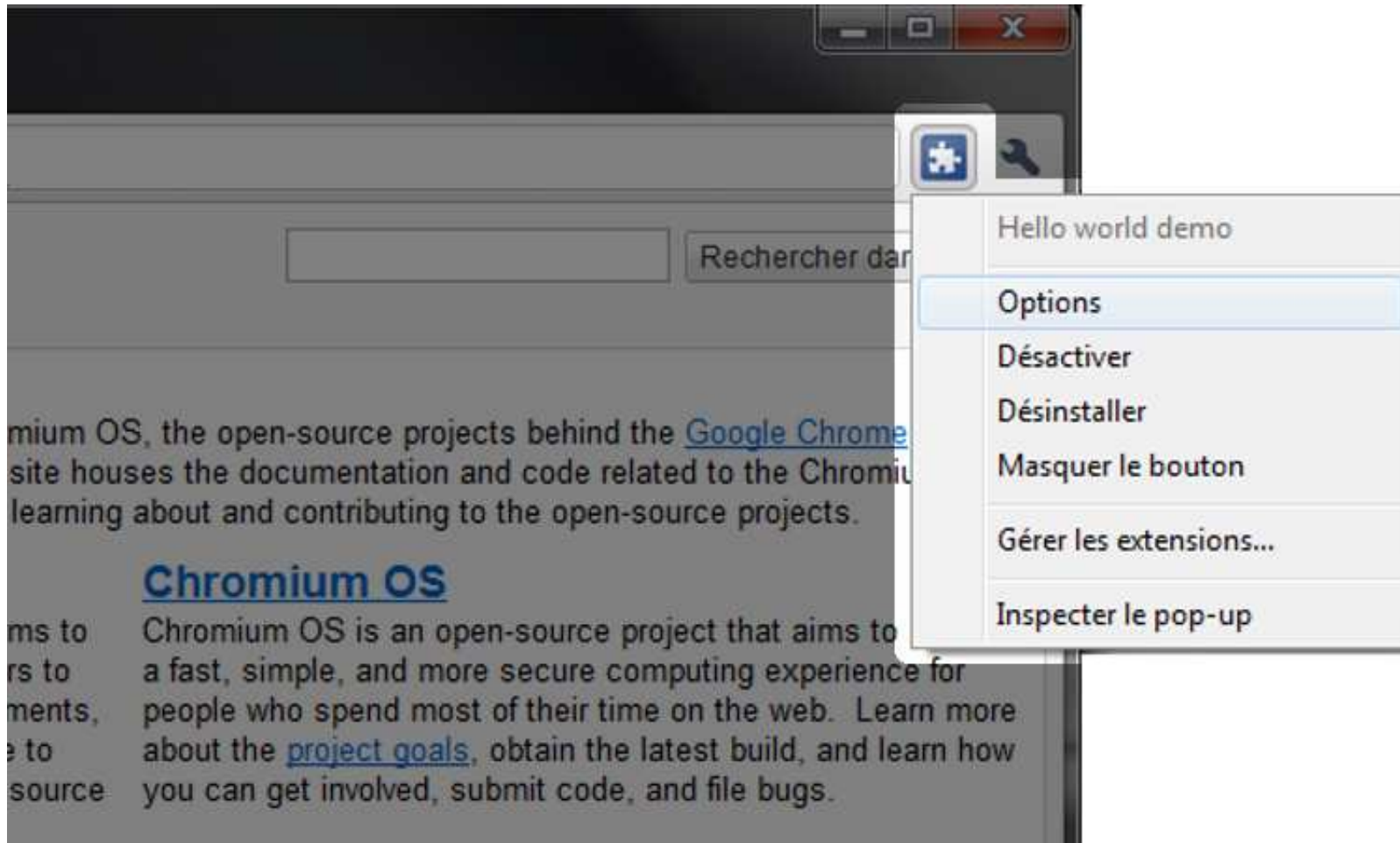
Intégration dans le navigateur

Omnibox



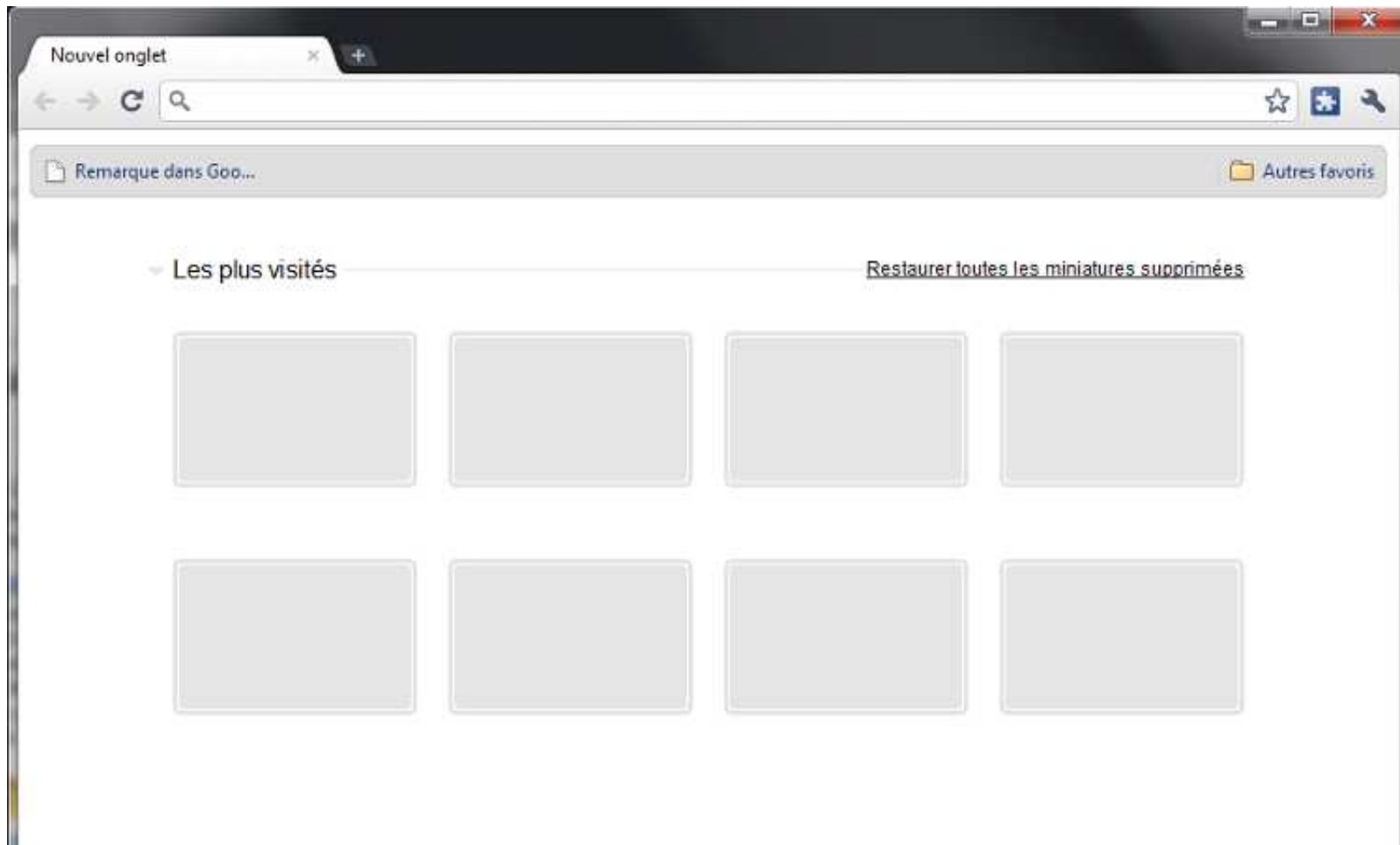
Intégration dans le navigateur

Page d'options



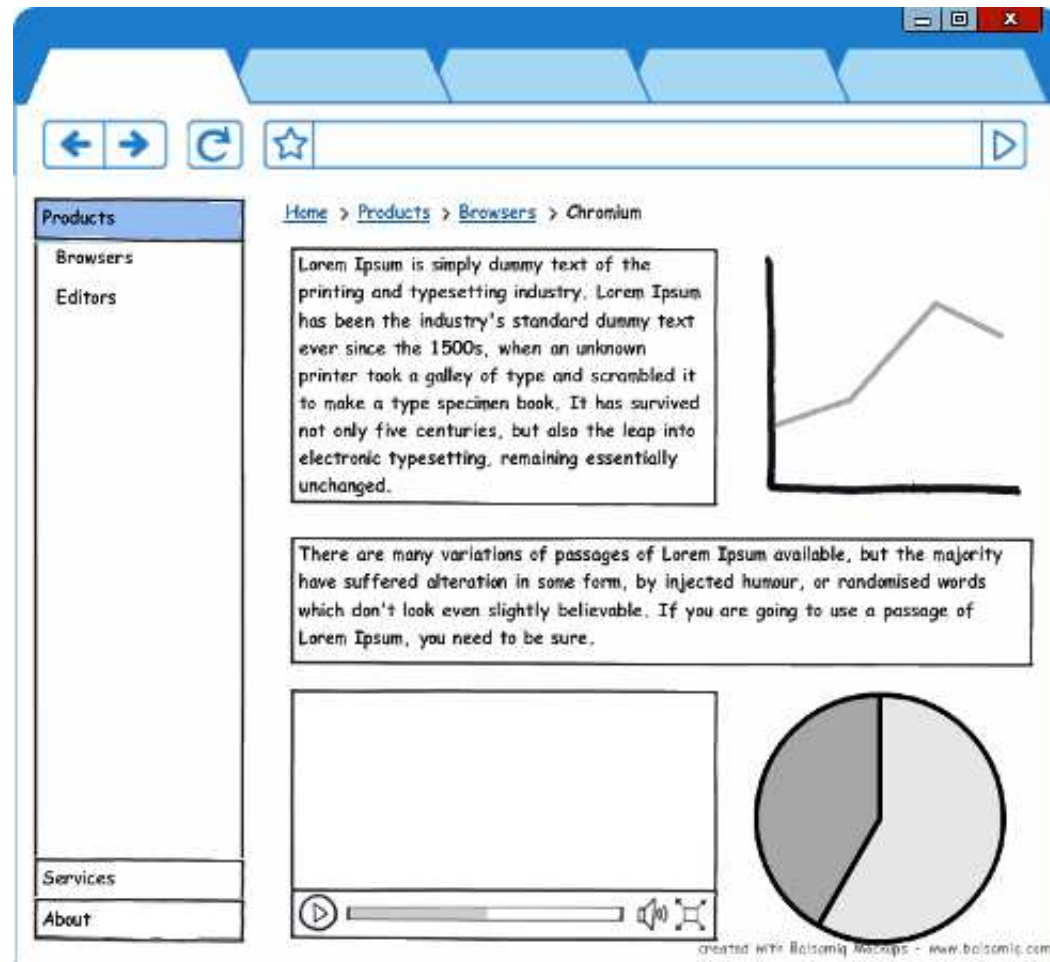
Intégration dans le navigateur

Page de remplacement : nouvel onglet



Architecture

Background page

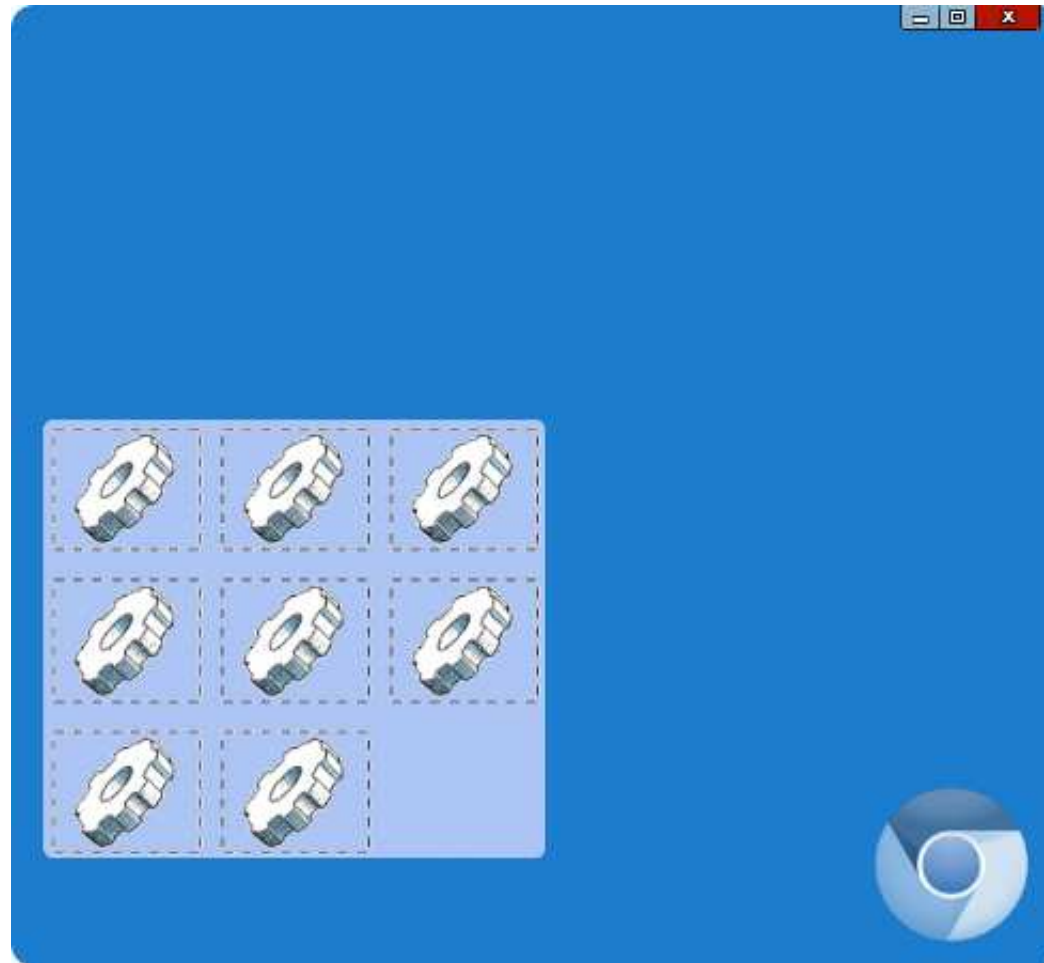


Background page



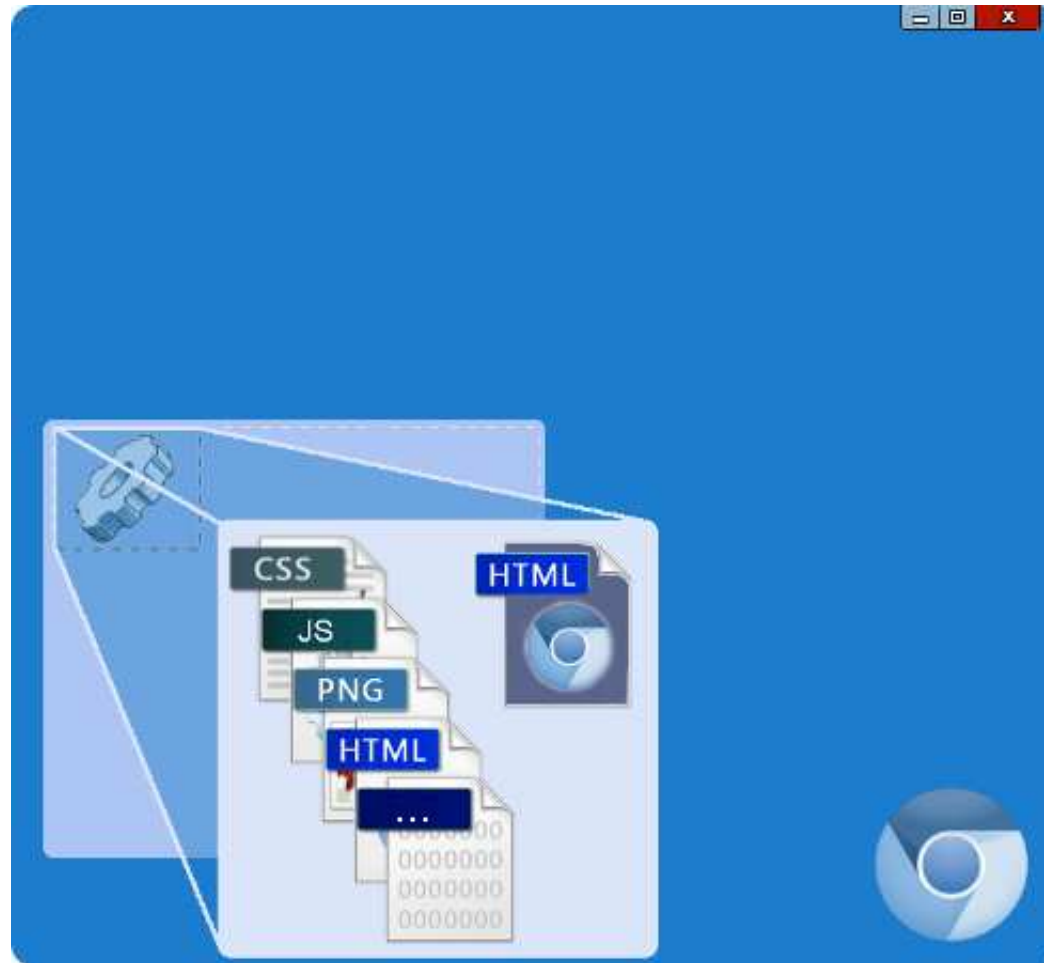
Background page

Extensions installées dans le navigateur



Background page

Extension = ressources web + background page

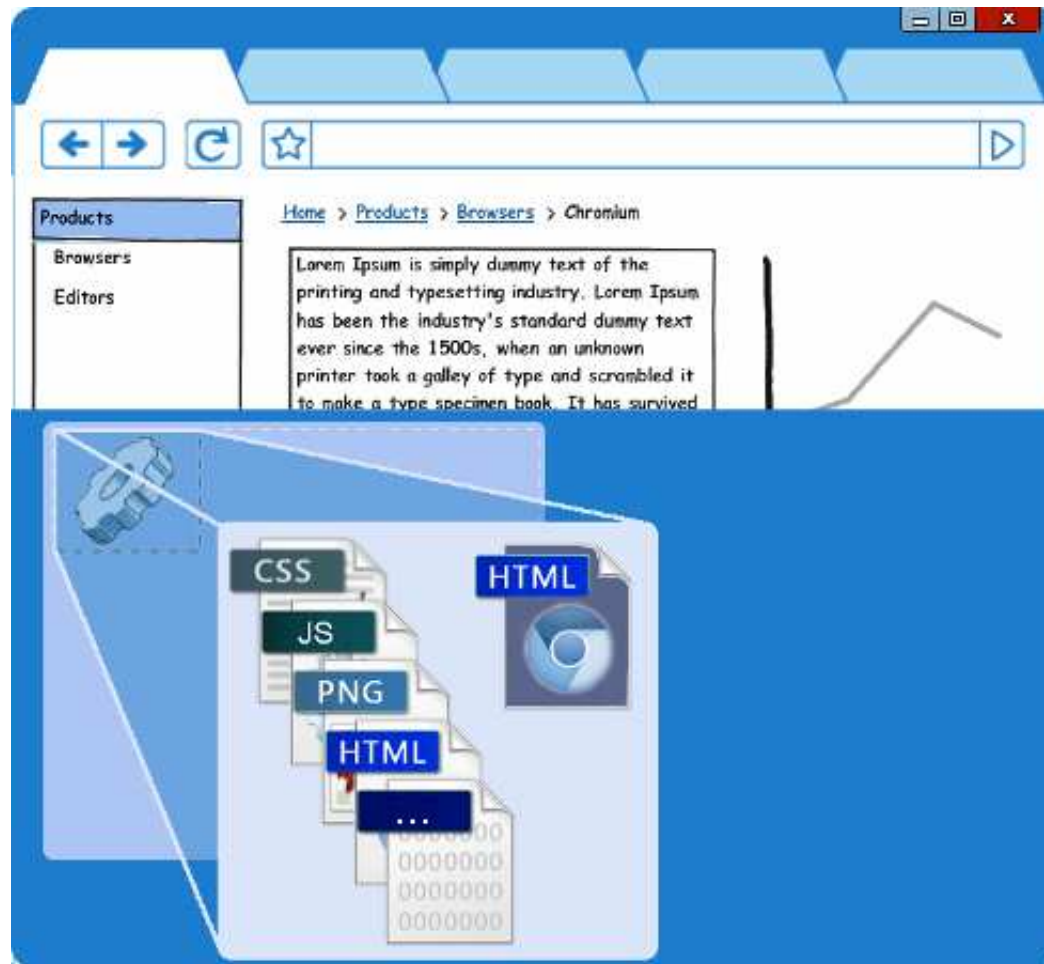


Background page

- page HTML associée à l'extension
- lancée dès le démarrage du navigateur en arrière plan
- jamais affichée
- exécutée dans un processus séparé

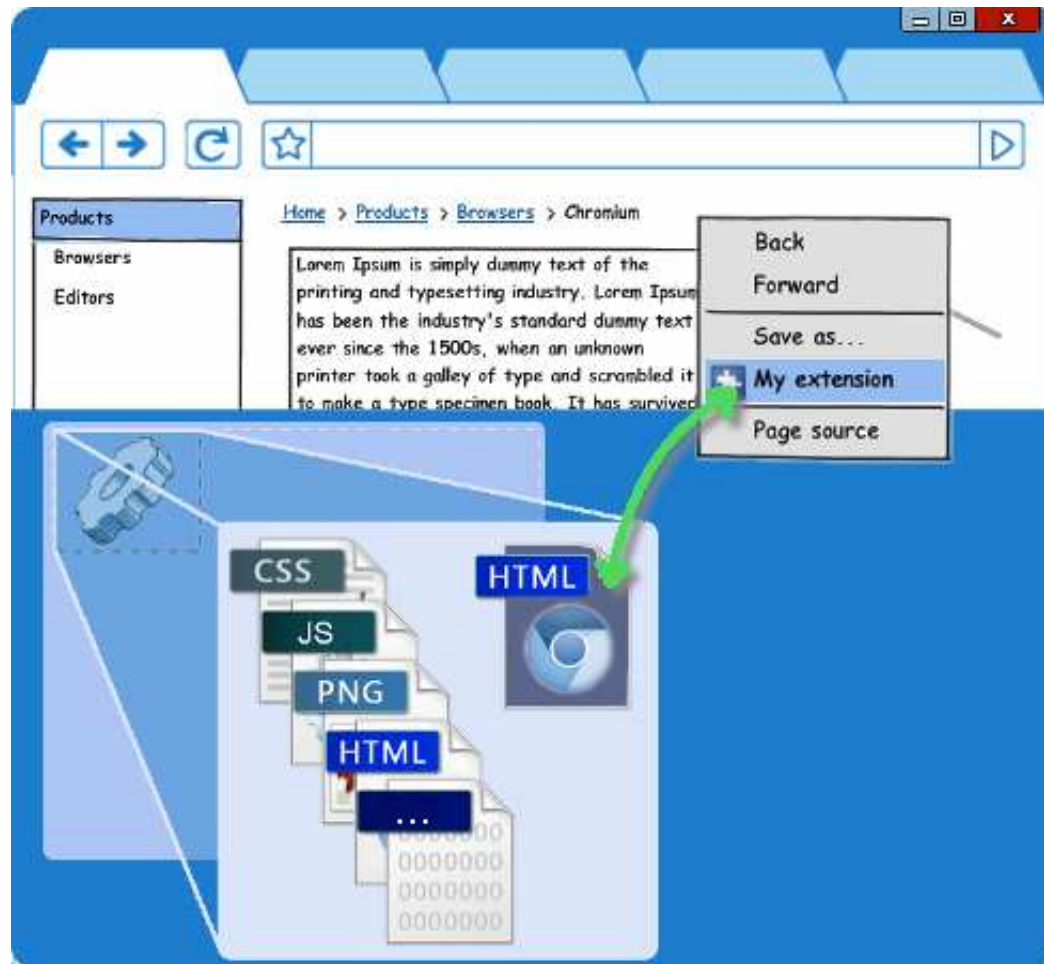
Background page

chef d'orchestre de l'extension



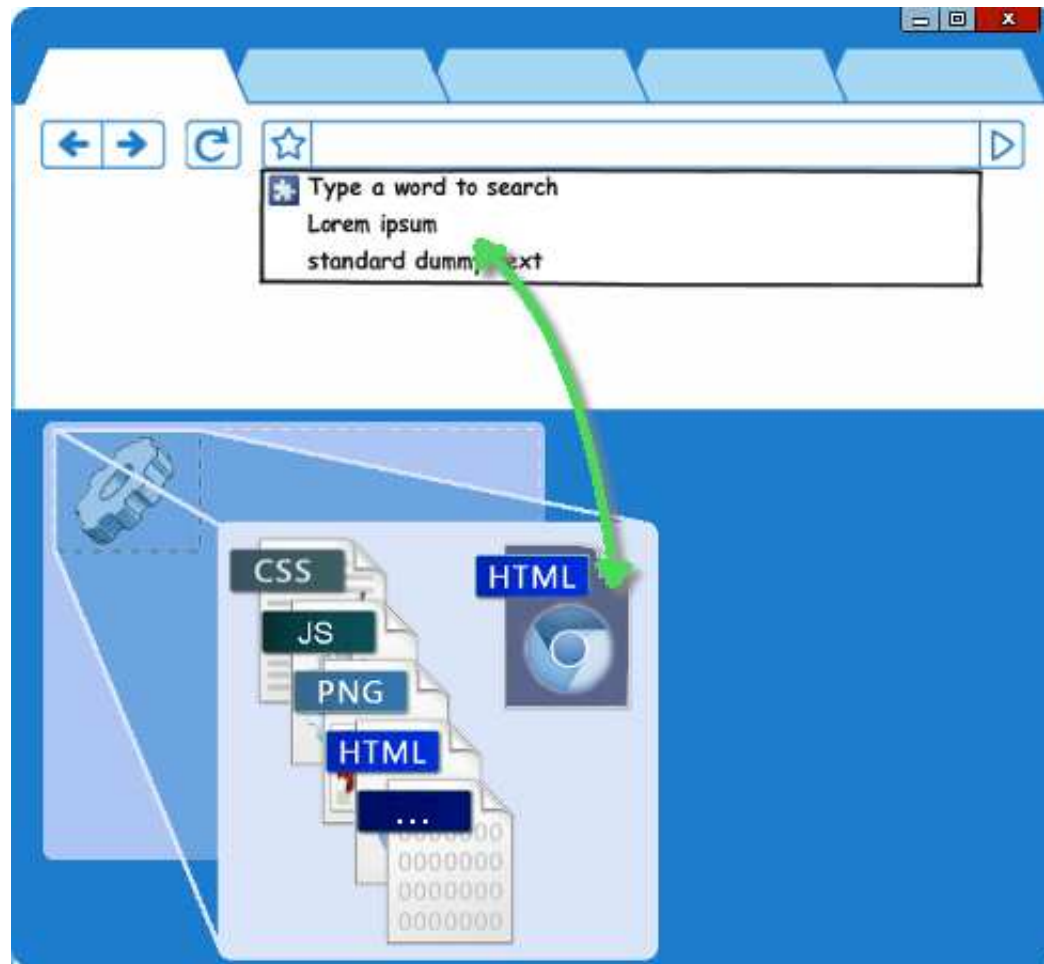
Background page

contrôle du menu contextuel



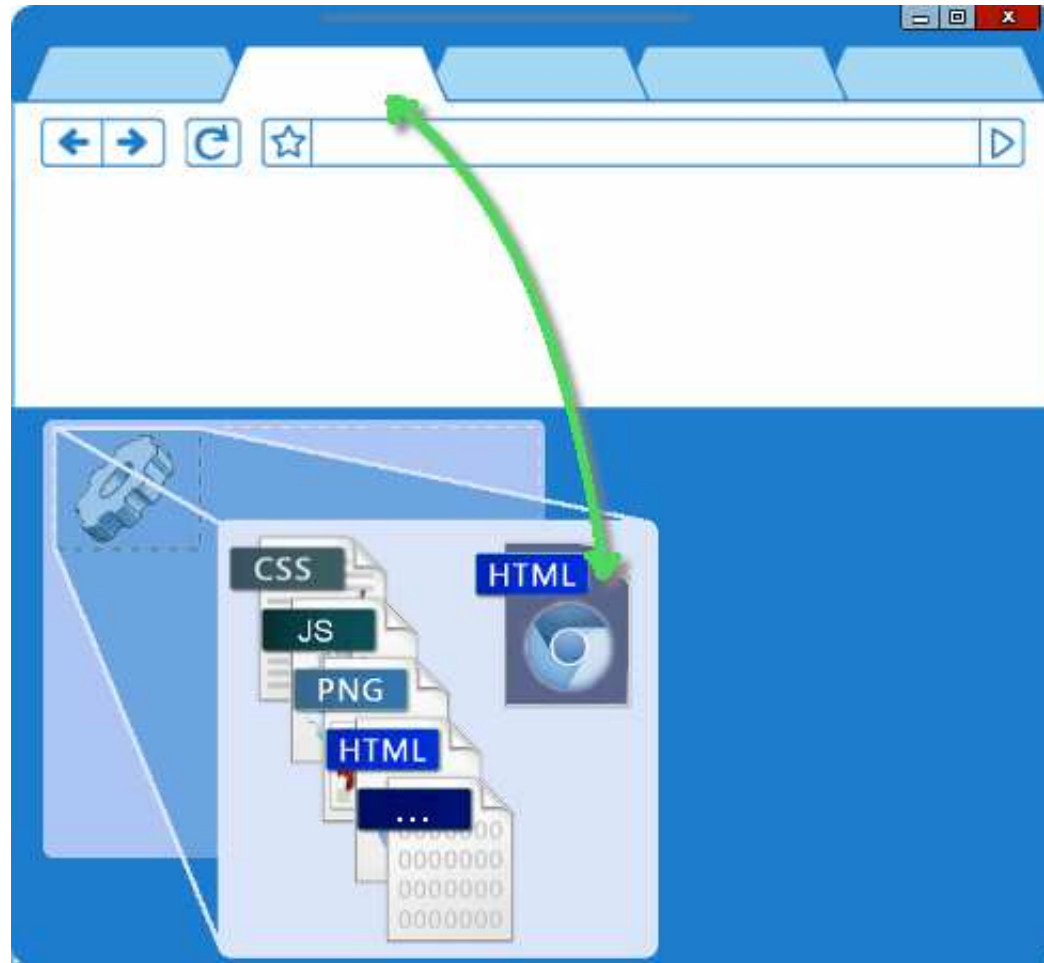
Background page

gestion des suggestions de l'omnibox



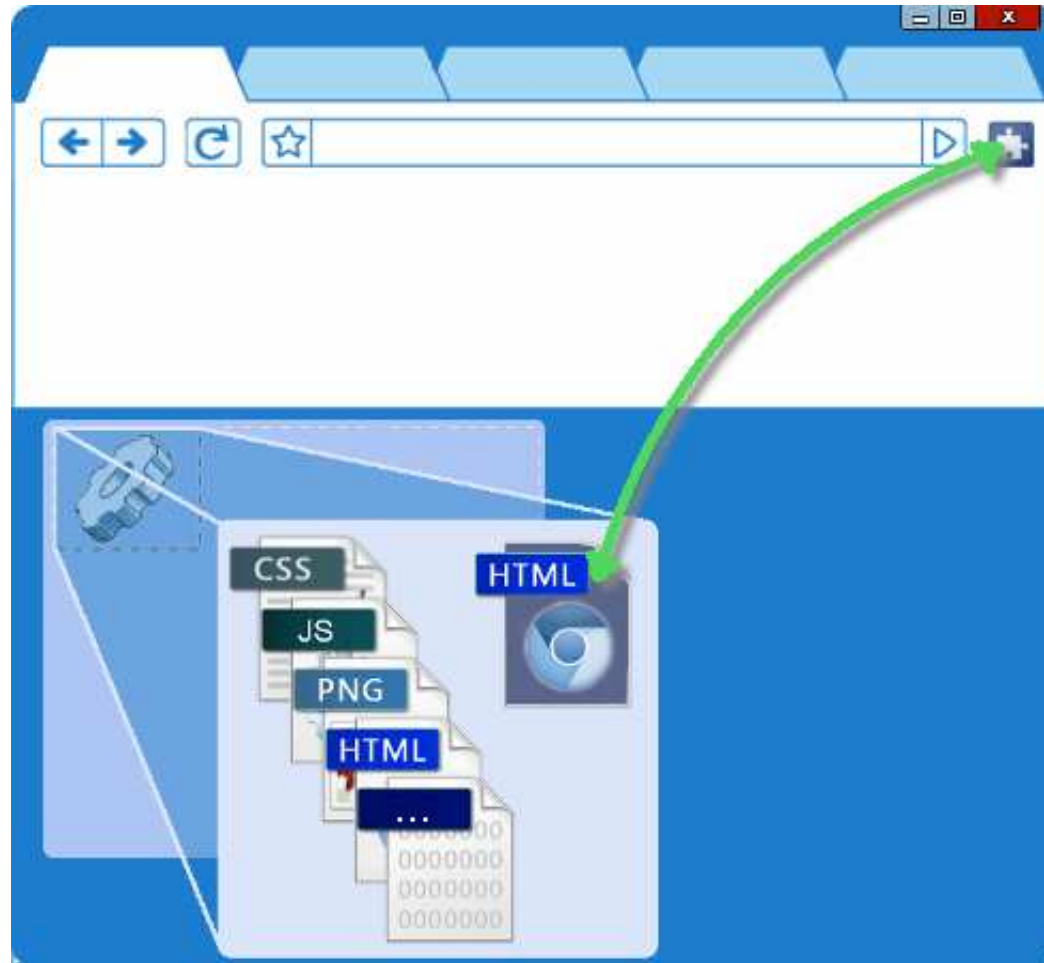
Background page

gestion des onglets et des fenêtres



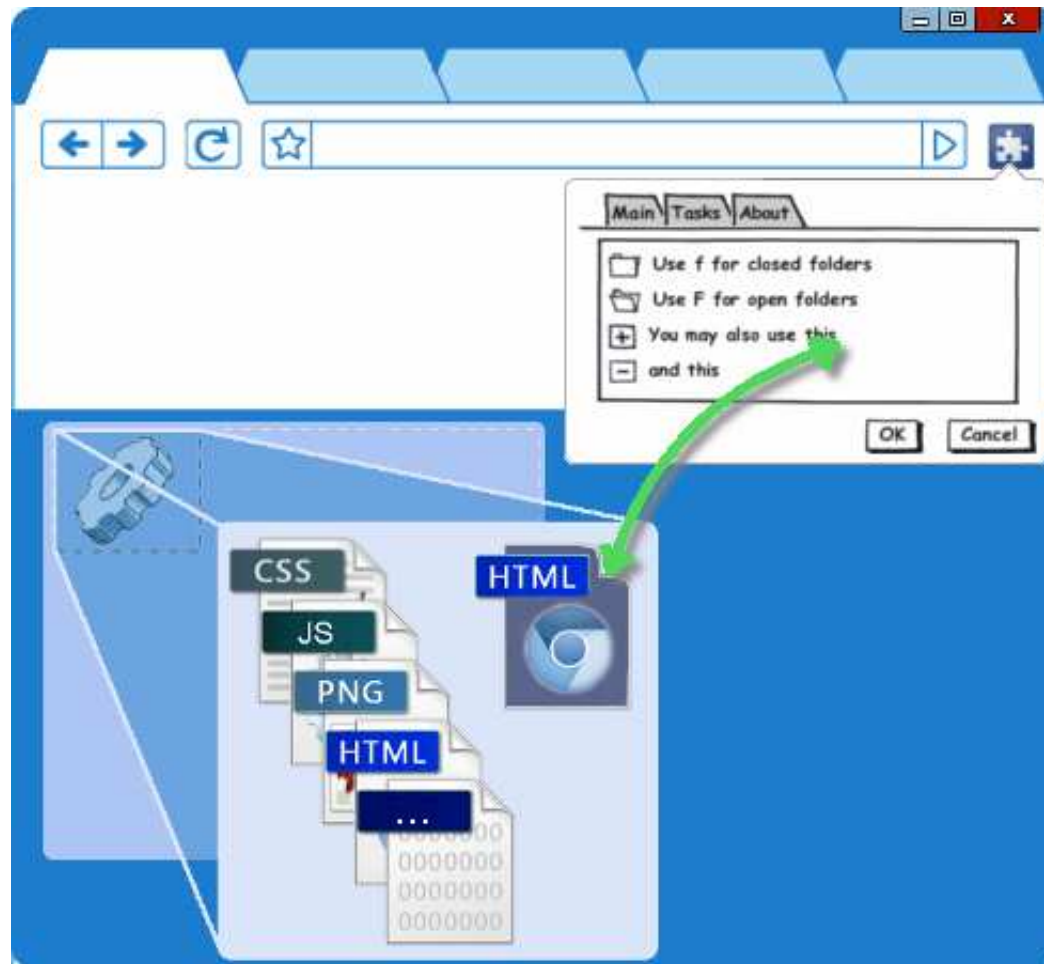
Background page

contrôle du bouton d'action



Background page

communication directe avec les pages de popup



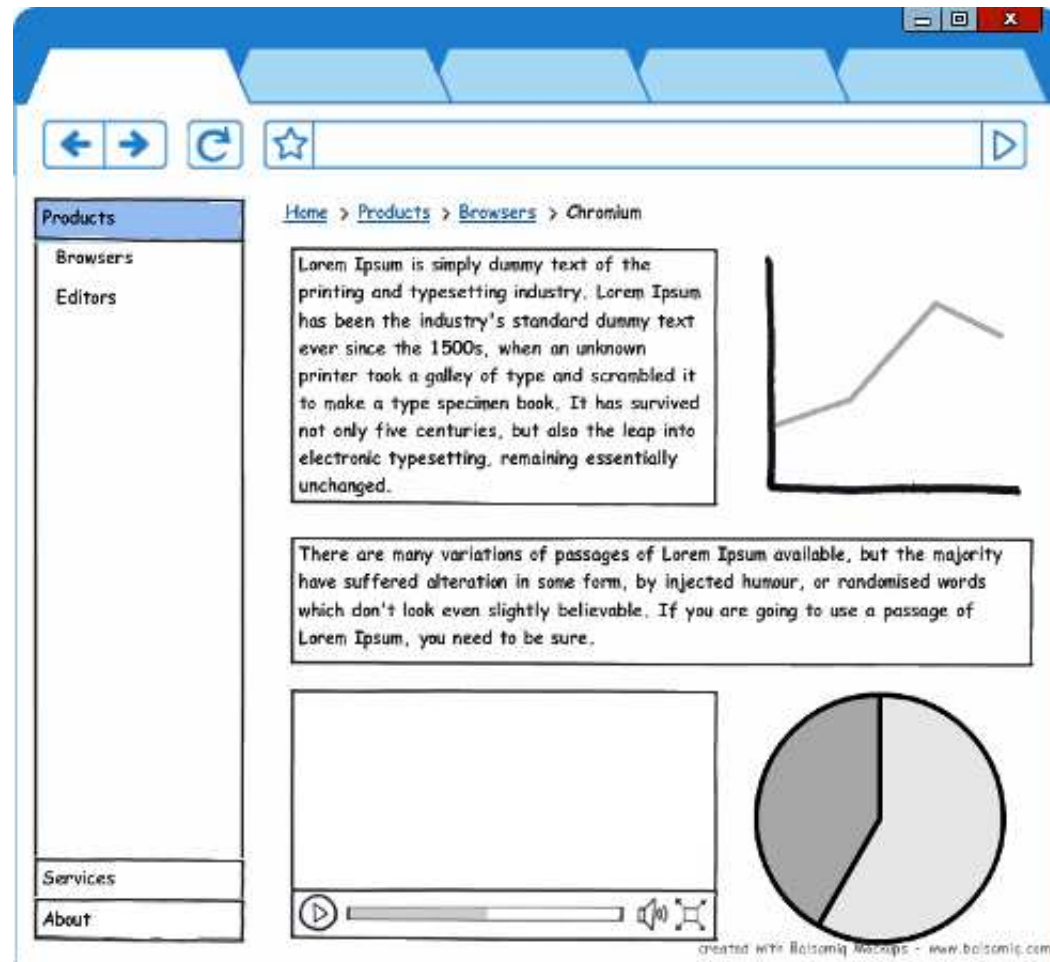
Background page

Autres permissions :

- communication avec des serveurs web
- gestion des marques-pages
- gestion de l'historique
- gestion des cookies
- gestion des extensions installées
- + APIs expérimentales ...

Content scripts

scripts associés à la page affichée



Content scripts

- exécutés dans le processus de l'extension
- possèdent leur propre contexte d'exécution :
 - pas d'accès direct aux variables et fonctions de la page
- contexte d'exécution temporaire :
 - scripts ré-exécutés à chaque chargement d'une page
- ne bénéficient pas des permissions de l'extension

Pages complémentaires

- mêmes permissions que la background page
- pages prédéfinies :
 - popup du bouton principal
 - popup de l'icône contextuelle
 - page d'options
 - pages de remplacement
(i.e. nouvel onglet, marque-pages, historique)
- page ouverte programmatically depuis la background page

Exemple d'extension pas à pas

en 10 étapes

Squelette de l'extension (1)

Manifeste d'installation

- fichier JSON
- contient les données nécessaires à l'installation

```
{  
  "name" : "Bloc notes",  
  "version" : "1.0",  
  "description" : "Prenez facilement vos notes !",  
  "icons" :  
    {  
      "16" : "icon_16.png",  
      "48" : "icon_48.png",  
      "128" : "icon_128.png"  
    }  
}
```

manifest.json

Background page (2)

Déclaration

```
{  
  ...  
  "background_page" : "background.html"  
}
```

manifest.json

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="utf-8">  
    <title>Background page</title>  
    <script src="background.js"></script>  
  </head>  
  <body>  
  </body>  
</html>
```

background.html

Content scripts (2)

Déclaration

```
{  
  ...  
  "content_scripts" :  
  [  
    {  
      "matches" : [ "http://*/*", "https://*/*" ],  
      "js" : [ "content_script.js" ]  
    }  
  ]  
}
```

manifest.json

Content scripts (2)

Affichage d'une popup de saisie (raccourci Alt+N)

```
document.addEventListener(  
  "keyup",  
  function(event) {  
    var note;  
    if (event.altKey && event.keyCode == 78) {  
      note = prompt("Saisissez votre note");  
    }  
  },  
  false  
);
```

content_script.js

Communication (3)

Envoi de la note vers la background page

```
document.addEventListener(  
  "keyup",  
  function(event) {  
    ...  
    if (note)  
      chrome.extension.sendRequest(  
        {  
          message : "sendNote",  
          note : note  
        },  
        function(response) {  
          console.log("response status", note,  
            response.status);  
        }  
      );  
    ...  
  }, false);
```

content_script.js

Communication (3)

Réception de la note dans la background page

```
var notes = [];  
  
function addNote(note) {  
    notes.push(note);  
}  
  
chrome.extension.onRequest.addListener(  
    function(request, sender, sendResponse) {  
        if (request.message == "sendNote") {  
            addNote(request.note);  
            sendResponse(  
                { status : "OK" }  
            );  
        }  
    }  
);
```

background.js

Bouton principal (4)

Déclaration

```
{  
  ...  
  "browser_action" :  
    {  
      "default_icon" : "icon_48.png",  
      "default_title" : "Vous n'avez aucune note",  
      "default_popup" : "popup.html"  
    }  
}
```

manifest.json

Bouton principal : popup (4)

Contenu

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Notes</title>
    <link rel="stylesheet" href="popup.css">
  </head>
  <body>
    <h3>Notes :</h3>
    <ul id="notes-list">
    </ul>
    <script src="popup.js"></script>
  </body>
</html>
```

popup.html

Bouton principal : popup (4)

Affichage des notes

```
var
  bgPage = chrome.extension.getBackgroundPage(),
  listElement = document.getElementById("notes-list");

function refresh() {
  listElement.innerHTML = "";
  bgPage.notes.forEach(
    function(note) {
      var itemElement = document.createElement("li");
      itemElement.innerText = note;
      listElement.appendChild(itemElement);
    }
  );
}

refresh();
```

popup.js

Bouton principal : info-bulle (5)

Mise à jour du libellé

```
...  
function addNote(note) {  
    notes.push(note);  
    refreshMainButton();  
}  
  
...  
function refreshMainButton() {  
    var title;  
    if (notes.length) {  
        title = "Vous avez " + notes.length + " note" +  
            (notes.length > 1 ? "s" : "");  
    } else {  
        title = "Vous n'avez aucune note";  
    }  
    chrome.browserAction.setTitle({  
        title : title  
    });  
}
```

background.js

Bouton principal : badge (5)

Mise à jour du libellé

```
...  
function refreshMainButton() {  
  var title, text;  
  if (notes.length) {  
    ...  
    text = String(notes.length);  
  } else {  
    ...  
    text = "";  
  }  
  ...  
  chrome.browserAction.setBadgeText({  
    text : text  
  });  
  chrome.browserAction.setBadgeBackgroundColor({  
    color : [ 4, 229, 36, 255 ]  
  });  
}
```

background.js

Menu contextuel (6)

Déclaration

```
{  
  ...  
  "permissions" :  
    [  
      "contextMenus"  
    ]  
}
```

manifest.json

Menu contextuel (6)

Création d'une entrée

```
...  
chrome.contextMenus.create(  
  {  
    title : "Ajouter dans les notes",  
    contexts :  
    [  
      "selection"  
    ],  
    onclick : function(info, tab) {  
      addNote(info.selectionText);  
    }  
  }  
);
```

background.js

Omnibox (7)

Déclaration

```
{  
  ...  
  "omnibox" :  
    {  
      "keyword" : "note"  
    }  
}
```

manifest.json

Suggestion par défaut

```
...  
chrome.omnibox.setDefaultSuggestion(  
  {  
    description: "cherchez une note"  
  }  
);
```

background.js

Omnibox (7)

Suggestion en cours de saisie

```
...
chrome.omnibox.onInputChanged.addListener (
  function(text, suggestCallback) {
    var suggestions = [];
    notes.forEach (
      function(note) {
        if (note.toLowerCase().indexOf(text.toLowerCase())
            != -1)
          suggestions.push({
            content : note,
            description : note.replace(/[<>]/g, "")
          });
      }
    );
    suggestCallback(suggestions);
  }
);
```

background.js

Utilisation de HTML5

Notes : sauvegarde (8)

localStorage.setItem et JSON.stringify

```
...  
function addNote(note) {  
    notes.push(note);  
    saveNotes();  
    refreshMainButton();  
}  
  
function saveNotes() {  
    localStorage.setItem(  
        "notes",  
        JSON.stringify(notes)  
    );  
}
```

background.js

Notes : chargement (8)

localStorage.getItem et JSON.parse

```
var notes;
```

```
...
```

```
function loadNotes() {  
    var storedNotes = localStorage.getItem("notes");  
    notes = storedNotes ? JSON.parse(storedNotes) : [];  
    refreshMainButton();  
}
```

```
...
```

```
loadNotes();
```

background.js

Notes : édition (9)

attributs hidden

```
<!DOCTYPE html>
<html>
  ...
  <body>
    ...
    <div class="links">
      <a href="#" id="edit-link">editor</a>
      <a href="#" id="save-link" hidden>sauver</a>
      <a href="#" id="cancel-link" hidden>annuler</a>
    </div>
  </body>
  ...
</html>
```

popup.html

Notes : édition (9)

attributs hidden et contenteditable

```
var ...  
    editLink    = document.getElementById("edit-link"),  
    saveLink    = document.getElementById("save-link"),  
    cancelLink  = document.getElementById("cancel-link");
```

```
function setEditMode(mode) {  
    listElement.contentEditable = mode;  
    editLink.hidden = mode;  
    saveLink.hidden = cancelLink.hidden = !mode;  
    listElement.focus();  
}
```

popup.js

Notes : édition (9)

```
var ...
    lastContent;
...
editLink.addEventListener("click",
    function() {
        if (listElement.children.length) {
            lastContent = listElement.innerHTML;
            setEditMode(true);
        }
    }, false
);

cancelLink.addEventListener("click",
    function() {
        listElement.innerHTML = lastContent;
        setEditMode(false);
    }, false
);
```

popup.js

Notes : édition (9)

```
...
saveLink.addEventListener("click",
    function() {
        bgPage.setNotes(
            Array.prototype.map.call(listElement.children,
                function(itemElement) {
                    return itemElement.innerText;
                }
            )
        );
        setEditMode(false);
    }, false);
```

popup.js

```
...
function setNotes(newNotes) {
    notes = newNotes;
    saveNotes();
    refreshMainButton();
}
```

background.js

Notes : export (10)

```
<!DOCTYPE html>
<html>
  ...
  <body>
    ...
    <a href="#" id="export-link">exporter</a>
  </body>
  ...
</html>
```

popup.html

Notes : export (10)

API File et FileWriter

```
var ...  
    exportLink = document.getElementById("export-link");  
    ...  
exportLink.addEventListener(  
    "click", bgPage.exportNotes, false  
);
```

popup.js

```
...  
function exportNotes() {  
    var blobBuilder = new WebKitBlobBuilder();  
    blobBuilder.append(notes.join("\r\n"));  
    open(  
        webkitURL.createObjectURL(  
            blobBuilder.getBlob()  
        )  
    );  
};  
}
```

background.js

Références & adresses

- Manuel de référence

<http://code.google.com/chrome/extensions/getstarted.html>

- Liste de discussion des extensions chromium

<http://groups.google.com/a/chromium.org/group/chromium-extensions>

- Web store

<https://chrome.google.com/webstore>

- Ressources HTML5

<http://www.html5rocks.com/en/resources.html>

Questions ?