

HTML5 en pratique

Javascript, le langage du web



- Introduction
- Les balises
- CSS 3
- Javascript, le langage du web
- Vers des application plus interactives
- Gestion des données
- Multimédia
- Conclusion

- En Javascript, presque tout est objet (Array, String, Function...)
- Les points à comprendre pour ne pas tomber dans les pièges :
 - Comment itérer sur des objets ou tableaux
 - Le *scope*, ou portée des variables
 - Les opérateurs de comparaison de variables
 - Les paramètres implicites et les types invocations de fonctions
 - Les bons outils pour travailler / débbugger

- En Javascript, un tableau est un objet. Il est possible de les parcourir l'un comme l'autre

- Parcourir un tableau

```
var chiffres = ['un', 'deux', 'trois', 'quatre'];  
var size = chiffres.length;  
for(var i=0; i<size; i++){  
    doSomething(chiffres[i]);  
}
```

- Parcourir les propriétés d'un objet

```
for (var i in obj) {  
    console.log(i + " = " + obj[i]);  
}
```

- La portée ("scope") d'une variable détermine sa visibilité au sein du programme
- Il existe deux scopes
 - Scope **global**
 - accessible dans tout le programme
 - pas de mot-clé
 - Scope **local**
 - accessible uniquement dans la fonction dans laquelle la variable est déclarée
 - mot-clé **var**

```
// Global
foo = "foo";

function f() {
    bar = "bar";    // Global
    var answer = 42; // Local à f()
}

f();

// Affiche "foo"
console.log(foo);

// Affiche "bar"
console.log(bar);

// undefined car "answer" n'est
// visible que dans la fonction f()
console.log(answer);
```

- Javascript offre deux façons de comparer des valeurs
 - `==` compare en forçant les types (égalité abstraite)
 - `===` compare les valeurs (égalité stricte). A préférer !

```
"1" == 1;    // true  
"1" === 1;   // false
```

- Les variables peuvent prendre certaines valeurs spéciales
 - `undefined` est l'état par défaut d'une variable non définie
 - `null` est l'état d'un objet défini mais qui n'a pas de valeur
 - `NaN` signifie "Not a Number" et caractérise généralement le résultat d'une opération sans résultat (ex : `Math.sqrt(-4)`)

- Les fonctions définissent dans leur contexte 2 variables implicites
 - **arguments** représente les paramètres passés à la fonction.
Note : il est légal de passer un nombre de paramètres différent de celui défini dans la signature de la fonction.

```
function myFunc(a, b) {  
    for (i in arguments) {  
        console.log(i);  
    }  
}  
  
myFunc(1, 2, 3, 4);
```

```
>>> function myFunc(a, b){  
4);  
1  
2  
3  
4
```


- Les fonctions définissent dans leur contexte 2 variables implicites
 - **this** représente le contexte d'appel d'une fonction

```
foo = "foo";  
function f() {  
    console.log(this.foo);    // Erreur ? Pas toujours.  
}
```

Attention !

Contrairement au "this" des langages objets traditionnels, le "this" de Javascript dépend de la façon dont la fonction est appelée.

- Le type de fonction détermine le contexte vu par this
 - Fonction *top-level* : this est l'objet window

```
function f() {  
    console.log("this="+this);  
}  
f(); // this=[object Window]
```

- Méthode appelée sur un objet : this est l'objet cible

```
var obj = {  
    f : function() {  
        console.log("this="+this);  
    }  
}  
obj.f(); // this=[object Object]
```

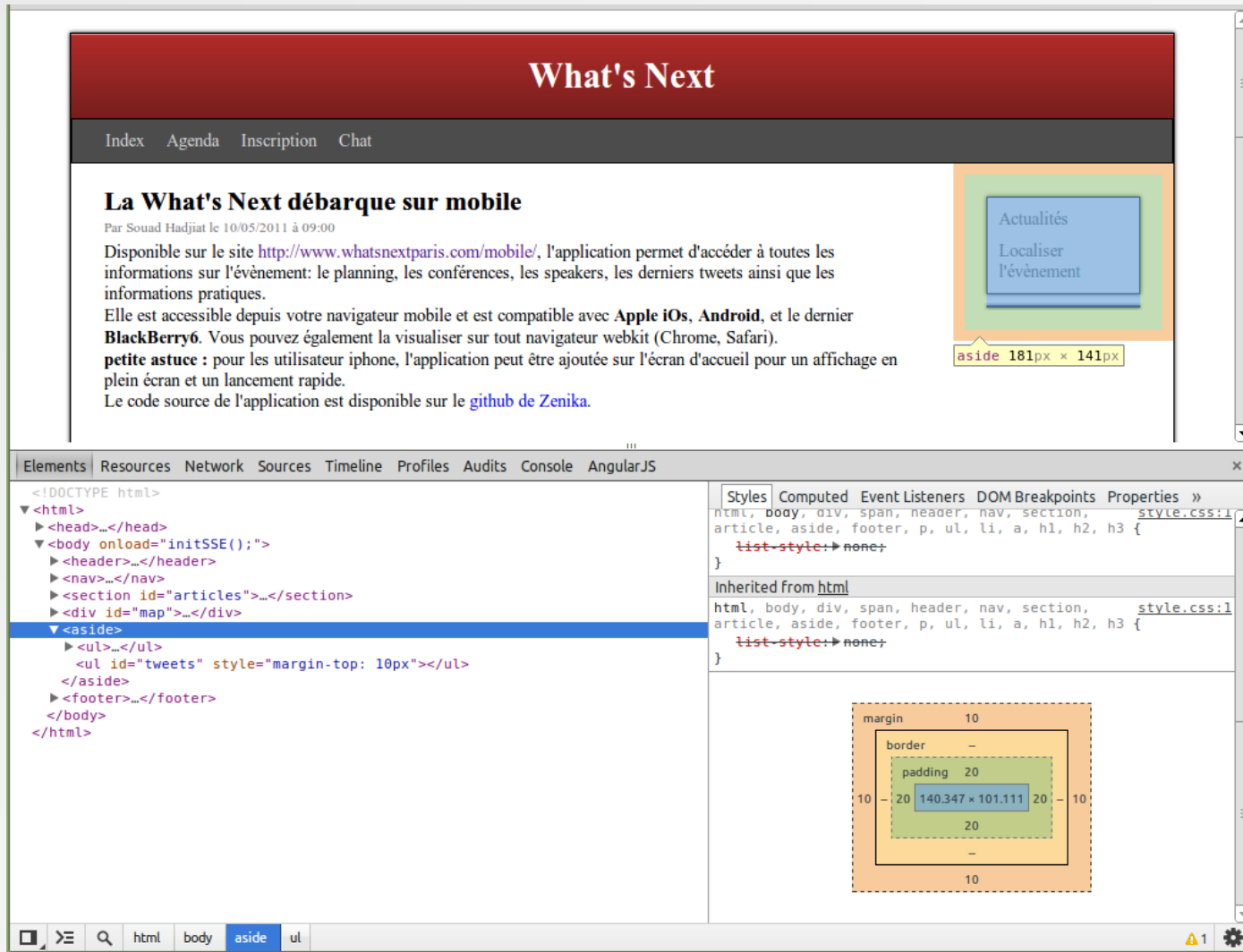
- Le type de fonction détermine le contexte vu par this
 - Constructeur : this est l'objet en cours de construction

```
function MyClass() {  
    console.log("this="+this);  
}  
var instance = new MyClass();           // this=[object Object]
```

- Fonction appelée par **apply** ou **call** : this réfère à l'objet passé en premier paramètre

```
Math.min.apply(Math, [1, 2, 3, 4]);
```

- **Webkit developer tools** : disponible sur tous les navigateurs basés sur Webkit (Chrome, Safari)
- **Firebug** : plugin de Firefox avec de nombreuses fonctionnalités (visionnage du html, des scripts, des requêtes en fonction du temps, une console d'exécution de javascript...)
- **JSLint** : outil d'analyse statique développé par D. Crockford. Valide le javascript soumis et teste de nombreuses mauvaises pratiques.
- **jsfiddle.net** : application web pour tester des fragments de code. L'application permet d'incorporer la plupart des librairies connues. (jquery, extjs...) pour tester directement leurs fonctionnalités.



The screenshot displays the Chrome DevTools interface. The top pane shows a web page titled "What's Next" with a navigation bar (Index, Agenda, Inscription, Chat) and a main content area. The main content area features a section titled "La What's Next débarque sur mobile" with a paragraph of text and a sidebar labeled "aside" containing a box with the text "Actualités Localiser l'évènement". A tooltip indicates the sidebar's dimensions as "aside 181px x 141px".

The bottom pane shows the "Elements" tab, displaying the DOM tree. The selected element is the "aside" element, which contains a "ul" element. The "Styles" pane on the right shows the default styles for the "aside" element, including "list-style-type: none".

The DOM tree structure is as follows:

```
<!DOCTYPE html>
<html>
  <head>...</head>
  <body onload="initSSE();">
    <header>...</header>
    <nav>...</nav>
    <section id="articles">...</section>
    <div id="map">...</div>
    <aside>
      <ul>...</ul>
      <ul id="tweets" style="margin-top: 10px">...</ul>
    </aside>
    <footer>...</footer>
  </body>
</html>
```

The "Styles" pane shows the default styles for the "aside" element:

```
html, body, div, span, header, nav, section,
article, aside, footer, p, ul, li, a, h1, h2, h3 {
  list-style-type: none;
}
```

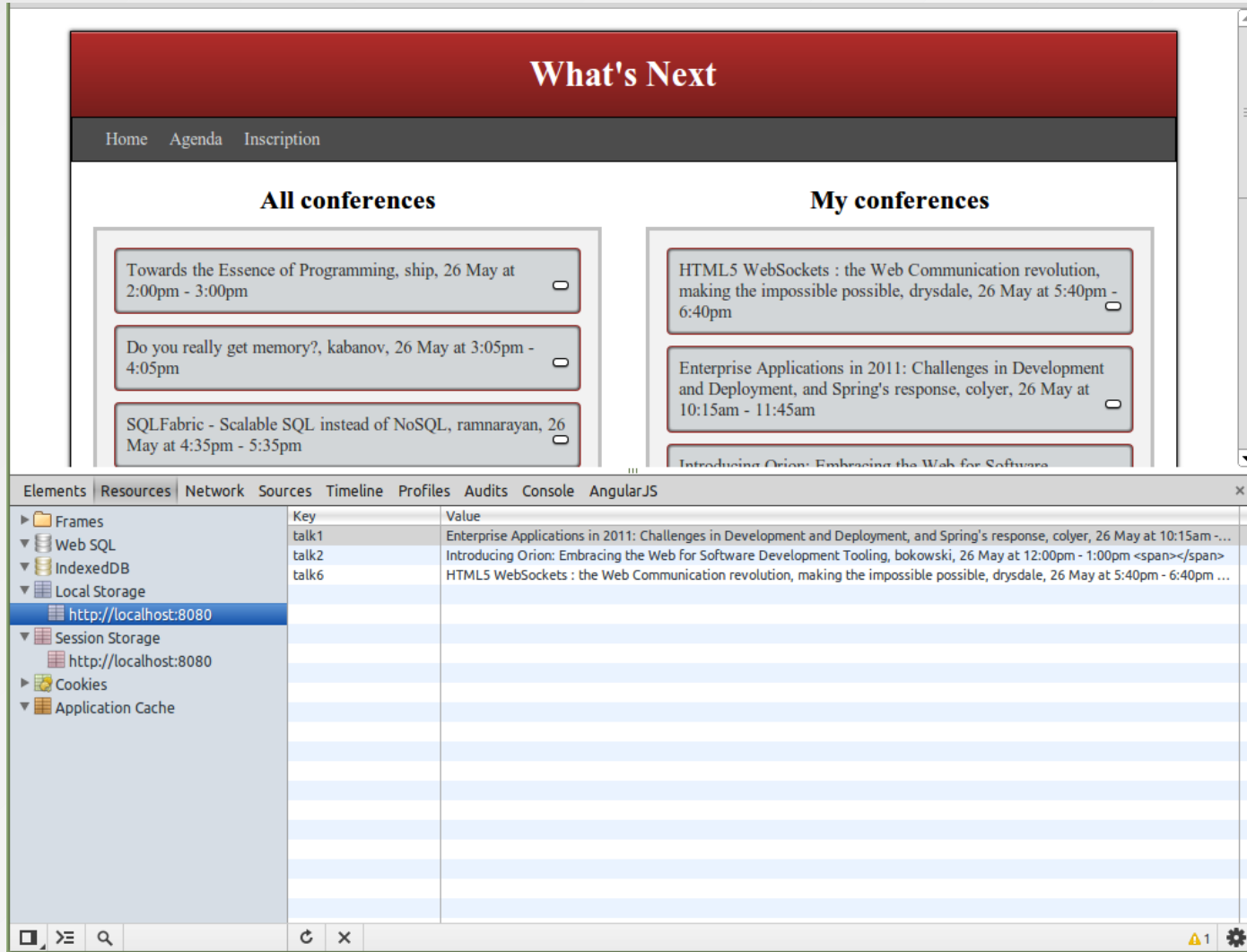
The "Computed" pane shows the default styles for the "aside" element:

```
html, body, div, span, header, nav, section,
article, aside, footer, p, ul, li, a, h1, h2, h3 {
  list-style-type: none;
}
```

The "Diagram" pane shows a visual representation of the "aside" element's box model, with dimensions of 140.347 x 101.111. The diagram includes a margin of 10, a border of 10, and a padding of 20.

- Edition des styles CSS à la volée, sauvegarde possible dans le fichier cible (source map)
- Accès à tous les override CSS, convertisseur hexa, rgb, hsl
- Métriques sur l'élément sélectionné
- Edition du DOM, déplacement d'éléments en mode DragnDrop
- Simulation de l'état (hover, focus, ...)
- Breakpoints sur subtree modification, attribute modification, node removal
- Via le sous-onglet "computed" : accès aux propriétés agrégées pour l'élément courant
- Accès aux Event Listener enregistrés sur chaque noeud

Chrome DevTools – Resources



The screenshot displays a web application titled "What's Next" with a navigation bar containing "Home", "Agenda", and "Inscription". The main content is divided into two columns: "All conferences" and "My conferences".

All conferences

- Towards the Essence of Programming, ship, 26 May at 2:00pm - 3:00pm
- Do you really get memory?, kabanov, 26 May at 3:05pm - 4:05pm
- SQLFabric - Scalable SQL instead of NoSQL, ramnarayan, 26 May at 4:35pm - 5:35pm

My conferences

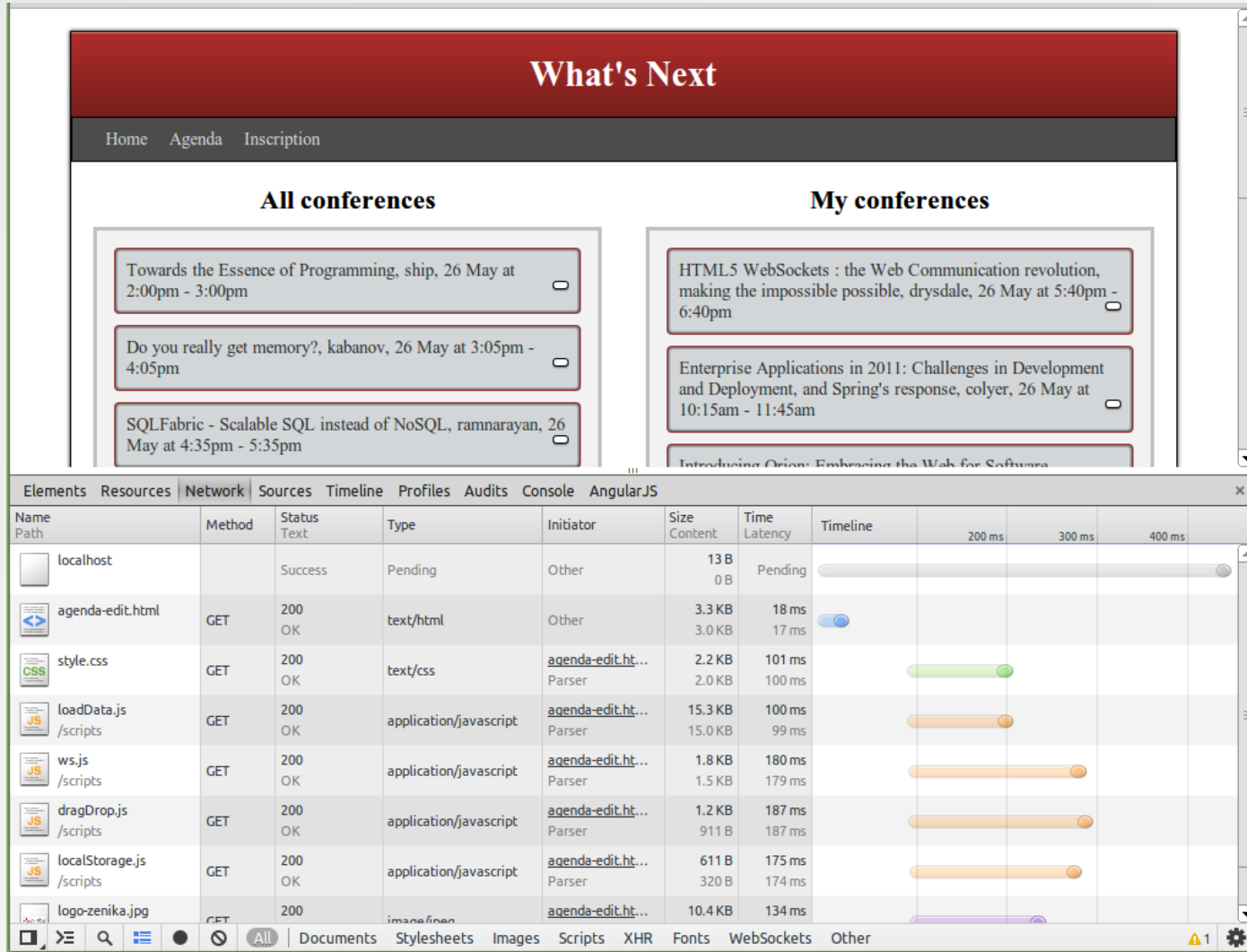
- HTML5 WebSockets : the Web Communication revolution, making the impossible possible, drysdale, 26 May at 5:40pm - 6:40pm
- Enterprise Applications in 2011: Challenges in Development and Deployment, and Spring's response, colyer, 26 May at 10:15am - 11:45am
- Introducing Orion: Embracing the Web for Software Development Tooling, bokowski, 26 May at 12:00pm - 1:00pm

The Chrome DevTools Resources panel is open, showing a tree view on the left with "http://localhost:8080" selected. The right pane displays a table of resources:

Key	Value
talk1	Enterprise Applications in 2011: Challenges in Development and Deployment, and Spring's response, colyer, 26 May at 10:15am - ...
talk2	Introducing Orion: Embracing the Web for Software Development Tooling, bokowski, 26 May at 12:00pm - 1:00pm
talk6	HTML5 WebSockets : the Web Communication revolution, making the impossible possible, drysdale, 26 May at 5:40pm - 6:40pm ...

- Frames : Ressources statiques de l'application
 - *Fichiers*
 - *Ressources*
 - *Javascript*
 - *Stylesheets*
 - *Images*
- Différents types de stockage :
 - *WebSQL Database*
 - *IndexedDB*
 - *LocalStorage*
 - *Sessionstorage*
 - *Cookies*
 - *Application Cache*
- La plupart des éléments accessibles ici sont modifiables et supprimables

Chrome DevTools – Network

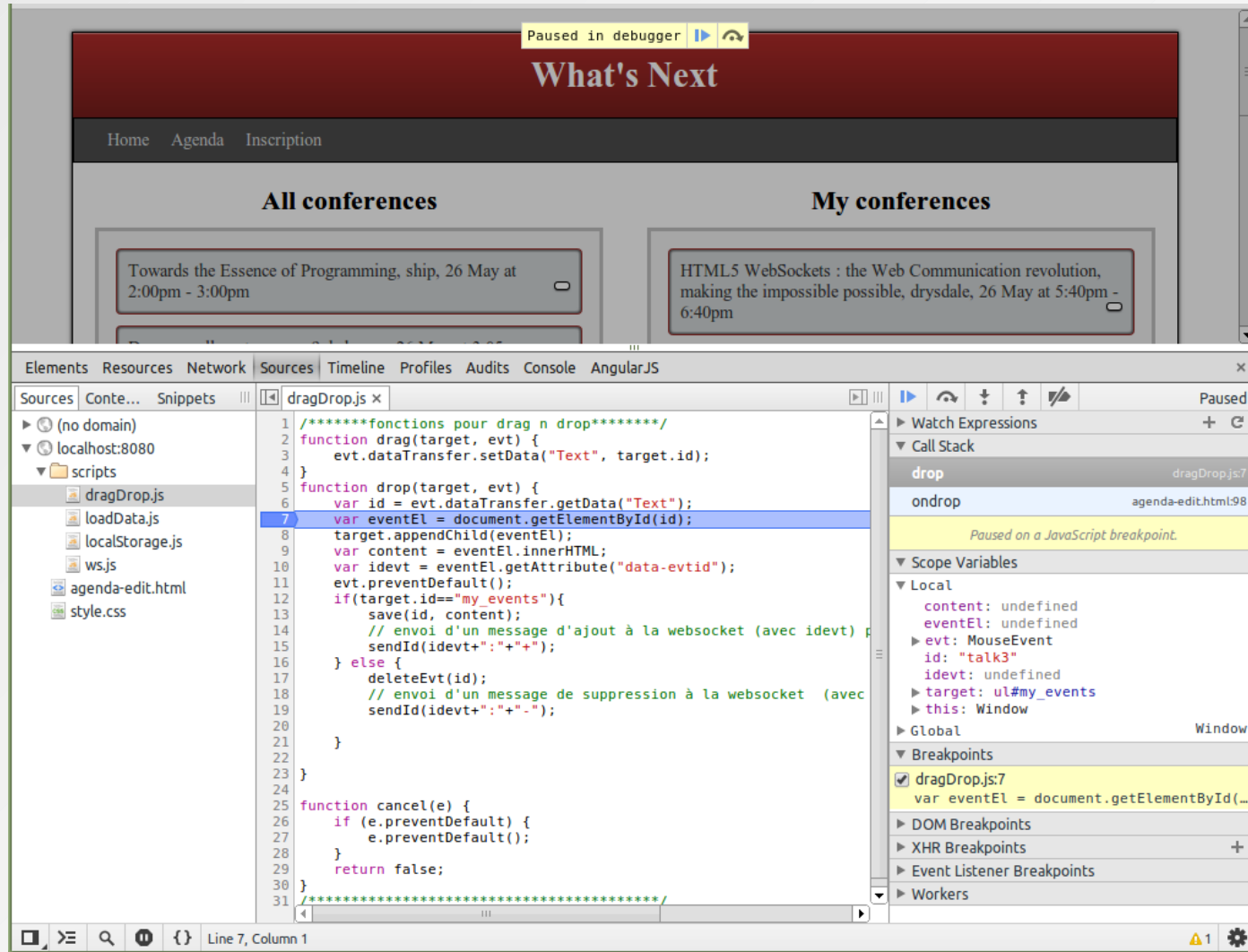


The screenshot displays the Chrome DevTools interface. The top part shows a web page titled "What's Next" with a navigation bar (Home, Agenda, Inscription) and two columns of conference listings: "All conferences" and "My conferences". The "All conferences" column lists three events, and the "My conferences" column lists two events. The bottom part of the screenshot shows the Chrome DevTools Network tab, which is currently selected. The Network tab displays a table of network requests, including the page itself and various resources like CSS, JavaScript, and images. The table columns are: Name Path, Method, Status Text, Type, Initiator, Size Content, Time Latency, and Timeline. The timeline view shows the sequence of requests and their durations. The bottom status bar indicates the current filter is "All" and shows a count of 17 items.

Name Path	Method	Status Text	Type	Initiator	Size Content	Time Latency	Timeline
localhost		Success	Pending	Other	13 B 0 B	Pending	
agenda-edit.html	GET	200 OK	text/html	Other	3.3 KB 3.0 KB	18 ms 17 ms	
style.css	GET	200 OK	text/css	agenda-edit.ht... Parser	2.2 KB 2.0 KB	101 ms 100 ms	
loadData.js /scripts	GET	200 OK	application/javascript	agenda-edit.ht... Parser	15.3 KB 15.0 KB	100 ms 99 ms	
ws.js /scripts	GET	200 OK	application/javascript	agenda-edit.ht... Parser	1.8 KB 1.5 KB	180 ms 179 ms	
dragDrop.js /scripts	GET	200 OK	application/javascript	agenda-edit.ht... Parser	1.2 KB 911 B	187 ms 187 ms	
localStorage.js /scripts	GET	200 OK	application/javascript	agenda-edit.ht... Parser	611 B 320 B	175 ms 174 ms	
logo-zenika.jpg	GET	200	image/jpeg	agenda-edit.ht...	10.4 KB	134 ms	

- Toutes les requêtes, filtrables par type
 - *Image*
 - *xhr / ajax*
 - *Scripts*
 - *Websockets*
- Détail de chacune des requêtes avec les entêtes, headers, réponse...
- Nombreuses options accessibles via clic droit
 - *Copy link*
 - *Copy as cURL*
 - *Copy as HAR*
- Les notifications de Server Sent Event ne sont pas visibles car elles sont gérées par le navigateur qui vous transmet seulement un évènement.

Chrome DevTools – Sources



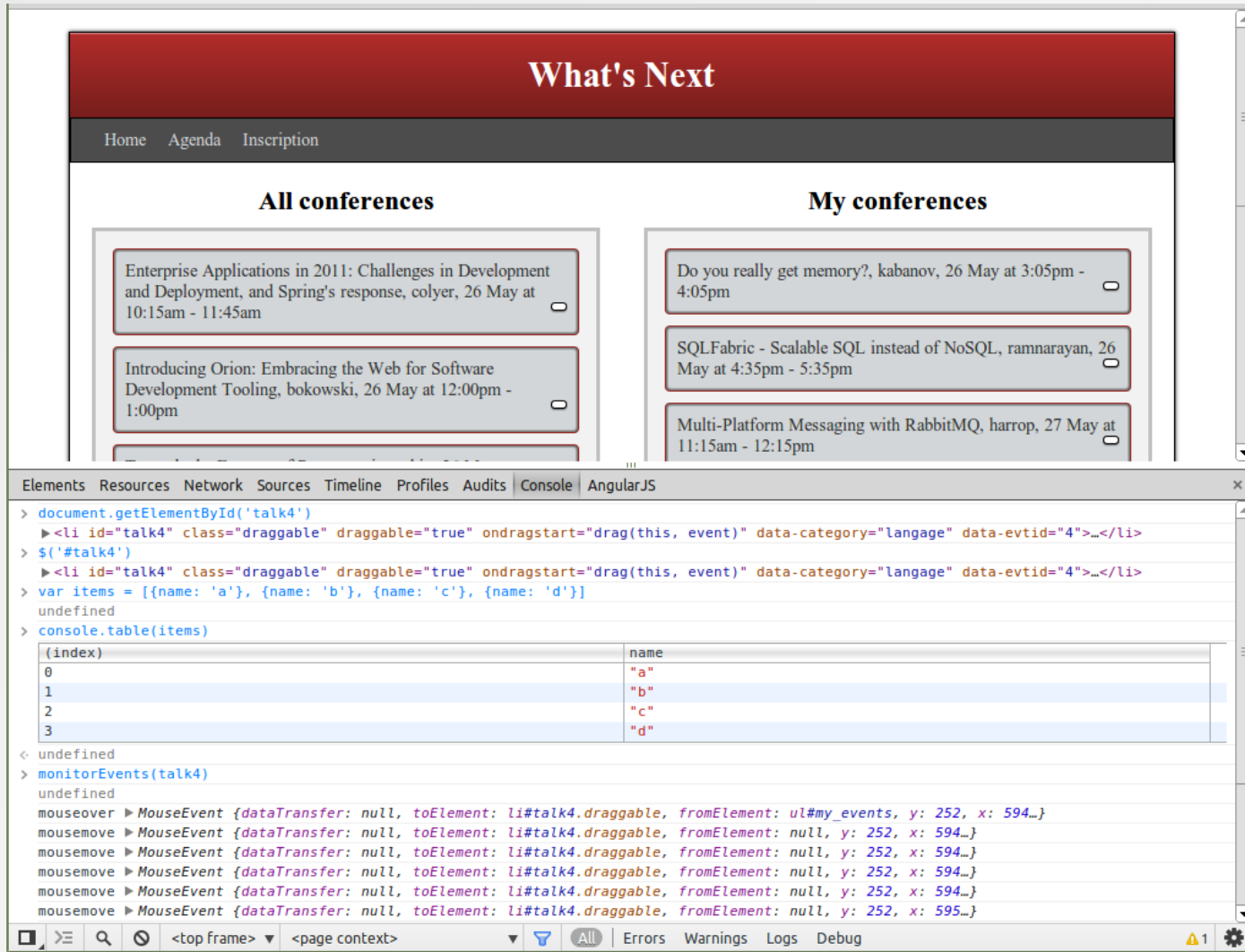
The screenshot displays the Chrome DevTools interface. The top pane shows a web application titled "What's Next" with a navigation bar (Home, Agenda, Inscription) and two main sections: "All conferences" and "My conferences". The "All conferences" section lists a conference titled "Towards the Essence of Programming, ship, 26 May at 2:00pm - 3:00pm". The "My conferences" section lists a conference titled "HTML5 WebSockets : the Web Communication revolution, making the impossible possible, drysdale, 26 May at 5:40pm - 6:40pm".

The bottom pane shows the "Sources" tab, which is paused in the debugger. The source file "dragDrop.js" is open, and the execution is paused at line 7, column 1. The code in "dragDrop.js" is as follows:

```
1 /*****fonctions pour drag n drop*****/
2 function drag(target, evt) {
3     evt.dataTransfer.setData("Text", target.id);
4 }
5 function drop(target, evt) {
6     var id = evt.dataTransfer.getData("Text");
7     var eventEl = document.getElementById(id);
8     target.appendChild(eventEl);
9     var content = eventEl.innerHTML;
10    var idevt = eventEl.getAttribute("data-evtid");
11    evt.preventDefault();
12    if(target.id=="my_events"){
13        save(id, content);
14        // envoi d'un message d'ajout à la websocket (avec idevt)
15        sendId(idevt+" "+"");
16    } else {
17        deleteEvt(id);
18        // envoi d'un message de suppression à la websocket (avec idevt)
19        sendId(idevt+" "+"-");
20    }
21 }
22
23
24
25 function cancel(e) {
26     if (e.preventDefault) {
27         e.preventDefault();
28     }
29     return false;
30 }
31 /*****fonctions pour drag n drop*****/
```

The right-hand pane shows the "Watch Expressions" and "Scope Variables" sections. The "Watch Expressions" section shows the current state of the variables: `drop` (undefined), `ondrop` (agenda-edit.html:98), and `Paused on a JavaScript breakpoint.` The "Scope Variables" section shows the local variables: `content` (undefined), `eventEl` (undefined), `evt` (MouseEvent), `id` ("talk3"), `idevt` (undefined), `target` (ul#my_events), and `this` (Window). The "Breakpoints" section shows a breakpoint set at line 7 of "dragDrop.js" with the condition `var eventEl = document.getElementById(...)`.

- Visualisation et édition (CSS, JS, HTML)
- Debugger
 - *Possibilité de placer des breakpoints*
 - *Accès au scope courant*
 - *Watch de variable*
 - *Callstack*
 - *Debugger spécifique Workers*
- Pretty print (bouton "{ }") pour formattage et déminification
- Pause on exceptions (bouton "pause")
- Historique des modifications via clic droit



What's Next

Home Agenda Inscription

All conferences

- Enterprise Applications in 2011: Challenges in Development and Deployment, and Spring's response, colyer, 26 May at 10:15am - 11:45am
- Introducing Orion: Embracing the Web for Software Development Tooling, bokowski, 26 May at 12:00pm - 1:00pm

My conferences

- Do you really get memory?, kabanov, 26 May at 3:05pm - 4:05pm
- SQLFabric - Scalable SQL instead of NoSQL, ramnarayan, 26 May at 4:35pm - 5:35pm
- Multi-Platform Messaging with RabbitMQ, harrop, 27 May at 11:15am - 12:15pm

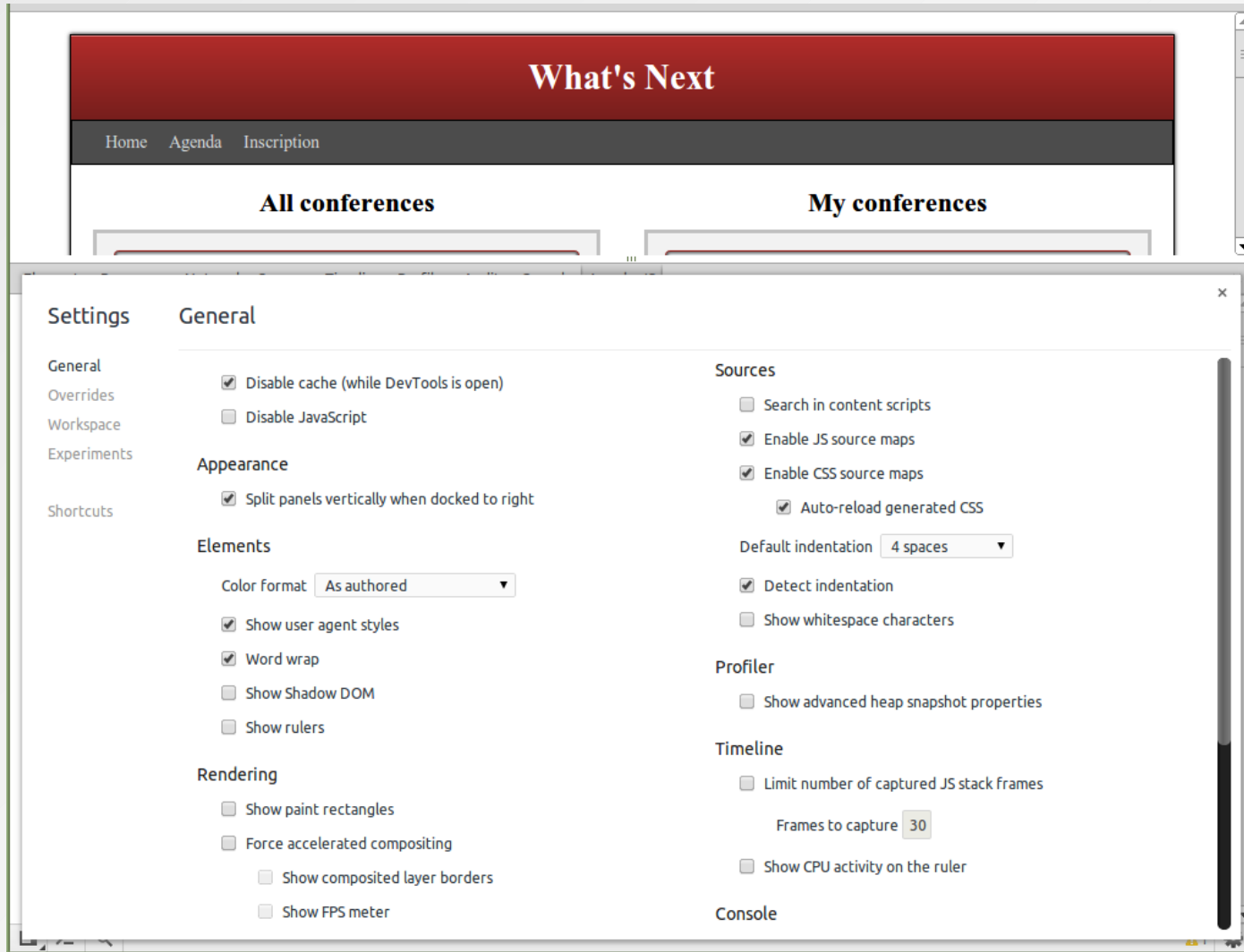
Elements Resources Network Sources Timeline Profiles Audits Console AngularJS

```
> document.getElementById('talk4')
  > <li id="talk4" class="draggable" draggable="true" ondragstart="drag(this, event)" data-category="langage" data-evtid="4">...</li>
  > $('#talk4')
  > <li id="talk4" class="draggable" draggable="true" ondragstart="drag(this, event)" data-category="langage" data-evtid="4">...</li>
  > var items = [{name: 'a'}, {name: 'b'}, {name: 'c'}, {name: 'd'}]
  undefined
  > console.table(items)
  (index) name
  0 "a"
  1 "b"
  2 "c"
  3 "d"
  < undefined
  > monitorEvents(talk4)
  undefined
  mouseover ▶ MouseEvent {dataTransfer: null, toElement: li#talk4.draggable, fromElement: ul#my_events, y: 252, x: 594...}
  mousemove ▶ MouseEvent {dataTransfer: null, toElement: li#talk4.draggable, fromElement: null, y: 252, x: 594...}
  mousemove ▶ MouseEvent {dataTransfer: null, toElement: li#talk4.draggable, fromElement: null, y: 252, x: 594...}
  mousemove ▶ MouseEvent {dataTransfer: null, toElement: li#talk4.draggable, fromElement: null, y: 252, x: 594...}
  mousemove ▶ MouseEvent {dataTransfer: null, toElement: li#talk4.draggable, fromElement: null, y: 252, x: 594...}
  mousemove ▶ MouseEvent {dataTransfer: null, toElement: li#talk4.draggable, fromElement: null, y: 252, x: 595...}
```

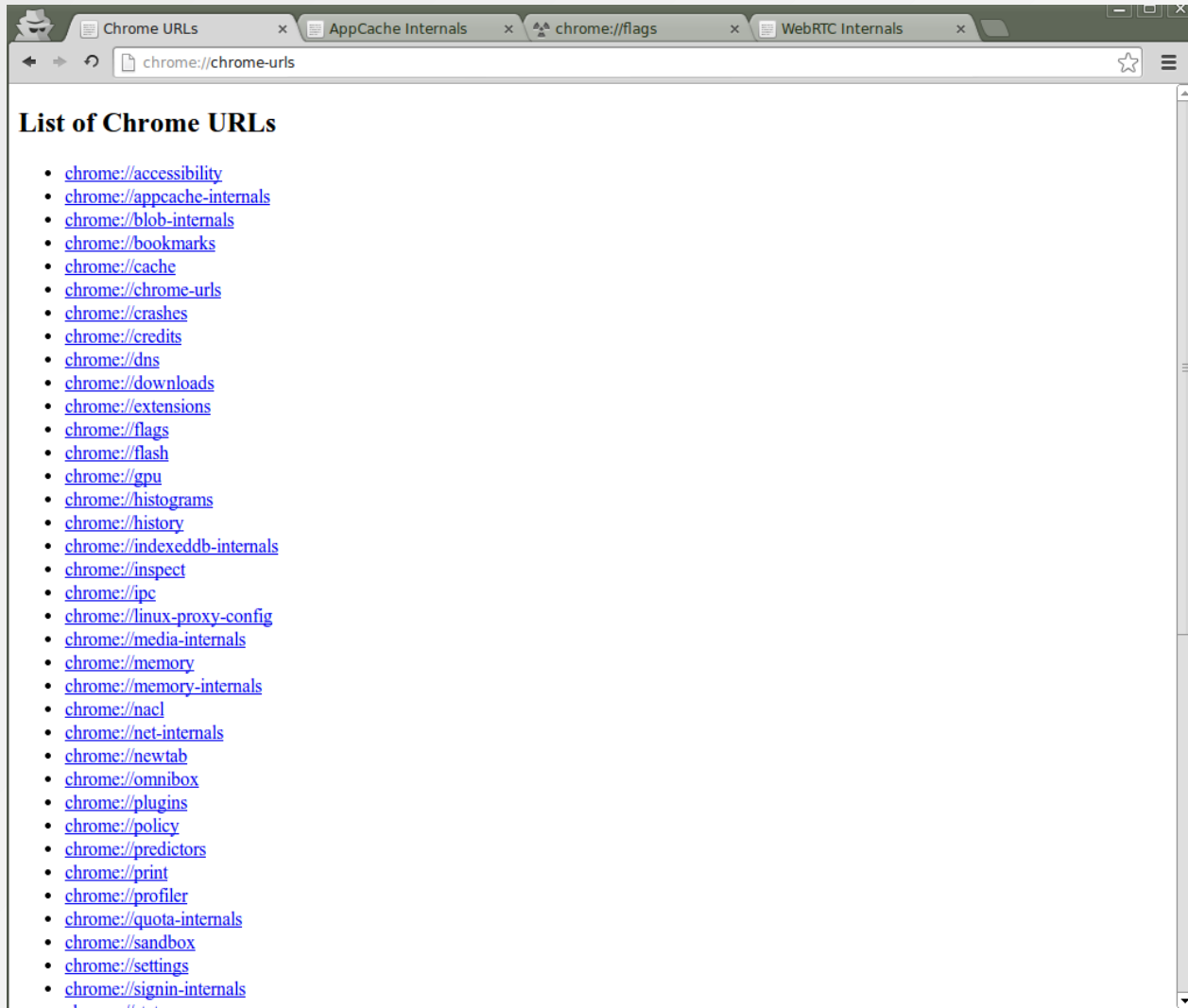
< top frame > < page context > All Errors Warnings Logs Debug 1

- Interprétation / exécution de code dans le scope courant (si script en pause, utilisation du contexte de l'endroit en pause)
- Visionnage d'un objet du scope et de toutes ses propriétés
- Sélection d'éléments avec `$('#element')` comme du jQuery (selecteur Bling). Possibilité d'accéder aux éléments précédemment sélectionnés via `$0`, `$1`, `$2...`
- Monitorer les événements sur un élément : `monitorEvents($('#element'))`
- `console.table(array)` permet de faire afficher un array de manière plus lisible

Chrome DevTools – General



- Raccourcis claviers
 - Experiments si activées dans about:flags
 - Désactivation du cache
 - Géoloc fictive
 - Source mapping
 - Emulation mobile
 - ...
-
- <http://anti-code.com/devtools-cheatsheet/>
 - <https://developers.google.com/chrome-developer-tools/>



- `chrome://chrome-urls/`
- Extension : Google Chrome Service Pages
 - `chrome://appcache-internals`
 - `chrome://flags/`
 - `chrome://indexeddbs-internals/`
 - `chrome://inspect`
 - `chrome://webRTC-internals`

