



FORMOSA

Javascript

" Cycle Développeur "

Dans ce document

Le plan de cette formation :

1. Introduction
2. Le langage Javascript
3. Environnement de débogage
4. Les Objets
5. Modification d'une page dynamiquement et événements
6. Annexes : Ajax



Introduction

" Cycle Développeur "

Présentation de JavaScript



Attention JavaScript est différent de Java !

Le **Java** est un langage de programmation serveur ou pour la création d'application de Bureau.

Javascript est orienté navigateur. Il s'exécute récemment coté serveur avec NodeJS

Coté navigateur :

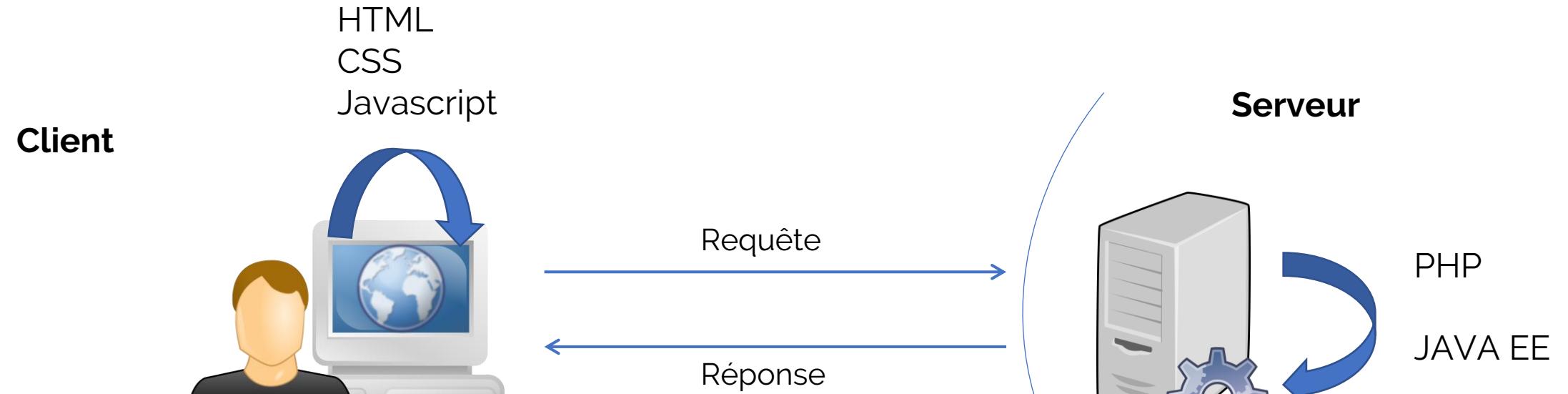
- Il permet de modifier dynamiquement des pages Web sans avoir besoin de recharger la page
- Permet de communiquer en tâche de fond avec des web services

Le javascript utilise donc les ressources du Client vu qu'il s'exécute sur le poste du client

Documentation de Référence : <https://www.w3schools.com/>

Javascript est en train de devenir le seul langage de programmation (hors HTML et CSS) qui s'exécute coté navigateur. Le Flash et le Java sur le navigateur sont en train disparaître

Présentation Architecture Client/Serveur



Editeurs



Pour développer en JavaScript nous avons besoin :

- D'un éditeur
- D'un navigateur

Ces outils permettent de programmer directement en ligne :

- JSFiddle <https://jsfiddle.net/>
- Codepen.io <http://codepen.io/>
- JSBin <http://jsbin.com/?js,output>
- Liveweave <http://liveweave.com/>
- Brackets.io

Télécharger Brackets IO



Pour télécharger

Brackets IO il faut aller
sur :

www.brackets.io

A modern, open source text editor that
understands web design.

[Download Brackets 1.10](#)

Other Downloads

Pour brackets.io

installer les plugins
jshint et beautify

```
index.html — Brackets
103 <div id="hero-wrapper">
104   <div id="hero" class="row">
105     <div class="large-12 columns">
106       <h1>Brackets is an open source code editor for web designers and front-end developers. <a class="not
107         <div id="download">
108           <a id="hero-cta-button" href="https://github.com/adobe/brackets/releases/latest" class="large rd
109             <span id="download-version" class="nowrap" data-i18n="index.page.hero.latest-release">Latest
110               </a>
111             <div id="os-alert" class="alert-box radius" data-alert>
112               <span>OS Alert</span>
113             </div>
114           </a>
115         </div>
116       </h1>
117       <div class="large-12 columns">
118         <div class="row">
119           <div class="large-12 columns">
120             <div class="row">
121               <div class="large-12 columns">
122                 <div class="row">
123                   <div class="large-12 columns">
124                     <div class="row">
125                       <div class="large-12 columns">
126                         <div class="row">
127                           <div class="large-12 columns">
128                             <div class="row">
129                               <div class="large-12 columns">
130                                 <div class="row">
131                                   <div class="large-12 columns">
132                                     <div class="row">
133                                       <div class="large-12 columns">
134                                         <div class="row">
135                                           <div class="large-12 columns">
136                                             <div class="row">
137                                               <div class="large-12 columns">
138                                                 <div class="row">
139                                                   <div class="large-12 columns">
140                                                     <div class="row">
141                                                       <div class="large-12 columns">
142                                                         <div class="row">
143               </div>
144             </div>
145           </div>
146         </div>
147       </div>
148     </div>
149   </div>
150 </div>
```

Navigateurs

Vous pouvez utiliser

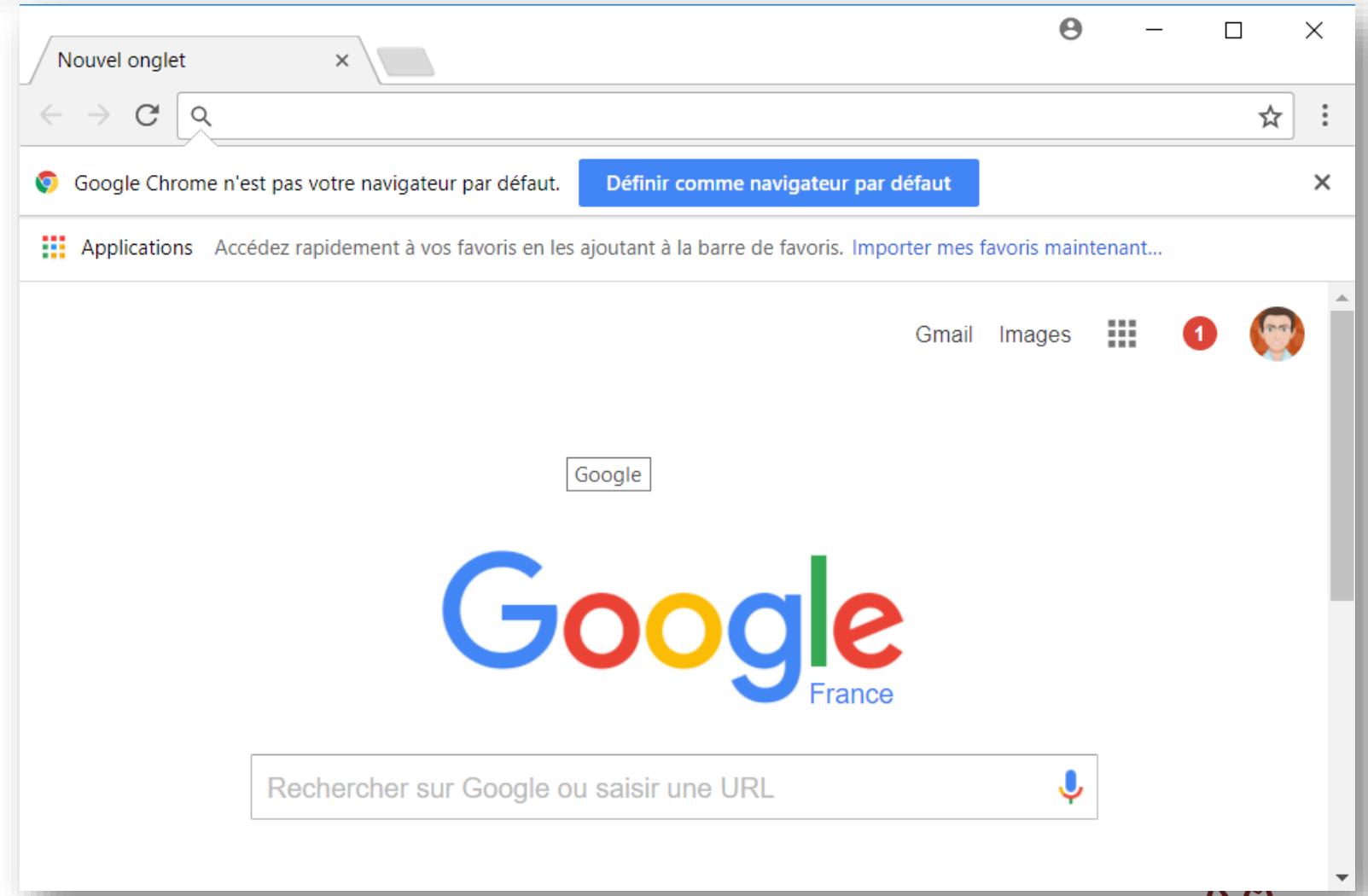
n'importe quel

navigateur

Chrome, Firefox etc.

Dans ces slides nous

utilisons Chrome



Editeurs



Les éditeurs hors ligne nous avons :

- Notepad++
- UltraEdit (sous licence)
- WebStorm pour 30 jours d'essais
- Visual Code
- ATOM



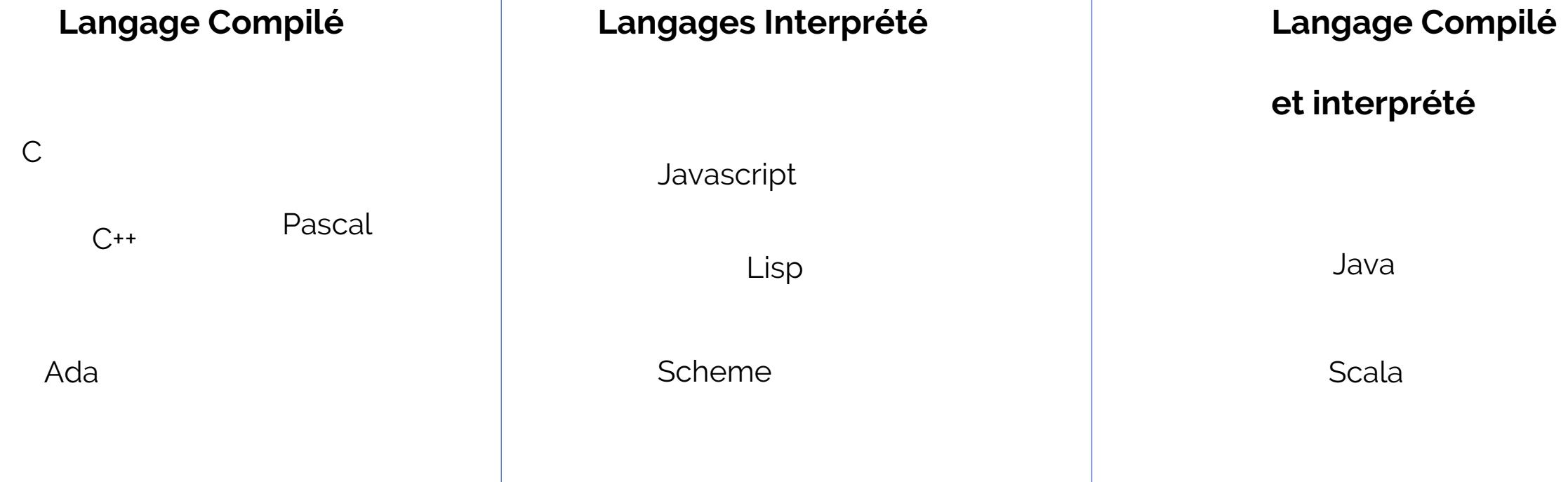
Langage

" Cycle Développeur "

Syntaxe du Langage



Le langage Javascript est un langage dit **interprété**



Syntaxe du Langage - Commentaires



Les commentaires :

Plusieurs lignes /* */

```
/*
 * Exemple de Commentaire sur
 * plusieurs
 * lignes
 */
```

Commentaire sur une ligne //

```
// Exemple de commentaires sur une lignes

// Second commentaire
```

Syntaxe du Langage - Commentaires



Les instructions et les blocs d'instructions :

```
{  
    console.log("Instruction1");  
    console.log("Instruction2");  
    console.log("Instruction3");  
}
```

```
{  
    {  
        //Exemple de bloc d'instruction  
    }  
  
    {  
        //Exemple pour un second bloc d'instructions  
    }  
}
```

Démarrage Rapide



The screenshot shows a browser window with two tabs: "abc - Recherche Google" and "PREMIERS PAS AVEC JS". The active tab displays the file path "file:///D:/Documents/formation/premier.html". The browser's developer tools are open, with the "Console" tab selected. The console output shows the word "Hello" followed by the file path "premier.html:13". Below the console, the HTML code of the page is visible, including the DOCTYPE declaration, HTML, head, meta, title, and body sections, along with a script tag that logs "Hello" to the console.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>PREMIERS PAS AVEC JS</title>
  </head>
  <body>
  </body>
  <script type="text/javascript">
    console.log("Hello");
  </script>
</html>
```

Syntaxe du Langage - Les Variables



Comme vu précédemment le langage Javascript est un langage faiblement typé. Ainsi une variable est exprimée par le mot clé "var".

```
var maVariable;
```

Une variable peut contenir des entiers, booléens, chaînes de caractères et des objets.

Syntaxe du Langage - Les Variables

Les variables :

```
//Les entiers  
var v1 = 5;
```

Déclaration d'une variable

```
//Les flottant  
var v2 = 5.5;
```

```
//Les chaines de caractère  
var v3 = "Voici une \"chaine\" ";  
v3 = 'Voici une \'chaine\' ';
```

Réutilisation d'une variable

```
//Les Booléens  
var v5 = true;  
var v6 = false;
```

Booléens

Syntaxe du Langage - Les Variables



Les variables :

```
//Les entiers  
var v1 = 5;  
  
//Les flottant  
var v2 = 5.5;  
  
//Les chaines de caractère  
var v3 = "Voici une \"chaine\" ";  
v3 = 'Voici une \'chaine\' ';  
  
//Les Booléens  
var v5 = true;  
var v6 = false;
```

Mémoire Statique

v1 5

v2 5.5

v3 α

v5 true

v6 false

Mémoire Dynamique

α
Voici une 'chaine'

Les variables

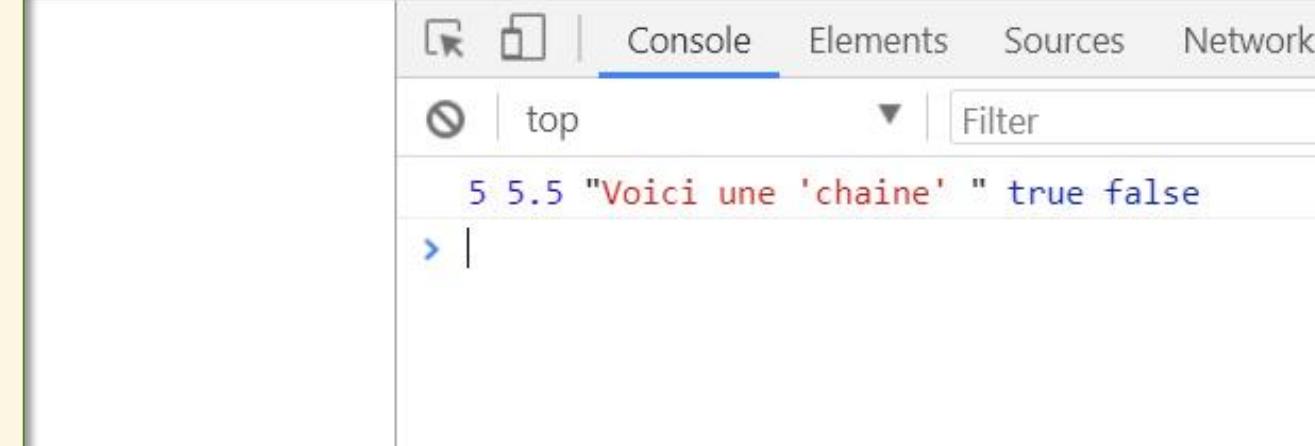
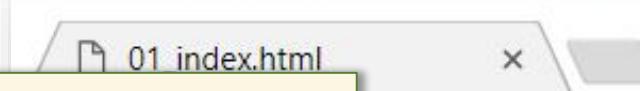


```
<script type="text/javascript">
    //Les entiers
    var v1 = 5;

    //Les flottant
    var v2 = 5.5;

    //Les chaines de caractère
    var v3 = "Voici une \"chaine\" ";
    v3 = 'Voici une \'chaine\' ';

    //Les Booléen
    var v5 = true;
    var v6 = false;
    console.log(v1,v2,v3,v5,v6);
</script>
```



Syntaxe du Langage - TypeOf



Les variables peuvent être réutilisées et peuvent contenir différents type, la fonction **typeof** permet de connaître le type

```
var v1 = 5;                                //Affiche : Number
console.log(typeof(v1));
v1 = 5.5;                                    //Affiche : Number
console.log(typeof(v1));
v1 = "Voici une \"chaine\" ";
console.log(typeof(v1));                      //Affiche : String
v1 = 'Voici une \'chaine\' ';
console.log(typeof(v1));                      //Affiche : String
v1 = true;                                     //Affiche : Boolean
console.log(typeof(v1));
v1 = false;                                    //Affiche : Boolean
console.log(typeof(v1));
```

Opérateur Arithmétique



Les opérateurs arithmétiques :

- + Addition
- Soustraction
- * Multiplication
- / Division
- % Modulo
- ++ Incrémentation
- Décrémantation

```
//OPERATEURS ARITHMETIQUES
var a = 5;
var b = 6;

//Multiplication
var c = a*b;    console.log(c);          //30
//Division
c = a/b;        console.log(c);          //0.83334
//Addition
c = a+b;        console.log(c);          //11
//Soustraction
c = a-b;        console.log(c);          //-1
//Modulo
c = b%a;        console.log(c);          //1
//incrémentation équivalent a = a+1
a++;            console.log(a);          //6
//incrémentation équivalent a = a-1
a--;            console.log(a);          //5
```

Opérateur Logiques



La comparaison de valeurs logique :

>	Plus grand
>=	Plus grand ou égal
<	Plus petit
<=	Plus petit ou égal
==	Pour l'égalité
!=	Pour la différence
&&	Pour le et logique
	Pour le ou logique

//OPERATEURS LOGIQUE et COMPARAISON

```
var a = 18;
var b = 5;

//Supérieur
var c = a>b;           console.log(c);      //true

//Inférieur
c = a<b;               console.log(c);      //false

//Supérieur ou égal
c = (a>=b);            console.log(c);      //true

//Inferieur ou égal
c = (a<=b);            console.log(c);      //false

//Égal
c = (a==b);             console.log(c);      //false

//Différent
c = (a!=b);             console.log(c);      //true

//négation
c = !(a==b);            console.log(c);      //true
```

Opérateur Logiques

La comparaison de valeurs

logique :

- > Plus grand
- >= Plus grand ou égal
- < Plus petit
- <= Plus petit ou égal
- == Pour l'égalité
- != Pour la différence
- && Pour le et logique
- || Pour le ou logique

```
//ET logique : age > 18 et taille<180
var age = 19;
var taille = 190;
var resultat = (age>18 && taille<180);
console.log("RESULTAT AGE ET TAILLE :" +resultat);
```

```
//OU logique ||
resultat = (age>18 || taille<180);
console.log("RESULTAT AGE ET TAILLE :" +resultat);
```

Des popup

Certains affichages par popup sont utiles pour le débogage ou la saisie de valeurs. Attention ces fenêtres sont bloquantes pour le code sous-jacent

Par exemple :

```
alert("Bonjour je suis une alerte");
```

```
var chiffre = prompt("Saisissez un chiffre");
```



Structures conditionnelles

Une structure conditionnelle se compose d'une condition, si la condition vaut vrai alors on exécute le premier bloc sinon le second.

```
var nom = prompt("Comment tu t'appelles ?");
var ageStr = prompt("Quel age as tu ?");
var age = parseInt(ageStr);

if(age>=18){
    alert("Tu es majeur");
}
else {
    alert("Tu es mineur");
}

alert("Au revoir");
```

Bien sûr le bloc **else** est facultatif

Structures conditionnelles - Imbrication

```
var nom = prompt("Comment tu t'appelles ?");
var ageStr = prompt("Quel age as tu ?");
var age = parseInt(ageStr);

if(age>=18)  {

    if (age>30){
        alert("Tu es trentenaire");
    }

    alert("Tu es majeur");
}

else{
    alert("Tu es mineur");
}

alert("Au revoir");
```

Il est bien sur possible d'imbriquer les structures conditionnelles entre elles

Les Boucles



Le boucles sont très utiles il existe trois types de boucles en JS :

- Le while
- Le For
- Le For Each
- Le Do while

Le **while** est une boucle dépendant d'une condition.

Le **Do While** respecte le même principe mais la condition est testée après une première exécution

Le **For** parcours un ensemble de valeurs

Le For Each parcours des tableaux et des listes, nous le verrons plus tard.

Le While

Le **while** va exécuter le bloc associé tant que la condition vaut vrai, attention au boucle infinies :

La syntaxe :

```
while (CONDITION) {  
}  
}
```

Instruction de
passage à la boucle
suivante

```
var ageStr = prompt("Quel age as tu ?");  
var age = parseInt(ageStr);  
  
while (age<18){  
    alert("Tu es mineur ! ");  
    ageStr = prompt("Quel age as tu ?");  
    age = parseInt(ageStr);  
}
```

Le do ... While



Le do while va exécuter le bloc associé en premier puis tester la condition

La syntaxe :

```
do {  
...  
} while (CONDITION);
```

```
var nb = 50;  
var chiffre, chiffreStr;  
  
do{  
    chiffreStr = prompt("Saisissez un chiffre");  
    chiffre = parseInt(chiffreStr);  
}while (chiffre!=nb)  
  
console.log("Bravo");
```

Le For est composé de 3 parties : initialisation, condition, incrémentation

FOR (INITIALISATION ; CONDITION ; INCREMENTATION ou DECREMENTATION){

...

}

```
for ( var i=0 ; i<10 ; i++ ){
    console.log("Affiche "+i);
}
```

Affiche 0	premier.html:22
Affiche 1	premier.html:22
Affiche 2	premier.html:22
Affiche 3	premier.html:22
Affiche 4	premier.html:22
Affiche 5	premier.html:22
Affiche 6	premier.html:22
Affiche 7	premier.html:22
Affiche 8	premier.html:22
Affiche 9	premier.html:22

TP1 : Devinette



Nous souhaitons créer un jeux de devinette :

- 1, Le programme demande de saisir un chiffre entre 0 et 100 (ne vous préoccupez pas des cas où l'utilisateur ne saisit pas bien le chiffre nous le traiterons plus tard)
- 2, Tant que le chiffre entré est différents du chiffre choisi par l'ordinateur, l'ordinateur redemande à l'utilisateur, il laisse cependant une indication à l'utilisateur. Il lui dit si le nombre est plus grand ou plus petit.*
- 3, Quand le nombre est trouvé, le programme s'arrête et indique à l'utilisateur en combien de coup il a trouvé

TP1 : Devinette (Correction)



```
var chiffre = 78;
var choix = parseInt(prompt("Saisissez un chiffre stp"));
var coup = 0;
while (choix != chiffre)
{
    if (choix < chiffre){
        alert("C'est plus grand");
    }
    else {
        alert("C'est plus petit");
    }
    coup = coup + 1;
    choix = parseInt(prompt("Saisissez un chiffre stp"));
}
alert("Bravo");
alert(coup);
```

Les Tableaux

FOR**MOSA**

Des tableaux sont des ensembles de valeurs indexées de 0 à n

```
var monTableau = new Array()
```

```
var monTableau = [];
```

```
var monTableau = [25,46,89];
```

```
var monTableau = [25, "Salut", true]
```

Le tableau index les objets de 0 à n

monTableau[0] nous donne 25

`monTableau[1]` nous donne l'objet "Salut"

`monTableau.length` nous donne 3

Les Tableaux



monTableau.push("Yo");	nous donne [25, "Salut", true, "yo"]
monTableau[1];	nous donne l'objet "Salut"
monTableau.length;	nous donne 4
monTableau.pop();	nous donne [25, "Salut", true]

Un tableau peut bien entendu stocker des objets

Les Tableaux



Parcourir un tableau

```
for (i=0;i<monTableau.length;i++){  
    alert(monTableau[i]);  
}
```

Parcourir un tableau

```
for (i in monTableau){  
    alert(monTableau[i]);  
}
```



Mémoire Statique et Dynamique



Un tableau est un objet et comme tout objet il est créé en mémoire dynamique.

```
var tab = new Array();
tab.push(15);
tab.push(20);
```

Mémoire Statique



Mémoire Dynamique

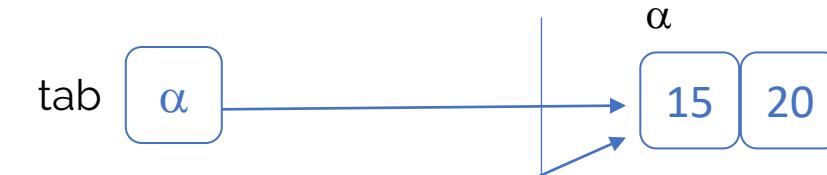
Mémoire Statique et Dynamique

Un tableau est un objet et comme tout objet il est créé en mémoire dynamique.

```
var tab = new Array();
tab.push(15);
tab.push(20);

var tab2 = tab;
```

Mémoire Statique



Mémoire Dynamique

Exercice sur les Tableau



A partir d'un tableau d'entier :

1. afficher l'entier maximum du tableau
2. afficher l'entier minimum du tableau
3. Trier le tableau

Les fonctions JS



Les fonctions nous permettent de factoriser du code, c'est-à-dire de réutiliser un bout de code.

Une fonction possède des arguments permettant de paramétrer le comportement de celle-ci.

Une fonction peut éventuellement retourner un résultat.

La Syntaxe la déclaration:

```
function maFonction (argument1, argument2, ... argumentN){  
    ...  
}
```

La Syntaxe pour l'appel :

```
maFonction(valeur1,valeur2,valeur3)
```

Les fonctions JS



Faisons une fonction retournant vrai si un nombre est pair et faux sinon :

```
function pair(nombre){  
    return nombre%2==0  
}  
  
var chiffre = prompt("S'il vous plait saisissez un chiffre");  
if (pair(chiffre)) console.log(chiffre+" est pair");  
else console.log(chiffre+" est impair");
```

Un code doit être DRY :

- Don't Repeat Yourself

Portée des Variables



Dans une fonction les variables créées ne sont pas disponibles en dehors de la fonction :

Les variables présentent en dehors de la fonction sont connues dans la fonction.

```
var externe=20;
maFonction(50,51);
console.log(externe);
console.log(locale);

function maFonction(argument1,argument2){
    console.log(argument1,argument2);
    console.log(externe);
    externe=25;
    var locale = 2;
    console.log(locale);
}
```

```
50 51
20
2
25
✖ ► Uncaught ReferenceError: locale is not defined
      at premier.html:32
premier.html:35
premier.html:36
premier.html:39
premier.html:31
premier.html:32
```

Hautement Dynamique



L'appel aux fonctions est fortement dynamique, le nombre d'arguments importe peu. Par exemple :

```
addition(1);
addition(1,2);
addition(1,2,3);
addition(1,2,3,4);

function addition(a1,a2,a3){
    console.log(a1,a2,a3);
}
```

```
1 undefined undefined
1 2 undefined
1 2 3
1 2 3
```

Hautement Dynamique



L'appel aux fonctions est fortement dynamique, le nombre d'arguments importe peu. Par exemple :

```
addition(1);
addition(1,2);
addition(1,2,3);
addition(1,2,3,4);

function addition(a1,a2,a3){
    console.log(a1,a2,a3);
}
```

```
1 undefined undefined
1 2 undefined
1 2 3
1 2 3
```

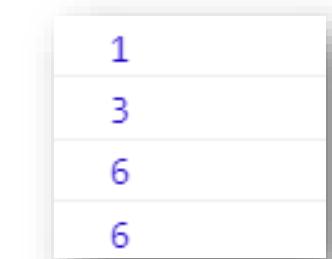
Hautement Dynamique

L'appel aux fonctions est fortement dynamique, le nombre d'arguments importe peu. Par exemple :

```
console.log(addition(1));
console.log(addition(1,2));
console.log(addition(1,2,3));
console.log(addition(1,2,3,4));

function addition(a1,a2,a3){
    var sum = 0;
    if (a1!=undefined) {      sum = sum + a1; }
    if (a2!=undefined) {      sum = sum + a2; }
    if (a3!=undefined) {      sum = sum + a3; }

    return sum;
}
```



1
3
6
10

Hautement Dynamique



Comment accéder à ce dernier paramètre ? Il existe un paramètre dynamique groupant tous les arguments. Cette argument utilise des notions pas encore abordées nous l'étudierons plus tard.

```
console.log(addition(1,2,3,4));  
  
function addition(a1,a2,a3){  
    console.log(arguments);  
    var sum = 0;  
    if (a1!=undefined) { sum = sum + a1; }  
    if (a2!=undefined) { sum = sum + a2; }  
    if (a3!=undefined) { sum = sum + a3; }  
  
    return sum;  
}
```

```
▶ (4) [1, 2, 3, 4, callee: f, Symbol(Symbol.iterator): f]  
6
```

Fonctions



Créer des fonctions :

1. Créer une fonction inverse, qui inverse les valeurs d'un tableau. La fonction modifie le tableau passé en paramètre et ne retourne aucun résultat
2. Créer une fonction inverse, qui inverse les valeurs d'un tableau et qui retourne un nouveau tableau
3. Retournant la valeur maximum entre 2 entiers, 3 entiers ou N entier
4. Retournant la valeur minimum entre 2 entiers, 3 entiers ou N entier

Les Fonctions en tant que valeurs



La propriété la plus intéressante avec le Javascript est que les fonctions sont aussi des valeurs. On peut donc les stocker dans des variables.

```
var f = function(a,b){ return a+b;};  
console.log(f);  
console.log(typeof f);  
  
var c = f(5,10)  
console.log(c);  
  
var f = 4;  
console.log(f);  
f(5,10);
```

```
f (a,b){ return a+b;}  
function  
15  
4  
✖ ► Uncaught TypeError: f is not a function  
at 25_fonction.html:16
```

Les Fonctions en tant que valeurs



Lorsque l'on déclare une fonction c'est donc équivalent à créer une variable avec comme nom le nom de la fonction :

```
function addition (a,b){  
    return a+b;  
};  
  
console.log(addition);  
console.log(addition(10,15));  
  
addition = 10;  
  
console.log(addition(10,15));
```

```
f addition(a,b){  
    return a+b;  
}
```

25

✖ **Uncaught TypeError: addition is not a function
at 25_fonction.html:16**

Faire une fonction



Faire une fonction qui va parcourir un tableau et qui applique un traitement sur chaque case du tableau :

- fonction **forEach(tab, fct)**

1. Par exemple à partir de forEach faire une fonction qui met en majuscule les chaînes de caractères d'un tableau
2. Faire une fonction qui calcule la racine carré d'un tableau d'entier

Fonction Eval



La fonction **eval** est une fonction permettant l'évaluation d'un bout de code javascript en Javascript :

Par exemple :

```
var a = eval("5+10");
```

"a" prendra la valeur de 15.

La fonction **eval** est puissante et permet aussi la définition de fonction :

```
eval("function add(a,b){return a+b;};");
```

Créera une fonction add, accessible après l'appel de la fonction eval

Variable locale et globales



Les conventions de nommage

nom de variable et de fonction

nomDeVariable, nomDeFonction

nom de Classes

NomDeClasse

La déclaration d'une variable locale ce fait comme cela :

```
var nomVariable;
```

Ou

```
var nomVariable=15;
```

Si on enlève le mot clé var, la variable devient alors globale

Différence Let et Var



Deux mots clés existent pour la définition des variables. Le mot clé **let** et le mot clé **var**.

Le mot clé **var** rend disponible la variable même après le bloc d'instruction.

Le mot clé **let** quand à lui supprime la variable après le bloc d'instruction

```
{  
    var toto = "Salut de toto";  
    let tata = "Salut de tata";  
}  
console.log(toto);  
console.log(tata);
```

Salut de toto

24_portee2.html:11

✖ Uncaught ReferenceError: tata is not defined
at 24_portee2.html:12

24_portee2.html:12

>

La Gestion des Erreurs



Il y a différents types d'erreurs en JavaScript :

- une Erreur de syntaxe
- une Erreur sémantique

Pour des erreurs de syntaxe nous ne pouvons rien faire, en revanche les erreurs de sémantiques peuvent être capturées par un bloc try catch :

```
try{  
    ...  
} catch (variable) { ... }
```

La Gestion des Erreurs



Voici un exemple de gestion d'erreurs ici :

```
try{
    yoyo
    window.alert("bonjour");
}catch (error){
    console.log("Une erreur a été soulevée");
}
console.log("Je suis passé");
```

Une erreur a été soulevée
Je suis passé

Lancer vos erreurs



Il est possible au développeur de générer des erreurs volontairement par exemple si une données saisie n'est pas valide nous passerons alors par l'instruction **throw**

```
throw ("Mon Erreur")
```

```
if (chiffre==50){  
    throw ("C'est horrible le chiffre = 50");  
}  
console.log("Salut je suis passé ?");
```

✖ ► Uncaught C'est horrible le chiffre = 50

[premier.html:23](#)

Switch

Le switch case est un instruction permettant d'exécuter différents block d'instruction en fonction de la valeur d'une expression :

```
switch(expression) {  
    case n:  
        CODE  
        break;  
    case n:  
        CODE  
        break;  
    default:  
        CODE  
}
```

```
switch (jour) {  
    default:  
        text = "Jour de la semaine";  
        break;  
    case 6:  
        text = "Le Samedi";  
        break;  
    case 0:  
        text = "Le Dimanche";  
}
```

L'instruction break stoppe l'exécution de l'évaluation de la suite du Switch case

Environnements de Débogage

" Cycle Développeur "



Chrome Dev Tools

Les navigateurs intègrent des outils pour le développement d'application web.

Le navigateur Chrome, intègre le Chrome Dev Tools en appuyant sur F12 vous obtenez :

- Un inspecteur d'éléments
 - Une émulateur de téléphone portable
 - La console web
 - Le DOM HTML
 - Les fichiers sources chargés (incluant le débogueur avec l'exécution pas à pas)
 - L'inspecteur des échanges réseaux
 - La partie performance
 - La partie Mémoire
 - Les parties Application, Sécurité et Audit

Chrome Dev Tools



The screenshot shows the Chrome Dev Tools interface with the Network tab selected. The main area displays a timeline of network requests, with a red vertical line indicating the point where the screenshot was taken. Below the timeline is a table of request details.

Name	Status	Type	Initiator	Size	Time	Waterfall
triangle-grey-sm.png	200	png	gerer-votre-relati...	(from dis...	121 ms	
collect?v=1&_v=j66&a=284449257...	200	gif	analytics.js:2	321 B	121 ms	
?wamid=1362&Wvar=%7B%22Co...	204	text/plain	wamfactory dpm...	40 B	360 ms	
comptage_wreport.fcgi?WRP_ID=4...	200	gif	wreport wcm.js:1...	772 B	869 ms	
?lid=115&payload=%7B%22source...	200	gif	sdk.js:74	196 B	212 ms	
datacollect.php?chID=5714a31785...	200	gif	5714a31...js:2	387 B	269 ms	
hit.xiti?s=547787&idclient=151486...	200	qif	VM190:30	258 B	291 ms	

Annotations in the image:

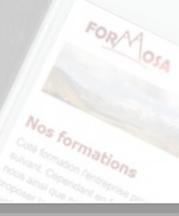
- Affichage du DOM HTML
- Code source des pages
- Affichage des échanges réseaux
- Affichage des performances
- Affichage de l'application
- Console web affichant les logs
- Inspecteur d'éléments
- Activation de la vue Adaptative
- Affichage de la vue adaptative
- Volet Audit
- Volet Sécurité
- Affichage de la mémoire

Chrome Dev Tools – Inspecteur d'éléments

L'inspecteur d'éléments permet la sélection des éléments HTML sur la page web. A la sélection nous basculons sur l'onglet "Elements" affichant l'arbre DOM HTML et les styles calculés de la page.

The screenshot shows the Chrome Dev Tools interface with the "Elements" tab selected. The left pane displays a mobile view of the La Poste website for an iPhone X with dimensions 375x812 at 75% zoom. The right pane shows the DOM tree for the current page. A specific element, a strong tag with the text "Communiquer à l'international", is selected. The "Styles" tab is active, showing the calculated styles for this element, including "strong.h1.space-mbs". The "Computed" tab shows the final CSS properties applied, including ".header-search-bar .vertical-center .vertical-center--wrapper { float: none; }". The "Properties" tab is also visible. At the bottom, there are tabs for "Console" and "What's New".

Chrome Dev Tools – Vue Adaptative



The screenshot shows the Chrome Dev Tools interface with the 'Elements' tab selected. At the top, it displays a simulated mobile device screen for an iPhone X with dimensions 375 x 812 pixels and a 75% zoom level. The page content is a La Poste Solutions Business website about international communication. The Dev Tools panel shows the DOM structure, with a specific element highlighted: `<strong class="h1 space-mbs"> Communiquer à l'international `. Below the DOM tree, the 'Styles' tab is active, showing the CSS rule `element.style { }` and its properties, such as `.space-mbs, .edito- companies-bless...513677782630:1`. The bottom of the Dev Tools panel includes tabs for 'Console' and 'What's New'.

Cette vue permet de simuler les téléphones et les tablettes pour voir le rendu de notre site sur un téléphone portable. Le menu déroulant en haut à gauche permet de choisir le téléphone et le pourcentage le niveau de zoom.

Chrome Dev Tools – Console

L'onglet Console, nous permet d'afficher la console Web de l'application. Nous affichons les logs de l'application et pouvons filtrer par granularité :

- Verbose
- Info
- Warning
- Errors

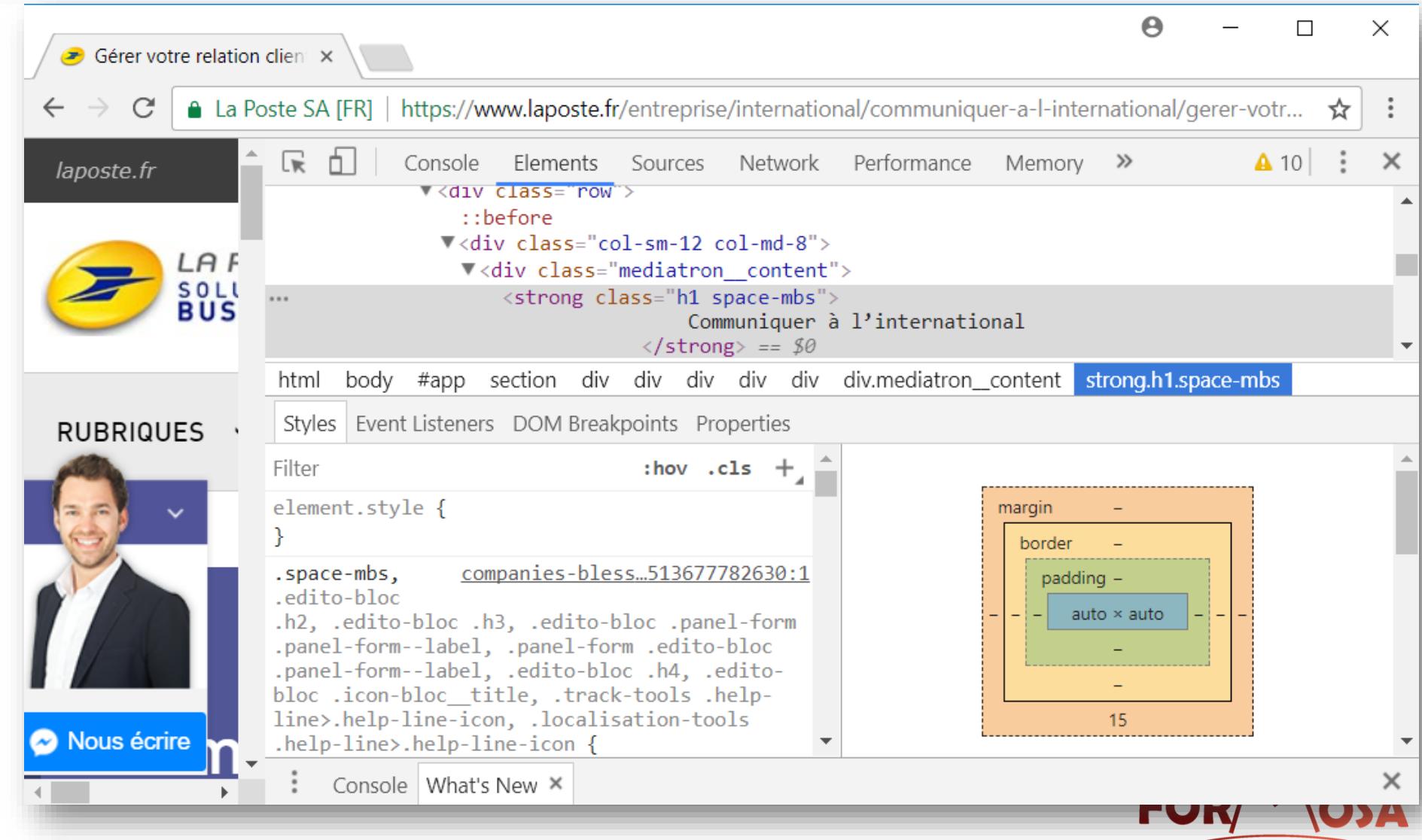
The screenshot shows the Chrome Dev Tools interface with the 'Console' tab selected. The main area displays several log entries:

- A warning message: `https://datacollect6.abtasty.com` will be distrusted in M70. Once distrusted, users will be prevented from loading these resources. See <https://g.co/chrome/symantecpkicerts> for more information.
- A deprecation warning: [Deprecation] Synchronous XMLHttpRequest on the main thread is deprecated because of its detrimental effects to the end user's experience. For more help, check <https://xhr.spec.whatwg.org/>.
- A warning about SSL certificates: The SSL certificate used to load resources from `gerer-votre-relation...a-l-international:1 https://hello.distribeo.com` will be distrusted in M70. Once distrusted, users will be prevented from loading these resources. See <https://g.co/chrome/symantecpkicerts> for more information.
- A warning about SSL certificates: The SSL certificate used to load resources from `gerer-votre-relation...a-l-international:1 https://logs1406.xiti.com` will be distrusted in M66. Once distrusted, users will be prevented from loading these resources. See <https://g.co/chrome/symantecpkicerts> for more information.
- A warning about SSL certificates: The SSL certificate used to load resources from `gerer-votre-relation...a-l-international:1 https://api.iadvize.com` will be distrusted in M70. Once distrusted, users will be prevented from loading these resources. See <https://g.co/chrome/symantecpkicerts> for more information.

The left sidebar shows a preview of the La Poste page, and the bottom navigation bar includes 'Console' and 'What's New' tabs.

Chrome Dev Tools – Elements

L'onglet Élément affiche l'arbre DOM HTML, la sélection d'un élément permet l'affichage et la modification temporaire de cet élément et de son style.



Débogage pas à pas

L'instruction **debugger** est un point d'arrêt pour le Script JavaScript



```
function Stylo (couleur){  
    this.couleur = couleur;  
    this.niveau = 100;  
    this.ouvert = false;  
  
    this.ouvrir= function() { this.ouvert = true; }  
    this.fermer= function() { this.ouvert = false; }  
  
    this.ecrire = function(s) {  
        if (this.ouvert){  
            console.log(s);  
        }  
        else {  
            console.log("Je ne peux pas écrire il faut ouvrir le stylo");  
        }  
    }  
  
}  
  
var stylo1 = new Stylo("rouge");  
debugger;  
stylo1.ecrire("Bonjour !");  
stylo1.ouvrir();  
stylo1.ecrire("Bonjour !");  
debugger;  
for (prop in stylo1){  
    console.log(prop);  
}
```

Débogage pas à pas



L'onglet Source permet d'afficher le code Javascript. Il permet aussi le positionnement de points d'arrêts et l'exécution pas à pas du débogueur.

The screenshot shows the Chrome DevTools interface with the 'Sources' tab selected. The left sidebar lists files: 'top', 'file://', 'D:/Documents/formation', 'premier.html' (which is highlighted), and '(no domain)'. The main pane displays the 'premier.html' file's code. A blue horizontal bar highlights line 35, which contains the line: `var stylo1 = new Stylo("rouge");`. A tooltip for 'Stylo' is open, showing its properties: 'couleur: "rouge"', 'écrire: f (s)', 'fermer: f ()', 'niveau: 100', 'ouvert: false', and 'ouvrir: f ()'. The right sidebar shows various browser properties like 'onwheel: null', 'open: f open()', etc.

```
Paused in debugger | ▶ | ⏪ | Sources | Network | Performance | Memory | Application | Security | Audits | × 3 | ⋮ | X |
```

Sources Content scripts > ⋮ premier.html x

```
20     this.ouvrir= function() { this.ouvert = true; }
21     this.fermer= function() { this.ouvert = false; }
22
23     this.ecrire = function(s) {
24         if (this.ouvert){
25             console.log(s);
26         }
27         else {
28             console.log("Je ne peux pas écrire il faut ouvrir");
29         }
30     }
31
32     var stylo1 = new Stylo("rouge");
33     debugger;
34     stylo1
35     Stylo
36     stylo1
37     stylo1
38     stylo1
39     stylo1
40     stylo1
41     stylo1
42     stylo1
43     stylo1
44     stylo1
45     </script>
46   </html>
```

onwheel: null
▶ open: f open()
▶ openDatabase: f openDatabase()
opener: null
origin: "file://"...
outerHeight: 824
outerWidth: 1536
pageXOffset: 0
pageYOffset: 0
▶ parent: Window {stop: f, open: f, ...}
▶ performance: Performance {onLine: f, ...}
▶ personalbar: BarProp {visible: f, ...}
▶ postMessage: f ()
▶ print: f print()
▶ prompt: f prompt()
▶ releaseEvents: f releaseEvent()
▶ removeEventListener: f removeEventListener()
▶ requestAnimationFrame: f requestAnimationFrame()
▶ requestIdleCallback: f requestIdleCallback()
▶ resizeBy: f resizeBy()
▶ resizeTo: f resizeTo()
▶ screen: Screen {availableWidth: 1536, availableHeight: 824, ...}

Chrome Dev Tools – Le réseau

Cet onglet est très précieux pour mes développeurs Front End et Back end. Nous pouvons voir le temps de chargement de chaque ressource, les entêtes envoyées, les entêtes en réponse et récupérer la ressource.

The screenshot shows the Network tab of the Chrome DevTools. At the top, there's a timeline chart with markers at 20000 ms, 40000 ms, 60000 ms, 80000 ms, 100000 ms, 120000 ms, and 140000 ms. Below the timeline, a list of resources is shown with their names and file types. A green callout bubble points to the 'index-produits.png' entry, which is highlighted with a blue selection bar. Another green callout bubble points to the 'index-produits.png' entry in the list, with a red icon containing 'A-Z' next to it. The bottom of the screenshot shows the status bar with '110 / 123 requests | 931 KB / 935 KB transferred | Finish: 2.3 min | DO...' and a preview of the image.

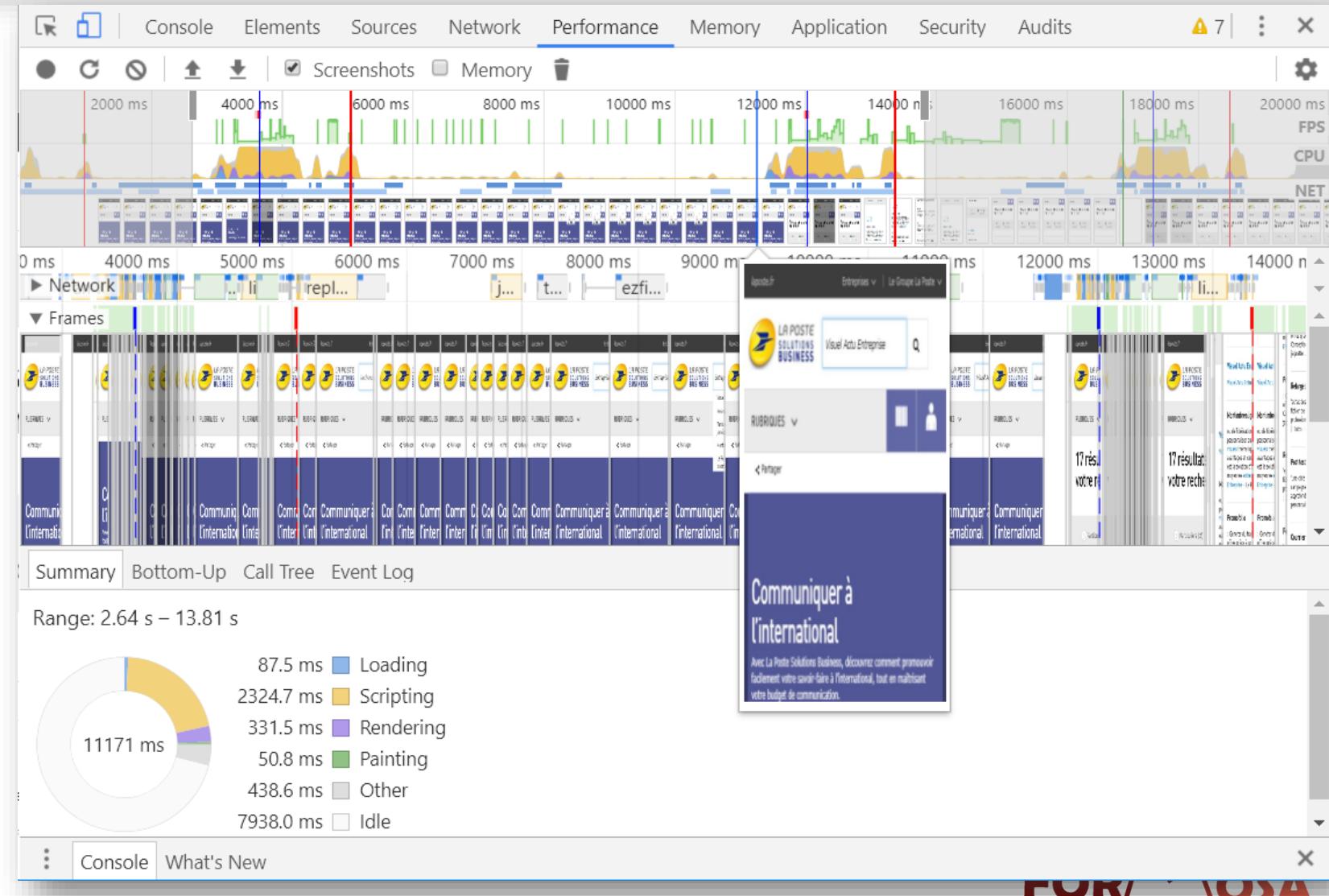
Sélection des ressources

Affichage du détail de la requête

Chrome Dev Tools - Performances



Cet onglet permet l'enregistrement d'une activité et mets en relation différentes informations. Network, frames, Interaction, GPU etc.



Chrome Dev Tools – La mémoire



Cet onglet permet de capturer dans le temps ou à l'instant T l'allocation mémoire

- Le nombre d'objets
- La taille de chaque objet

The screenshot shows the Chrome DevTools Memory tab interface. At the top, there's a summary chart titled "Summary" with a "Class filter" set to "Selected size: 13.5 MB". Below the chart, the "Profiles" section lists "HEAP SNAPSHOTS" (Snapshot 1, 30.2 MB) and "ALLOCATION PROFILES" (Profile 1). The main area displays a table of memory usage details:

Constructor	Distance	Objects Count	Shallow Size	Retained Size			
►(system)	-	94 127	26 %	4 758 344	18 %	8 299 608	31 %
►(compiled code)	3	11 962	3 %	1 880 040	7 %	5 748 384	21 %
►(array)	-	36 866	10 %	4 134 712	15 %	5 612 976	21 %
►(closure)	2	17 670	5 %	1 129 616	4 %	4 972 304	18 %
►Object	-	9 474	3 %	501 520	2 %	4 007 944	15 %
►system / Context	3	3 672	1 %	274 160	1 %	2 475 624	9 %
►Array	2	3 556	1 %	114 224	0 %	1 307 728	5 %
►Window / https://www.laposte.fr	1	1	0 %	56	0 %	1 057 456	4 %
►(string)	-	19 984	6 %	892 768	3 %	892 768	3 %
►Window / https://s7.addthis.com	1	1	0 %	56	0 %	589 584	2 %

Below the table, there's a section for "Retainers" with a table header "Object" and columns "Distance", "Shallow Size", and "Retained Size".

At the bottom of the DevTools window, there are tabs for "Console" and "What's New".

Chrome Dev Tools – l'Application



Affiche le stockage dans le navigateur comme les cookies.

The screenshot shows the Chrome Dev Tools interface with the "Application" tab selected. On the left, a sidebar lists storage types: Manifest, Service Workers, Clear storage; Storage (Local Storage, Session Storage, IndexedDB, Web SQL, Cookies); and Cache (Cache Storage, Application Cache). The Local Storage section is expanded, showing items for the domain <https://www.laposte.fr>. The table lists key-value pairs:

Key	Value
ABTastyData	{"VisitedPages": [{"url": "https://www.laposte.fr/entreprise..."}]}
_at.cww	{"value": true, "expires": 1515036150497}
_at.hist.0102	{"m": 600, "k": 2, "l": {"0": 378, "1": 57}, "b": {"0": 0, "1": 33554432, "2": 0}}
_at.hist.0104	{"m": 600, "k": 2, "l": {"0": 311, "1": 329}, "b": {"0": 0, "1": 0, "2": 0, "3": 0}}
at-lojson-cache-ra-56a2543ab3f50232	{"pc": "newsletter,shin", "subscription": {"active": true, "editi...}}
at-rand	0.30721816271334657
cookiesAlert	{"laPoste": 1549087308729, "MGC": "on", "adverline": "on"}
debug	undefined
tC_Sync	1514872908766

At the bottom of the sidebar, there are links to "Console" and "What's New".

Chrome Dev Tools - Sécurité

Cette page affiche les informations en relation avec la sécurité.

Il présente aussi les certificats de sécurité et l'encryptage des informations.

The screenshot shows the Chrome DevTools interface with the "Security" tab selected. The main panel displays a "Security overview" with a green lock icon and a warning triangle. It states "This page is secure (valid HTTPS)". Below this, three bullet points are listed: "Valid certificate", "Secure connection", and "Secure resources". A "View certificate" button is located under the first point. To the right, a modal dialog titled "Certificat" is open, showing the "Général" tab with detailed information about the certificate, including its purpose (to protect user identity), the issuing authority (CA de Certificados SSL EV), and the validity period (Valid from 03/07/2017 to 03/07/2019). Buttons for "OK" and "Déclaration de l'émetteur" are at the bottom right of the dialog.

ste SA [FR] | <https://www.laposte.fr/entreprise/service-clients/service-clients-france-metropolitaine>

Console Elements Sources Network Performance Memory Application **Security** Audits

Overview

Main origin Reload to view details

Security overview

This page is secure (valid HTTPS).

- Valid certificate
- Secure connection
- Secure resources

[View certificate](#)

Certificat

Général Détails Chemin d'accès de certification

Informations sur le certificat

Ce certificat est conçu pour les rôles suivants :

- Garantit l'identité d'un ordinateur distant
- 1.3.6.1.4.1.14777.6.1.1
- 0.4.0.2042.1.4

* Consultez la déclaration de l'autorité de certification pour p

Délivré à : www.laposte.fr

Délivré par CA de Certificados SSL EV

Valide du 03/07/2017 au 03/07/2019

Déclaration de l'émetteur

OK

Console What's New

Chrome Dev Tools - Audit

Audits teste l'application sur différents critères de qualité.

Attention cependant au conseil spécifique à Chrome. Comme l'utilisation du format WebP développé par google et compatible que sur Google Chrome.

The screenshot shows the Chrome DevTools Audits panel for the website www.laposte.fr. The Audits tab is selected, displaying four categories with their respective scores: Progressive Web App (45), Performance (53), Accessibility (89), and Best Practices (63). The Progressive Web App section is expanded, showing 6 Failed Audits:

- ▶ Does not register a service worker
- ▶ Does not respond with a 200 when offline
- ▶ User will not be prompted to Install the Web App
Failures: No manifest was fetched, Site does not register a service worker, Service worker does not successfully serve the manifest's start_url.
- ▶ Is not configured for a custom splash screen
Failures: No manifest was fetched.
- ▶ Address bar does not match brand colors
Failures: No manifest was fetched, No `<meta name="theme-color">` tag found.
- ▶ Content is not sized correctly for the viewport
The viewport size is 561px, whereas the window size is 412px.

Below this, there is a section for 5 Passed Audits, which is currently collapsed.

Firefox Developer



Firefox Developer Edition est une version optimisée pour les développeurs offrant de meilleures performances.

Des captures d'écrans simplifiées

Il apparaît sur un fond noir pour le confort des yeux.

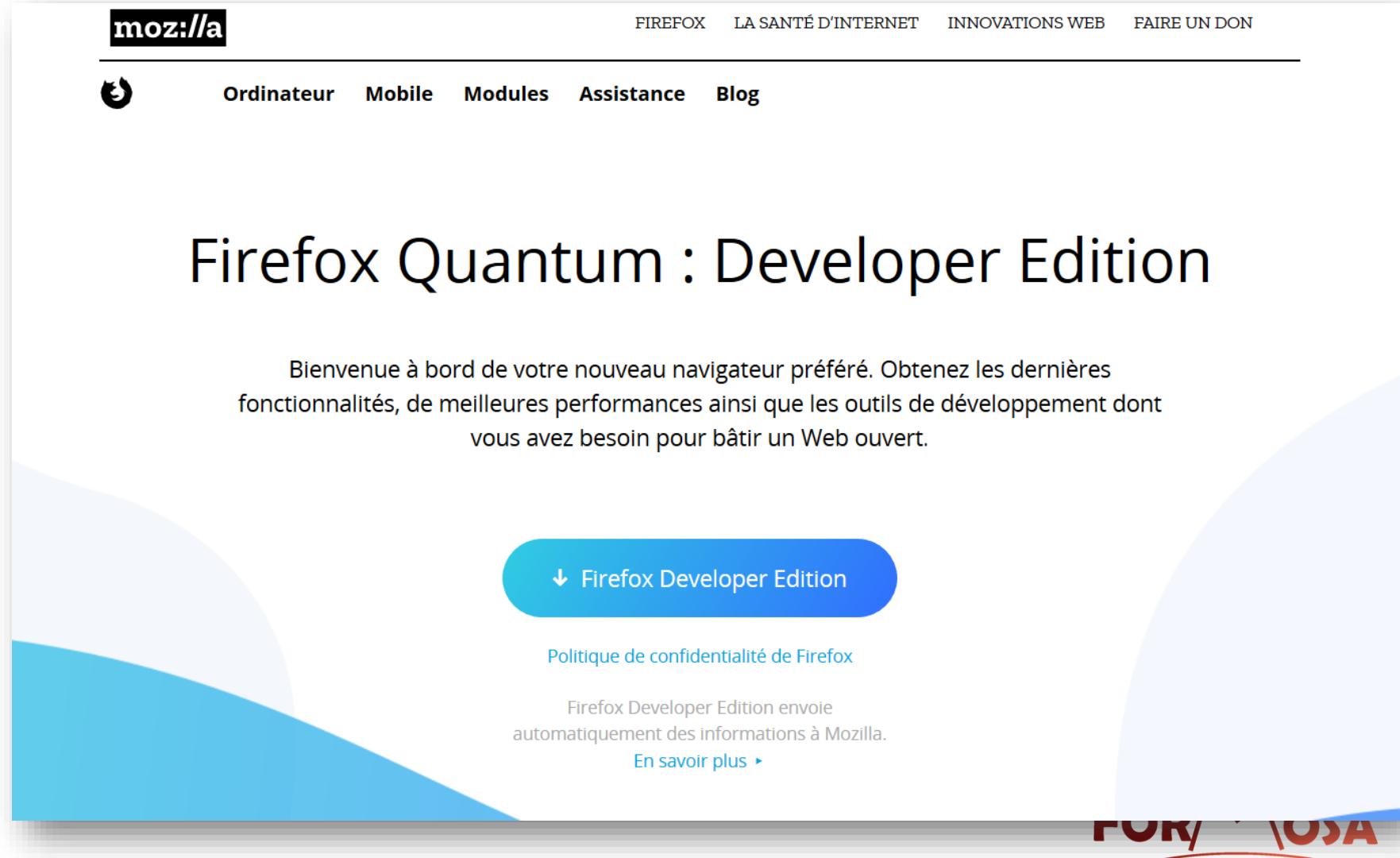
L'outil intégré pour les développeurs reste cependant le même.

Télécharger Firefox Developer



Le téléchargement de Firefox Developer est possible sur :

[https://www.mozilla.org/fr/
firefox/developer/](https://www.mozilla.org/fr/firefox/developer/)



The screenshot shows the Mozilla Firefox Developer Edition download page. At the top, there's a navigation bar with the Mozilla logo, links for FIREFOX, LA SANTÉ D'INTERNET, INNOVATIONS WEB, and FAIRE UN DON, and a search bar. Below the navigation is a menu with links for Ordinateur, Mobile, Modules, Assistance, and Blog. The main title "Firefox Quantum : Developer Edition" is prominently displayed. A welcome message in French encourages users to get the latest features, better performance, and development tools. A large blue button with a downward arrow and the text "Firefox Developer Edition" is centered. To the right, there's a link to the "Politique de confidentialité de Firefox". Below that, a note states that Firefox Developer Edition sends information to Mozilla and includes a "En savoir plus" link.

moz://a

FIREFOX LA SANTÉ D'INTERNET INNOVATIONS WEB FAIRE UN DON

Ordinateur Mobile Modules Assistance Blog

Firefox Quantum : Developer Edition

Bienvenue à bord de votre nouveau navigateur préféré. Obtenez les dernières fonctionnalités, de meilleures performances ainsi que les outils de développement dont vous avez besoin pour bâtir un Web ouvert.

↓ Firefox Developer Edition

Politique de confidentialité de Firefox

Firefox Developer Edition envoie automatiquement des informations à Mozilla.

[En savoir plus ▶](#)

Nos formations

FORMOSA

Firefox Developer



Sélection d'éléments

Code source des pages et débogueur

Onglet pour l'édition des styles

Calcul des performances

Affichage de l'allocation mémoire

Stockage des cookies et des sessions

Firefox Quantum : Developer Edition

Bienvenue à bord de votre nouveau navigateur préféré.

Obtenez les dernières fonctionnalités, de meilleures performances ainsi que les outils de développement dont vous

Affichage du DOM HTML

Affichage de la vue adaptative

Onglet Gestion du réseau

Activation de la vue adaptative

État	Métho...	Fichier	Domaine	Type	Transfert	Taille	
○ 200	GET	icon-menu-dark.d12612893396.s...	www.mozilla.org	img	svg	mis en cache	268 o
○ 200	GET	down-arrow-light-sprite.3c238d...	www.mozilla.org	img	svg	mis en cache	1,22 Ko
○ 200	GET	logo-developer-quantum.48752c...	www.mozilla.org	img	svg	mis en cache	19,06 Ko
○ 200	GET	newsletter-background.44c7b34c...	www.mozilla.org	img	svg	mis en cache	382 o
○ 200	GET	moz-wordmark-light-reverse.cb1...	www.mozilla.org	img	svg	mis en cache	1,43 Ko
○ 200	GET	social-icon-sprite.bf2ae0cd0f01.s...	www.mozilla.org	img	svg	mis en cache	3,61 Ko
○ 200	GET	moz-wordmark-dark-reverse.2cb...	www.mozilla.org	img	svg	mis en cache	1,55 Ko
○ 200	GET	stylo-engine.44d72db41ee9.svg	www.mozilla.org	img	svg	mis en cache	6,17 Ko
○ 200	GET	hero-debugger-ani.7e04d95e76e...	www.mozilla.org	img	gif	mis en cache	192,58 Ko
○ 200	GET	hero-cssgrid-ani.c0a7ace9dbf7.gif	www.mozilla.org	img	gif	mis en cache	75,76 Ko
○ 200	GET	feature-inspector.1bea11fb2286....	www.mozilla.org	img	svg	mis en cache	1,92 Ko
○ 200	GET	feature-console.921a5b1b4ff3.svg	www.mozilla.org	img	svg	mis en cache	824 o

Firefox Developer - Inspecteur



Un inconvénient de Firefox est le problème pour adapter la vue.

Mis à part cela la fonctionnalité de Firefox est similaire à Chrome

The screenshot illustrates the Firefox Developer Tools Inspector interface. On the left, a responsive view of the La Poste mobile website is displayed at 320x480 pixels with GPRS network conditions. The main content area shows the La Poste logo and some placeholder text. On the right, the DOM tree and CSS styles for the current element are shown. The DOM tree starts with the root element `<html>`, followed by `<body>`, and then various header, container, and brand elements. The CSS styles panel shows numerous rules, many of which are overridden by !important declarations. The styling for the main content area includes a background-size of 1791px auto, a background-color of #fff, and a background-image of `url("https://ads.adverline.com/ba...")`. The `body` element has a min-width of 35em and a font-family of Verdana, Tahoma, Helvetica, Arial, sans-serif. The `h1` element has a font-size of 16px and a line-height of 1.31. The overall layout is centered and scrollable.

Firefox Developer - Console



The screenshot shows the Firefox Developer Tools interface with the 'Console' tab selected. On the left, there's a mobile device simulation showing a web page from 'Le Groupe La Poste'. The console output on the right displays several error messages and some object introspection results:

```
ADYOU LIKE LOAD : NATIVE LIBRARY
⚠ unreachable code after return statement [En savoir plus]
⚠ unreachable code after return statement [En savoir plus]
⚠ SyntaxError: 08 is not a legal ECMA-262 octal constant [En savoir plus]
⚠ unreachable code after return statement [En savoir plus]
⚠ Using //@ to indicate sourceMappingURL pragmas is deprecated. Use //## instead [En savoir plus]

NativePush();
Zone : adyoulike
nativePushReady()
nativePushInit()

Critical: Visit Tag found in Cross domain Iframe
⚠ unreachable code after return statement [En savoir plus]

SMARTTAG plc2
{
  account_id: 10677
  ads: Array [ ... ]
  error_code: "16"
  inventory: Object { sections: "lang_fr,home,geoip_unknown" }
  reason: "internal-error"
  site_id: 28342
  size_id: 68
  status: "error"
  tracking: ""
  zone_id: 263106
  __proto__: Object { ... }

sskz s Visit Tag found
CRITICAL : Accurate Visit time measure IMPOSSIBLE, loader was loaded in cross domain
Loader ready 2.027s
Tracker ready 1.03312s

lopstique
```

Options de Filtrage

Les fonctionnalités offertes par la console restent similaires aux fonctionnalités de Google Chrome.

Affichage détaillé des objets



Firefox Developer - Débogueur



Le Débogueur de Firefox

Développer est équivalent à

l'onglet "source" sur Google

Chrome il permet de

positionner des points d'arrêts

sur les scripts et d'exécuter le

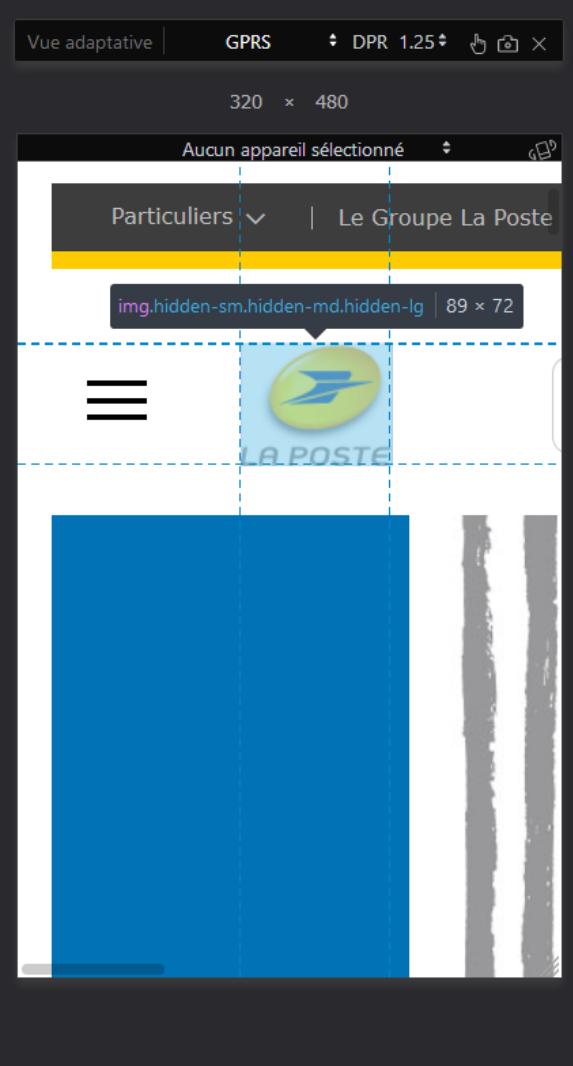
débogueur.

A screenshot of the Firefox Developer Tools interface. The top bar shows tabs for 'Inspecteur', 'Console', 'Débogueur' (which is selected), 'Éditeur de style', and 'Performances'. The main area has three panes: 'Sources' (listing files like app-min.js, modernizr.js, vendor.js, etc.), 'Structure' (showing the DOM tree), and a code editor pane showing the source code for 'app-min.js'. On the left, there's a mobile device simulation showing a webpage from 'www.laposte.fr'. A tooltip over an element on the page reads 'img.hidden-sm.hidden-md.hidden-lg | 89 x 72'. A blue dashed box highlights a specific element on the page, likely where a breakpoint was set.

Firefox Developer - Style

L'onglet Style permet de modifier les feuilles de styles directement. Et d'enregistrer la feuille dans un fichier séparé.

L'éditeur affiche aussi les styles intégrés.



The screenshot shows the Firefox Developer Tools interface with the "Style" tab selected. On the left, there's a mobile device simulation window titled "Vue adaptative" showing a dark-themed mobile application interface with a yellow header bar and a logo. The main panel on the right is the "Éditeur de style" (Style Editor), which lists various CSS files and their rule counts. The file "sprites.css" is currently selected, highlighted with a blue background. The code editor on the right displays the CSS rules for this file, which define background images and positions for various badge classes. The code is as follows:

```
.badges-sprite,.badges-a-partir-de-deux-cent-cinquante-cinq-euros,.badges-a-p  
background-image:url('../images/generated/badges-s451054f892.png');  
background-repeat:no-repeat  
}  
.badges-a-partir-de-deux-cent-cinquante-cinq-euros {  
background-position:0 0  
}  
.badges-a-partir-de-deux-cent-dix-huit-euros {  
background-position:0 -56px  
}  
.badges-a-partir-de-deux-cent-quarante-sept-euros {  
background-position:0 -124px  
}  
.badges-a-partir-de-trois-cent-trente-euros {  
background-position:0 -186px  
}  
.badges-attributs-produits-calendar {  
background-position:0 -248px  
}  
.badges-avis-de-reception {  
background-position:0 -310px  
}  
.badges-cent-grammes-maxi {  
background-position:0 -372px  
}  
.badges-cinq-dix-jours {  
background-position:0 -434px  
}  
.badges-cinq-kilo-maxi {  
background-position:0 -496px  
}  
.badges-cinqcent-grammes-maxi {  
background-position:0 -558px  
}  
.badges-code-barre {  
background-position:0 -620px  
}  
.badges-coliposte {  
background-position:0 -682px  
}  
}
```

Firefox Developer - Performance



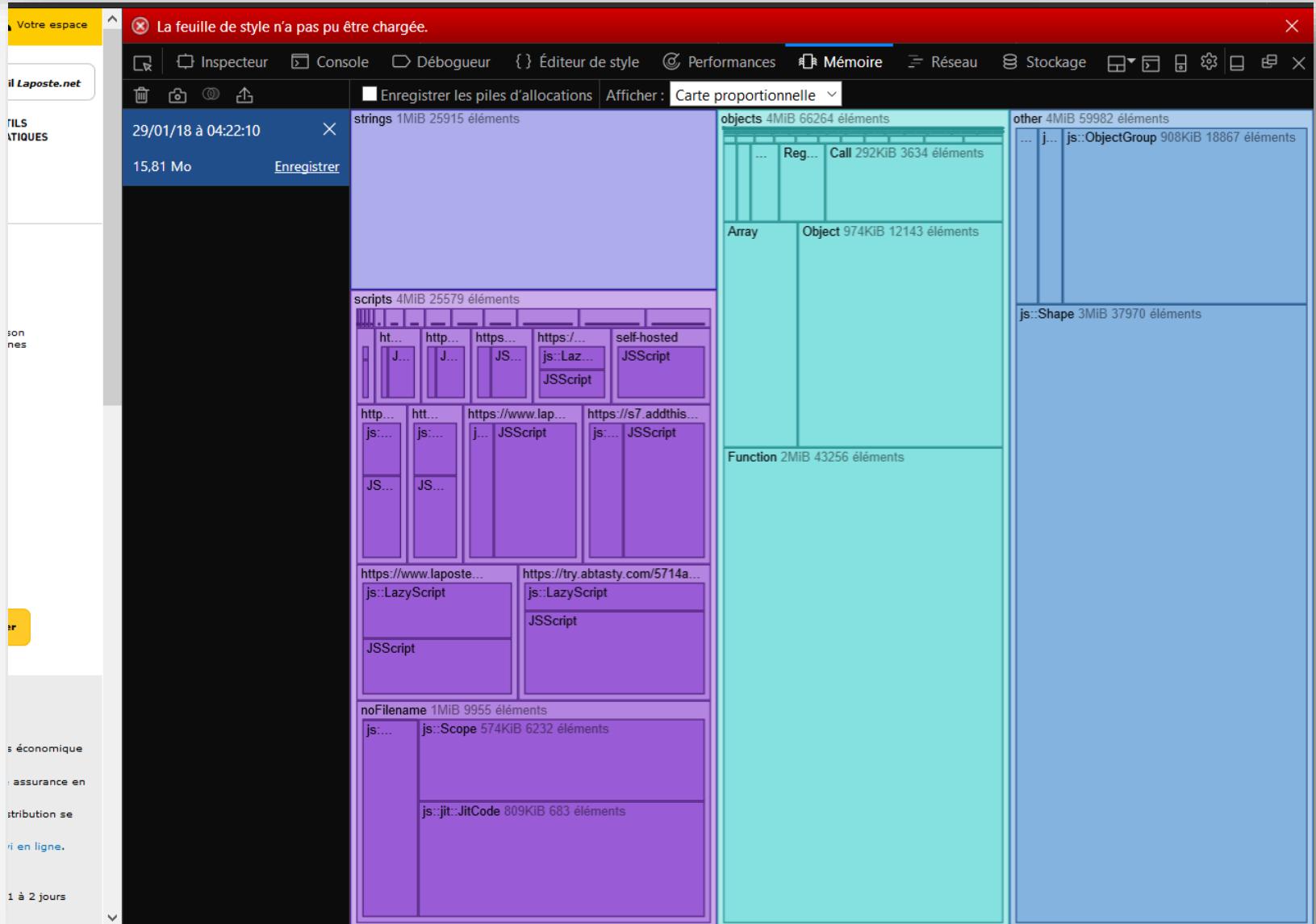
L'onglet performance permet d'enregistrer une séquence d'interactions, de restituer le temps d'affichage et d'afficher les événements.



Firefox Developer - Mémoire



L'onglet mémoire permet de capturer un instantané, il permet aussi de ressortir l'état de la mémoire,



Firefox Developer - Réseau

L'onglet réseau permet comme pour Google Chrome d'afficher les appels réseaux, leur réponses, statuts, informations et de les filtrer par type ou nom.

La feuille de style n'a pas pu être chargée.

État	Métho...	Fichier	Domaine	Source	Type	Transfert	Taille
200	GET	layers.ccbc92d4047235e1c1a9.js	s7.addthis.com	script	js	mis en cache	0 o
200	GET	fre-fr	www.laposte.fr	document	html	25,22 Ko	371,09 Ko
204	POST	100eng.json?sh=0&ph=5777&iv...	m.addthis.com	beacon	json	245 o	0 o
200	GET	vendor.css	www.laposte.fr	stylesheet	css	mis en cache	34,03 Ko
200	GET	sprites.css	www.laposte.fr	stylesheet	css	mis en cache	4,55 Ko
200	GET	sprite-picto.css	www.laposte.fr	stylesheet	css	mis en cache	2,15 Ko
200	GET	print.css	www.laposte.fr	stylesheet	css	mis en cache	721 o
200	GET	consumers.css?version=v1.3	www.laposte.fr	stylesheet	css	mis en cache	52,66 Ko
200	GET	modernizr.js	www.laposte.fr	script	js	mis en cache	0 o
200	GET	yui-min.js	www.laposte.fr	script	js	mis en cache	0 o
200	GET	tc_Laposteportail_4.js	cdn.tagcommun...	script	js	mis en cache	0 o
200	GET	consumers-blessed2.css?z=1515...	www.laposte.fr	stylesheet	css	mis en cache	194,44 Ko
200	GET	consumers-blessed1.css?z=1515...	www.laposte.fr	stylesheet	css	mis en cache	269,55 Ko
200	GET	affranchissement-chrono-expres...	www.laposte.fr	img	png	4,93 Ko	4,51 Ko
200	GET	vendor.js	www.laposte.fr	script	js	mis en cache	0 o
200	GET	app-min.js?version=v1.3	www.laposte.fr	script	js	mis en cache	0 o

Firefox Developer - Stockage



L'onglet Stockage affiche l'état des variables de sessions et des cookies.

La feuille de style n'a pas pu être chargée.

Inspecteur Console Débogueur Éditeur de style Performances Mémoire Réseau Stockage

Filtrer les éléments

Cookies

Nom	Domaine	Chemin	Expire le	Dernier accès le	Valeur
_as_rng	.laposte.fr	/	Wed, 02 Jan 2019 06:36:5...	Mon, 29 Jan 2018 03:17:1...	894
_atvc	www.laposte.fr	/	Fri, 01 Mar 2019 03:17:27...	Mon, 29 Jan 2018 03:17:2...	6%7C1%2C0%7...
_atuvs	www.laposte.fr	/	Mon, 29 Jan 2018 03:47:2...	Mon, 29 Jan 2018 03:17:2...	5a6e8e3e6a77...
_gat	.laposte.fr	/	Mon, 29 Jan 2018 03:17:4...	Mon, 29 Jan 2018 03:16:4...	1
_ga	.laposte.fr	/	Wed, 29 Jan 2020 03:17:2...	Mon, 29 Jan 2018 03:17:2...	GA1.2.1883292...
_gid	.laposte.fr	/	Tue, 30 Jan 2018 03:17:2...	Mon, 29 Jan 2018 03:17:2...	GA1.2.1207747...
ABTastySession	.laposte.fr	/	Mon, 29 Jan 2018 03:47:2...	Mon, 29 Jan 2018 03:17:2...	referrer%3Dhtt...
ABTasty	.laposte.fr	/	Sat, 23 Feb 2019 03:17:2...	Mon, 29 Jan 2018 03:17:2...	uid%3D180102...
ADRUM_BT	www.laposte.fr	/	Mon, 29 Jan 2018 03:17:4...	Mon, 29 Jan 2018 03:17:1...	R%3A34%7Ccli...
ayl_vfvt	www.laposte.fr	/	Mon, 29 Jan 2018 23:00:0...	Mon, 29 Jan 2018 03:17:1...	ayl_vfvt
cap_apn_ttlbdb...	www.laposte.fr	/	Tue, 30 Jan 2018 03:00:1...	Mon, 29 Jan 2018 03:17:1...	1
cap_dsp_tuck_tt...	www.laposte.fr	/	Tue, 30 Jan 2018 03:00:1...	Mon, 29 Jan 2018 03:17:1...	1
cap_seg_dsp_ttl...	www.laposte.fr	/	Wed, 03 Jan 2018 06:36:5...	Tue, 02 Jan 2018 06:48:0...	1
cap_seg_dsp_ttl...	www.laposte.fr	/	Mon, 29 Jan 2018 03:31:4...	Mon, 29 Jan 2018 03:16:4...	1
cap_seg_dsp_ttl...	www.laposte.fr	/	Mon, 29 Jan 2018 03:31:4...	Mon, 29 Jan 2018 03:16:4...	1
edge_xua_device	www.laposte.fr	/	Wed, 02 Jan 2019 06:36:5...	Mon, 29 Jan 2018 03:17:1...	desktop

Stockage de session

Stockage cache

Indexed DB

Stockage local

Firefox Developer vs Chrome Dev Tools



Le fonctionnement est sensiblement le même que pour Chrome Dev Tools.

Les onglets sont un peu renommés cependant le principe est présent.

ChromeDev Tools possède cependant plus de fonctionnalités. De plus, la vue adaptative de Firefox Developer possède quelques bugs.

Nous recommandons donc fortement Google Chrome pour le débogage de vos applications.

Node Debug



Node debug est un utilitaire NodeJS pour le débogage d'application Node JS

Pour l'installation

```
npm install-g node-debug
```

Pour l'utilisation

```
node-debug--web-port 9001 --debug-port 4200
```

Test Multinavigateurs



Le Problème

Nous effectuons des développement Web qui vont être visualisés par différents appareils, Système d'exploitation et navigateurs possédant différentes versions et des bugs.

Ainsi il est impossible que les navigateurs possèdent tous un rendu exactement identique.

En tant que développeur Web, nous constatons que la majorité des problèmes de compatibilité proviennent d'Internet Explorer mais pas que d'internet explorer

Test Multinavigateurs



Le Besoin

Le besoin est de pouvoir visualiser le rendu de notre site sur les navigateurs et plateforme posant problèmes.

Différentes solutions existent.

Test Multinavigateurs



Mode compatibilité

Internet Explorer à développé une fonctionnalité nommé "Mode Compatibilité". Le problème est que cette fonctionnalité ne fait que simuler le navigateur et ne reproduit pas les bugs correctement.

Par exemple l'oubli d'une virgule à la fin d'un JSON est un problème fréquent :

```
{  
    attribut1 : valeur1,  
    attribut2 : valeur2,  
}
```

IE6 et IE7 plante, alors qu'IE8 ainsi que les autres navigateurs passent.

Le mode compatibilité ne reproduit pas ce bug

Test Multinavigateurs



Installation de versions multiples

L'installation de versions multiples semble être une bonne idée mais tout les navigateurs ne supporte pas cela. De plus certains bugs sont présents sur certains OS mais pas sur d'autres.

Par exemple Internet Explorer est associé à des bibliothèque de bas niveau du Système d'exploitation qui fait que le changement de version du navigateur est quasiment impossible.

Test Multinavigateurs

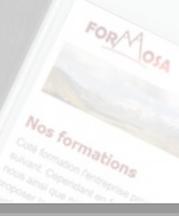


Meilleures Solutions : Les machines virtuelles

La meilleure solution reste l'utilisation de machines virtuelles, simulant appareil, navigateur et versions.

Différentes compagnies proposent des solutions allant du screenshot, jusqu'à l'interaction en passant par la comparaison des rendus

Utilisation de Machines Virtuelle



Browser Stack (Free Trial)

<https://www.browserstack.com>

Prix : 29\$ à 149\$

Test en Live

Capture d'écrans

Test du Responsive

Capture d'écrans via une API

Automate pour application de Bureau

Automate pour Mobile

Mobile App Testing on 1 device

Sécurisé | https://www.browserstack.com/app-live

BrowserStack Live Automate App Live App Automate More Beta Pricing Resources Sign in FREE TRIAL

Interactive Mobile App Testing

Instant access to 1000+ real iOS and Android devices.
Say goodbye to your device lab.

Sign Up for free

Logs

```
> 01-04 04:06:03.861 2749-2749/com.app.touristGuide D/TapAndSwipe: Article opened : Red Sea
> 01-04 04:06:06.362 2749-2749/com.app.touristGuide D/TapAndSwipe: Received text : Map Section
```

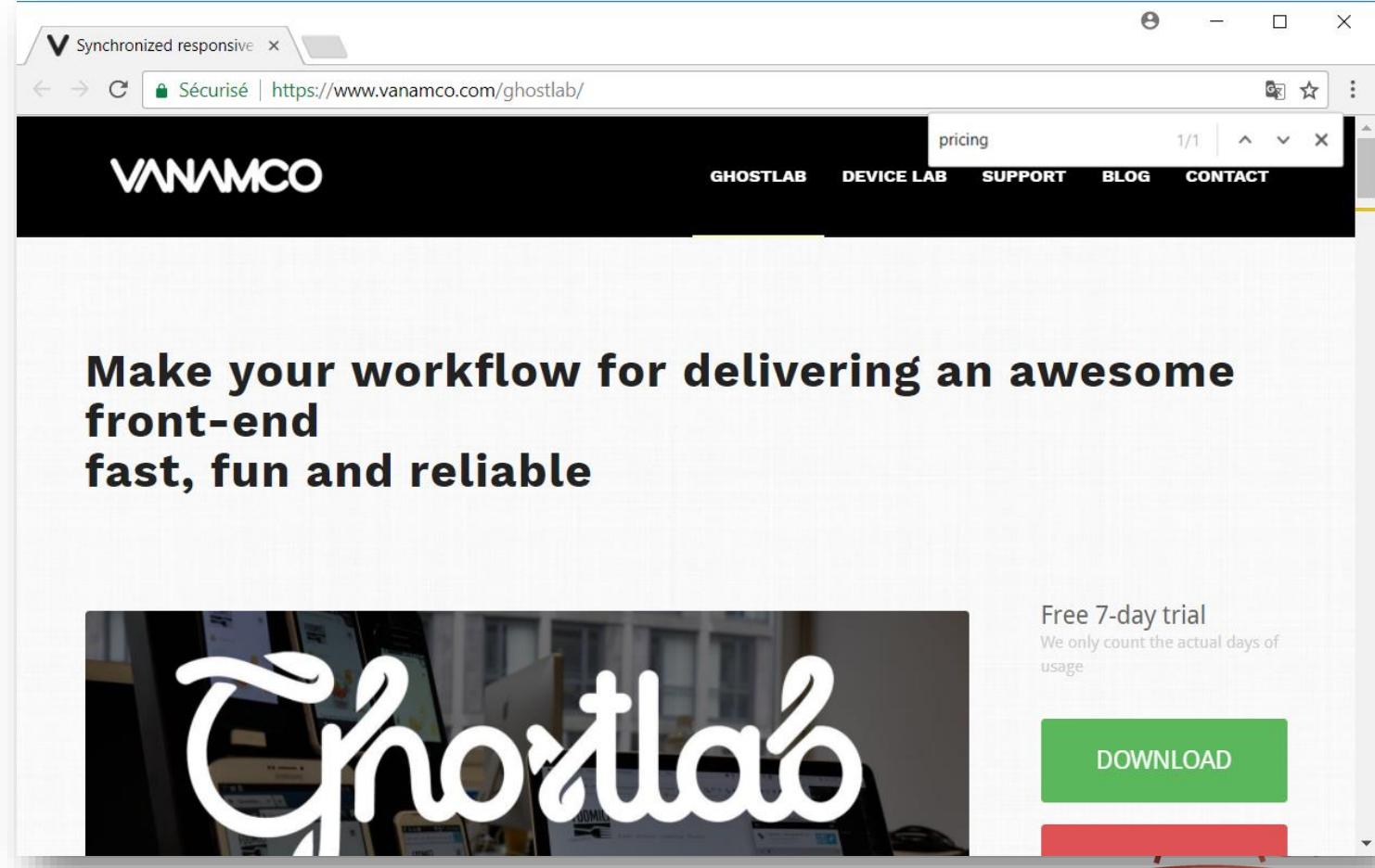
Utilisation de Machines Virtuelle

GhostLab (Free Trial 7 jours)

Prix : 49\$

<https://www.vanamco.com/ghostlab/>

Test multi navigateur et répercussions des actions sur différents navigateurs



Utilisation de Machines Virtuelle



CrossbrowserTesting

<https://crossbrowsertesting.com/>

Prix : 29\$ à 100\$

Test en live

Automatisation (Selenium, Appium)

Test visuel du responsive

A screenshot of a web browser displaying the CrossBrowserTesting website. The URL in the address bar is https://crossbrowsertesting.com/pricing. The page has a dark blue header with the CrossBrowserTesting logo and navigation links for Product, Enterprise, Customers, Pricing, Resources, and Browsers. It features a main heading "Join Thousands Of Happy Customers Testing Faster On Real Browsers and Devices." Below this are two main service offerings: "Live Testing" (represented by a computer monitor icon) and "Automated Testing" (represented by a gear icon). A live chat window is visible on the right side, showing a message from "Shane from CrossBrowserTesting...." asking if there are any pricing or upgrade questions, with a response indicating an easy pricing model. A message input field says "Write a reply..." with icons for GIF, smiley face, and attachment.

Utilisation de Machines Virtuelle



Browsershots

<http://browsershots.org/>

Prix : Gratuit à 29\$

Création de Screenshots

The screenshot shows the Browsershots.org homepage. At the top, there's a navigation bar with links for Accueil, Queue, Factories, Recent, Free Icons, and Vec. Below the navigation, there's a banner for 'WebStorm' with the text 'L'IDE JavaScript le plus intelligent'. On the left, there's a 'Free 7-Day Trial' button for 'CrossBrowserTesting' with a list of features: 'Live interactive tests plus screenshots', 'Test dev and login sites behind a firewall', and 'CrossBrowserTesting'. In the center, there's a search bar with the placeholder 'Enter URL Here:' and a 'Submit' button. Below the search bar, there are four sections for different operating systems: 'Linux', 'Windows', 'Mac', and 'BSD'. Each section has a list of browser versions with checkboxes next to them. A 'Contribute' button is located in the bottom right corner of the Linux section. A 'Free 7-Day Trial' banner is also visible in the bottom right corner.

BrowserLing

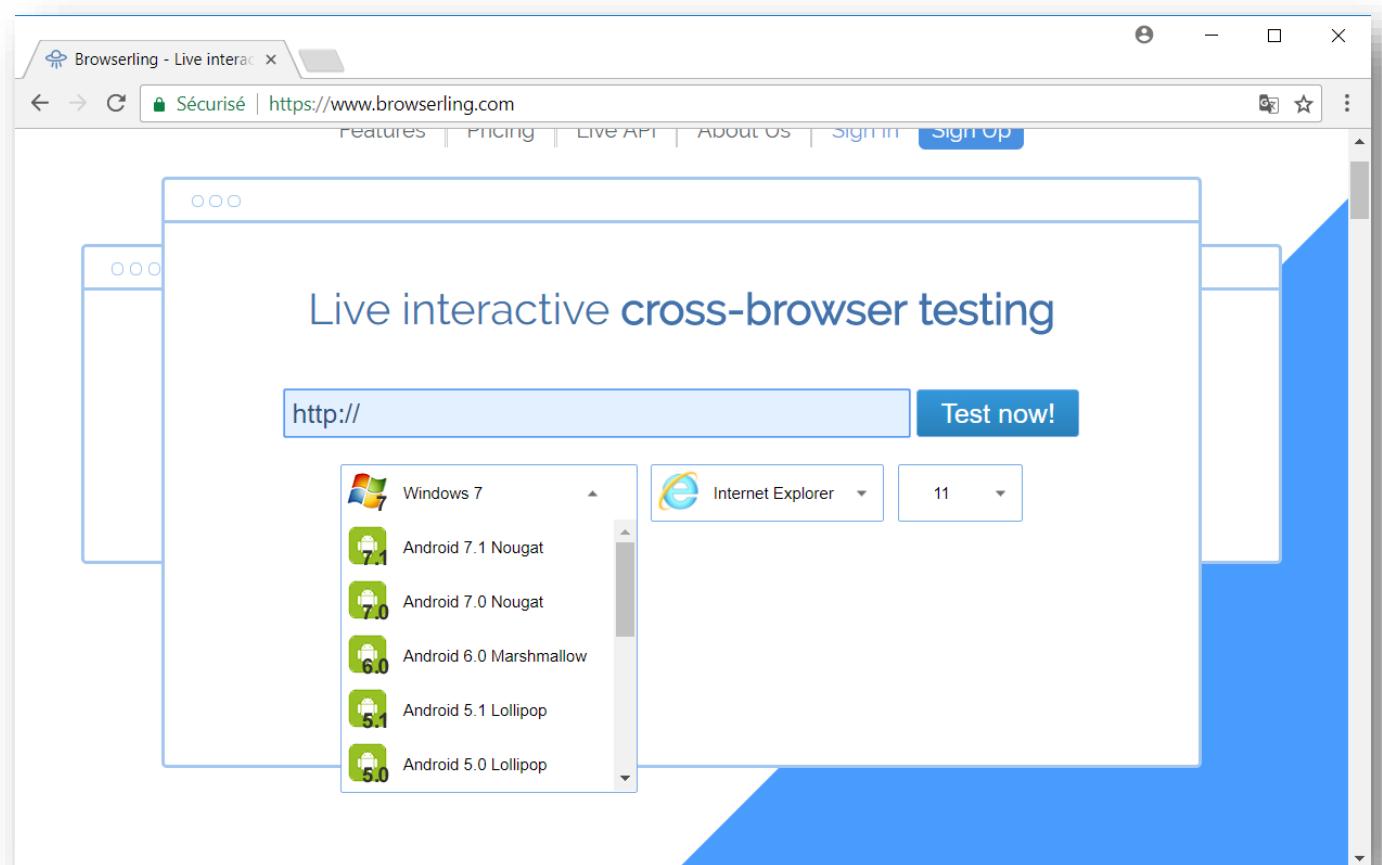
BrowserLing

<https://www.browserling.com/>

Gratuit pour 3 minutes de test en live

Plus de 80 combinaisons
(Navigateur/OS)

Prix : Gratuit à 250\$



Autre Solutions



Browsera

<http://www.browsera.com/>

Prix : Gratuit à 99\$

IE Net Renderer

<http://netrenderer.com/>

Prix : Gratuit

BrowserMall

<https://www.browsemall.com/>

Prix : 69\$

Spoon

<http://spoon.net/browsers/>

Prix : Gratuit à 99\$



Gestion des Logs



Le problème

La gestion des logs est un problème récurrent en informatique :

- Il diminue les performances de l'application
- C'est un problème de sécurité
- Dans certaines versions d'Internet Explorer IE8 et IE9 l'utilisation de console.log peut provoquer des erreurs si les outils de développement ne sont pas actifs

Comment traiter ces logs ?

Gestion des Logs



1^{ière} solution : La suppression manuelle

Il s'agit de supprimer manuellement les occurrences :

- console.log
- console.dir
- console.group

Il est possible de détecter plus facilement les problèmes de codes à l'aide d'outils comme :

- JSLint
- JSHint
- ESLint

Gestion des Logs



2^{ième} solution : Remplacer le méthode de gestion des Logs

La seconde solution consiste à remplacer les méthodes permettant de générer les logs par une méthode vide.

Ainsi tous les appels seront alors redirigés vers cette méthode ne faisant rien.

Gestion des Logs



3^{ième} solution : L'utilisation d'une bibliothèque comme Javascript Debug

<http://benalman.com/projects/javascript-debug-console-log/>

Ce genre de solution permet d'organiser les logs en différents niveau :

- debug.log
- debug.debug
- debug.info
- debug.warn
- debug.error

La méthode debug.setLevel permet de spécifier le niveau de debuggage et ainsi choisir le niveau en production, en recette et en développement.

Pratique



- 1- Analyse des performances sur un site en utilisant les Chrome Dev Tools
- 2- Comparaison des affichages d'un site web en utilisant les outils de virtualisation



La programmation orientée objets en JS

" Cycle Développeur "

Paradigmes de la programmation Orientée Objet

La programmation orientée objet est clairement la programmation la plus populaire actuellement.

Indépendamment des langages de programmation elle introduit les concepts :

- d'Objets
- D'encapsulation des données (Attributs et Méthodes)
- d'Héritage (Simple et Multiple)
- de polymorphisme
- de redéfinition de méthodes

...

La création d'un objet se fait dans le langage JSON :

```
var stylo1 = {  
    couleur : "rouge",  
    niveauEncre : 100  
};  
  
var stylo2 = {  
    couleur : "noir",  
    niveauEncre : 50  
};  
  
console.log(stylo1);  
console.log(stylo2);
```

```
▶ {couleur: "rouge", niveauEncre: 100}  
▶ {couleur: "noir", niveauEncre: 50}
```

[premier.html:26](#)

[premier.html:27](#)

La syntaxe du JSON est la suivante :

```
{  
    Attribut1 : Valeur1,  
    Attribut2 : Valeur2,  
    ...  
    AttributN : ValeurN  
}
```

Une valeur pouvant être :

- Un entier
- Un décimal
- Une chaîne de caractère
- Un tableau composés de valeurs
- Un sous objet

POO - Objet



Nous présentons ici un exemple de JSON :

```
{  
    nom : "Guy",  
    prenom : "Bastien",  
    notes : [{  
        matiere : "Math",  
        note : 15  
    },  
    {  
        matiere : "philo",  
        note : 12  
    }  
]
```

POO - Encapsulation



En programmation nous pouvons être les développeurs des classes instanciant les objets, ou nous pouvons être les utilisateurs des objets.

Pour utiliser un stylo nous n'avons pas besoin de savoir comment il est fabriqué.

La notion de programmation orientée objet repose sur le principe d'encapsulation. L'encapsulation c'est le fait de ne pas rendre accessible les données directement depuis l'extérieur de l'objet mais de rajouter des méthodes qui accèdent à l'objet.

Ainsi un Objet peut être vu comme :

DONNEES + METHODES

Définir un objet Attributs et Méthodes



Un objet dans la théorie des langage à Objet se compose d'un ensemble d'attributs et d'un ensemble de méthodes .

- Un objet par exemple un Stylo possède des caractéristiques : couleur, niveau d'encre
- Un objet possède des actions : décapuchonner, écrire, capuchonner.

La syntaxe pour accéder à une méthode :

```
objet.methode()
```

La syntaxe pour accéder aux propriétés

```
objet.nomProprietes
```

Définir un objet Attributs et Méthodes



Une chaîne de caractère est un objet, ainsi concat est une méthode des chaînes de caractères length est une propriété.

```
var chaîne = "Bonjour "
var nom = "Guillaume",
typeof chaîne;
typeof nom;
var résultat = chaîne.concat(nom);
console.log(résultat);
console.log(résultat.length),
```

Type "string"

Méthode

Propriété

Création d'objet



Il est possible de créer un objet en définissant directement les attributs et les méthodes. Pour créer un modèle d'objets, nous devrons passer par les classes :

```
var obj = {  
    nom : "Arthur",  
    prenom : "Guy",  
    maMethode : function(a,b){ return a+b}  
}  
  
console.log(obj);
```

Les "classes" JS



Il est possible de créer des classes en JavaScript :

```
function Stylo (couleur){  
    this.couleur = couleur;  
    this.niveau = 100;  
    this.ouvert = false;  
  
    this.ouvrir= function() { this.ouvert = true; }  
    this.fermer= function() { this.ouvert = false; }  
  
    this.ecrire = function(s) {  
        if (this.ouvert){  
            console.log(s);  
        }  
        else {  
            console.log("Je ne peux pas écrire il faut ouvrir le stylo");  
        }  
    }  
}  
  
var stylo1 = new Stylo("rouge");  
stylo1.ecrire("Bonjour !");  
stylo1.ouvrir();  
stylo1.ecrire("Bonjour !");
```

Je ne peux pas écrire il faut ouvrir le stylo
Bonjour !

Les propriétés d'un objet



Pour parcourir les propriétés d'un objet nous pouvons utiliser :

```
for (prop in Objet){  
    console.log(prop);  
}
```

```
for (prop in stylo1){  
    console.log(prop);  
}
```

couleur
niveau
ouvert
ouvrir
fermer
ecrire

premier.html:40
premier.html:40
premier.html:40
premier.html:40
premier.html:40
premier.html:40

Réaliser des Objets



1. Réaliser une classe Sac permettant l'ajout d'élément dans un ensemble seulement si il n'existe pas. On doit avoir les méthodes suivantes :
 - Ajouter
 - Choisir aléatoirement
 - Taille
 - Vide

Réaliser des Objets



1. Nous souhaitons réaliser une grille de Loto, une grille de loto est un ensemble de 5 chiffres + d'un numéro complémentaire
2. Une grille possède une mise de départ par exemple 2€
3. Une grille peut être validée avec un tirage (un tableau d'entier) et retournant la somme gagnée

Réaliser des Objets

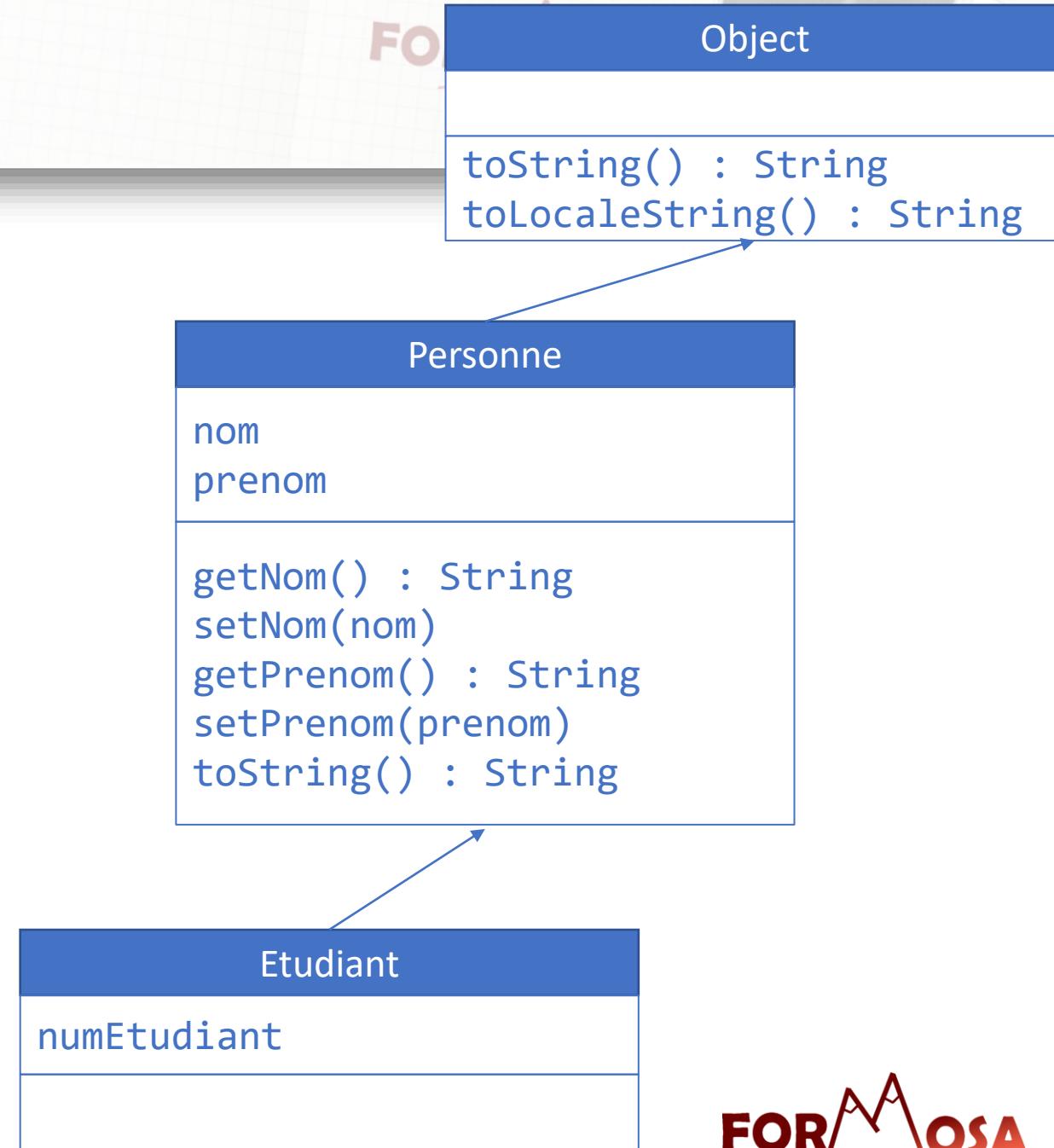


1. Nous souhaitons modéliser des personnes chaque personne joue généralement la même grille de loto chaque jour et possède de l'argent.
2. Simuler un jeu de loto sur 10 ans à raison d'une grille par jour.

Le concept d'héritage

Le concept d'héritage est un concept fondamental en programmation orienté objets et tout langages orientés objets qui se respecte doit implémenter ce concept.

L'héritage est la création de classes héritant des propriétés et des méthodes d'autres classes.



Le concept d'héritage

Javascript, implémente le concept d'héritage en utilisant le concept de prototype.

Personne
nom
prenom
getNom() : String
setNom(nom)
getPrenom() : String
setPrenom(prenom)
toString() : String

```
function Personne(){  
    this.nom="";  
    this.prenom="";  
  
    this.getNom = function(){ return this.nom; }  
    this.getPrenom = function(){return this.Prenom;}  
    this.setNom = function(nom){this.nom=nom;}  
    this.setPrenom = function(prenom){this.prenom=prenom;}  
  
    this.toString = function(){  
        return "Je m'appelle "+this.nom+" "+this.prenom;  
    }  
}
```

Constructeur

Propriétés

Méthode

Surcharge

Le concept d'héritage



Javascript, implémente le concept d'héritage en utilisant le concept de prototype.

```
function Personne(){
    this.nom="";
    this.prenom="";
    ...
}

var p1 = new Personne();
p1.setNom("Arthur");
p1.setPrenom("Guy");

console.log(p1);
console.log(p1.toLocaleString());
```

117

26_prototype.html:56

```
▼ Personne {nom: "Arthur", prenom: "Guy", getNom: f, getPrenom: f, setNom: f, ...} ⓘ
  ► getNom: f ()
  ► getPrenom: f ()
  nom: "Arthur"
  prenom: "Guy"
  ► setNom: f (nom)
  ► setPrenom: f (prenom)
  ► toString: f ()
  ▼ __proto__:
    ► constructor: f Personne()
    ▼ __proto__:
      ► constructor: f Object()
      ► hasOwnProperty: f hasOwnProperty()
      ► isPrototypeOf: f isPrototypeOf()
      ► propertyIsEnumerable: f propertyIsEnumerable()
      ► toLocaleString: f toLocaleString()
      ► toString: f toString()
      ► valueOf: f valueOf() function toString() { [native code] }
      ► __defineGetter__: f __definegetter_()
      ► __defineSetter__: f __definesetter_()
      ► __lookupGetter__: f __LookupGetter_()
      ► __lookupSetter__: f __LookupSetter_()
      ► get __proto__: f __proto__()
      ► set __proto__: f __proto__()
```

Je m'appelle Arthur Guy

26_prototype.html:57

Le concept d'héritage

```
function Personne(){...}

function Etudiant(){
    this.numEtudiant="";
}

Etudiant.prototype = new Personne();

var p1 = new Personne();
p1.setNom("Arthur"); p1.setPrenom("Guy");

var e1 = new Etudiant();
var e2 = new Etudiant();

console.log(e1); console.log(e2);

e1.setNom("Robinson"); e1.setPrenom("Crusoe");

e2.setNom("Kent"); e2.setPrenom("Clark");
console.log(p1);
console.log(p1.toLocaleString());
console.log(e1.toLocaleString());
console.log(e2.toLocaleString());
```

```
▼ Etudiant {numEtudiant: ""} ⓘ
  nom: "Robinson"
  numEtudiant: ""
  prenom: "Crusoe"
  ► __proto__: Personne

▼ Etudiant {numEtudiant: ""} ⓘ
  nom: "Kent"
  numEtudiant: ""
  prenom: "Clark"
  ► __proto__: Personne

▼ Personne {nom: "Arthur", prenom: "Guy", getNom: f, getPrenom: f}
  ► getNom: f ()
  ► getPrenom: f ()
  nom: "Arthur"
  prenom: "Guy"
  ► setNom: f (nom)
  ► setPrenom: f (prenom)
  ► toString: f ()
  ► __proto__: Object

Je m'appelle Arthur Guy
Je m'appelle Robinson Crusoe
Je m'appelle Kent Clark
```

Classe String



Classe String offre de nombreuses méthodes :

charAt(i)
charCodeAt(i)
concat(str)
endsWith(str)
fromCharCode(n1...nn)
includes(str)
indexOf(str)
lastIndexOf(str)
localeCompare(str)
match(regex)
repeat(count)
replace(search, replace)

Retourne le caractère à la position i
Retourne le caractère unicode à la position i
Concatène deux chaînes et retourne la chaîne concaténée
Retourne vrai si la chaîne termine par str faux sinon
Convertit les valeurs unicode en caractère
vérifie si la chaîne contient la sous chaîne str
retourne l'indice de la première occurrence trouvée de str
retourne l'indice de la dernière occurrence trouvée de str
Compare deux chaînes dans la locale courante
Cherche une expression régulière dans la chaîne de caractère
Répète la chaîne de caractère plusieurs fois
Cherche et remplace une chaîne/expression régulière par une autre chaîne

Classe String



Classe String offre de nombreuses méthodes :

charAt(i)
charCodeAt(i)
concat(str)
endsWith(str)
fromCharCode(n1...nn)
includes(str)
indexOf(str)
lastIndexOf(str)
localeCompare(str)
match(regex)
repeat(count)
replace(search, replace)

Retourne le caractère à la position i
Retourne le caractère unicode à la position i
Concatène deux chaînes et retourne la chaîne concaténée
Retourne vrai si la chaîne termine par str faux sinon
Convertit les valeurs unicode en caractère
vérifie si la chaîne contient la sous chaîne str
retourne l'indice de la première occurrence trouvée de str
retourne l'indice de la dernière occurrence trouvée de str
Compare deux chaînes dans la locale courante
Cherche une expression régulière dans la chaîne de caractère
Répète la chaîne de caractère plusieurs fois
Cherche et remplace une chaîne/expression régulière par une autre chaîne

Classe String



search(str)	Recherche la position d'une chaîne de caractère ou d'une expression régulière et retourne sa position
slice(start, end)	Extrait une partie de la chaîne et retourne la nouvelle chaîne
split(délimiteur)	Découpe une chaîne en sous chaîne par un délimiteur
startsWith(str)	Retourne vrai si la chaîne commence par str, faux sinon
substr(debut,length)	Retourne une sous chaîne entre une position de début sur une longueur donnée
substring(debut, fin)	Retourne une sous chaîne située entre deux indices, début et fin
toLocaleLowerCase()	Convertit une chaîne de caractère en minuscule dans la locale de l'hôte
toLocaleUpperCase()	Convertit une chaîne de caractère en majuscule dans la locale de l'hôte
toLowerCase()	Convertit une chaîne de caractère en minuscule
toString()	Retourne une chaîne de caractère décrivant l'objet
toUpperCase()	Convertit une chaîne de caractère en majuscule
trim()	Supprime les espaces avant et après la chaîne de caractère
valueOf()	Retourne la valeur primitive de l'objet string

Classe Date



Les constructeurs pour construire une Date :

```
var d = new Date();
d = new Date(milliseconds);
d = new Date(dateString);
d = new Date(year, month, day, hours, minutes, seconds, milliseconds);
```

Quelques méthodes

`Date.now()`
`d.getTime()`
`Date.parse(str);`
`d.getHours()`
`d.getMilliseconds()`
`d.getMinutes()`
`d.getMonth()`
`d.getSeconds()`

...

retourne la date courante au format milliseconds
d'une date existante retourne le nombre de milliseconde
à partir d'une chaîne de caractère retourne une date
retourne les heures au format 0-23
retourne les milliseconds 0-999
retourne les minutes 0-59
retourne les mois 0-11
retourne les secondes 0-59

Classe Regexp



Pour la création des Regexp, les expressions suivantes sont équivalentes

```
/ab+c/i;  
new RegExp('ab+c', 'i');  
new RegExp(/ab+c/, 'i');
```

Une description des différentes caractères spéciaux est présente ci-dessous :

- | | | | |
|-------------|--|----|------------------------------|
| - . | Tout les caractère sauf les caractères de fin de ligne | | |
| - \d | Tout les chiffres [0-9] | \D | Tout sauf les chiffre [^0-9] |
| - \w | Caractère AlphaNumerique [A-Za-z0-9_] | \W | L'inverse [^A-Za-z0-9_] |
| - \s | Les caractères d'espacement | \S | Les autres caractères |
| - \t | les tabulation | | |
| - \r | les retours chariot | | |
| - \n | les sauts de ligne | | |
| - \f | les sauts de page | | |
| - abc def | abc ou def | | |

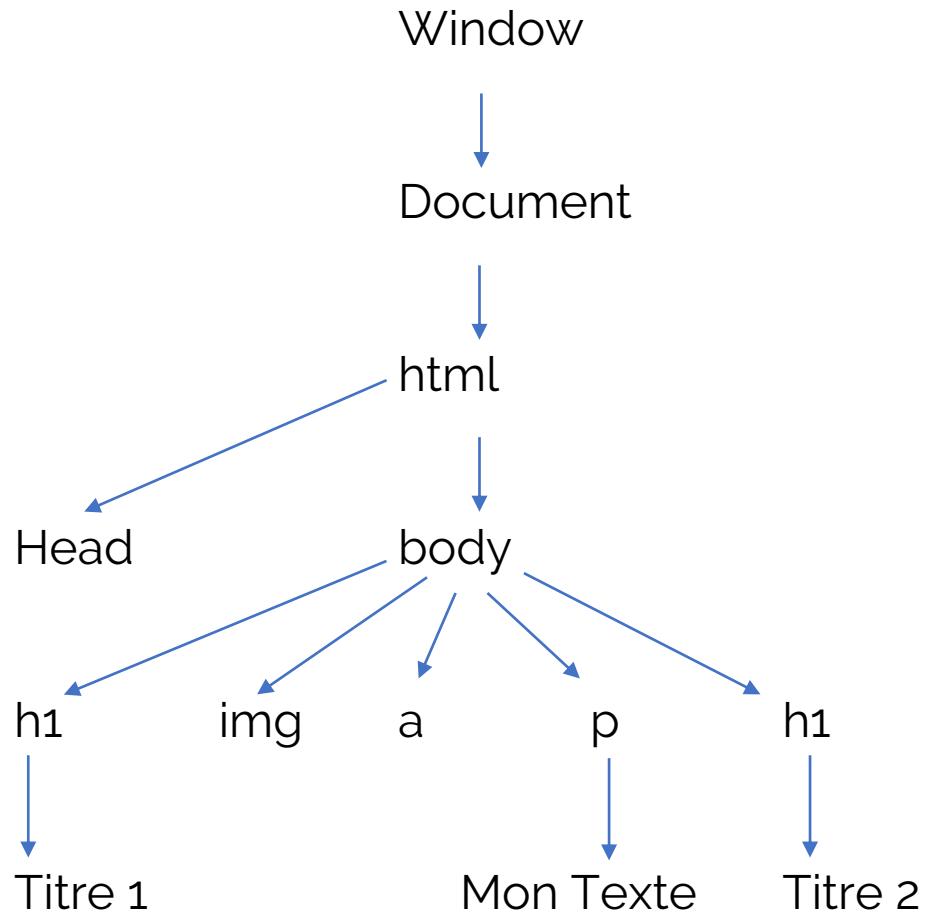
https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Objets_globaux/RegExp



Modifier une page et événements

" Cycle Développeur "

L'arbre DOM HTML



```
<body>
  <h1>Titre1</h1>
  <a href="www.google.fr">Google</a>
  <p>Mon Texte</p>
  <h1>Titre2</h1>
  
</body>
```

L'arbre DOM HTML



Il y a différents type de nœuds :

- Les éléments
- Les Textes
- Les commentaires
- Les attributs

`document.getElementById(id)`

`document.getElementsByName(name)`

`element.getElementsByClassName(classname)`

`element.children()`

`element.childNodes();`

Retrouve un élément par son id

Retrouve un élément par son nom

Retrouve un élément par le nom de la classe

Retourne la liste des enfants

Retourne la liste des nœuds enfants

Ajouter/modifier des Attributs

```
<body>
  <h1 id="titre1">Titre1</h1>
    <a id="ancre1" href="www.google.fr">Google</a>
    <p>Mon Texte</p>
  <h1 id="titre2">Titre2</h1>
  
</body>
```

Titre1

Google

Mon Texte

Titre2



```
var v = document.getElementById("ancre1");
console.log(v.attributes);
v.setAttribute("style","background-color:red;");
v.setAttribute("href","http://www.yahoo.fr");
```

Ajouter des Éléments

Pour créer des éléments avec les méthodes :

- `document.createElement(tagName)`
`element.appendChild(element)`

ou

- `element.innerHTML = "code html"`

```
var body = document.getElementsByTagName("body")[0];
var h1 = document.createElement('h1');
var text = document.createTextNode("Nouveau titre dynamique");

h1.appendChild(text);
body.appendChild(h1);
```

Ajoute un Noeud à
l'arborescence



Titre1

Google

Mon Texte

Titre2



Nouveau titre dynamique



Supprimer des Éléments

Titre1

Mon Texte

Titre2



Nouveau titre dynamique

```
var v = document.getElementById("ancre1");
console.log(v.attributes);
v.setAttribute("style","background-color:red;");
v.setAttribute("href","http://www.yahoo.fr");

var body = document.getElementsByTagName("body")[0];
var h1 = document.createElement('h1');
var text = document.createTextNode("Nouveau titre dynamique");

h1.appendChild(text);
body.appendChild(h1);
body.removeChild(v);
```

Ajouter des écouteurs



Nous pouvons écouter les événements en positionnant des écouteurs sur les éléments HTML. Se positionnement peut se faire de trois manières :

- Directement sur l'élément HTML
- En affectant la fonction sur l'attributs en JS
- En utilisant le Gestionnaire d'événements

Directement sur le HTML – Evénement souris

Nous pouvons positionner les attributs suivant sur les balises HTML :

```
<button onclick="myFunction()">Click me</button>
<div oncontextmenu="myFunction()" contextmenu="mymenu">
<p ondblclick="myFunction()">Double-click me</p>
<p onmouseup="mouseUp()">Click the text!</p>


<p onmousedown="myFunction()">Click the text!</p>

<div onmousemove="myFunction()">Move the cursor over me</div>

```

Gérer un événement



Titre1

[Google](http://www.google.fr)

Mon Texte

Titre2



```
Elements Console Sources Network Performance Memory »
top Filter Default levels ▾ 1 item hidden by filters


```
▶ MouseEvent {isTrusted: true, screenX: 54, screenY: 110, clientX: 53, clientY: 44, ...}
>
```


```

```
<body>
    <h1 id="titre1" onclick="appelClick(event)">Titre1</h1>
        <a id="ancre1" href="http://www.google.fr">Google</a>
        <p>Mon Texte</p>
    <h1 id="titre2">Titre2</h1>
    
</body>
<script type="text/javascript">
    function appelClick(event){
        console.log(event);
        alert("Hello");
    }
</script>
```

Gérer un événement



Titre1

[Google](http://www.google.fr)

Mon Texte

Titre2



```
Elements Console Sources Network Performance Memory »
top Filter Default levels ▾ 1 item hidden by filters


```

▶ MouseEvent {isTrusted: true, screenX: 54, screenY: 110, clientX: 53, clientY: 44, ...}
>
```


```

```
<body>
    <h1 id="titre1" onclick="appelClick(event)">Titre1</h1>
        <a id="ancre1" href="http://www.google.fr">Google</a>
        <p>Mon Texte</p>
    <h1 id="titre2">Titre2</h1>
    

</body>
<script type="text/javascript">

    var header = document.getElementById("titre1");
    header.onclick = function (event){
        console.log(event);
        alert("Hello");
    }
</script>
```

Les gestionnaires d'événements



Les gestionnaires d'événements sont des méthodes permettant d'ajouter des événements contrairement au cas précédent ou nous avons utilisé la propriété onClick.

```
element.addEventListener('click', function(event) {  
    ...  
});
```

Cette méthode nous permet aussi d'ajouter plusieurs événement sur le click d'un élément.

Tester les différents événements



1. Tester par des consoles log l'appel des différents événement souris.
2. Tester les événements d'entrées et de sortie dans un élément
3. Nous souhaitons marquer les éléments que l'on souhaite déplacer par une classe "movable"
4. Implémenter le déplacement des éléments

Directement sur le HTML – Événement Clavier

Nous pouvons positionner les attributs suivant sur les balises HTML :

```
<input type="text" onkeydown="myFunction()">  
<input type="text" onkeyup="myFunction()">  
<input type="text" onkeypress="myFunction()">
```

Tester les différents événements Claviers



1. Tester par des consoles log l'appel des différents événement clavier.
2. Observer la relation entre le focus et l'interaction clavier
3. Nous souhaitons marquer les éléments que l'on souhaite modifier par une classe "editable"
4. Implémenter l'édition d'éléments HTML

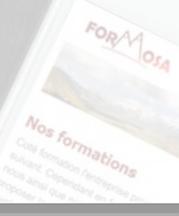
Directement sur le HTML – Les Événements Fenêtre

Nous pouvons positionner les attributs suivant sur les balises HTML :

```
<input type="text" onkeyup="myFunction()">
<body onunload="myFunction()">
<div onscroll="myFunction()">
<body onpagehide="myFunction()">
<body onpageshow="myFunction()">
<body onload="myFunction()">
<body onhashchange="myFunction()">

<body onbeforeunload="return myFunction()">
```

Directement sur le HTML – Les Formulaires



Nous pouvons positionner les attributs suivant sur les balises HTML :

```
<form onsubmit="myFunction () ">  
<input type="text" onselect="myFunction () ">  
<input type="search" onsearch="myFunction () ">  
<form onreset="myFunction () ">  
  
<input type="text" oninvalid="alert ('You must fill out the form! ') ;" required>  
<input type="text" oninput="myFunction () ">  
<input type="text" onfocusout="myFunction () ">  
<input type="text" onfocusin="myFunction () ">  
<input type="text" onfocus="myFunction () ">  
  
<select onchange="myFunction () ">  
<input type="text" onblur="myFunction () ">
```

Les formulaires



Nous pouvons utiliser javascript avec des formulaires

```
<form action="" id="myForm">
    <label for="nom">Nom:</label>
    <input name="nom" type="text">
    <label for="prenom">Prénom:</label>
    <input name="prenom" type="text">
    <input type="submit" value="Envoyer">
</form>
```

```
window.addEventListener('load', function(event) {
    var myForm = document.getElementById('myForm');
    myForm.addEventListener('submit', function(event) {
        ...
    });
});
```

Tester les différents événements Formulaire



- 1, Faire des opérations de contrôles sur les champs d'un formulaire age entre 18 et 60 ans
- 2, Format d'une adresse mail
3. Format de dates
- 4, Intercepter l'appel d'un submit pour effectuer les opérations de contrôle

Directement sur le HTML – Les Presse papiers

Nous pouvons positionner les attributs suivant sur les balises HTML :

```
<input type="text" oncopy="myFunction()" value="Try to copy this text">
<input type="text" oncut="myFunction()" value="Try to cut this text">
<input type="text" onpaste="myFunction()" value="Paste something in here">
```

Gérer les événements



On a la possibilité de traiter des événement. Nous pouvons configurer les événements directement dans le HTML ou dans le JS.

```
element.onclick = function(event){  
}
```

Les événements sont **annulable**

```
event.preventDefault()
```

Les événement utilise la **propagation**

La propagation peut être annulée

```
event.stopPropagation()
```

Gérer les événements - Event



Les propriétés

`event.type`

`event.target`

`event.currentTarget`

Événement Multi-navigateurs



Internet Explorer et le reste du monde

Dans le cadre des développements Web il est bien connu qu'Internet est le navigateur qui se distingue entre tous les navigateurs voir même entre différentes versions de lui-même.

Ainsi la majorité des problèmes d'incompatibilité viennent d'Internet Explorer avec les autres navigateurs.

Microsoft est leader sur le marché du Système d'exploitation pour les utilisateurs finaux. Et Internet Explorer est imposé à ceux-ci

Événement Multi-navigateurs



Objet Event (Modélisation)

Le problème principal d'incompatibilité est lié à l'objet Event. Cette objet possède en effet une modélisation différente sur Internet Explorer par rapport aux autres navigateurs :

Ce bout de code est compatible : Firefox et Chrome

```
alert('Vous avez cliqué sur le ' + event.target.value + ' bouton');
```

Ce bout de code est compatible : Internet Explorer et Chrome

```
alert('Vous avez cliqué sur le ' + window.event.srcElement.value + ' bouton');
```

Quelle solutions s'offrent à nous ?

Connaître le navigateur et sa version



Navigateur et Version

Une solution consiste à récupérer la version du navigateur et le nom du navigateur.

Puis en fonction de ces deux paramètres créer un code spécifique.

```
var myBrowser = navigator.appCodeName + " " + navigator.appVersion;
```

Cette solution longtemps utilisée n'est pas la meilleure solution. Elle peut cependant s'avérer utile dans des cas très spécifiques

Tester les propriétés existantes



Test les propriétés existantes

Ce test là est plus évolutif plutôt que de tester la version du navigateur on peut tester si la propriété ou la fonction que l'on souhaite appeler existe.

```
if (window.event) {
    //Nous sommes ici dans le cas d'internet explorer
    eventObject = window.event;
    eventTarget = window.event.srcElement;
}
else {
    //Nous sommes ici dans le cas d'autres navigateurs
    eventObject = event;
    eventTarget = event.target;
}
```

JQuery : Bibliothèque Cross Browser



Jquery : garantie cross browser

La dernière option est l'utilisation de bibliothèque qui elles sont CrossBrowser, c'est le cas par exemple de JQuery.

JQuery offre un moteur de requêtage remarquable se basant sur les sélecteurs CSS.

Il permet d'écrire moins de lignes de code qu'avec un code pur Javascript.

De plus de nombreuses bibliothèques comme Bootstrap ou encore D3JS se basent sur Jquery bénéficiant ainsi de sa propriété CrossBrowser et de son moteur de requêtage.



AJAX

" Cycle Développeur "

Ajax - Asynchronous JavaScript and XML



Ajax est un framework qui permet de communiquer avec un serveur et un serveur web de manière asynchrone..

Principalement AJAX permet de communiquer avec une API de WebService et de récupérer les résultats. Cela permet donc d'effectuer des appels sans recharger la page.

Intercepter les liens



En HTML :

```
<a href="#"> lien 1 </a>
<a href="#"> lien 2 </a>
<a href="#"> lien 3 </a>
```

Au click sur des liens nous pouvons intercepter l'appel :

```
event.preventDefault() // Nous pouvons arrêter le comportement par default
```

Nous pouvons ensuite effectuer l'appel avec AJAX :

```
var url = event.target.href;
var req = new XMLHttpRequest();
req.open("GET", url, true);
req.addEventListener('readystatechange', function(event) {
    ...
})
req.send();
```

Intercepter les liens



```
req.addEventListener('readystatechange', function(event) {  
    ...  
})
```

Une fois les données récupérées nous pouvons affecter ces données aux éléments du HTML.

Bloquer l'exécution d'un lien



```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>PREMIERS PAS AVEC AJAX</title>
  </head>
  <body>
    <a onclick="follow(event)" href="maPage.html">
      Lien vers une page HTML
    </a>
    <section id="result"></section>
  </body>
  <script type="text/javascript">
    function follow(event){
      event.preventDefault();
    }
  </script>
</html>
```

Bloquer l'exécution d'un lien



```
function follow(event){  
    event.preventDefault();  
    var cible = event.target.href;  
}
```

Ici nous bloquons le comportement par défaut du navigateur et nous récupérons le lien à charger.

Ajax



```
var req = new XMLHttpRequest();

req.open('GET', url, true);

req.send()

req.addEventListener('readystatechange', function(event) {

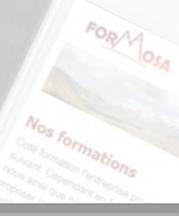
    ...

});
```

Nous requêtons le lien avec AJAX et récupérons le résultat. Pour finir nous affichons le résultat dans une DIV :

```
function follow(event){  
    event.preventDefault();  
    var cible = event.target.href;  
    var req = new XMLHttpRequest();  
    req.open('GET',cible, true);  
    req.addEventListener('readystatechange', function(event) {  
        console.log(event.currentTarget.response);  
        document.getElementById("result").innerHTML=event.currentTarget.response;  
    });  
    req.send();  
}  
}
```

Les formulaires



Il est possible d'intercepter les appels d'un formulaire de la même façon.

```
<form id="monFormulaire" action="maPage.php">  
    <label for="nom" >Nom : </label>  
    <input name="nom" type="text"/>  
    <input type="submit" value="envoyer"/>  
</form>
```

Dans le script JS :

```
var monFormulaire = Document.getElementById("monFormulaire");  
monFormulaire.addEventListener("submit", function(event){  
    ...  
});
```

Faire une application qui affiche la Météo



A l'aide du WebService de Météo de Yahoo,

<https://developer.yahoo.com/weather/>

Faire une application qui affiche la météo en fonction d'une liste de choix de ville de France.