**Cours Javascript, éditions ENI**

*Nota Bene:*

Ces extraits sont personnellement complétés afin de servir de base pour développer et se former au Javascript. En terme de volume, ils ne dépassent ici pas plus de 1% du livre (218 pages). Il y a volontairement des oublis par rapport à ce qui m'est déjà connu. Le site de référence largement mentionné ici est *W3Schools.com*

**2. Insertion du code JavaScript**

Langage sensible à la casse

**Commentaires**

<script type="text/javascript">

//Ceci est un commentaire monoligne

</script>

<script type="text/javascript">

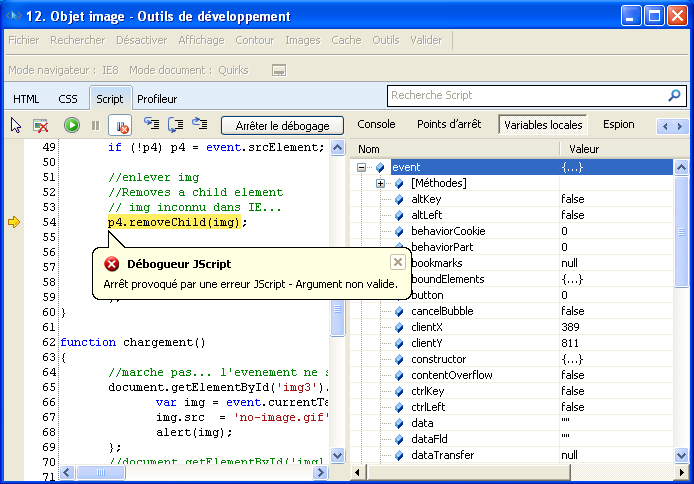
/\* Ceci est un commentaire

Sur plusieurs lignes \*/

</script>

**Outil de développement Internet Explorer**

Intégré à Internet Explorer 8, il permet de débuger le Javasript (onglet Script, Démarrer le débogage). L'intérêt est alors de consulter le contenu des variables locales. Il est accessible via le menu *Outils* ou *F12*.



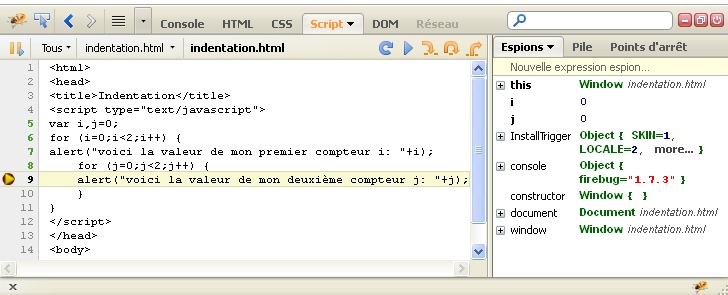
**Console Web Firefox**

Intégrée à Firefox, une console web permet d'afficher les erreurs et avertissement Javascript. Elle est accessible via le menu *Outils*.

**Firebug**

Extension de Firefox utile pour debugger des scripts Javascript. <https://addons.mozilla.org/fr/firefox/addon/firebug/>

L'intérêt est de rajouter des points d'arrêt (onglet Script, marge de gauche) afin d'observer le contenu des variables (onglet Espions):



Il est aussi possible d'executer les instructions en mode "pas à pas" (icones flechées orange).

**jQuery Lint**

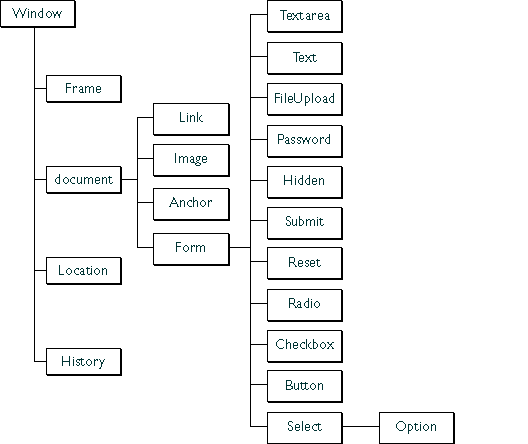
Script pour utiliser conjointement à jQuery et Firebug. Il permet d'afficher des messages d'erreur dans la console de Firebug. <http://james.padolsey.com/javascript/jquery-lint/>

**3. Initiation à la Programmation Orientée Objet et JavaScript**

**DOM: Document Object Model**

Le DOM est une API fondée sur les arbres (qui contient l'ensemble des objets/noeuds (*nodes*) de la page web).

Diagramme de hiérarchie des objets de Window:



Dans une page comportant un formulaire nommé *formu*, contenant lui-même un champ *nom*, si l'on désire accéder à ce champ, la syntaxe suivante sera utilisée:

document.formu.nom;

En plus d'atteindre un objet précis dans une hiérarchie, la syntaxe pointée désigne l'accès à des méthodes ou des propriétés qui permettent de manipuler ou de décrire ces objets.

L'objet *window* étant le plus élevé dans la hiérarchie, il n'est pas nécessaire de le nommer à chaque manipulation.

**confirm()**

Affiche une boite de dialogue avec les boutons *OK* et *Annuler*. Renvoie respectivement *true* et *false*.

<script language="javascript">

var reponse=window.confirm("ok = true, Annuler = false");

alert("la valeur de la variable réponse est: "+reponse);

</script>

**Exercice 1**: lors de la fermeture de la page, détecter l'évènement et demander la confirmation. Si on annule, la page reste ouverte.

**Solution**: [Exercices/1.confirm\_fermer\_page.html](http://papierpeintox.free.fr/formation_javascript/Exercices/1.confirm_fermer_page.html)

**4. Utilisation des constantes, variables et opérateurs**

**Constantes**

constante\_numerique=l.45226;

constante\_texte="ma constante texte";

**isNaN()**

is Not a Number: renvoie true si la valeur n'est pas un nombre.

**Variables**

JavaScript est faiblement typé c'est-à-dire que les variables n'ont pas besoin de correspondre à un type (texte, numérique, booléen) pour fonctionner.

Une variable est *locale* (définie dans une fonction), ou *globale* (définie dans le script).

Déclaration:

var mavariable1, mavariable2, mavariable3;

Les variables n'ayant pas reçu de valeurs, elles sont dites *undefined* (par exemple, les arguments qui sont oubliés lors d'un appel de fonction); test avec *typeof*

var numerique=l5;

var texte="ma variable";

var booleen=false;

document.write("La variable numerique est de type: "+ typeof numerique +"<br />"); // number

document.write("La variable texte est de type: "+ typeof texte +"<br />"); // string

document.write("La variable booleen est de type: "+ typeof booleen +"<br />"); // boolean

**Conversion de chaînes en nombres**

var variablenumerique=parselnt(variabletexte);

var variablenumerique=parseFloat(variabletexte);

**Conversion de nombres en chaînes**

var variabletexte=variablenumerique.toString();

var n = 123456.789;

n.toFixed(0); // "123457"

n.toFixed(2); // "123456.79"

**Opérateurs de comparaison**

var egal = 1 == '1'; // true

var strictement\_egal = 1 === '1'; // false

var different = 1 != '1'; // false

var strictement\_different = 1 !== '1'; // true

var indefini = undefined == null; // true

var strictement\_indefini = undefined === null; // false

**Test**: [Exercices/2.operateurs\_stricts.html](http://papierpeintox.free.fr/formation_javascript/Exercices/2.operateurs_stricts.html)

***delete*, *new* et *this***

Respectivement, supprime une variable, crée une instance d'un objet et référence l'objet courant.

**5. Structures de contrôle**

***if* ternaire**

suite=confirm("Voulez-vous poursuivre? ");

(suite==true)? alert("J'en suis heureux") : alert("C'est dommage");

***else if***

Avec un espace entre les deux mots...

***for ... in***

Pour manipuler un objet (et non un tableau!).

var tabProprietes = new Array();

var i = 0;

for (var propriete in document) {

tabProprietes[i] = propriete;

i++;

}

**Test**: [Exercices/3.for\_in\_document.html](http://papierpeintox.free.fr/formation_javascript/Exercices/3.for_in_document.html)

**6. Fonctions et évènements**

**L'instruction return**

Retourner le résultat d'une fonction. Si la fonction n'a rien à renvoyer, la valeur *undefined* sera retournée.

**Utiliser une fonction pour créer un objet**

méthode du prototypage:

function voiture (propMarque, propModele, propAnnee) {

this.marque=propMarque;

this.modele=propModele;

this.annee=propAnnee;

}

var mavoiture=new voiture("Peugeot","307","2007");

alert("Voici les caractéristiques de mon objet voiture :"

+ "\n la marque : " + mavoiture.marque + "\n le modèle : " + mavoiture.modele + "\n l'année: " + mavoiture.annee);

**Test**: [Exercices/8.creer\_objet\_prototype.html](http://papierpeintox.free.fr/formation_javascript/Exercices/8.creer_objet_prototype.html)

**Evènements**

Trois méthodes pour créér des évènements:

<script language="javascript">

function MaFonction()

alert("Ceci est ma première fonction");

</script>

<input type="submit" name="Submit" value="Fonctionner" onClick="MaFonction()" />

window.OnLoad=alert("Cette fenêtre s'affiche au moment du chargement de la page");

Troisième méthode: l'enregistrement de gestionnaire d'évènement:

Appel de la méthode *addEventListener()* pour un élément particulier. Trois arguments:  
1. nom du type d'évènement, sans le "on" (ex.: *click* pour *onClick*)  
2. fonction gestionnaire  
3. booleen (false, gestionnaire normal)

Le mot clé *this* se réfère à l'élément sur lequel l'évènement est survenu. Pour la troisième méthode, on préfèrera la propriété *currentTarget* de l'objet *Event* qui est passé à la fonction gestionnaire.

function getEventTrigger(event) {

var x=event.currentTarget;

alert("The id of the triggered element: " + x.id);

}

document.getElementById('p3').addEventListener('mousedown', getEventTrigger, false);

**Ressource**:  
<http://stackoverflow.com/questions/807878/javascript-that-executes-after-page-load>  
<http://www.w3schools.com/jsref/dom_obj_event.asp>  
<http://www.w3schools.com/html5/html5_ref_eventattributes.asp>

**7. Les formulaires**

**Exemple**: [Exercices/5.formulaire\_reservation.html](http://papierpeintox.free.fr/formation_javascript/Exercices/5.formulaire_reservation.html)

**8. Les principaux objets JavaScript en détail**

**Objet Window**

**Ressource**:  
<http://www.w3schools.com/jsref/obj_window.asp>

**Exercice**: afficher les propriétés de la fenetre.

**Solution**: [Exercices/6.objet\_window.html](http://papierpeintox.free.fr/formation_javascript/Exercices/6.objet_window.html)

**Objet Document**

**Ressource**:  
<http://www.w3schools.com/jsref/dom_obj_document.asp>

**Exercice**: afficher les propriétés du document.

**Solution**: [Exercices/7.objet\_document.html](http://papierpeintox.free.fr/formation_javascript/Exercices/7.objet_document.html)

**Objet String**

**Ressource**:  
<http://www.w3schools.com/jsref/jsref_obj_string.asp>

**Exercice**: afficher la propriété longueur et utiliser les méthodes courantes.

**Solution**: [Exercices/9.objet\_string.html](http://papierpeintox.free.fr/formation_javascript/Exercices/9.objet_string.html)

**Objet Date**

getter: fonctions qui commencent par get\*

setter: fonctions qui commencent par set\*

conversion: fonctions qui commencent par to\*

**Ressource**:  
<http://www.w3schools.com/jsref/jsref_obj_date.asp>

**Exercice**: afficher la date du jour en français; calculer un intervalle de temps; executer une fonction à intervalle régulier

**Solution**: [Exercices/10.objet\_date.html](http://papierpeintox.free.fr/formation_javascript/Exercices/10.objet_date.html)

**Objet Array**

**Ressource**:  
<http://www.w3schools.com/jsref/jsref_obj_array.asp>

**Exercice**: afficher la propriété longueur et utiliser les méthodes courantes.

**Solution**: [Exercices/11.objet\_array.html](http://papierpeintox.free.fr/formation_javascript/Exercices/11.objet_array.html)

**Objet DOM Image**

**Ressource**:  
<http://www.w3schools.com/jsref/dom_obj_image.asp>  
<http://www.w3schools.com/jsref/dom_obj_all.asp>

**Exercice**: afficher les dimensions; créér un effet rollover; utiliser *appendChild()* et *removeChild()* pour afficher une image dynamiquement.

**Solution**: [Exercices/12.objet\_image.html](http://papierpeintox.free.fr/formation_javascript/Exercices/12.objet_image.html)

**Objet Regexp**

var patt=new RegExp(pattern,modifiers);

//or more simply:

var patt=/pattern/modifiers;

**Ressource**:  
<http://www.w3schools.com/jsref/jsref_obj_regexp.asp>

**Exercice**: chercher s'il y a "[\*]" dans une chaine

**Solution**: [Exercices/13.objet\_regexp.html](http://papierpeintox.free.fr/formation_javascript/Exercices/13.objet_regexp.html)

**Objet Math**

**Ressource**:  
<http://www.w3schools.com/jsref/jsref_obj_math.asp>

**Exercice**: Loto - utilisation de *random()*, *ceil()* et *eval()*.

**Solution**: [Exercices/14.objet\_math\_loto.html](http://papierpeintox.free.fr/formation_javascript/Exercices/14.objet_math_loto.html)

**9. Améliorer l'interactivité avec JavaScript et CSS**

**JavaScript et les cookies**

document.cookie="moncookie=mon nom";

alert(document.cookie) ;

**Ressource**:  
<http://www.w3schools.com/js/js_cookies.asp>

**Interaction JavaScript/CSS**

Effet onmouseover: penser à la pseudo classe CSS *:hover*

**Exercice**: Créér un menu déroulant et utiliser la classe *:hover*

**Solution**: [Exercices/15.css\_hover.html](http://papierpeintox.free.fr/formation_javascript/Exercices/15.css_hover.html)