

JAVASCRIPT C'EST QUOI ?

- Langage de script indépendant de toute plateforme.
- Son utilisation la plus connue est dans les pages web.
- Il est aussi très utilisé dans le développement sous la plateforme NodeJS.
- Il est aussi utilisé dans les documents « pdf ». Dans des logiciels comme « photoshop » etc

JAVASCRIPT DANS UNE PAGE WEB

JavaScript est utilisé en complément des pages HTML pour :

- Accéder à des informations variables telles que la date, cookie, etc.
- Interagir avec les utilisateurs par le biais des éléments de l'interface graphique.
- Validation de formulaires avant leurs soumissions.
- Gestion de l'interface utilisateur. (menus dynamiques, panneaux, onglets, carousels ...)
- etc.

JAVASCRIPT - WEB

◆ JavaScript est :

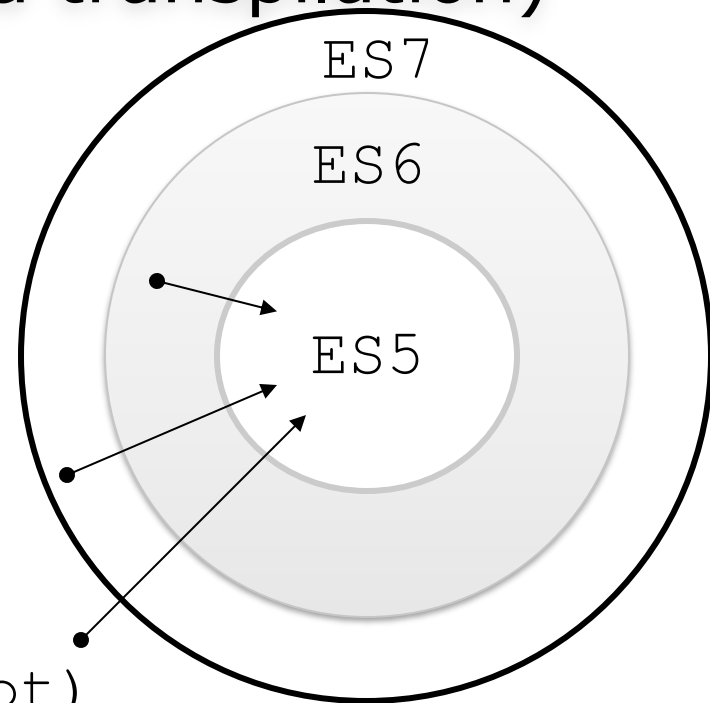
- un langage orienté objet (à sa façon)
- destiné à créer des scripts
- interprété par le moteur de script du navigateur
 - **NS, FF -> spiderMonkey , rhino, ...**
 - **Chrome -> V8 (Utilisé dans NodeJS et ...)**
 - **Opera -> Carakan**
 - **IE -> Chakra**
 - **Safari -> SquirrelFish/Nitro**
- JavaScript n'a rien à voir avec Java
- Créé par et intégré à NetScape en 1996 (LiveScript)

JAVASCRIPT - WEB

- ◆ standardisé par l'ECMA = organisme de standardisation, en 1997 (Suisse)
 - Norme de base ECMAScript-262
 - ECMAScript 4 = ES4 (juin 2007) **Désuet**
 - ECMAScript 5 = ES5 (déc 2009) **Courant**
 - ECMAScript 6 = ES6 (Harmony juin 2015) **En cours d'implémentation**
 - ECMAScript 7 = ES7 Elaboration des nouvelles spécifications

PEUT-ON UTILISER ES6 MAINTENANT ?

- ◆ OUI. En faisant appel à un transpileur tel que « babel » pour compiler ES6 vers ES5.
- ◆ JS n'étant pas typé, ajoute le support des types (la commande **tsc** permet la transpilation)
- ◆ Vision globale :

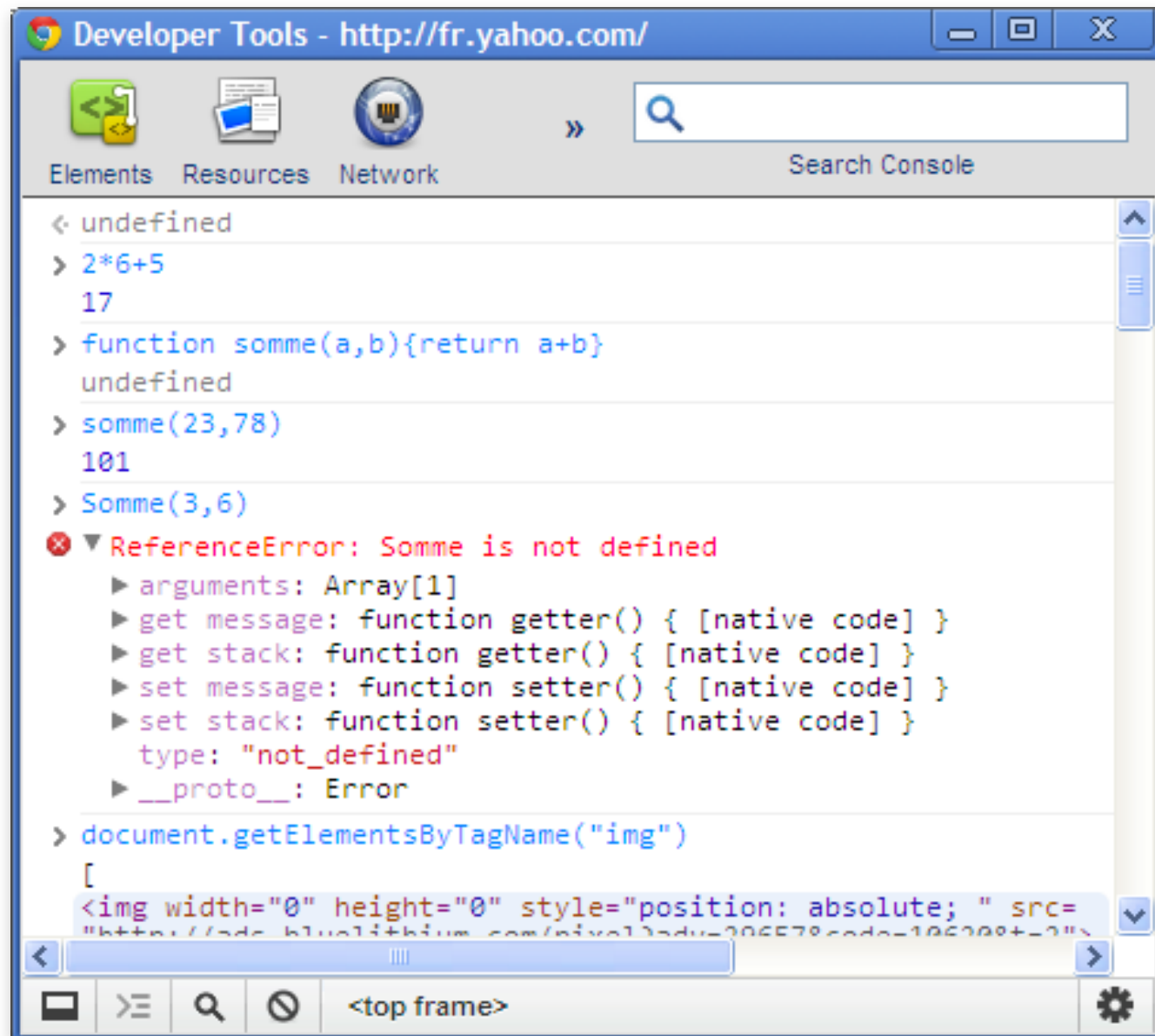


DE QUOI A-T-ON BESOIN ?

- ◆ Javascript est du texte (Interprété)
- ◆ On a besoin d'un bon éditeur de texte
 - Quelques IDE osent (!)
 - **Aptana**
 - **Komodo**
 - **Brackets**
 - **Eclipse**
 - **webStorm**
 - **Visual Studio Code**
 - JS n'étant pas typé, les IDE font ce qu'il peuvent pour assister le développeur.

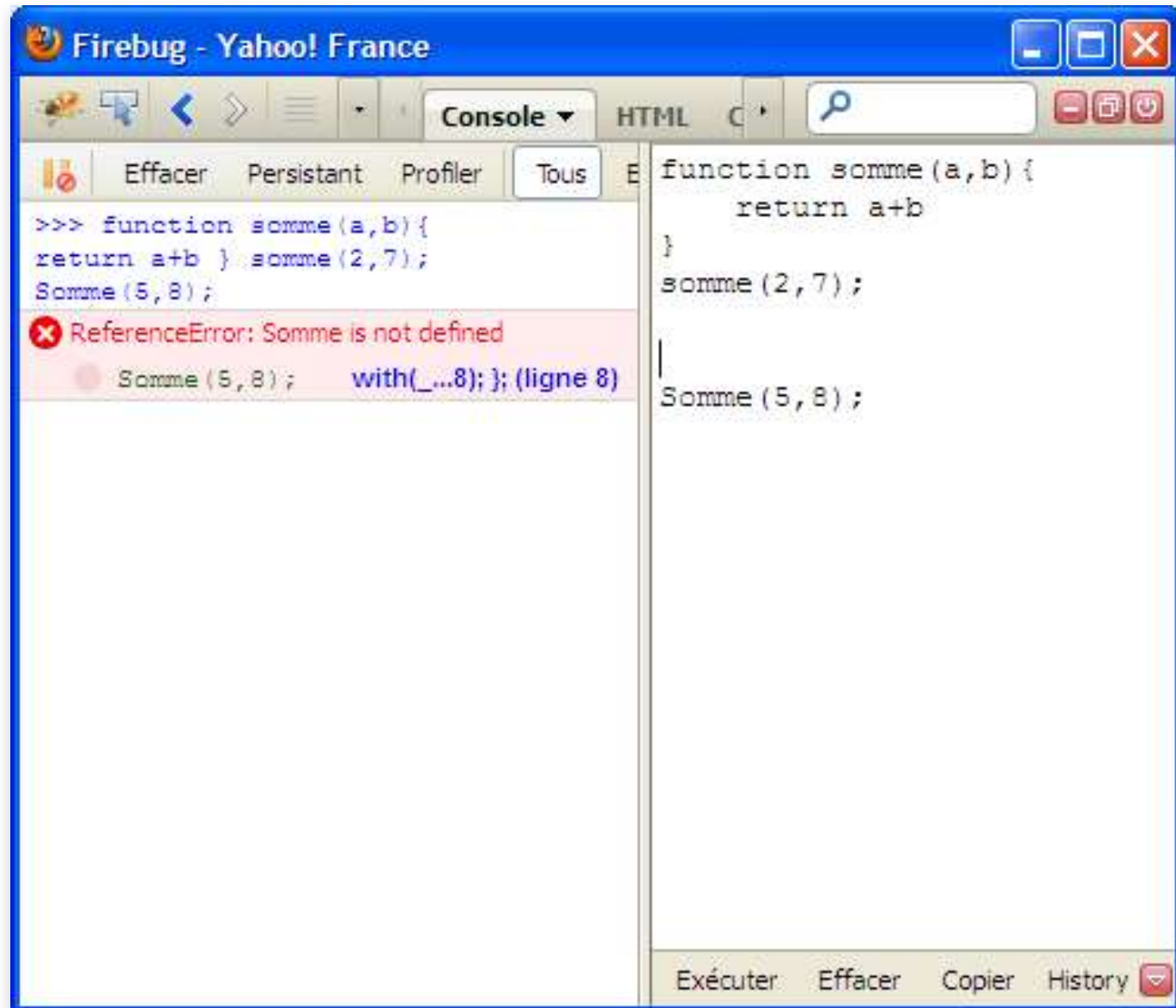
OUTILS D'AIDE AU DÉVELOPPEMENT

GOOGLE CHROME



OUTILS D'AIDE AU DÉVELOPPEMENT

L'extension Firebug



EXEMPLE 1

INSÉRER UN SCRIPT DANS DU CODE HTML

```
<html>
<head>
</head>
<body>
<br />
Ceci est une page HTML.
<br />
```

```
<script>
  document.write("Ceci est un script JavaScript
                 inséré dans cette page !")
```

```
</script>
<br />
Le HTML reprend la main.
</body>
</html>
```

On signale le script à l'aide de la balise script

On appelle la méthode « write » de l'objet « document »

EXEMPLE 2

AJOUTER L'INTERACTIVITÉ À UNE PAGE HTML

```
<html>
<head>
<script type="text/JavaScript">
    function bonjour() { alert("Bonjour!"); }
</script>
</head>
<body>
    <input type="button" name="unBouton"
        value="Cliquez ici"
        onClick="bonjour()">
</body>
</html>
```

Définition de la
fonction
« bonjour () »
« alert() »
affiche un
message

L'événement « onclick » est
un attribut HTML.
Au click, on appelle la
fonction « bonjour() »

EXEMPLE 3

RÉCUPÉRER DES DONNÉES TRANSMISES PAR UN UTILISATEUR

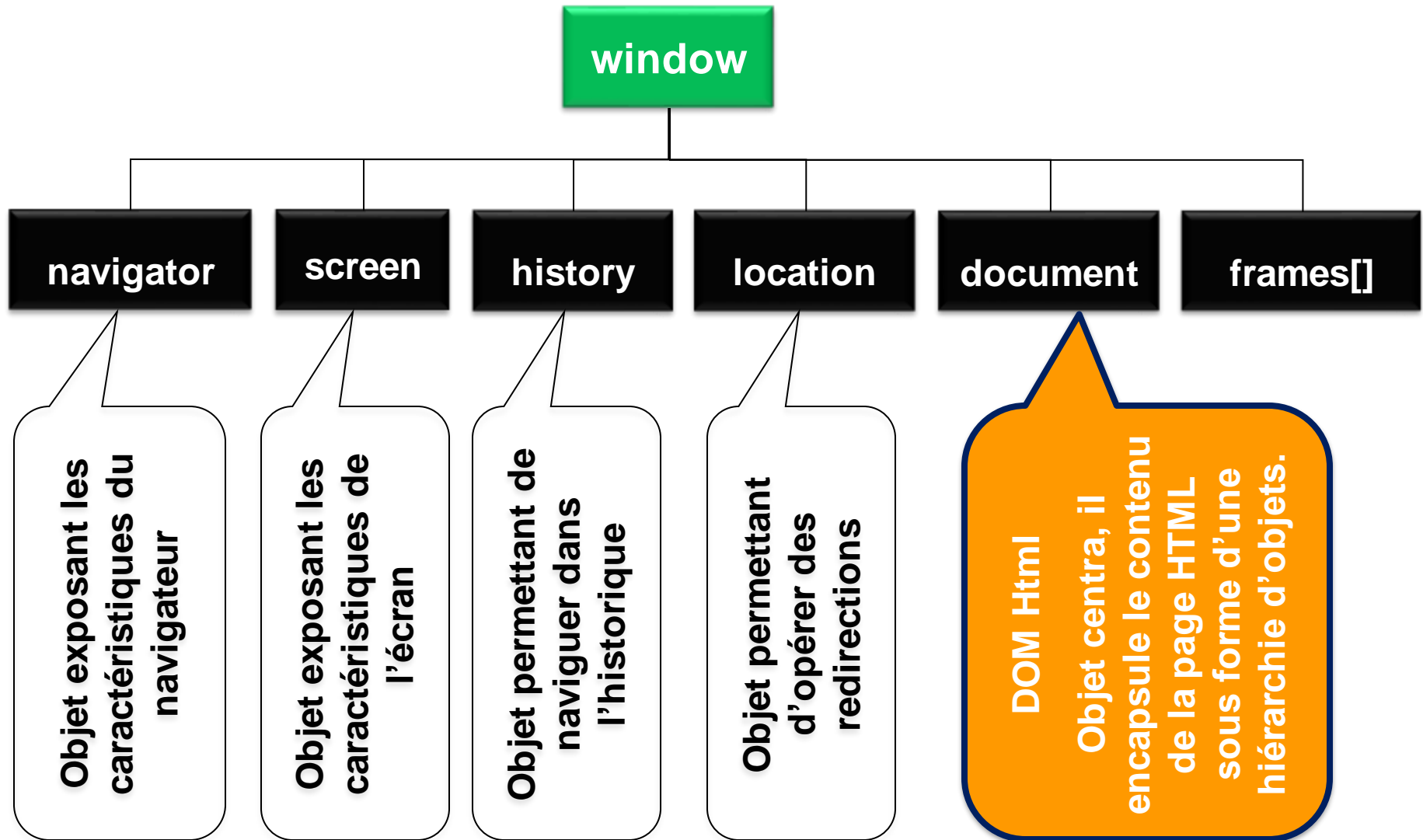
```
<html>
<head>
<script type="text/javascript">
  function nom(chaine) {
    alert("Bonjour, " + chaine + "!");
  }
</script>
</head>
<body>
  Entrez votre nom s'il vous plait:
  <input type="text" name="EntreNom"
    onBlur="nom(this.value)" value="">
</body>
</html>
```

L'opérateur « this » fait référence à l'objet courant.

Ici la balise « input »

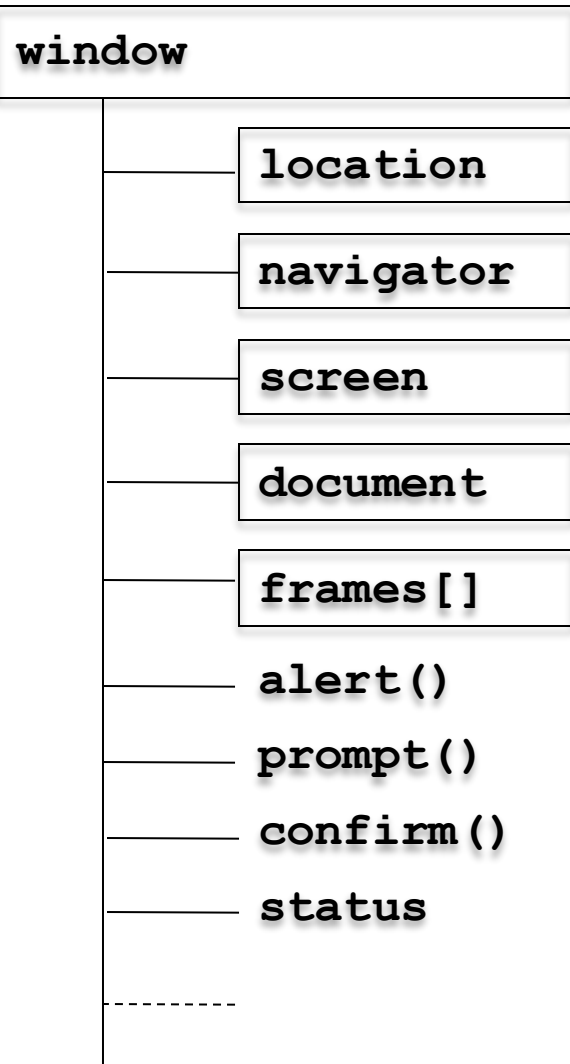
L'attribut « value » correspond à l'attribut html de la balise input.

BOM = BROWSER OBJECT MODEL

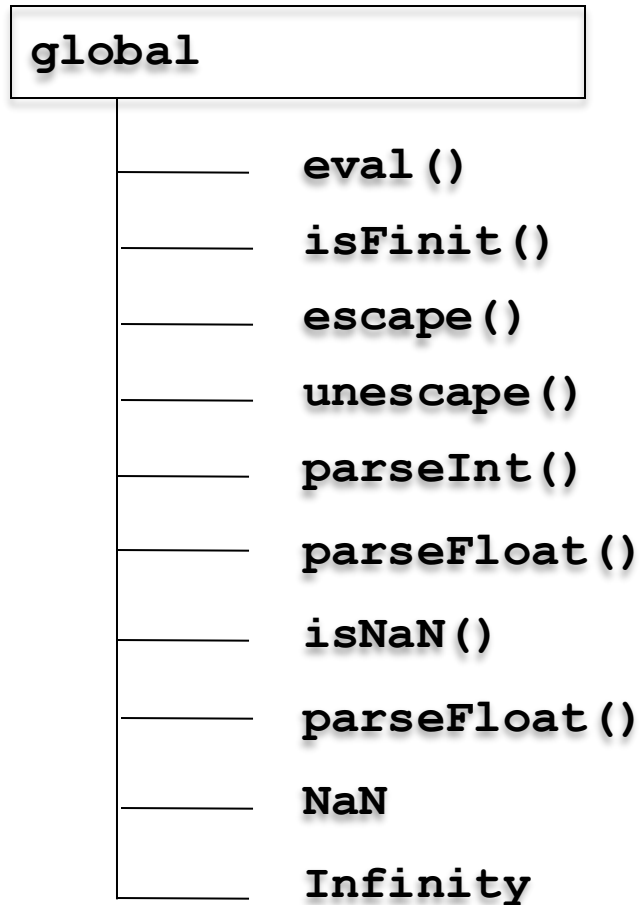


BROWSER COMME APPLICATION HÔTE

BROWSER



JavaScript Engine



BROWSER COMME APPLICATION HÔTE

JavaScript Engine

`window = self = global`

`location`

`navigator`

`screen`

`document`

`frames[]`

`alert()`

`prompt()`

`confirm()`

`status`

`eval()`

`isFinite()`

`escape()`

`unescape()`

`parseInt()`

`parseFloat()`

`isNaN()`

`parseFloat()`

`NaN`

`Infinity`

`variable_globale`

`Fonction_utilisateur`

L'OBJET WINDOW

- ◆ A chaque chargement de la page (ex. F5) l'environnement Javascript est **réinitialisé**.
- ◆ Les scripts embarqués dans la page se greffent à l'objet window.
 - Les variables globales deviennent des attributs de window
 - Les fonctions utilisateur deviennent des méthodes de window
- ◆ L'objet **document** encapsule le contenu de la page HTML sous forme d'un arbre d'objets.
- ◆ Voici les caractéristiques des objets window et document (Réf. => <http://www.w3schools.com>)

OBJET WINDOW

Property	Description
<u>closed</u>	Returns a Boolean value indicating whether a window has been closed or not
<u>defaultStatus</u>	Sets or returns the default text in the statusbar of a window
<u>document</u>	Returns the Document object for the window (See Document object)
<u>frames</u>	Returns an array of all the frames (including iframes) in the current window
<u>history</u>	Returns the History object for the window (See History object)
<u>innerHeight</u>	Sets or returns the the inner height of a window's content area
<u>innerWidth</u>	Sets or returns the the inner width of a window's content area
<u>length</u>	Returns the number of frames (including iframes) in a window
<u>location</u>	Returns the Location object for the window (See Location object)
<u>name</u>	Sets or returns the name of a window
<u>navigator</u>	Returns the Navigator object for the window (See Navigator object)
<u>opener</u>	Returns a reference to the window that created the window
<u>outerHeight</u>	Sets or returns the outer height of a window, including toolbars/scrollbars
<u>outerWidth</u>	Sets or returns the outer width of a window, including toolbars/scrollbars
<u>pageXOffset</u>	Returns the pixels the current document has been scrolled (horizontally) from the upper left corner of the window
<u>pageYOffset</u>	Returns the pixels the current document has been scrolled (vertically) from the upper left corner of the window
<u>parent</u>	Returns the parent window of the current window
<u>screen</u>	Returns the Screen object for the window (See Screen object)
<u>screenLeft</u>	Returns the x coordinate of the window relative to the screen
<u>screenTop</u>	Returns the y coordinate of the window relative to the screen
<u>screenX</u>	Returns the x coordinate of the window relative to the screen
<u>screenY</u>	Returns the y coordinate of the window relative to the screen
<u>self</u>	Returns the current window
<u>status</u>	Sets the text in the statusbar of a window
<u>top</u>	Returns the topmost browser window

OBJET WINDOW (SUITE)

Method	Description
<u>alert()</u>	Displays an alert box with a message and an OK button
<u>blur()</u>	Removes focus from the current window
<u>clearInterval()</u>	Clears a timer set with setInterval()
<u>clearTimeout()</u>	Clears a timer set with setTimeout()
<u>close()</u>	Closes the current window
<u>confirm()</u>	Displays a dialog box with a message and an OK and a Cancel button
<u>createPopup()</u>	Creates a pop-up window
<u>focus()</u>	Sets focus to the current window
<u>moveBy()</u>	Moves a window relative to its current position
<u>moveTo()</u>	Moves a window to the specified position
<u>open()</u>	Opens a new browser window
<u>print()</u>	Prints the content of the current window
<u>prompt()</u>	Displays a dialog box that prompts the visitor for input
<u>resizeBy()</u>	Resizes the window by the specified pixels
<u>resizeTo()</u>	Resizes the window to the specified width and height
<u>scroll()</u>	
<u>scrollBy()</u>	Scrolls the content by the specified number of pixels
<u>scrollTo()</u>	Scrolls the content to the specified coordinates
<u>setInterval()</u>	Calls a function or evaluates an expression at specified intervals (in milliseconds)
<u>setTimeout()</u>	Calls a function or evaluates an expression after a specified number of milliseconds

L'OBJET DOCUMENT

Collection	Description	W3C
<u>anchors[]</u>	Returns an array of all the anchors in the document	Yes
<u>forms[]</u>	Returns an array of all the forms in the document	Yes
<u>images[]</u>	Returns an array of all the images in the document	Yes
<u>links[]</u>	Returns an array of all the links in the document	Yes
Property	Description	W3C
<u>cookie</u>	Returns all name/value pairs of cookies in the document	Yes
<u>documentMode</u>	Returns the mode used by the browser to render the document	No
<u>domain</u>	Returns the domain name of the server that loaded the document	Yes
<u>lastModified</u>	Returns the date and time the document was last modified	No
<u>readyState</u>	Returns the (loading) status of the document	No
<u>referrer</u>	Returns the URL of the document that loaded the current document	Yes
<u>title</u>	Sets or returns the title of the document	Yes
<u>URL</u>	Returns the full URL of the document	Yes
Method	Description	W3C
<u>close()</u>	Closes the output stream previously opened with document.open()	Yes
<u>getElementById()</u>	Accesses the first element with the specified id	Yes
<u>getElementsByName()</u>	Accesses all elements with a specified name	Yes
<u>getElementsByTagName()</u>	Accesses all elements with a specified tagname	Yes
<u>open()</u>	Opens an output stream to collect the output from document.write() or document.writeln()	Yes
<u>write()</u>	Writes HTML expressions or JavaScript code to a document	Yes
<u>writeln()</u>	Same as write(), but adds a newline character after each statement	Yes

EXEMPLE DE REPRESENTATION

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head><title>Ma page XHTML
```

```
</title></head>
```

```
  <body>
```

```
    <h1>Contenu d'entête</h1>
```

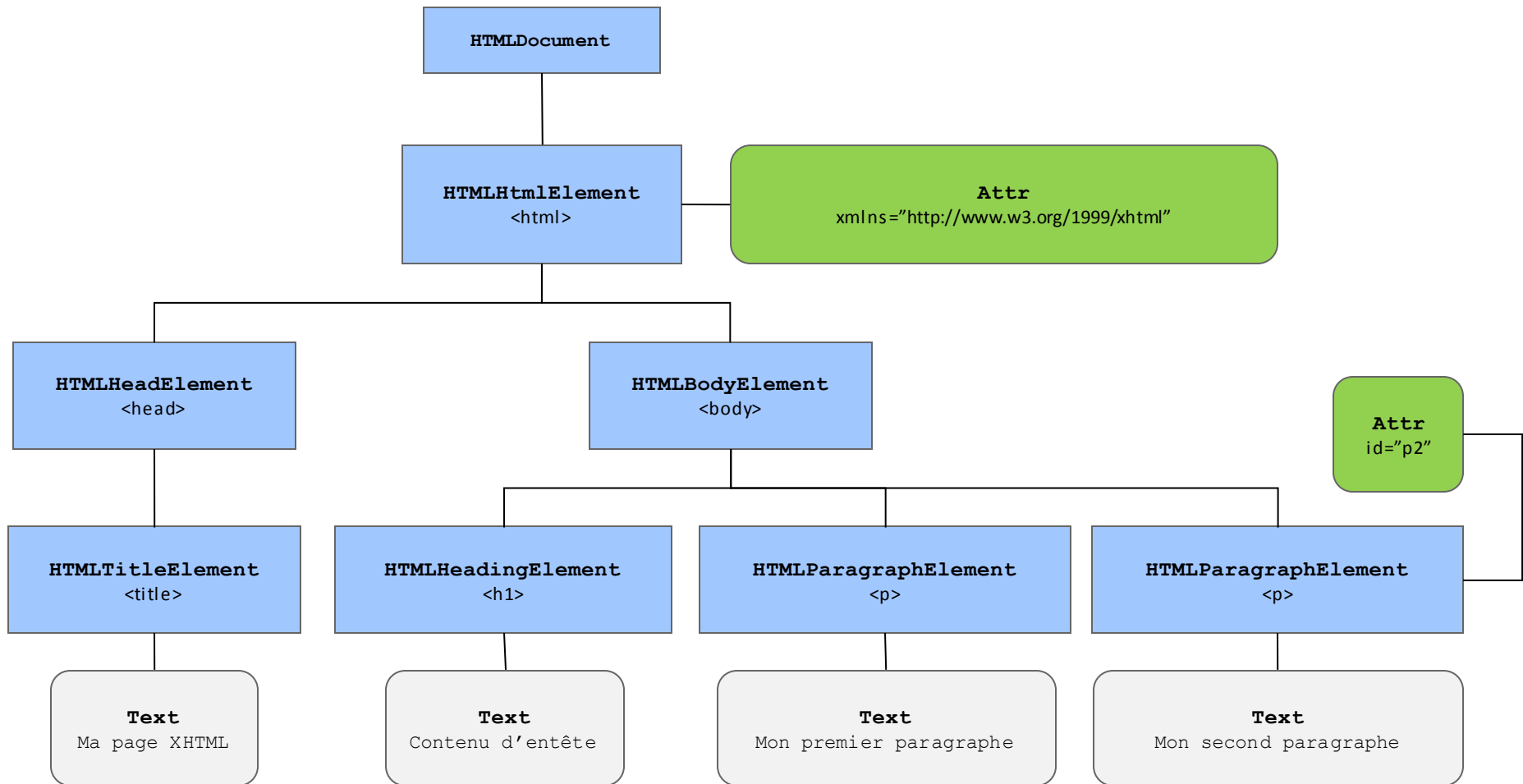
```
    <p>Mon premier paragraphe</p>
```

```
    <p id="p2">Mon second paragraphe</p>
```

```
  </body>
```

```
</html>
```

EXEMPLE DE REPRÉSENTATION (SUITE)



DOCUMENT INTERFACE - NODE INTERFACE

◆ Accéder aux éléments du DOM

- `document.querySelector("Slecteur CSS");` // retourne un élément
- `document. querySelectorAll("Slecteur CSS") ;` // retourne liste d'éléments
- `document.getElementById("id")`
- `document.getElementsByTagName("tr")`
- `document.getElementsByName("nom")`
- `document.getElementsByClassName("class")`

◆ Ces attributs retournent un nœud ou un tableau de nœuds (NodeList). Ces Nœuds (éléments) pourront ensuite être manipulés :

- `node.NodeChilds` (Liste des enfants)
- `node.FirstChid` (Premier enfant)
- `node.innerHTML` (Ecrire ou lire du HTML dans l'élément)
- `node.attributes` (Liste des attributs de l'élément)