Institut National Universitaire Champollion

Programmation événementielle

JAVASCRIPT

Elisabeth Pecatte

elisabeth.pecatte@iut-tlse3.fr

D’après le cours de Sylvie Trouilhet / Jean-Marie Pecatte

Institut National Universitaire Champollion

1- Généralités

- 1995 – Brendan Eich de Netscape Communications (Netscape > Firefox de Mozilla)
- Langage de programmation de script
- Interprété et exécuté dans un navigateur
- https://developer.mozilla.org/en-US/docs/Web/JavaScript

- Standard ECMAScript Edition 5 **(ES5)** [décembre 2009]
- Standard ECMAScript Edition 6 (ES5) [juin 2015]

Institut National Universitaire Champollion

1- Généralités

Classement des langages de programmation

- TIOBE *(moteurs de recherche)* :
  javascript en 6ème position*(01/2018)*
- REDMONK *(site stack overflow)* :
  javascript en 1ère position (*06/2017*)
- IEEE Spectrum *(classement le plus complet)* : javascript en 6ème

Institut National Universitaire Champollion

2. Scripts en JavaScript

- **Dans le fichier HTML :**
- **Dans l'entête du document** (entre <head> </head>)

  Le code Javascript n'est pas exécuté au chargement, il est juste stocké dans le navigateur
- **Dans le corps du document** (entre <body> </body>)

  Le code JavaScript est exécuté une fois lors du chargement de la page dans le navigateur.

Pour le distinguer du code HTML, dans ces 2 cas, le code javascript doit être saisi entre les 2 balises <script> </script>

```
<script type="text/javascript" >
...
</script>
```

type facultatif en HTML5

Institut National Universitaire Champollion

2. Scripts en JavaScript

- **Dans des balises HTML** : permet d'exécuter des bouts de code javascript quand certains événements (chargement de la page, clic de souris, …) surviennent

  mon lien

  Mauvaise pratique : ne pas utiliser !

- **Dans un fichier externe**
- Le fichier est au format ".js"

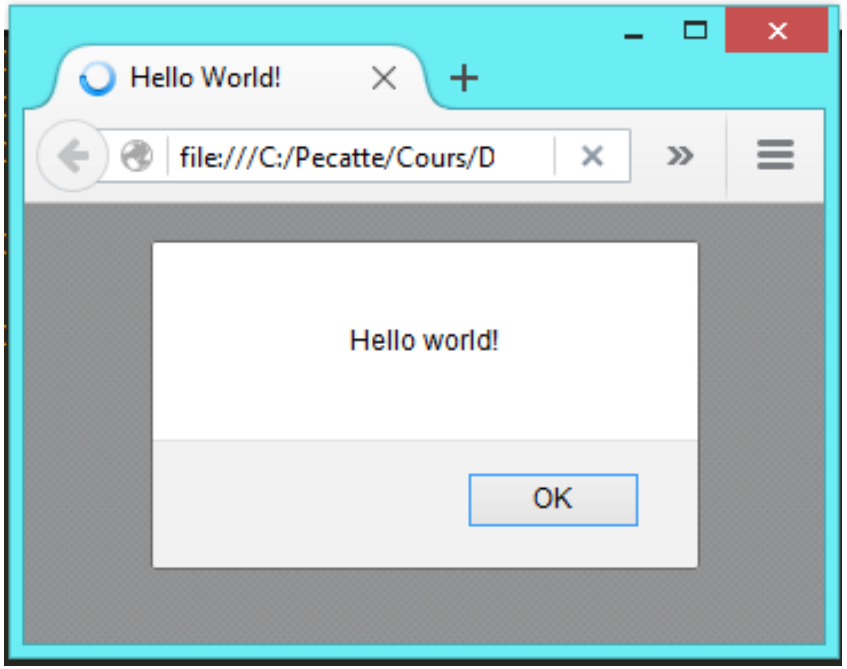
  Il faut ensuite appeler le fichier dans le HTML, à l'aide de la balise :

<script type="text/javascript" src="script.js" ></script>

2. Scripts en JavaScript

```
<!DOCTYPE html>
<html>
<head>
    <title>Hello World!</title>
</head>
<body>
    <script>
        alert('Hello world!');
    </script>
</body>
</html>
```

L'écran d'affichage des résultats est la fenêtre du navigateur



3. Structures de base du langage

3.1 Les commentaires

```
// commentaire sur une ligne        // imbrication possible

/* commentaire sur une ou
        plusieurs lignes */          /* sans imbrication */
```

3. Structures de base du langage

3.2 Mots-clés réservés (liste non exhaustive)

boolean	break	byte		
case	char	const		
default	delete	do	double	
else	enum			
false	final	float	for	function
if	in	int	interface	
long				
new	null			
package	private	protected	public	
return				

3. Structures de base du langage

3.2 Mots-clés réservés (liste non exhaustive)

short	static	switch
this	true	typeof
var	void	
while	with	

NB : distinction entre majuscule et minuscule

3. Structures de base du langage

Convention d'écriture

CamelCase (casse de chameau) : la première lettre de chaque mot est en majuscule

MonMot

Et plus précisément du **lowerCamelCase** : la première lettre du premier mot reste en minuscule

monMot

Identificateur de fonction : **lowerCamelCase**

Identificateur de variable : **lowerCamelCase**

Identificateur de constantes : **TOUT_EN_MAJUSCULES**

3. Structures de base du langage

3.3 Variable

Déclaration : variable désignée par le mot-clé **var** et un identificateur

```
var maVariable ;  // maVariable a la valeur undefined
```

Types de valeurs :

type de base	valeurs	exemple
Boolean	true et false	
Number	valeur numérique sur 64 bits (entiers et réels)	10 -3 3.14 0723 (octal) 0x723 (hexadécimal)
String	chaîne de caractères (caractère codé sur 16 bits)	"bonjour" 'bonjour' "\t"

3. Structures de base du langage

3.3 Variable (suite)

Déclaration et affectation d'une variable

- dans un script : variable globale
- à l'intérieur d'une fonction : variable locale
  (portée limitée à la fonction)

Valeurs équivalentes à faux :

false, null, undefined, '', 0, NaN (Not a Number)

Déclaration d'une constante : const

```
const NB_MOIS=12 ;
```

3. Structures de base du langage

3.4 Les opérateurs du langage :

- arithmétiques **+ - * / % *(modulo)***
- de comparaison **< <= > >= == !=**
- logiques ***&& (et)*** **||** ***(ou)*** **!** ***(négation)***
- opérateur **typeof** pour connaître le type d'une variable :

  'number' 'string' 'boolean' 'undefined' 'function' 'object'
- concaténation : **+**

"salut" + "monde" ➡ **"salutmonde"**

3. Structures de base du langage

3.5 Instructions

- Instruction simple : terminée par un point virgule
- Séquence d'instructions : Blocs d'instructions délimitées par des accolades **{ }**
- Structure de sélection :

```
if ( condition )
    {      ...          }
```

```
if ( condition )
    {      ...          }
else
    {      ...          }
```

3. Structures de base du langage

Exemple : afficher la mention d'une note (P, AB, B, TB)

```
if ( note >= 16 )
      {   alert("Très bien") ;   }
else {
    if ( note >= 14 )
          {   alert("Bien") ;   }
   else  {
         if ( note >= 12 )
               {  alert("Assez bien") ;   }
         else {  alert("Passable") ;   }
   }
}
```

3. Structures de base du langage

Structures de répétition :

```
while ( condition )
    {       ...         }
```

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

```
for (var i=1; i<=10; i=i+1 )
    {       ...         }
```

```
for ( variable in objet )
    {       ...         }
```

3. Structures de base du langage

Exercice : calculer la somme des 10 premiers nombres entiers positifs

```
var somme = 0 ;
var nombre = 1 ;
while (nombre <= 10) {
  somme = somme + nombre ;
  nombre = nombre + 1 ;
}
```

```
var somme = 0 ;
for ( var nombre = 1 ; nombre <= 10 ; nombre = nombre + 1 )
 { somme = somme + nombre ; }
```

3. Structures de base du langage

Branchement multiple :

```
switch ( expression )
    {
      case label1 :      …      ; break ;
      case label2 :      …      ; break ;
      default :   …   ;           break ;
    }
```

3. Structures de base du langage

3.5 Les tableaux

- Tableau = regroupement de plusieurs valeurs de même type (ensemble de notes, de produits, …)
- Déclaration

```
var tableau = [ ] ;   // déclaration d'un tableau vide
var tNotes = [12,20,11] ; // déclaration d'un tableau contenant 3 notes ; les indices des cases sont 0,1,2

var tab ; // déclaration d'une variable, pas encore un tableau

tab = [ ]; // initialisation de la variable tab comme un tableau vide
```

3. Structures de base du langage

3.5 Les tableaux (suite)

var tNotes = [12,20,11] ;

- Utilisation

```
alert( tNotes[1] ) ; // affiche 20
tNotes[1] =  34 ; // modifie la valeur de la case d'indice 1
alert( tNotes[1] ) ; // affiche 34
alert( tNotes.length) ; // affiche 3, la taille du tableau
```

3. Structures de base du langage

Exercice : créer un tableau contenant les valeurs 9,11,2,21,16

chercher la valeur la plus petite

```
var tabInt = [9, 11, 2, 21, 16] ;
```

```
var petite = tabInt[0] ;
for ( var i = 1 ; i < tabInt.length ; i ++) {
      if ( tabInt[i] < petite ) { petite = tabInt[i] ; }
}
```

3. Structures de base du langage

3.6 Les objets

Objet : regroupement des propriétés *(des attributs)* qui caractérisent une entité *(les propriétés d'un produit)*

→ Objets existants (prédéfinis) : ils ne sont pas à déclarer

Exemple : l'objet **document** correspond au document ouvert dans un onglet d'un navigateur

→ Objets créés : il faut les déclarer

Déclaration

```
var objet = { attribut1 : valeur1, attribut2 : valeur2, …} ;
```

Exemple :

```
var etudiant1 = { nCarte : 1, nom : "toto", masculin : true } ;
```

3. Structures de base du langage

3.6 Les objets (suite)

Utilisation : notation pointée pour indiquer l'objet pour lequel on veut la propriété

```
alert(etudiant1.nom) ;   // affiche toto
```

3. Structures de base du langage

3.7 Les fonctions

- Une fonction regroupe un ensemble d'instructions que l'on peut utiliser plusieurs fois dans un programme

```
function nom(arguments) { … }
```

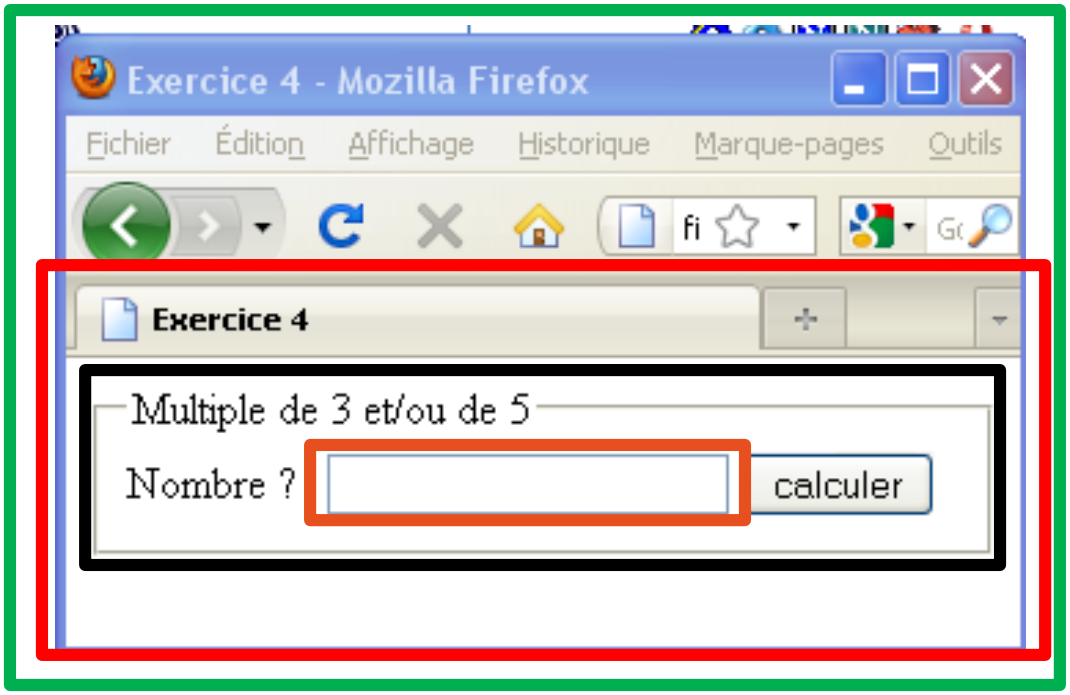
Exemple :

```
function somme(a,b) { return a + b ; }
        // l'instruction return définit la valeur renvoyée
result = somme(6,8) ;
document.write("le resultat est " + result) ;
```

4. Objets prédéfinis

4.1 Généralités

- Javascript est un langage basé sur des objets ; tous les éléments du navigateur et du document HTML sont des objets.



4. Objets prédéfinis

4.1 Généralités (suite)

- Les objets sont organisés sous la forme d'une arborescence
- Pour accéder à un objet, il faut donner le chemin d'accès depuis la racine (window)

Exemple : **window.document.form.valeur**

- Pour accéder à un objet du document HTML, il est aussi possible de le rechercher si l'élément à un **ID en HTML**, méthode getElementById()

```
<img id="img1" src="…" />
var img=document.getElementById("img1");
```

4. Objets prédéfinis

4.2 DOM (Document Object Model)

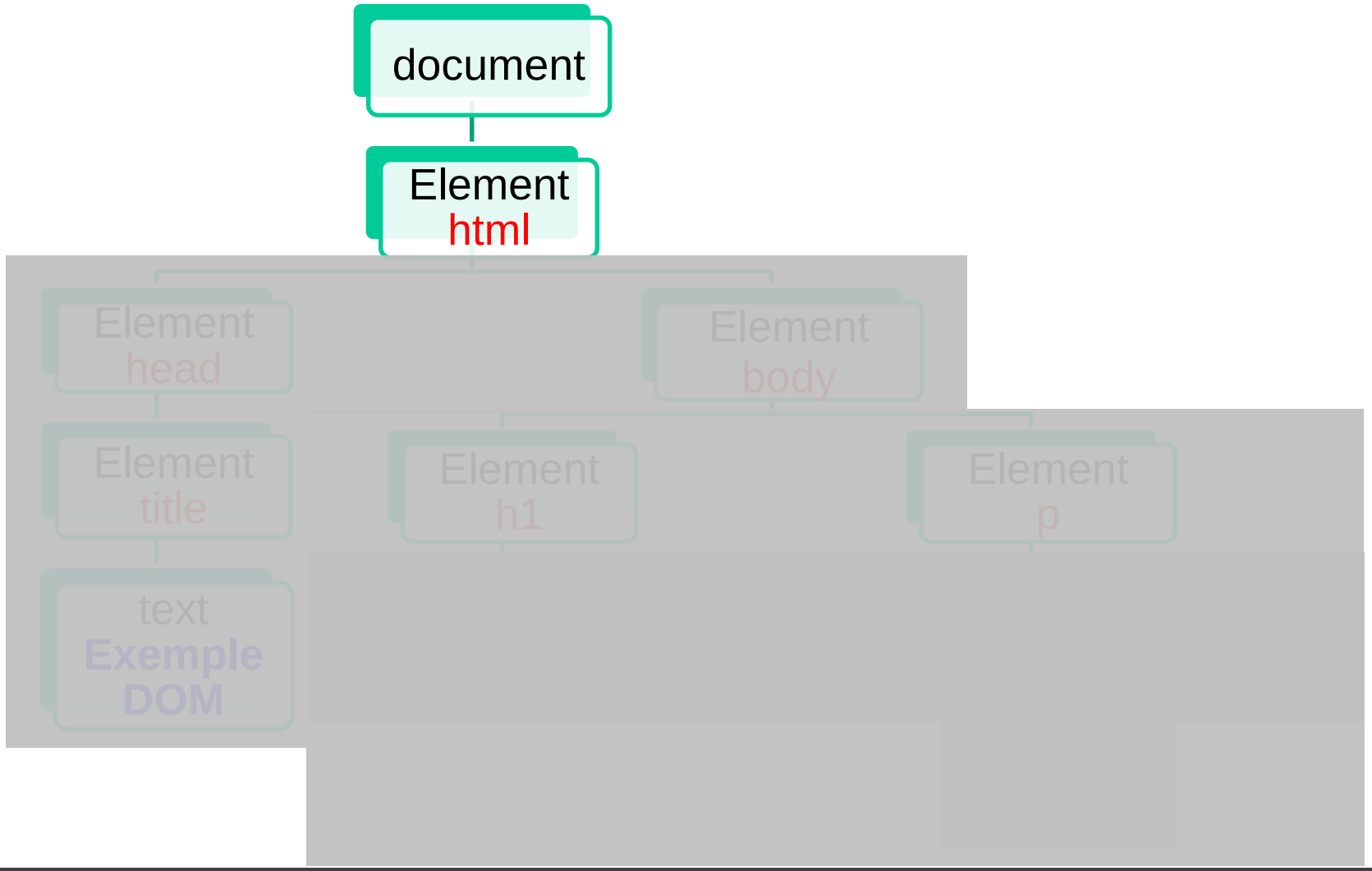
- modèle standardisé par le W3C *(World Wide Web Consortium)*
- représente un document sous la forme d'un arbre
- toutes les balises HTML sont donc des nœuds de l'arbre
- les feuilles sont soit des balises sans contenu, soit le texte de la page HTML
- plusieurs types de nœud :
  1. Nœud élément (balise HTML)
  2. Nœud attribut (attribut HTML)
  3. Nœud texte (contenu d'un élément)

4. Objets prédéfinis

```
<!DOCTYPE html>
<html>
<head>
    <title>Exemple DOM</title>
</head>
<body>
        <h1 class="titre">Titre</h1>
        <p>Début<img src="img1.jpg"/>fin</p>
</body>
</html>
```

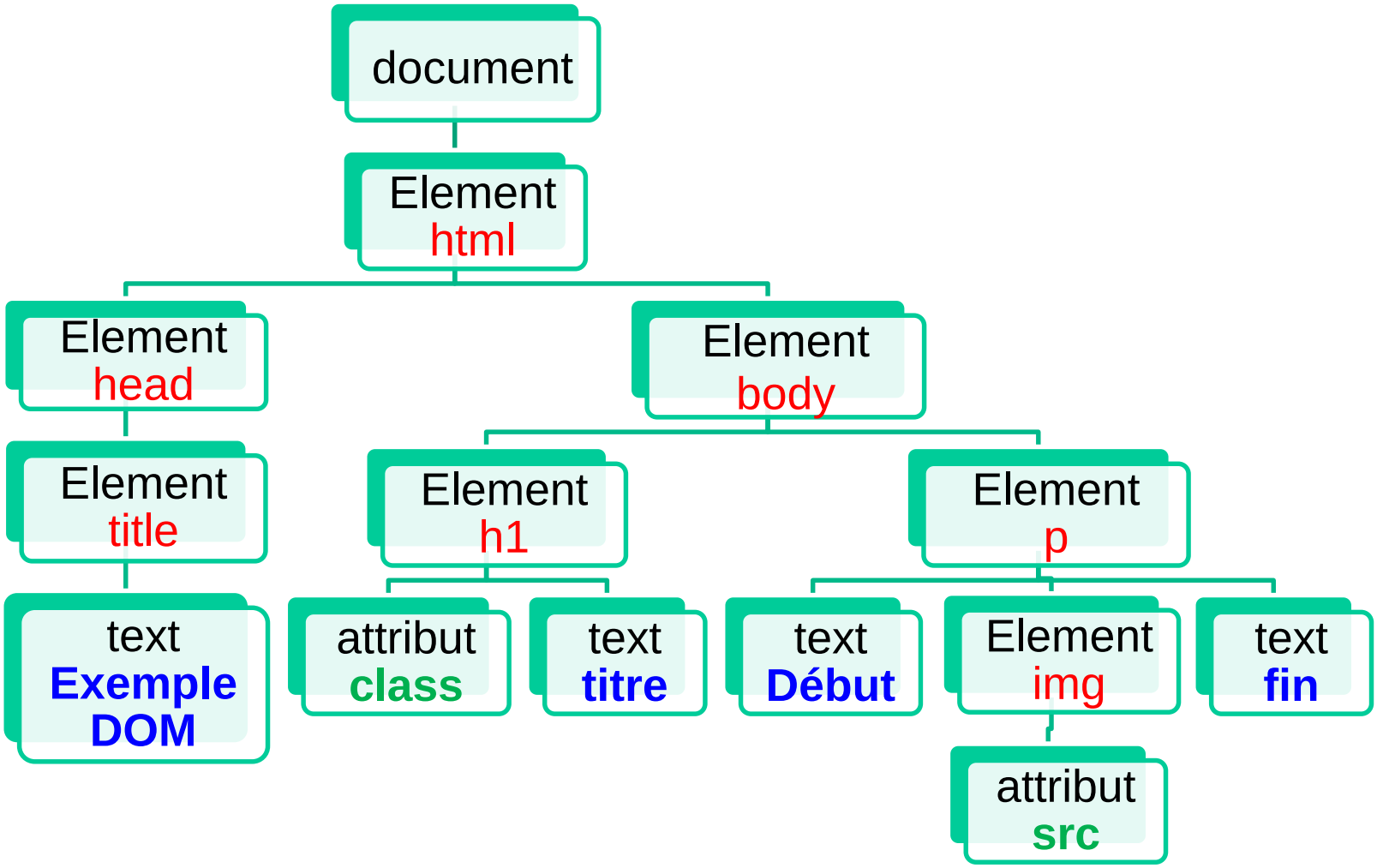
Institut National Universitaire Champollion

4. Objets prédéfinis



Institut National Universitaire Champollion

4. Objets prédéfinis



4. Objets prédéfinis

4.2 DOM (gestion en javascript)

- rechercher un élément à partir de son ID *(attribut id HTML)*

```
<img id="img1" src="…" />
var img=document.getElementById("img1");
```

- rechercher d'éléments à partir d'un sélecteur

```
<div class="myclass" > … </div>
var div=document.querySelector(".myclass");
```

- childNodes → tableau des fils d'un objet
- parentNode → nœud parent
- nextSibling → prochains frères