

Plan du reste du cours

1

• CSS

2

• Java script

3

• PHP

CHAPITRE 3 : JAVASCRIPT

Web dynamique ?

- Le modèle du Web présentée précédemment est **statique**. Les documents sont stockés sous forme de fichiers physiques, sur le disque dur d'un serveur.
- Très tôt on a souhaité générer **dynamiquement** le contenu d'une page.
- La programmation Web **côté serveur** évolue ensuite (apparition de PHP en 1994, puis possibilité ensuite de programmer le côté serveur dans des langages plus robustes, comme Java, ...)
- Un problème subsiste : le manque d'interactivité. En effet, on est contraint par le modèle :
formulaire HTML ↔ envoi au serveur ↔ calcul de la réponse ↔ retour au client ↔ rechargement de page. Problème d'interactivité (latence réseau, rendu graphique d'une nouvelle page, ...).

Web dynamique côté client

- Avec l'arrivée de Java (1995) la notion d'Applet fait son apparition. Ils sont (pour l'époque) une manière portable d'exécuter du code côté client.
- Problème : Java est trop lour à l'époque (c'est un vrai langage, il fait peur aux créateurs de site, les performances sont médiocres, ...).
- 1995 : Brendan Eich (Netscape) crée Javascript en 10 jours. Il emprunte de la syntaxe à Java/C, et Netscape Navigator 2.0 embarque un interpréteur Javascript en standard
- Le langage est rapidement adopté, mais chaque navigateur implémente sa propre variante. Le langage lui-même est standardisé en 1996 (ECMAScript, standardisé par l'ECMA).

Introduction (1)

- Javascript permet de rendre dynamique un site internet développé en HTML.
- Javascript permet de développer de véritables applications fonctionnant exclusivement dans le cadre d'Internet.
- Le Javascript est un langage de script simplifié orienté objet dont la syntaxe est basée sur celle du Java.
- Javascript a été initialement élaboré par Netscape en association avec Sun Microsystem.
 - Plus tard, Microsoft développera son propre langage Javascript officiellement connu sous le nom de *JScript*.

Introduction (2)

- Contrairement à un applet Java qui est un programme compilé, les scripts écrits en Javascript sont **interprétés**
 - Le Java, représenté par un ou plusieurs fichiers autonomes dont l'extension sera **.class* ou **.jar*, est invoqué par une balise HTML spécifique
 - Le JavaScript est **écrit directement au sein du document HTML** sous forme d'un script encadré par des balises HTML spéciales.
- Javascript est standardisé par un comité spécialisé, l'ECMA (European Computer Manufacturers Association).

Le saviez-vous?

- **JavaScript** est le langage de programmation le plus populaire au monde.
- JavaScript est le langage de programmation du Web.
- JavaScript est facile à apprendre.
- **JavaScript** et Java sont des langages complètement différents, à la fois dans le concept et la conception.
- JavaScript a été inventé par Brendan Eich en 1995 et est devenu une norme ECMA en 1997.

Pourquoi étudier JavaScript?

JavaScript est l'un des **3 langages** que tous les développeurs Web **doivent** apprendre:

1. HTML pour définir le contenu des pages Web
2. CSS pour spécifier la mise en page des pages Web
3. **JavaScript** pour programmer le comportement des pages Web

Questions fréquemment posées

- ? Comment obtenir JavaScript ?
- ? Où puis-je télécharger JavaScript ?
- ? JavaScript est-il gratuit ?
- **Vous n'avez pas besoin d'obtenir ou de télécharger JavaScript.**
- **JavaScript est déjà en cours d'exécution dans votre navigateur sur votre ordinateur, sur votre tablette et sur votre smartphone.**
- **JavaScript est gratuit pour tous.**

Java script Exécution

- Côté client : le code javascript est exécuté par le navigateur Web. Il doit donc être référencé dans une page HTML :

- Soit en utilisant l'attribut src d'une balise script (externe)
- Soit en mettant le code directement dans une balise script (interne)

```

3 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" >
4 ...
5 <script type="text/javascript" src="toto.js"></script>
6 ...
7 <script type="text/javascript">
8   alert("Hello, World!");
9 </script>
10 </html>

```

- Côté serveur : on peut maintenant utiliser Javascript comme un langage généraliste grâce à l'interpréteur Node.js

JavaScript Où aller

- La balise **<script>**
 - En HTML, le code JavaScript est inséré entre les balises **<script>** et **</script>**.

Exemple 1

- JavaScript est le langage de script par défaut en HTML.

Exemple 2

Fonctions et événements JavaScript

- Un JavaScript **function** est un bloc de code JavaScript, qui peut être exécuté lorsqu'il est «appelé» pour.
- Par exemple, une fonction peut être appelée lorsqu'un **événement** se produit, comme lorsque l'utilisateur clique sur un bouton.

Liste non exhaustive d'événements :

```

Abort //l'utilisateur interrompt le chargement de l'image
Change //l'utilisateur modifie le contenu d'un champ de données.
Click //l'utilisateur clique sur l'élément associé à l'événement
Dbclick //l'utilisateur double clique sur l'élément associé à l'événement
Dragdrop //l'utilisateur effectue un glisser déposer sur la fenêtre
Error //se déclenche lorsqu'une erreur apparaît
Focus //l'utilisateur donne le focus à un élément
Keydown //l'utilisateur appuie sur une touche de son clavier
keypress //l'utilisateur maintient une touche de son clavier enfoncée
KeyUp //l'utilisateur relâche une touche préalablement enfoncée
Load //le navigateur de l'utilisateur charge la page en cours
MouseOver //l'utilisateur positionne le curseur de la souris au
//dessus d'un élément
MouseOut //le curseur de la souris quitte un élément
Reset //l'utilisateur efface les données d'un formulaire à l'aide
//du bouton Reset
Select //l'utilisateur sélectionne un texte dans un champ de type "text"
//ou "textarea"
Submit //l'utilisateur clique sur le bouton de soumission d'un formulaire
MouseMove, MouseUp, MouseDown ...

```

Langage de script : Javascript

Inclusion du code:

```
<script></script>
```

1. Dans l'élément **head** de la page html

```
<head>
```

```
<script> alert('Bonjour');
```

```
</script>
```

```
</head>
```

2. Dans l'élément **body** de la page html

```
<body>
```

```
<script> alert('Bonjour');
```

```
</script>
```

```
</body>
```

3. Dans l'élément **un fichier externe** de la page html

```
<head>
```

```
<script src="test.js"> </script>
```

```
</head>
```


Langage de script : Javascript

alert(): fonction qui permet d'afficher une boîte de dialogue

confirm(): fonction qui permet d'afficher une boîte de dialogue avec les boutons ok et annuler

m=prompt(): afficher une zone de saisie qui récupère du texte dans la variable

NB : `parseInt()` et `parseFloat()` permettent de convertir

Affichage des résultats:

1. À l'aide de la console

`console.log('message à afficher');`

2. Dans la page html à l'aide de

`document.write('Message à afficher');`

15

JavaScript dans <head> ou <body>

- Vous pouvez placer n'importe quel nombre de scripts dans un document HTML.
- Les scripts peuvent être placés dans <body>, ou dans la section <head> d'une page HTML, ou dans les deux.
 - [Exemple1](#)
 - [Exemple2](#)
- Le fait de placer des scripts en bas de l'élément <body> améliore la vitesse d'affichage, car l'interprétation des scripts ralentit l'affichage.

JavaScript externe

- Les scripts peuvent également être placés dans des fichiers externes:
- Les scripts externes sont pratiques lorsque le même code est utilisé dans de nombreuses pages Web différentes.
- Les fichiers JavaScript ont l'extension de fichier **.js**.
- Vous pouvez placer une référence de script externe dans **<head>** ou **<body>** comme vous le souhaitez.
- Pour utiliser un script externe, placez le nom du fichier de script dans l'attribut **src** (source) d'une balise **<script>** :
- [Exemple](#)
- Fichier externe: [myScript.js](#)

Avantages JavaScript externes

- Il **sépare** HTML et code
- Il **facilite** la lecture et la maintenance du HTML et du JavaScript
- Les fichiers JavaScript mis en cache peuvent **accélérer** le chargement des pages
- Pour ajouter plusieurs fichiers de script à une page, utilisez **plusieurs balises** de script.

Formulaires JavaScript

- Validation de formulaire JavaScript
 - La validation du formulaire HTML peut être effectuée par JavaScript.
 - Si un champ de formulaire (**fname**) est vide, cette fonction alerte un message, et renvoie false, pour éviter que le formulaire ne soit soumis:
- La fonction peut être appelée lorsque le formulaire est soumis.
- [Exemple](#)

JavaScript peut valider une entrée numérique

- JavaScript est souvent utilisé pour valider les entrées numériques:
- [Exemple](#)

Validation automatique du formulaire HTML

- La validation du formulaire HTML peut être effectuée automatiquement par le navigateur:
- Si un champ de formulaire (**fname**) est vide, l'attribut **required** empêche ce formulaire d'être soumis:
- [Exemple](#)

La validation des données

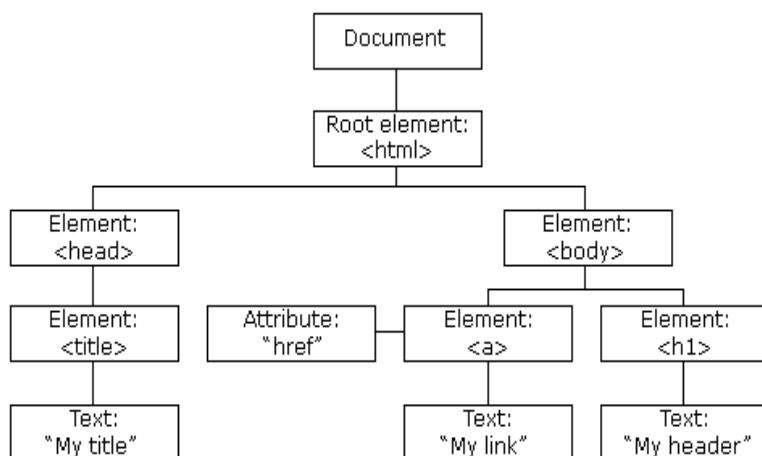
- Les tâches de validation typiques sont:
 - L'utilisateur a-t-il rempli tous les champs obligatoires?
 - l'utilisateur a-t-il entré une date valide?
 - l'utilisateur a-t-il saisi du texte dans un champ numérique?
- Le plus souvent, le but de la validation des données est d'assurer une saisie correcte de l'utilisateur.
- La validation peut être définie par de nombreuses méthodes différentes et déployée de différentes manières.
- **La validation côté serveur** est effectuée par un serveur Web, une fois que l'entrée a été envoyée au serveur.
- **La validation côté client** est effectuée par un navigateur Web, avant que l'entrée ne soit envoyée à un serveur Web.

DOM HTML JavaScript

Le DOM HTML (Document Object Model)

- Avec le DOM HTML, JavaScript peut accéder et modifier tous les éléments d'un document HTML.
- Lorsqu'une page Web est chargée, le navigateur crée un **Document Objet Modèle** de la page.
- Avec le modèle objet, JavaScript obtient toute la puissance dont il a besoin pour créer du HTML dynamique:
- En d'autres termes: **le DOM HTML est une norme pour savoir comment obtenir, modifier, ajouter ou supprimer des éléments HTML.**
- Le modèle **HTML DOM** est construit comme une arborescence d' **objets** :

L'arbre d'objets HTML DOM



Langage de script : Javascript – Variables & Opérateurs

Toute instruction se termine par ;	
commentaires	// ou /* */
Variables : commencent par une lettre, Sensibilité à la casse Type : number, string, booléen, null, NaN, undefined	var a=100, b='100', c=true; var e=null; //faiblement typé typeof() : retourne le type de la variable
Opérateurs arithmétiques Opérateurs de concaténation Opérateurs de comparaison Opérateurs logiques	+, -, *, /, +=, -=, /=, *= + <, >, <=, >=, ==, === &&, , !

10

Langage de script : Javascript – Conditions et boucles

Conditions	Boucles
if (condition){ //code }	initialisation while (condition_répétition){ //code pas }
if (condition) { //code } else { //code }	do { //code pas } while (condition_repetition);
switch (condition){ case val1:; break; case val2 :; break; }	for (initialisation; condition_rpet;pas){ //code }

26

Langage de script : Javascript - Fonctions

Fonctions :

```
<script>
```

```
//Définition
```

```
function test(a, b)
{
    return a+b;
}
```

```
var m=test(1,2);
```

```
</script>
```

Portée des variables :

Toute variable déclarée dans une fonction n'est valable que dans cette fonction

27

Langage de script : Javascript - Fonctions

Il est possible de définir des **fonctions anonymes** : sans nom

```
<script>
```

```
//Définition
```

```
var test=function (a, b)
{
    return a+b;
};
```

```
var m=test(1,2);
```

```
</script>
```

Pour l'instant affecter la fonction à une variable pour l'exécuter

28

Langage de script : Javascript – classes et objets

Classe : une représentation abstraite d'un ensemble d'éléments ayant des attributs (propriétés) et méthodes (fonctions) communs

Etudiant
CNE
Nom
Prenom
Etudier()
estenVacance()

Une instance d'une classe est appelée **objet**.
Par exemple

E1 : Etudiant
2016/ER1
Lahcen
Ahmed
Etudier()
estenVacance()

Pour accéder à un attribut/méthode
E1.CNE ----- 2016/ER1
E1.Etudier()

29

Langage de script : Javascript – classes et objets

En Javascript tout est évalué comme un objet, même si c'est une valeur que vous avez affectée à une variable.

Création d'objets : Méthode 1 via la classe object.

- a- créer la variable : `var E1 = new Object();`
 - b- affecter /remplir les attributs `E1.nom='Ali';`
les attributs seront automatiquement ajoutés
 - c - utiliser les attributs `document.write(E1.nom);`
- NB : L'ajout de méthodes se fait à l'aide d'une propriété qui recevra comme valeur une fonction anonyme

30

Langage de script : Javascript – classes et objets

Création d'objets : Méthode 1 via la classe object.

```
var E1= new Object();
E1.nom='Ali';
E1.prenom='Ahmed';
E1.CNE= '2012/RET';
E1.info=function(){
    return this.CNE+ ' ' + this.nom+ ' '+this.prenom;
};
document.write(E1.info ());
```

31

Langage de script : Javascript – classes et objets

Création d'objets : Méthode 2 - objet littéral.

Un objet littéral est un objet qui contient déjà les valeurs de ses attributs .

Il reste à définir les propriétés et les méthodes.

```
var E1 = {
    CNE : '2012/RET',
    Nom : 'Ali',
    prenom : 'Ahmed',
    infoCNE : function() {
        return this.CNE;
    }
};
document.write(E1.infoCNE());
```

Une méthode sera mentionnée comme un attribut
this permet d'accéder à une propriété interne à l'objet

32

Langage de script : Javascript – classes et objets

Création d'objets : Méthode 3 – constructeur/fonction

Il s'agit de définir un constructeur/méthode qui sera utilisé pour créer un objet et initialiser les valeurs de l'objet.

```
function Etudiant(numCNE, Nome, Prenome) {
  this.CNE = numCNE; this.nom=Nome; this.prenom = Prenome;
  this.info = function(){
    return this.CNE+ ''+ this.Nom+ ''+this.prenom;
  }
}
var E1=new Etudiant('2012/RET', 'Ali', 'Ahmed'); //Creation objet
document.write(E1.info()); //appel d'une fonction
```

33

Langage de script : Javascript – classes et objets

Création de classes : Méthode 4 – constructeur/classe

Il s'agit de définir une classe avec ses propriétés et ses fonctions parmi lesquelles le constructeur.

```
class Etudiant {
  constructor(numCNE, Nome, Prenome) {
    this.CNE = numCNE;
    this.nom=Nome;
    this.prenom = Prenome;}
  info (){
    return this.CNE+ ''+ this.Nom+ ''+this.prenom;
  }
}
var E1=new Etudiant('2012/RET', 'Ali', 'Ahmed'); //Creation objet
document.write(E1.info()); //appel d'une fonction
```

34

Langage de script : Javascript – Tableaux

Création de tableaux:

- | | |
|---|---|
| 1. | 2. |
| <code>var t=[1, 2, 3, 4];</code> | <code>var t=new Array(1, 2, 3, 4);</code> |
| <code>var liste=['Ali', 'Car', 'maison'];</code> | <code>var liste=new Array('Ali', 'Car', 'maison');</code> |

Accès aux champs `t[0] , liste[2]`

Taille du tableau `liste.length => 3`

Parcours

1. `for (i=0; i<t.length;i++){ }`
2. `for (i in t){ }` la boucle s'arrête à la fin du tableau, i prend la valeur des indices

35

Langage de script : Javascript – Tableaux

Méthodes spécifiques :

concat(ch2) : concatène 2 tableaux retourne la nouvelle chaîne
pop() : supprime le dernier élément
push(val1, val2, etc..) : ajoute à la fin
shift() : retire le premier élément
unshift() : ajoute un élément au début
slice(debut [,fin]) : retourne une portion du tableau
splice(debut , nbr_el, val1, val2, etc..)] : ajoute/remplace les éléments situés à partir de début
includes(val), reverse()

36

Langage de script : Javascript – Number

Méthodes spécifiques:

isInteger() : vérifie si c'est un entier

toFixed(nbr) : spécifie le nombre de chiffres après la virgule

toString() : transforme en chaîne de caractères

isNaN() : is not a number

37

Langage de script : Javascript – Math

Attributs spécifiques:

PI: valeur de PI

LN2: valeur de log 2

SQRT2 : la racine carrée de 2

Méthodes spécifiques:

abs(), sin(), cos() :

min(val1, val2, ..., valn), max(val1, val2, ..., valn) : min et max,

floor() : l'entier précédent, **ceil()** : l'entier suivant

random() : nombre aléatoire entre 0 et 1

38

Langage de script : Javascript – String

Attributs spécifiques:

length: longueur de la chaîne

Méthodes spécifiques:

charAt(pos): retourne le caractère de la position

ch1.concat(ch2): concatène la ch1 et ch2 et retourne la chaîne concaténée

includes(ch) : vérifie si la chaîne1 contient la chaîne ch

search(ch) : retourne la position de ch dans la chaîne

toUpperCase() : retourne la chaîne en majuscules

toLowerCase() : retourne la chaîne en minuscules

split(motif) : sépare la chaîne en fonction du motif transmis

39

Langage de script : Javascript – Date

```
var d = new Date();
```

```
var d = new Date(milliseconds);
```

```
var d = new Date(dateString);
```

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

Méthodes spécifiques:

getDate(): retourne le jour du mois entre 1-31

getFullYear/Day/Hours/Minutes/Seconds/Milliseconds(): retourne la valeur demandée

setFullYear/Day/Hours/Minutes/Seconds/Milliseconds(val): modifie un objet Date

now() : retourne le nombre de millisecondes écoulées depuis le 1 Janvier 1970

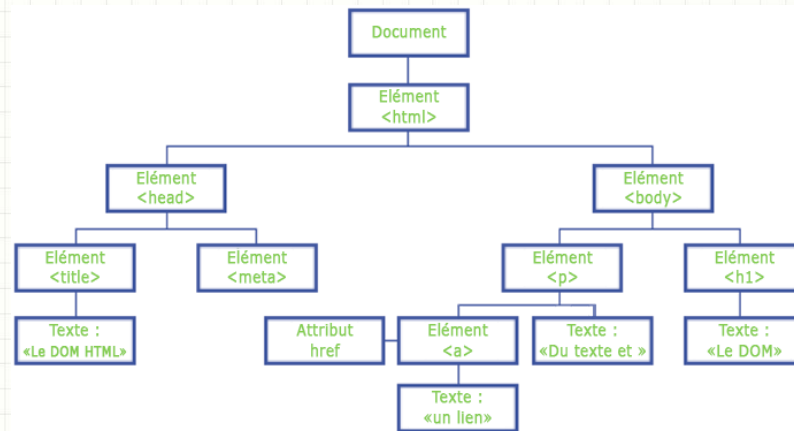
NB **getDay()** : retourne le jour de la semaine

40

Javascript DOM - HTML

DOM : Document Object Model

DOM HTML : Considérer tout le document ainsi que les éléments comme des objets/noeuds auquel je peux accéder à partir de javascript



Javascript DOM - HTML

En HTML chaque balise est considérée comme un noeud et peut posséder des attributs

` Page d'accueil `

En présence de plusieurs nœuds du même type, je peux attribuer un identifiant unique pour y accéder après

`<div id='par1'> ceci est le paragraphe 1</div>`

`<div id='par2'> ceci est le paragraphe 2</div>`

`<div id='par3'> ceci est le paragraphe 3</div>`

Javascript DOM - HTML

A partir de Document je peux accéder aux éléments à l'aide des méthodes

getElementById('valeur_id') : accéder à un élément via son Id

Exemple : `document.getElementById('par1');`

getElementsByTagName('nom_element') : accéder à tous les éléments

Exemple : `document.getElementsByTagName('div');` retourne un tableau contenant tous les éléments de type div

getElementsByClassName('nom_classe') : accéder à tous les éléments sur la base de l'attribut class

43

Javascript DOM - HTML

A partir de Document je peux accéder aux éléments à l'aide des méthode

querySelector("") : accéder au premier élément correspondant

Exemple : `document.querySelector('div');`

querySelectorAll("") : accéder à tous les éléments

Exemple : `document.querySelectorAll('div');` retourne un tableau contenant tous les éléments de type div

Ces deux méthodes se basent sur le sélecteur CSS

44

Javascript DOM - HTML

Pour **accéder** au contenu de l'élément il faut utiliser

- la méthode **innerHTML**

Exemple : `document.getElementById('par1').innerHTML;`

- La méthode **textContent**

Exemple : `document.getElementById('par1').textContent();`

Pour **modifier** le contenu de l'élément il faut utiliser

- la méthode **innerHTML**

Exemple :

`document.getElementById('par1').innerHTML='bienvenue';`

45

Javascript DOM - HTML

Pour **modifier** la valeur d'un attribut

`document.querySelector('a').href='http://www.emsi.ma';`

Pour **modifier** la valeur d'un attribut class

`document.querySelector('.para').className='class1';`

Pour **modifier** le css

`document.getElementById('p').style.color='blue';`

46

JavaScript Gestion des évènements

Un évènement est une action de l'utilisateur prise en compte par le navigateur

Exemple : Clic droit, touche enfoncée, position de la souris, etc.....

Le gestionnaire d'évènements permet d'associer une action à un évènement sur un objet ou un élément de la page

`onEvenement="Action_Javascript_ou_Fonction();"`

Pour associer un évènement à un élément :

`<img src='une_image.jpg'`

`onclick='nom_de_lafunction(parametre1,parametre2)' />`

Et si j'ai plusieurs fonctions

``

47

JavaScript Gestion des évènements

Pour associer un évènement à un élément après le chargement de la page:

Méthode 1 : DOM-0

A- via une fonction

`document.getElementById('mon_element').onclick=nom_de_la_fonction;`

B- Via une fonction anonyme

`document.getElementById('mon_element').onclick=function("un_parametre"){
du code du code...etc}`

Méthode 2 : via addEventListener - DOM-2

`document.getElementById('mon_element').addEventListener("click",
nom_de_la_fonction, true));`

- Il est possible d'ajouter plusieurs fonctions à un évènement

Il est possible de retirer un évènement en utilisant `removeEventListener()`;

`document.getElementById('mon_element').removeEventListener("click",
nom_de_la_fonction, true));`

48

Javascript Gestion des évènements

Pour associer un évènement à un élément après le chargement de la page:

Méthode 3 : L'objet Event

Cet objet contient les informations liées à un évènement déclenché

Il faut transmettre cet objet lors de l'appel de la fonction.

```
A- element.onclick = function(e) {
    // L'argument « e » va récupérer une référence vers l'objet « Event »
    alert(e.type); // Ceci affiche le type de l'évènement (click, mouseover, etc.)
};
```

```
B- element.addEventListener('click', function(e) {
    // L'argument « e » va récupérer une référence vers l'objet « Event »
    alert(e.type); // Ceci affiche le type de l'évènement (click, mouseover, etc.)
});
```

49

Introduction à AJAX

- AJAX = JavaScript et XML asynchrones.
- **AJAX** est le rêve d'un développeur, car vous pouvez:
 - Lire les données d'un serveur Web - après le chargement de la page
 - Mettre à jour une page Web **sans recharger la page**
 - Envoyer des données à un serveur Web - en arrière-plan
- AJAX consiste à mettre à jour **des parties d'une page** Web, sans recharger la page entière.
- Exemples d'applications utilisant AJAX: onglets Google Maps, Gmail, Youtube et Facebook.

Comment fonctionne AJAX

1. Un événement se produit dans une page Web (la page est chargée, un bouton est cliqué)
2. Un objet **XMLHttpRequest** est créé par JavaScript
3. L'objet XMLHttpRequest envoie une requête à un serveur Web
4. Le serveur traite la demande
5. Le serveur renvoie une réponse à la page Web
6. La réponse est lue par JavaScript
7. Une action appropriée (comme la mise à jour de la page) est effectuée par JavaScript.

[Exemple](#)

Exemple PHP AJAX

- AJAX est utilisé pour créer des applications plus interactives.
- L'exemple suivant montre comment une page Web peut communiquer avec un serveur Web pendant qu'un utilisateur saisit des caractères dans un champ de saisie.
- Dans l'exemple ci-dessus, lorsqu'un utilisateur tape un caractère dans le champ de saisie, une fonction appelée **showHint()** est exécutée.
- La fonction est déclenchée par l'événement **onkeyup**.
- [Exemple](#)

jQuery vs JavaScript "écrire moins, faire plus"

- [jQuery](#) a été créé en 2006 par John Resig. Il a été conçu pour gérer les incompatibilités de navigateur et pour simplifier la manipulation HTML DOM, la gestion des événements, les animations et Ajax.
- Depuis plus de 10 ans, jQuery est la **bibliothèque JavaScript** la plus populaire au monde.
- Cependant, après la [version 5 de](#) JavaScript (2009), la plupart des utilitaires jQuery peuvent être résolus avec quelques lignes de JavaScript standard:

Syntaxe jQuery

- La syntaxe jQuery est faite sur mesure pour **sélectionner des** éléments HTML et effectuer certaines **actions** sur le ou les éléments.
- La syntaxe de base est: **\$ (*sélecteur*). *action* ()**
 - Un signe \$ pour définir / accéder à jQuery
 - Un (*sélecteur*) pour "interroger (ou rechercher)" des éléments HTML
 - Une *action* jQuery () à effectuer sur le ou les éléments

Exemples

- `$(this).hide()` - masque l'élément courant.
- `$("p").hide()` - masque tous les éléments `<p>`.
- `$(".test").hide()` - masque tous les éléments avec `class = "test"`.
- `$("#test").hide()` - masque l'élément avec `id = "test"`.

Recherche d'un élément HTML par ID

- Renvoie l'élément avec `id = "id01"`:

jQuery

```
var myElement = $("#id01");
```

Essayez vous-même »

JavaScript

```
var myElement = document.getElementById("id01");
```

Recherche d'éléments HTML par nom de balise

- Renvoie tous les éléments <p>:

jQuery

```
var myElements = $("p");
```

Essayez vous-même »

JavaScript

```
var myElements = document.getElementsByTagName("p");
```

Recherche d'éléments HTML par nom de classe

- Renvoie tous les éléments avec class = "intro".

jQuery

```
var myElements = $(".intro");
```

Essayez vous-même »

JavaScript

```
var myElements = document.getElementsByClassName("intro");
```


Recherche d'éléments HTML à l'aide des sélecteurs CSS

- Renvoie une liste de tous les éléments <p> avec class = "intro".

jQuery

```
var myElements = $("p.intro");
```

Essayez vous-même »

JavaScript

```
var myElements = document.querySelectorAll("p.intro");
```

Ressources

- w3schools.com
- fr.openclassrooms.com
- [Cours HTML, IUT Montpellier, (http://www.lirmm.fr/~croitoru/teaching/iut_special_year.html)]
- [Écrire ses pages Web en quelques leçons..., (<http://lita.univ-metz.fr/~paris/>)]
- [CSS : vos premiers pas, (<http://www.css-faciles.com/premiers-pas-css.php>)]
- [Dynamisez vos sites web avec Javascript !, (http://www.fr.capgemini.com/carrieres/technology_services/)]