

Programmation web

PARTIE 1 : **Programmation Client (suite)** **(CSS-JAVASCRIPT)**

Nhan LE THANH

IUT de Nice

Plan

- **DEFINITIONS**
 - SERVICE HTTP ET NOTION URL
 - MODELE CLIENT/SERVEUR
 - MODELE CLIENT SERVEUR VIA LE SERVICE HTTP
 - ARCHITECTURE 3-TIERS
- **LANGAGE (x)html**
 - INTRODUCTION
 - BALISES DE BASE
 - BALISES MULTIMEDIAS
 - IMAGES CLIQUABLES
 - FORMULAIRES
 - CADRES
- **FEUILLES DE STYLE CSS2**
 - INTRODUCTION
 - DEFINITIONS
 - CLASSES, IDENTIFIANT
 - UNITES
 - BOITES ET COUCHES
- **LANGAGE JAVASCRIPT**
 - INTRODUCTION
 - JAVASCRIPT ET HTML
 - ELEMENTS DE JAVASCRIPT
 - OBJETS DOM
 - AUTRES OBJETS
 - EVENEMENTS
 - DHTML

PROGRAMMATION CLIENT WEB

Langage de styles CSS2

PROGRAMMATION CLIENT WEB

Langage de styles CSS2

• INTRODUCTION

- .CSS (Cascading Style Sheet)
- Possibilité d'utiliser plusieurs feuilles de style en les combinant
- Notion d'héritage entre feuilles de styles
- permet de déclarer des feuilles de style en utilisant la syntaxe suivante :
 - BALISE { Propriété : valeur}
 - Ex: H2 { color : green } indique que tous les titres de niveau 2 seront en vert.

PROGRAMMATION CLIENT WEB

Langage de styles CSS2

Une feuille de style CSS peut être définie de 4 façons :

- à l'extérieur du document par l'utilisation de la balise LINK dans la zone du document marquée par <HEAD> </HEAD>
`<LINK rel="stylesheet" TYPE="text/css" HREF="..." TITLE="...">`
- par utilisation de balises dans le document :
`<STYLE TYPE="text/css"> H1 { color: green }</STYLE>.`
- par utilisation de balises STYLE + import de la feuille de style
`<STYLE TYPE="text/css">
 @import url(http://...);
 H1 { color: blue }
</STYLE>`
- À l'intérieur d'une balise supportant la propriété style
`<P STYLE="Propriétés:valeur">`

PROGRAMMATION CLIENT WEB

Langage de styles CSS2

INTRODUCTION : UTILITE ET AVANTAGES

- Séparation du contenu et de la mise en forme.
- Cohésion de la présentation tout au long du site
- Modifier l'aspect d'un page ou d'un site sans en modifier le contenu et cela en quelques lignes plutôt que de devoir changer un grand nombre de balises.
- Un "langage" neuf, compréhensible, simple et logique par rapport au Html et à ses différentes versions.

PROGRAMMATION CLIENT WEB

Langage de styles CSS2

- Réduire le temps de chargement des pages.
- Palier certaines insuffisances du langage Html (contrôle des polices, contrôle de la distance entre les lignes, contrôle des marges et des indentations (sans devoir utiliser de tableaux ou de balise
- enfin, et surtout, moins de code, mieux structuré, et donc des sites plus faciles à maintenir et à faire
- évoluer

PROGRAMMATION CLIENT WEB

Langage de styles CSS2

• INTRODUCTION

Notions de base

- Jeu de caractères : Unicode (ISO 10646)
- Identificateurs : indépendants de la casse ; ne doivent pas être entourés par des guillemets ou des apostrophes
text-traNSFORM white FONt-style italiC
- Règles (de style) :
composées à gauche d'un sélecteur, et à droite d'un bloc (de déclarations de styles)
H1 + P {color: blue; margin-left: 1em}
- Règles "at" : le premier identificateur de la partie gauche (sélecteur) est immédiatement précédée d'un @
@import "feuille.css";

PROGRAMMATION CLIENT WEB

Langage de styles CSS2

- **Déclarations :**
dans le bloc, un identificateur, suivi de deux-points, suivi d'une ou plusieurs valeurs suivi d'un pt-virgule

ex text-align: center;
- **Chaînes de caractères :**
"entre guillemets comme ceci" ou *'en utilisant l'apostrophe, comme ceci'* ;
- **Nombres :** 1.414 +5 .4 -0.9
- **Longueurs :** exprimées en cm, mm, in (2.54cm), pt (1/72in), pc (12pt) pour les longueurs absolues, et en em, ex et px pour les longueurs relatives, voire en pourcentages <HR style="width: 50%">
- **Uniform Resource Locators (URL)**
 - BODY { background: url("http://somewhere.to/nowhere.jpg") }

PROGRAMMATION CLIENT WEB

Langage de styles CSS2

- **DEFINITIONS : ATTRIBUT STYLE** (sans fichier css)
 - L'attribut STYLE peut être utilisé dans toutes les balises de HTML pour définir un style local à cette balise
 - Syntaxe : STYLE="propriété:valeur; propriété:valeur;...; "
 - Exemple :

```
<!DOCTYPE .... >
<HTML xmlns=...>
<HEAD>
  <title>exemple</title>
</head>
<BODY>
<H1 style="font-family: Arial; font-style: italic;"> Style H1 </H1>
</BODY>
</HTML>
```

PROGRAMMATION CLIENT WEB

Langage de styles CSS2

• DEFINITIONS : BALISE STYLE

- Balise STYLE dans la partie entête du document (<head></head>) permet de définir des règles de style pour tout le document
- Syntaxe : <STYLE type=text/css> les règles de styles </STYLE>
- Exemple :
 - <STYLE type="text/css" >
 - <!--
 - div.slide { background-color: white; margin: 1em; padding: .5em; border: solid; border-width: 3px; border-color: #ff9999; }
 - -->
 - </STYLE>
- Les règles de style se mettent comme commentaires pour être ignorées par les navigateurs qui ne possèdent pas l'interpréteur

PROGRAMMATION CLIENT WEB

Langage de styles CSS2

DEFINITION :

les feuilles de style externes permettent de définir les règles de style pour plusieurs documents

- Une feuille de style est un fichier texte qui ne contient que des règles de style
- Le suffixe du fichier de style ou extension est .css
- L'intégration d'une feuille de style dans un document HTML est faite grâce à la balise LINK dans la partie entête du document :

```
<LINK rel="STYLESHEET" href="URL du fichier de style" type="text/css">
```

PROGRAMMATION CLIENT WEB

Langage de styles CSS2

- **LES CLASSES DE STYLES : permettent d'affecter les différents styles à une balises dans un même document**
 - Syntaxe :
 - `.nom_de_classe { propriété de style: valeur }`
(Attention! L'emploi du point (.) devant le nom de classe est indispensable)
 - Utilise dans une balise :
 - `<balise class="nom_de_classe"> </balise>` (sans le .)
 - Exemple :
 - Définition : `.essentiel { font-weight: bold; font-color: #000080 }`
 - Utilisation : `<P class="essentiel"> ... blabla ... </P>`
 - `<H1 class="essentiel">Titre 1</H1>`
 - `<TABLE><TR><TD class="essentiel">cellule</TD></TD>...`

PROGRAMMATION CLIENT WEB

Langage de styles CSS2

- Importation de règles:
Dans une feuille de style, il est possible d'importer les déclarations d'une autre feuille. On utilise pour ça une règle "at" @import :

```
/* fichier default.css */  
  
@import "special.css"; /* noter le ; à la fin */  
  
@import url('http://somewhere.to/nowhere.css');
```

P {color: blue}

/ noter que les @import doivent précéder les règles classiques sous peine d'invalidité de la feuille de style */*

PROGRAMMATION CLIENT WEB

Langage de styles CSS2

• LES CLASSES DE STYLES : les ID (Identifiants) sont des classes de styles spéciales qui permettent

- de nommer de la manière unique une zone de texte dans un document.
- Ce qui permet le manipuler dynamiquement la zone comme un objet DOM avec JavaScript
- **Syntaxe :**
 - Définition #nom_de_ID { propriété de style: valeur }
 - Utilisation : un seul appel pour identifier une balise texte quelconque :
<balise id="nom_de_ID"> </balise>
 - Manipulation les propriétés de la balise identifiée depuis JavaScript en particulier la propriété "visibility"
- Exemple : Pour #essentiel{ ... }
<P id=essentiel> est correct. Mais si on rencontre dans la même page
<H1 id=essentiel> ce n'est plus correct !

PROGRAMMATION CLIENT WEB

Langage de styles CSS2

• LES CLASSES DE STYLES : Les balises SPAN et DIV sont utilisées

uniquement pour le style, elles permettent un découpage plus fin que la notion de Balises

- SPAN permet de isoler une parcelle de texte à l'intérieur d'une balise et applique un style différent à cette parcelle
- DIV calque regroupant un ensemble de balises <DIV id=nom>...</DIV>
- EXEMPLE de SPAN

Un monde de géants.

```
<HTML> <HEAD>
<STYLE type="text/css">
.element{font-size: x-large; color: navy;}
</STYLE> </HEAD>
<BODY>
<P>Un monde de <SPAN class=element>géants</SPAN>.</P>
</BODY>
</HTML>
```


PROGRAMMATION CLIENT WEB

Langage de styles CSS2

- **LES UNITES Unité de longueur**

Les longueurs généralement positives représentent des valeurs relatives (en pourcentage) ou absolues

- Les valeurs relatives peuvent être :
 - em : exemple 0.5em : pourcentage par rapport à la taille de la police
 - ex : exemple 1ex : pourcentage par rapport à la taille de la lettre x
 - px : exemple 12px : pourcentage par rapport à la résolution d'une image en pixel
 - % : exemple 50% : pourcentage par rapport à la totalité de l'espace
- Les valeurs absolues peuvent être :
 - pt : les points
 - in : inches (2,54cm)
 - cm : les centimètres
 - mm : les millimètres
 - pc : les picas

PROGRAMMATION CLIENT WEB

Langage de styles CSS2

- **LES UNITES : les couleurs**

- Les couleurs peuvent être désignées par leur nom en langue anglaise : *aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white et yellow*
- Elles peuvent aussi être désignées en valeur hexadécimale représentant la quantité de couleur en RGB : #FFFF00. Les valeurs à trois digits sont permises. Ex #FFF au lieu de #FFFFFF
- Elles peuvent enfin être représentées par la fonction *rgb* qui renseigne sur les valeurs des couleurs élémentaire rouge vert bleu : *rgb(10, 20, 255)* ou même sur les pourcentages *rgb(10%,20%,70%)*

- Les URL doivent être précédées de l'appel de fonction `url()`

Exemple :

```
BODY { background:url('http://www.ungj.com/gif/caution.gif') }
```

PROGRAMMATION CLIENT WEB

Langage de styles CSS2

- **LES BOITES/CALQUES ET COUCHES :**

- Les boîtes sont des zones de présentation (souvent d'un paragraphe P) associées à un style
- Exemple

```
P.essai {  
    position:absolute; /* Absolute ou relative*/  
    background-color: #dddddd;  
    top:200px; /*200px en partant du haut de la fenêtre*/  
    margin-top: 10px; /* s'ajoute a top*/  
    margin-right: 50px;  
    margin-bottom: 20px;  
    margin-left: 30px;  
    border-style : double;  
    border-width: 4pt;  
    border-color: blue;  
    padding: 20pt; /*20 pt autour du texte*/  
}
```

PROGRAMMATION CLIENT WEB

Langage de styles CSS2

- **LES BOITES/COUCHES : Les raccourcis de style**

- padding: 20pt; équivaut à padding-top:20pt; padding-right:20pt; padding-bottom:20pt;
- padding left:20pt; padding: 20pt 40pt; équivaut à padding-top:20pt; padding-right:40pt; padding-bottom:20pt;
- padding-left:40pt;
- border: 10px solid blue; équivaut à border-width: 10px; border-style: solid; border-color: blue;

PROGRAMMATION CLIENT WEB

Langage de styles CSS2

• **BOITES : Display - Types de boîtes**

Display:attribut permet de gérer le calque/boite.

Liste des attributs de display :

- **inline** : pour définir un flot de texte normal sur une même ligne
- **block** : pour définir une boîte de type bloc, comme une division, un paragraphe, un `<pre>...</pre>`
- **list-item** : pour définir un article dans une liste (boîte de type bloc) ainsi que la boîte nécessaire à l'éventuelle numérotation
- **marker** : une boîte spéciale pour numéroter et générer d'autres contenus : boîte formatée sur une seule ligne et sans marges extérieures
- **compact** : une boîte qui est de type bloc ou de type flot de texte selon la largeur de la boîte suivante et de sa marge extérieure (observer le dernier article de cette liste...)
- **run-in** : comme pour compact, sans la condition sur la marge
- **none** : aucune boîte n'est créée : la place de l'élément concerné n'est pas réservée et l'élément n'est pas restitué. À ne pas confondre avec visibility: hidden pour lequel la place est réservée et l'élément n'est pas restitué

PROGRAMMATION CLIENT WEB

Langage de style CSS2

☞ **BOITE: Positions position:attribut;**

- Il est possible de positionner des éléments où l'on veut dans une page web, grâce aux attributs suivants :
- **static** : correspondant à la restitution "classique" d'un document HTML
- **relative** : en utilisant les valeurs de 4 autres propriétés (top right bottom left) l'élément est positionné relativement à sa position classique (telle que static l'aurait rendu)
- **absolute** : cette fois-ci l'élément est positionné par rapport à sa boîte parente
- **fixed** : et cette fois par rapport à la boîte parente de toutes les autres, c'est-à-dire la fenêtre de restitution

PROGRAMMATION CLIENT WEB

Langage de styles CSS2

• Tableau de propriétés de style

<i>Couleurs</i>	color	La couleur d'avant-plan
	background-color	La couleur d'arrière plan
	background-image	Une image à utiliser comme fond
<i>Polices</i>	font-family	Une liste de familles de la police à utiliser
	font-style	normal, italic, oblique
	font-weight	normal, bold, bolder, lighter
	font-size	La taille de la police exprimée soit par une taille absolue déterminée par le visualiseur (xx-small, x-small, small, medium, large, x-large, xx-large), soit par une mesure relative (larger, smaller) soit par une mesure de longueur soit par un pourcentage par rapport à l'élément englobant

PROGRAMMATION CLIENT WEB

Langage de styles CSS2

• Tableau de propriétés de style

<i>Polices</i>	font	Un raccourci pour toutes les propriétés ci-dessus P { font: bold italic large Palatino, serif }
<i>Texte</i>	text-indent	Indentation de la première ligne d'un bloc
	text-align	Alignement horizontal d'un bloc de texte : left, right, center, justify. Peut être également utilisé pour les cellules de tableau.
	text-decoration	Attributs additionnels de décoration : underline, overline, line-through, blink
	text-transform	Modification de la capitalisation d'un bloc de texte : capitalize, uppercase, lowercase
<i>Espacement</i>	margin-top margin-right margin-bottom margin-left	Largeur de marge. La valeur est soit une longueur, soit un pourcentage de la boîte englobant. margin est un raccourci pour ces quatre spécifications individuelles.

PROGRAMMATION CLIENT WEB

Langage de styles CSS2

Tableau de propriétés de style

<i>Espacement</i>	padding-top padding-right padding-bottom padding-left	Largeur de remplissage (cette zone de remplissage entoure la zone d'affichage et est elle-même entourée par la zone de marge). padding est un raccourci pour ces quatre spécifications individuelles.
<i>Bordures</i>	border-top-width border-right-width border-bottom-width border-left-width	Largeur de bordure. La valeur est soit une longueur soit thin, medium ou thick. border-width est un raccourci pour ces quatre spécifications individuelles
	border-style	Le style de cadre : dotted (pointillé), dashed (pointillé long), solid, double, groove (creusé), ridge (relief)
	border-top-color border-right-color border-bottom-color border-left-color	Couleur de bordure. border-color est un raccourci pour ces quatre spécifications individuelles.
	border	Raccourci pour les trois propriétés précédentes

PROGRAMMATION CLIENT WEB

Langage de styles CSS2

Tableau de propriétés de style

<i>Listes</i>	list-style-type	Type de numérotation de la liste : disc, circle, square, decimal, lower-roman, upper-roman, lower-latin, upper-latin, ...
<i>Tableaux</i>	caption-side	Position de la légende du tableau : top, bottom, left ou right.
	vertical-align	La hauteur d'une rangée dépend de la taille et du placement des cellules qu'elle contient. Une cellule est donc souvent moins haute que la hauteur de la rangée à laquelle elle appartient. Cette propriété détermine l'alignement vertical d'une cellule par rapport à sa rangée : baseline (aligne la ligne de base texte de la cellule avec la ligne de base de la rangée), bottom (le bas de la cellule coïncide avec le bas de la rangée), top, middle
	border-spacing	Espacement des bordures de cellules. La spécification comprend un ou deux longueurs (une seule longueur s'applique à l'espacement horizontal et vertical), deux longueurs correspondent respectivement à l'espacement horizontal et vertical. TABLE { border-spacing: 15pt }
	empty-cells	Attributs de bordure pour les cellules vides : border ou noborder.

PROGRAMMATION CLIENT WEB

Langage JavaScript

PROGRAMMATION CLIENT WEB

Langage JavaScript

- INTRODUCTION

- JavaScript est un langage de scripts, qui, incorporé aux balises Html, permet d'améliorer la présentation et l'interactivité des pages Web.
- JavaScript n'est pas JAVA
 - Java : véritable langage de programmation, développé par Sun, compilé, réellement objet (classes, instances, héritage), permettant de créer des modules (applets) accessibles par html, code protégé, professionnel (typage fort, performances, sécurité,...).

PROGRAMMATION CLIENT WEB

Langage JavaScript

- INTRODUCTION

- JavaScript : simple, facile à comprendre et à utiliser, intégré dans HTML, interprété, basé sur des objets, plus amateur (faiblement typé), sans confidentialité possible du source.
- JavaScript partage les mêmes objets DOM (Document Object Model) que ceux de document HTML. On peut manipuler de la manière dynamique tous les objets comportés dans un documents HTML (mais non pas des texte et des balises HTML en général)

PROGRAMMATION CLIENT WEB

Langage JavaScript

- JavaScript : un tour du langage

- Les lignes commençant par //, plusieurs lignes peuvent être mises en commentaire si elles sont encadrées par les symboles /* */
- Types : JavaScript comprend les types de données suivants :
 - les nombres : 2, 2.90, 314E-2 . Ils peuvent être en base 10, 16 (0x.. ou 0X..) ou 8 (commençant par 0). Les réels flottants peuvent contenir un point décimal, un exposant (E)
 - les booléen : *true* ou *false*

PROGRAMMATION CLIENT WEB

Langage JavaScript

- JavaScript : un tour du langage
 - Types (suite) :
 - les chaînes de caractères : "coucou" ou 'coucou'. Les caractères spéciaux peuvent être utilisés dans les chaînes (\b backspace, \f form feed , \n new line , \r CR, \t tabulation). Le caractère \ permet d'insérer une double quote dans une chaîne
 - les tableaux : tableau[0], tableau[1]
 - les tableaux associatifs : X["Poitier"] , Y["Poitier"]
 - Déclaration de variable : var nom = valeur

PROGRAMMATION CLIENT WEB

Langage JavaScript

- JavaScript : un tour du langage (2)
 - Objets prédéfinis
 - String :
 - Propriétés : length
 - Méthodes : italics - lastIndexOf - link - small - strike - sub - substring - sup - toLowerCase - toUpperCase
 - Syntaxe :
 - var-chaine.propriété,
 - var-chaine.méthode(paramètres)

PROGRAMMATION CLIENT WEB

Langage JavaScript

- JavaScript : un tour du langage (3)
 - Objets prédéfinis
 - Math :
 - Propriétés : E - LN2 - LN10 - LOG2E - LOG10E - PI - SQRT1_2 - SQRT2
 - Méthodes : abs - acos - asin - atan - ceil - cos - exp - floor - log - max - min - pow - random - round - sin - sqrt - tan
 - Syntaxe : Math.Attribut - Math.Méthode(paramètres)

PROGRAMMATION CLIENT WEB

Langage JavaScript

- JavaScript : un tour du langage (4)
 - Types (objets) prédéfinis
 - Date :
 - Propriétés :
 - Méthodes : getDate - getDay - getMinutes - getMonth - getSeconds - getTime - getTimeZoneoffset - getYear - parse - setDate - setHours - setMinutes - setMonth - setSeconds - setTime - setYear - toGString - toLocaleString - prototype - etc.
 - Syntaxe : new Date() , new Date(year, month, day),
 - new Date("month day, year hours:minutes:seconds"),
 - new Date(year, month, day, hours, minutes, seconds)

PROGRAMMATION CLIENT WEB

Langage JavaScript

- JavaScript : un tour du langage (5)
 - Les variables commencent par des lettres ou par le caractère souligné, dans le reste de l'identifiant des valeurs décimales sont autorisées. Les noms en minuscule ou en majuscule sont différents
 - Les expressions du langage JavaScript sont de trois ordres :
 - arithmétiques : opérations sur les entiers, les réels.
 - sur les chaînes de caractères
 - booléennes ou logiques

PROGRAMMATION CLIENT WEB

Langage JavaScript

- JavaScript : un tour du langage (6)
 - Les expressions conditionnelles
 - Syntaxe : (condition) ? val1 : val2
 - évalue la condition et exécute *val1* si vrai, ou *val2* si faux
 - Exemple :
 - `var message = (fin = true) ? "bonjour" : "au revoir"`

PROGRAMMATION CLIENT WEB

Langage JavaScript

- JavaScript : un tour du langage (7)
 - Opérateurs : Comme en C ou en Java les opérateurs suivants sont valides :
 - + (addition), - (soustraction), * (multiplication), ++ (incrément), -- (décrément), / (division), % (modulo),
 - & (ET bit à bit), | (OU bit à bit), ^ (XOR bit à bit), << (décalage de bits vers la gauche), >> (décalage de bits vers la droite), >>> (idem mais en mettant les nouveaux digits à 0), && (ET logique entre deux booléens),
 - || (OU logique entre deux booléens), ! (négation d'un booléen)

PROGRAMMATION CLIENT WEB

Langage JavaScript

- JavaScript : un tour du langage (8)
 - Les comparaisons sont faites par les opérateurs : == (test d'égalité), >, >=, <, <=, != (test d'inégalité)
 - Les opérateurs peuvent être assignés à l'affectation, par exemple :
 - $x += 4$ équivaut à $x = x + 4$
 - $x++$ signifie $x+=1$ c'est à dire $x = x + 1$
 - $x--$ signifie $x-=1$ c'est à dire $x=x-1$

PROGRAMMATION CLIENT WEB

Langage JavaScript

- JavaScript : un tour du langage (9)
 - Instructions de contrôle :
 - for : permet de répéter une séquence d'instructions tant qu'une condition est vraie; elle peut prendre deux formes :
 - La condition de boucle, prend une valeur initiale, une valeur de fin et un pas d'incrément
 - Syntaxe : for ([expression initiale]; [condition]; [expression finale]) { instructions }
 - Exemple : for (i = 0; i < 255; i++) { fonction(n) }
 - La boucle se répète pour un ensemble de valeurs de i =0 à 254 par incrémente de i
 - Syntaxe : **for** (var in obj)**in** { instructions }

PROGRAMMATION CLIENT WEB

Langage JavaScript

- JavaScript : un tour du langage (10)
 - Les instruction de contrôle :
 - Les instructions break et continue :
 - L'instruction *break* permet de sortir de la boucle ou du bloc en cours.
 - L'instruction *continue* termine le bloc en cours pour se positionner sur l'itération suivante
 - L'instruction *while* permet de répéter un bloc d'instructions tant qu'une expression est vraie.
 - Syntaxe : while (condition) { instructions ; }

PROGRAMMATION CLIENT WEB

Langage JavaScript

- JavaScript : un tour du langage (11)
 - Les instruction de contrôle :
 - L'instruction de condition : permet d'effectuer une séquence d'instruction si la condition exprimée est vraie. La partie *else* qui est optionnelle permet d'effectuer la séquence suivante si la condition est fausse.
 - Syntaxe : `if (condition) { instructions } [else { instructions }]`
 - Exemple: `if (prix == 0) { x++ } else { result = result + prix`

PROGRAMMATION CLIENT WEB

Langage JavaScript

- JavaScript : un tour du langage (12)
 - Les fonctions : Les fonctions en JavaScript comme dans tous les autres langages désignent des suites d'instruction répétitives, qui sont exécutés plusieurs fois mais écrites une seule fois.
 - Syntaxe : `nom (param1, param2,...;) { ... }`
 - A partir de la version 1.2 du langage, une fonction peut être appelée récursivement.
 - Remarques :
 - Il est de bon ton de définir toutes les fonctions à l'intérieur du couple de balises `<HEAD> </HEAD>`
 - La fonction sera donc déclarée entre un couple de balise `<SCRIPT> </SCRIPT>` par un appel du genre `nom (param1, param2)`.
 - Une fonction peut appeler des arguments de tout type, ainsi que des fonctions et pour les fous de l'algorithmique les fonctions peuvent être appelées récursivement.

PROGRAMMATION CLIENT WEB

Langage JavaScript

- JavaScript : un tour du langage (13)
 - Remarques :
 - Une fonction retourne une valeur quand elle rencontre le mot clé *return* suivie de la valeur retournée qui peut être de tout type. Les fonctions suivantes permettent de manipuler les chaînes de caractères :
 - eval convertit une chaîne de caractères en valeur entière
 - parseInt convertit une chaîne de caractères en valeur entière dans la base spécifiée
 - parseFloat convertit une chaîne de caractères en valeur réelle flottante
 - A partir de la version 1.2 du langage, une fonction peut être appelée récursivement.

PROGRAMMATION CLIENT WEB

Langage JavaScript

- JavaScript et HTML :

JavaScript *peut être implanté dans un document HTML de différentes façons :*

 - par la balise SCRIPT
 - par les événements
 - par un fichier source
 - par les variables JavaScript

PROGRAMMATION CLIENT WEB Langage JavaScript

- JavaScript et HTML(2):
 - Exemple : Afficher la date du jour dans le document

```
<SCRIPT LANGUAGE="JavaScript">
var jour = new Date()
document.write(« Nice, le ", jour.getDate(), "/",
jour.getMonth() + 1, "/", jour.getYear(), "<P>")
</SCRIPT>
```

PROGRAMMATION CLIENT WEB Langage JavaScript

- JavaScript et HTML (3) : La balise Script
 - Syntaxe : <SCRIPT> code JavaScript </SCRIPT>
 - Spécification de la version par l'attribut LANGUAGE
 - LANGUAGE= " JavaScript " pour JavaScript 1.0
 - LANGUAGE= " Javascript1.1 " pour JavaScript 1.1
 - LANGUAGE= " Javascript1.2 " pour Javascript1.2
- ceci permet de définir des fonctions dans les 3 versions

PROGRAMMATION CLIENT WEB

Langage JavaScript

- JavaScript et HTML (4) : La balise Script
 - Exemple : redirection d'un site vers une autre adresse

```
<SCRIPT LANGUAGE= "JavaScript">  
//remplacer le site actuel par un autre site  
location.replace(" http://nyyx.unice.fr/autresite/ " )  
</SCRIPT>
```

PROGRAMMATION CLIENT WEB

Langage JavaScript

- JavaScript et HTML (5): Par les évènements
 - Une action utilisateur (déplacement de la souris sur une zone, clic sur un bouton souris, ...) crée un événement que JavaScript peut alors traiter par un handler associé
 - le document HTML deviendra réactif à un événement si à cet événement est associée une fonction JavaScript : Ces événements sont surveillés dans certaines balises HTML en particulier dans un formulaire ou bien sur un lien hypertexte.

PROGRAMMATION CLIENT WEB Langage JavaScript

- JavaScript et HTML (6): Par les évènements

- Syntaxe :

<Balise-HTML Attributs eventhandler= "Code JAVASCRIPT ">

- Exemple : déclencher une fonction JavaScript, appelé control() pour contrôler le donnée d'un champ INPUT d'un formulaire :

```
<FORM>  
<INPUT NAME="telephone" onChange= "control(this); ">  
</FORM>
```

PROGRAMMATION CLIENT WEB Langage JavaScript

- JavaScript et HTML(7) : Les événements

onBlur	l'utilisateur après avoir cliqué sur une zone d'entrée clique à l'extérieur
onClick	l'utilisateur clique sur une zone hypertexte ou un élément de formulaire
onChange	l'utilisateur après avoir cliqué sur une zone de formulaire la quitte après avoir changé son contenu
onFocus	l'utilisateur clique sur une zone de formulaire
onMouseOver	la souris passe sur un lien hypertexte

PROGRAMMATION CLIENT WEB Langage JavaScript

- JavaScript et HTML(8) : Les événements

onSelect	l'utilisateur sélectionne un champ élément d'un formulaire
onSubmit	l'utilisateur soumet un formulaire (clique sur bouton submit)
onLoad	une page est chargée par le navigateur (pour les images IMG aussi)
onUnload	l'utilisateur quitte la page.

PROGRAMMATION CLIENT WEB Langage JavaScript

- JavaScript et HTML(9) : Les événements

En JavaScript 1.2, les évènements suivants ont été rajoutés	
onMouseOut	le pointeur quitte un lien
onMouseDown onMouseUp onDbClick	pour les liens et boutons de formulaires (button, checkbox, radio, submit, reset)
onKeyDown onKeyPress onKeyUp	pour les zones de saisie de texte d'un formulaire(text, textarea, password)

PROGRAMMATION CLIENT WEB Langage JavaScript

- JavaScript et HTML(10) : Les événements (suite)

En JavaScript 1.2, les événements suivants ont été rajoutés	
onAbort onError	pour les images
onDragDrop onMove onResize	pour la fenêtre du browser (tag Body ou Frameset) ;

PROGRAMMATION CLIENT WEB Langage JavaScript

- JavaScript et HTML (6) : Exemples d'utilisation des événements

```
<FORM>  
<INPUT VALUE=onBlur onBlur= "alert('onBlur de la zone') ">  
l'utilisateur après avoir cliqué sur une zone clique à l'extérieur  
de la zone active. <BR><l>cliquez sur onBlur puis n'importe où  
à l'extérieur de la zone</l>  
</FORM>
```

```
<A HREF= "" onClick="alert('onClick sur le lien') ">onClick </A>  
l'utilisateur clique sur une zone hypertexte. <BR>  
<l>cliquez sur onClick</l>
```

PROGRAMMATION CLIENT WEB Langage JavaScript

- JavaScript et HTML (7) : Exemples d'utilisation des événements
 - Exemple 3 : **L'exemple suivant affiche dans une fenêtre d'alerte le type (qui est mousedown) et la position dans la page de l'événement déclenché quand un bouton de souris est pressé.**

```
<SCRIPT>
function souris(evt) {
  alert ("Le document a reçu un événement : " + evt.type +
  " en x = " + evt.layerX + " et y = " + evt.layerY);
  return true;
}
document.onMouseDown = souris;
</SCRIPT>
```

PROGRAMMATION CLIENT WEB Langage JavaScript

- JavaScript et HTML (8) : Charger un fichier de code JavaScript
 - A partir de la version 1.1, il est possible de placer le script dans un fichier source annexe en utilisant la balise SCRIPT
 - Syntaxe:
<SCRIPT LANGUAGE= " JavaScript " SRC= "source.js " >
</SCRIPT>
 - Remarque:
le serveur HTTP doit être configuré pour un type MIME
"application/x JavaScript "
pour l'extension js (dans le fichier httpd.conf)

PROGRAMMATION CLIENT WEB Langage JavaScript

- JavaScript et HTML (9) : Variables JavaScript dans HTML :
 - on peut utiliser les variables JavaScript pour paramétrer la valeur des attributs dans les balise HTML.
 - La page devient donc dynamique selon le changement de la valeur des attributs par celle de ces variables

PROGRAMMATION CLIENT WEB Langage JavaScript

- JavaScript et HTML (10) : Variables JavaScript dans HTML :
 - Exemple : si taillefont est une variable JavaScript (défini avant son appel dans une balise SCRIPT par exemple) :
` texte de taille variable`
permettra d'afficher texte de taille variable dans une taille définie par JavaScript

PROGRAMMATION CLIENT WEB

Langage JavaScript

- OBJETS DOCUMENTAIRES

- JavaScript dispose d'un ensemble d'objets prédéfinis spéciaux dits objets documentaires décrivant les pages HTML sur un navigateur. Ces objets font partie du modèle DOM (Document Object Model)
- Ces objets sont prédéfinis et accessibles sans que vous ayez à les créer dans toute page HTML.

PROGRAMMATION CLIENT WEB

Langage JavaScript

- OBJETS DOCUMENTAIRES (2)

- ce sont :
 - *Windows* : l'objet de plus haut niveau, c'est la fenêtre que vous voyez dans le navigateur.
 - *child window* : ce sont les fenêtres filles que l'on retrouve dans chacune des frames. On verra leur appel dans les déclarations des frames
 - *location* : c'est l'URL courante et ses propriétés
 - *history* : ce sont les URL précédemment visitées.
 - *document* ce sont les propriétés du document courant, comme le titre, les couleurs de fonds, les formulaires
 - *navigator* ce sont le nom et la version du navigateur ainsi que les types MIME et les PLUG-INS supportés par le client.

PROGRAMMATION CLIENT WEB

Langage JavaScript

- OBJETS DOCUMENTAIRES (3) :
- Hiérarchie d'objets (composition mais non héritage !)
 - window (attributs : parent, frames, self, top)
 - location
 - history
 - document
 - Form : INPUT (text fields, textarea, checkbox, password, radio, select, button, submit, reset)
 - links, images, anchors, applets, arguments, embeds, mimeTypees, options, plugins, URL
 - navigator

PROGRAMMATION CLIENT WEB

Langage JavaScript

PROGRAMMATION CLIENT WEB Langage JavaScript

- **OBJETS DOCUMENTAIRES (4) :**
Tableaux d'objets prédéfinis

anchors	comprend toutes les balises <A> qui contiennent un argument NAME
applets	comprend toutes les balises <APPLET> du document
arguments	comprend tous les arguments d'une fonction
elements	comprend toutes les balises <FORM> dans l'ordre de leur définition
embeds	comprend toutes les balises <EMBED> dans l'ordre de leur définition
forms	comprend toutes les balises <FORM> du document
frames	comprend toutes les balises <FRAME> contenant un FRAMESET dans l'ordre de leur définition

PROGRAMMATION CLIENT WEB Langage JavaScript

- **OBJETS DOCUMENTAIRES (5) :** Tableaux d'objets prédéfinis

history	comprend tous l'historique du navigateur
images	comprend toutes les balises dans l'ordre de leur définition
links	comprend tous les liens <AREA HREF="...">, et les objets Link créés par la méthode linkwith the link
mimeTypes	Comprend tous les types MIME supportés par le navigateur (helpers ou plug-ins)
options	comprend tous les éléments OPTION d'une balise SELECT
plugins	comprend tous les plug-ins installés sur votre navigateur

PROGRAMMATION CLIENT WEB Langage JavaScript

- OBJETS DOCUMENTAIRES (6) : Exemple

```
Le document :  
<HTML>  
<BODY>  
<FORM NAME="formulaire"  
METHOD="Post"  
ACTION="http://nyx.unice.fr/test.cgi">  
<INPUT TYPE="text"  
NAME="nom"><P>  
<INPUT TYPE="button"  
NAME=" b" VALUE=" aaa">  
<INPUT TYPE="submit"  
VALUE="Envoyer">  
</FORM>  
</BODY>  
</HTML>
```

Les objets connus dans ce document :

- document.URL :
l'URL de ce document
- document.bgColor : sa couleur de fond
- document.title : son titre
- document.lastModified :
sa date de dernière modification
- document.formulaire : ce formulaire
- document.formulaire.action : l'action exécutée
- document.formulaire.length :
le nombre de composants, 3
- document.formulaire.bouton1 : le bouton
- document.formulaire.bouton1.value :
la chaîne échappée sur ce bouton
- document.formulaire.nom : le champ textuel
- document.formulaire.nom.value :
la chaîne écrite par l'utilisateur dans ce champ

PROGRAMMATION CLIENT WEB Langage JavaScript

- CADRES :

- **JavaScript est particulièrement utile pour manipuler les différents panneaux d'un cadre**
- Objets window
 - Propriétés : defaultStatus - frames - length - name - parent - self - status - top - window
 - Méthodes : alert - close - confirm - open - prompt - setTimeout - clearTimeout
 - Syntaxe :
 - windowVar = window.open("URL", "Nom" [, "options"])

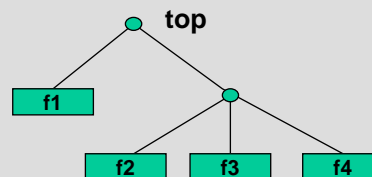
PROGRAMMATION CLIENT WEB Langage JavaScript

- CADRES (2) :
 - Les balises Frameset définit une hiérarchie des fenêtres dont les feuilles sont des panneaux d'affichage (frame)
 - Chaque panneau correspond à objet windows qui peut-être manipulé depuis des fonctions JavaScript

PROGRAMMATION CLIENT WEB Langage JavaScript

- CADRES (3) : Hiérarchie de fenêtres - Exemple

```
<FRAMESET COLS="50,**">  
<FRAME NAME=" f1"  
SRC="tdm.html"  
NORESIZE SCROLLING="no">  
<FRAMESET ROWS="20%,70%,10%">  
<FRAME NAME=" f2"  
SRC="titre.html" NORESIZE>  
<FRAME NAME=" f3"  
SRC="chap1.html">  
<FRAME NAME=" f4"  
SRC="navig.html" NORESIZE>  
</FRAMESET>  
</FRAMESET>
```



- La racine correspond à top
- Chaque FRAMESET crée un nœud intermédiaire
- frames permet d'accéder aux nœuds de même niveau
- parent permet de remonter dans l'hiérarchie
- self permet de retrouver le nœud courant
- es panneaux sont accessibles par son nom

PROGRAMMATION CLIENT WEB

Langage JavaScript

- CADRES (4) : Ouvrir simultanément plusieurs documents
 - Quand on travaille avec plusieurs fenêtres (notamment des panneaux), on peut vouloir ouvrir simultanément plusieurs documents en suivant un seul lien.
 - Les scripts suivants réalisent ceci à l'aide d'un bouton dans f4

PROGRAMMATION CLIENT WEB

Langage JavaScript

- CADRES (5) : Exemple

```
<HTML><HEAD>
<SCRIPT LANGUAGE="JavaScript">
function suivre_liens(page1, page2) {
top.f1.location = page1;
parent.f3.location = page2;
}
</SCRIPT></HEAD>
<BODY>
<FORM NAME="boutons">
<INPUT TYPE="button"
VALUE="Chapitre 1"
onClick="suivre_liens('p1.html','p2.html')">
</FORM></BODY></HTML>
```

Comportement :
-Le click sur le bouton
«chapitre1 »provoque le chargement
de documents p1.html dans f1 et
p2.html dans f3

Utiliser un lien à la place du bouton :
<A HREF="p1.html" TARGET="f1"
onClick="parent.f3.location =
'p2.html'"> Chapitre 1

On peut aussi utiliser une URL dont le protocole est JavaScript (au lieu de http, mailto ou ftp, etc), contenant une expression JavaScript.
Chapitre 1

PROGRAMMATION CLIENT WEB Langage JavaScript

- Traitement des formulaires
 - Radio :
Un groupe de boutons à choix exclusif doit porter le même nom.
 - Checkbox :
Contrairement aux boutons radio, chaque checkbox a son nom.

PROGRAMMATION CLIENT WEB Langage JavaScript

- Traitement des formulaires (2): Exemple

```
<FORM NAME="f" ...>
<INPUT TYPE="radio" NAME="q4" VALUE="0" onClick="..."> Non <BR>
<INPUT TYPE="radio" NAME="q4" VALUE="1" onClick="..."> Oui <BR>
</FORM>
```

 - Ces deux boutons radio sont désignés en JavaScript par
 - `document.f.q4[0]` et `document.f.q4[1]`.
 - La propriété `checked` indique si le bouton est sélectionné: `document.f.q4[0].checked` vaut `true` ou `false`.
 - L'événement `onClick` permet d'exécuter une action quand ce bouton est sélectionné.

PROGRAMMATION CLIENT WEB Langage JavaScript

- Traitement des formulaires (3)
 - **Select** : Chaque liste de sélection, dans un formulaire, est un objet. Ses propriétés sont :
 - name : son nom
 - options : le tableau de ses options, désignées par options[0], options[1], etc.
 - length : le nombre de ses options
 - selectedIndex : l'indice de l'option sélectionnée (ou de la première sélectionnée en cas de sélection multiple).
 - Chaque option d'une liste de sélection est un objet. Ses propriétés sont :
 - name : son nom, qui est la valeur de l'attribut NAME
 - index : son indice
 - text : le texte
 - value : la valeur de l'attribut VALUE
 - selected : booléen qui indique si l'option est sélectionnée
 - L'événement onChange permet de traiter la sélection faite par l'utilisateur.

PROGRAMMATION CLIENT WEB Langage JavaScript

- Traitement des formulaires (5): **Exemple d'additionner deux champs**

```
<FORM>
A = <INPUT TYPE = "text" NAME = "A"><BR>
B = <INPUT TYPE = "text" NAME = "B"><BR>
<INPUT TYPE = "button" VALUE = "Additionner"
onClick = "this.form.S.value =
parseInt(this.form.A.value) + parseInt(this.form.B.value)"
<INPUT TYPE="text" NAME="S">
```

parseInt est une fonction qui évalue une expression sous format texte et retourne le résultat de l'évaluation en entier

PROGRAMMATION CLIENT WEB Langage JavaScript

- Animation graphique : Temporisation
 - La méthode `setTimeout` d'une fenêtre permet d'évaluer une expression à l'expiration d'un délai (spécifié en millisecondes) :
 - Exemple : `affaire = setTimeout(tropTard(), 10000)`
 - On peut l'utiliser par exemple pour laisser un certain temps à l'utilisateur pour répondre à un formulaire ; une fois le temps écoulé, la fonction `tropTard()`, dénie par l'utilisateur (par exemple un appel à `alert()`), sera exécutée.

PROGRAMMATION CLIENT WEB Langage JavaScript

- Animation graphique (2) : Temporisation
 - La méthode `clearTimeout` annule le compte à rebours. Une fonction de traitement local du questionnaire (par exemple traitant l'événement `onSubmit`) devra exécuter
 - Exemple : `clearTimeout(affaire)`

pour empêcher l'exécution de `tropTard()`.

PROGRAMMATION CLIENT WEB Langage JavaScript

- Animation graphique (3) : Images animées
 - Les images contenues dans un document sont accessibles comme des propriétés du document. En modifiant l'URL d'une image, on peut facilement faire défiler une suite d'images.

PROGRAMMATION CLIENT WEB Langage JavaScript

- Animation graphique (4) : Exemple d'Images animée

```
<SCRIPT language="JavaScript">
var temps = 100, numero = 1
// Chargement des images
lesImages = new Array()
for (i = 1; i < 6; i++) {
  lesImages[i] = new Image()
  lesImages[i].src = "images/animation"
  + i + ".gif"
}
function animer() {
  document.anim.src = lesImages[numero].src
  numero = (numero < 5) ? numero + 1 : 1
}
</SCRIPT>
```

```
<IMG NAME="anim" SRC="images/critic1.gif"
ALT="[Animation]"
onLoad="setTimeout('animer()', temps)">
```

PROGRAMMATION CLIENT WEB Langage JavaScript

- Fenêtres et documents dynamiques
 - Une fenêtre, la fenêtre principale, est ouverte quand le navigateur est lancé ; quand un document, qui consiste en la définition d'un cadre (<FRAMESET> est chargé, une fenêtre est créée pour chacun des panneaux (<FRAME>) spécifiés. Quand l'attribut TARGET d'un lien a pour valeur _blank, une fenêtre indépendante est également créée. Toutes les façons précédentes de créer une fenêtre sont du ressort d'HTML. Il y a d'autres moyens de créer des fenêtres grâce à JavaScript.

PROGRAMMATION CLIENT WEB Langage JavaScript

- Fenêtres et documents dynamiques (2)
 - Fenêtre de dialogue :

Les méthodes alert, confirm et prompt, de l'objet window créent des fenêtres indépendantes spéciales :

 - Fenêtre Alert
`alert("Erreur : cette réponse n'est pas permise")`
 - Fenêtre Confirm
`confirm("voulez-vous vraiment quitter cette page ?")`
 - prompt("Quel est votre nom ?", nom)
 - où nom est une variable ou une propriété JavaScript.
 - Alert affiche d'un message, par contre confirme affiche et saisie une valeur booléenne de retour (true ou false)

PROGRAMMATION CLIENT WEB Langage JavaScript

• Fenêtres et documents dynamiques (3) : Création de fenêtres

- Les méthodes open et close permettent de créer et de fermer une fenêtre :
 - `var fenetre = open("truc.html", "truc")`
- La valeur retournée par la méthode open est affectée à la variable fenetre, qui aura les propriétés et les méthodes des objets de type window.
 - Le premier argument est une URL qui désigne le document qui doit être chargé dans cette fenêtre ; si la chaîne est vide, le contenu du document devra être créé.
 - Le deuxième argument donne à la fenêtre un nom qui pourra être utilisé comme valeur de l'attribut TARGET. Un troisième argument, optionnel, est une chaîne de caractères qui spécifie des propriétés de la fenêtre créée, par exemple :
 - `var fenetre = open("", "truc", "width=300,height=100,status,scrollbar")`

PROGRAMMATION CLIENT WEB Langage JavaScript

• Fenêtres et documents dynamiques (4) : Création de DOCUMENTS

- Addition de texte : La seule façon d'opérer sur le texte d'un document est d'y ajouter du texte, ce qui sera exécuté une seule fois, au chargement du document:

```
<SCRIPT LANGUAGE="JavaScript">
```

```
document.write("<FONT SIZE=-1>Dernière mise à jour  
: " + document.lastModified + "</FONT>")
```

```
</SCRIPT>
```

PROGRAMMATION CLIENT WEB Langage JavaScript

- Fenêtres et documents dynamiques (5) :
Création de DOCUMENTS

- Documents dynamiques : Il s'agit de l'équivalent, côté client, des documents construits par un programme CGI, côté serveur.
- Les méthodes open, write et close, appliquées à un document, permettent de créer un document, d'y écrire et de le fermer. Le contenu du document peut ainsi être calculé (à partir de données saisies dans un formulaire, la date courante, etc).
 - `var fenetre = open("", "truc")`
 - `fenetre.document.open("text/html")`
 - `fenetre.document.write("<HTML> ... </HTML>")`
 - `fenetre.document.close()`

PROGRAMMATION CLIENT WEB Langage JavaScript

- COUCHES DYNAMIQUES

Si un document définit des couches à l'aide de styles nommés, les propriétés de ces couches sont accessibles par JavaScript. Ainsi, avec le style

PROGRAMMATION CLIENT WEB Langage JavaScript

• COUCHES DYNAMIQUES (2)

```
<STYLE>
#couche1 {
position: absolute;
top: 2cm;
left: 2cm;
width: 6cm;
z-index: 1;
}
#couche2 {
position: absolute;
top: 3cm;
left: 3cm;
width: 6cm;
z-index: 1
}
</STYLE>
```

Les propriétés document.couche1 et document.couche2 sont créées. Ce sont des objets qui ont les propriétés suivantes :

- document : le contenu de la couche (qui figurera par exemple dans un ...)
- left, top : position relative à l a couche
- Parente pageX, pageY : position absolue (relative à la fenêtre)
- zIndex
- bgColor
- src : l'URL du document chargé dans la couche

Les méthodes suivantes s'appliquent :

- moveBy(dx, dy)
- moveTo(x,y)
- moveToAbsolute(x,y)
- resizeTo(width, height)
- load(sourcestring, width)

Un objet couche peut être créé dynamiquement par

```
couche3 = new Layer(200)
// 200 est la largeur
couche4 = new Layer(200, couche3)
// couche3 est parente de couche4
```

Après l'avoir créée, les diverses propriétés de la couche peuvent être spécifiées.

PROGRAMMATION CLIENT WEB Langage JavaScript

• COUCHES DYNAMIQUES (3)

- Une couche a les mêmes capacités de traitement d'événement qu'un document séparé.
- Par exemple, la couleur de fond d'une couche peut être modifiée par le script local suivant

PROGRAMMATION CLIENT WEB Langage JavaScript

- COUCHES DYNAMIQUES (4)

```
<SPAN ID="couche1">  
<P>Une couche de couleur variable</P>  
<SCRIPT>  
function changeColor(newcol) {  
  bgColor=newcol;  
  return false;  
}  
function onMouseOver() {changeColor("blue");}  
function onMouseOut() {changeColor("red");}  
</SCRIPT>  
</SPAN>
```

PROGRAMMATION CLIENT WEB Langage JavaScript

- Annexes 1 : Les évènements

- onBlur : se produit quand un textarea, un text ou un select perd la main sur les entrées clavier
- onChange : se produit quand un textarea, un text ou un select est modifié par l'utilisateur
- onClick : se produit quand un button, checkbox, radio, link, reset ou un submit reçoit un click de la souris
- onFocus : se produit quand un textarea, un text ou un select prend la main sur les entrées clavier
- onLoad : se produit quand le navigateur a fini de charger une fenêtre ou toutes les frames d'un FRAMESET. L'évènement *onLoad* se positionne dans la balise BODY ou dans la balise FRAMESET

PROGRAMMATION CLIENT WEB

Langage JavaScript

- Annexes 1 : Les évènements (2)
 - onmouseover : se produit quand la souris passe sur un link ou un layer (version 1.2)
 - onSelect textarea, un text: se produit quand un textarea, un text ou un text est sélectionné.
 - onsubmit : se produit quand une form est soumise au serveur par l'appui du bouton *Submit*.
 - onunload : se produit quand un document est quitté. L'évènement *onUnload* se positionne dans la balise BODY ou dans la balise FRAMESET
 - onabort : (*version 1.1*) se produit quand l'utilisateur avorte le chargement d'une image

PROGRAMMATION CLIENT WEB

Langage JavaScript

- Annexes 1 : Les évènements (3)
 - onerror : (*version 1.1*) se produit quand le chargement d'une page ou d'une image produit une erreur.
 - onmouseout : (*version 1.1*) se produit quand la souris quitte une zone area , un link ou un layer (version 1.2)
 - onreset : (*version 1.1*) se produit quand on clique sur le bouton *reset* d'un formulaire
 - ondblclick : (*version 1.2*) se produit quand on produit un double click sur la souris (ne fonctionne pas sur Macintosh ou le double clic a sa signification propre)
 - ondragdrop : (*version 1.2*) se produit lorsqu'on fait un glisser lâcher vers le navigateur, par exemple pour ouvrir une page HTML depuis son disque dur.

PROGRAMMATION CLIENT WEB Langage JavaScript

• Annexes 1 : Les évènements (4)

- `onKeyDown` : (*version 1.2*) se produit quand une touche du clavier est lâchée.
- `onKeyPress` : (*version 1.2*) se produit quand une touche du clavier a été utilisée
- `onKeyUp` : (*version 1.2*) se produit quand une touche du clavier est appuyée
- `onMouseDown` : (*version 1.2*) se produit quand une touche de la souris est lâchée.
- `onMouseMove` : (*version 1.2*) se produit quand le curseur de la souris est bougé
- `onMouseUp` : (*version 1.2*) se produit quand un bouton de la souris est relâché
- `onMove` : (*version 1.2*) se produit quand une fenêtre est déplacée
- `onResize` : (*version 1.2*) se produit quand une fenêtre subit un changement de taille

PROGRAMMATION CLIENT WEB Langage JavaScript

• Annexes 2 : Les propriétés

propriété	objets	description
<code>action</code>	form	détermine l'URL appelée après appui sur le bouton <i>Submit</i>
<code>alinkColor</code>	document	détermine la couleur des liens visités
<code>appName</code>	navigator	donne le nom de code du navigateur
<code>appVersion</code>	navigator	donne le nom du navigateur
<code>appCodeName</code>	navigator	donne une chaîne de caractères contenant la version, la plateforme et la langue du navigateur
<code>bgColor</code>	document	donne la couleur de fond du document

PROGRAMMATION CLIENT WEB Langage JavaScript

- Annexes 2 : Les propriétés (2)

propriété	objets	description
checked	checkbox, radio	Renvoie la valeur de l'objet True/False
cookie	document	permet d'accéder à des variables internes
defaultChecked	Checkbox, radio	indique l'état par défaut de l'objet concerné
defaultSelected	option	Retourne une valeur indiquant l'état par défaut d'un objet sélectionnable
defaultStatus	window	est le message par défaut affiché dans la région de statut de votre navigateur
defaultValue	hidden, pword, text, textarea	chaîne de caractères indiquant la valeur par défaut d'une zone texte, mot de passe ou zone texte

PROGRAMMATION CLIENT WEB Langage JavaScript

- Annexes 2: Les propriétés (3)

propriété	objets	description
E	Math	Constante e (environ 2.718)
encoding	form	chaîne spécifiant l'encodage MIME dans une forme c'est à dire la valeur ENCTYPE
fgColor	document	chaîne spécifiant la couleur du texte d'un document
getselection	document	renvoi le texte sélectionné dans le navigateur
hash	link, location	chaîne commençant par le caractère # qui spécifie une ancre dans une URL

PROGRAMMATION CLIENT WEB Langage JavaScript

- Annexes 2: Les propriétés (4)

propriété	objets	description
host	link, location	chaîne de caractères déterminant la partie <i>host</i> d'une URL
hostname	link, location	chaîne de caractères déterminant la partie <i>hostname</i> d'une URL
href	link , location	chaîne de caractères déterminant l'URL
index	option	entier représentant l'index d'une option de l'objet sélectionné
lastModified	document	la date de dernière modification d'un document

PROGRAMMATION CLIENT WEB Langage JavaScript

- Annexes 2: Les propriétés (5)

propriété	objets	description
length	frame, history, radio, select, string, window, anchors, elements, forms, frames, links, options	longueur de l'objet ou nombre d'objet
linkColor	document	Spécifie la couleur des liens hypertextes
LN2	Math	logarithme de 2 (0,693)
LN10	Math	logarithme de 10 (2,302)
location	document	l'URL complet du document

PROGRAMMATION CLIENT WEB Langage JavaScript

- Annexes 2: Les propriétés (6)

propriété	objets	description
LOG2E	Math	Logarithme base 2 de e (1.442)
LOG10E	Math	Logarithme base de 10 de e (0,434)
name	objet	window.name est le nom de la fenêtre
method	form	spécifie la méthode (POST,GET) utilisée pour envoyer un formulaire

PROGRAMMATION CLIENT WEB Langage JavaScript

- Annexes 2: Les propriétés (7)

propriété	objets	description
parent	Window, frame	c'est la fenêtre contenant les frames
pathname	location link	C'est la partie répertoire d'un URL
preference	navigator	permet de connaître ou de positionner les préférences du navigateur. Voir éléments avancés de JavaScript
protocol	location	protocole de l'URL (http, ftp etc..)
referrer	document	l'URL du document appelant le lien cliqué

PROGRAMMATION CLIENT WEB Langage JavaScript

- Annexes 2: Les propriétés (8)

propriété	objets	description
search	location	chaîne commençant par un point d'interrogation et correspondant aux informations d'une requête
selectedIndex	Select, options	donne le rang de l'objet sélectionné dans une liste
self	window, frame	désigne la frame ou la fenêtre courante
SQRT1_2	Math	racine carrée de 1/2 (0,707)
SQRT2	Math	racine carrée de 2 (1,414)

PROGRAMMATION CLIENT WEB Langage JavaScript

- Annexes 2: Les propriétés (9)

propriété	objets	description
status	window	c'est la fenêtre status en bas à droite de Netscape
target	form, link, location	c'est le nom de la fenêtre qui va afficher le résultat d'un clic sur un URL ou un bouton de soumission
text	options	texte suivant une balise <OPTION> dans une sélection
title	document	c'est le titre d'un document
top	window	c'est la fenêtre de plus haut niveau, c'est à dire celle initiale

PROGRAMMATION CLIENT WEB Langage JavaScript

- Annexes 2: Les propriétés (10)

propriété	objets	description
userAgent	navigator	nom d'agent du logiciel de navigation
value	button, checkbox, hidden, password, radio, reset, submit, text, textarea, options,	valeur de l'objet
vlinkColor	document	couleur des liens visité
window	window	c'est la fenêtre courante