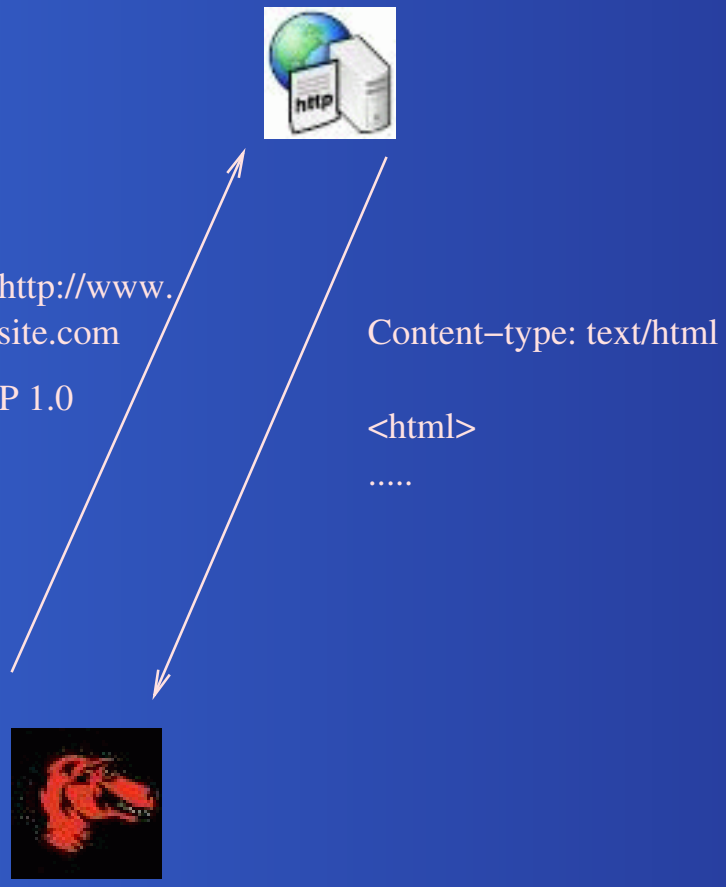
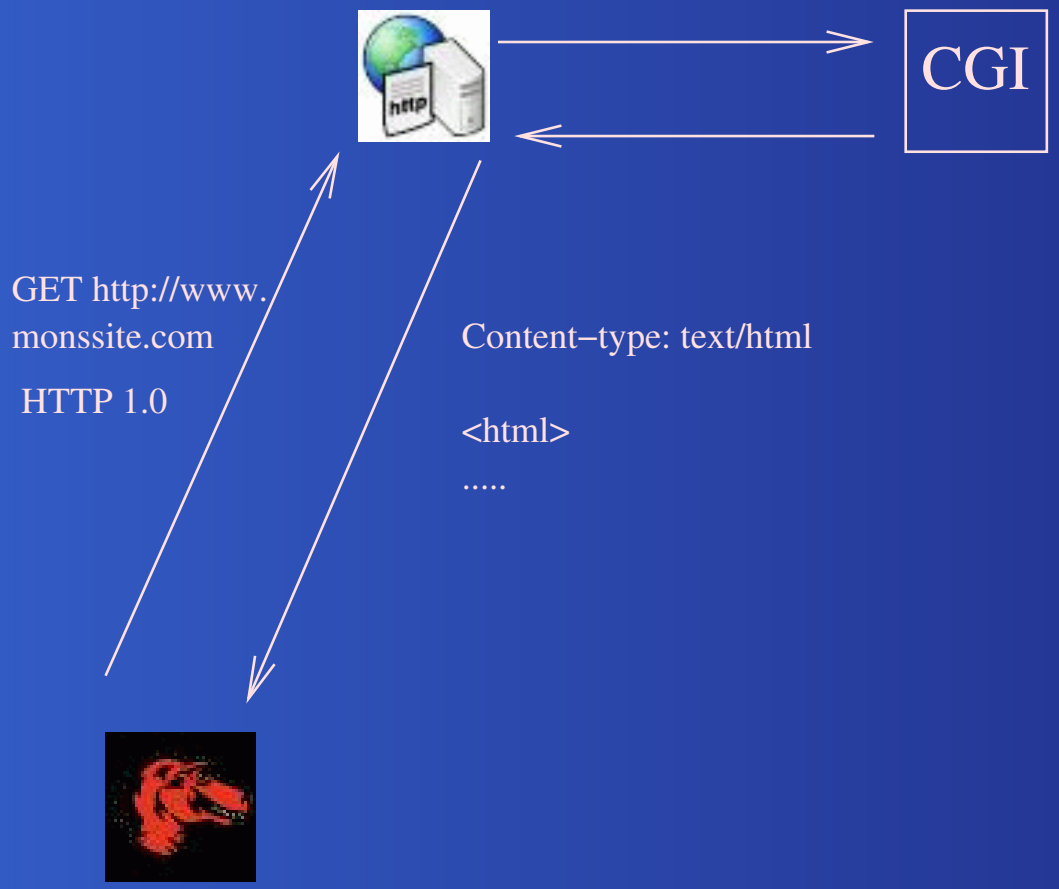
Création de pages Web Dynamiques

Luc Brun

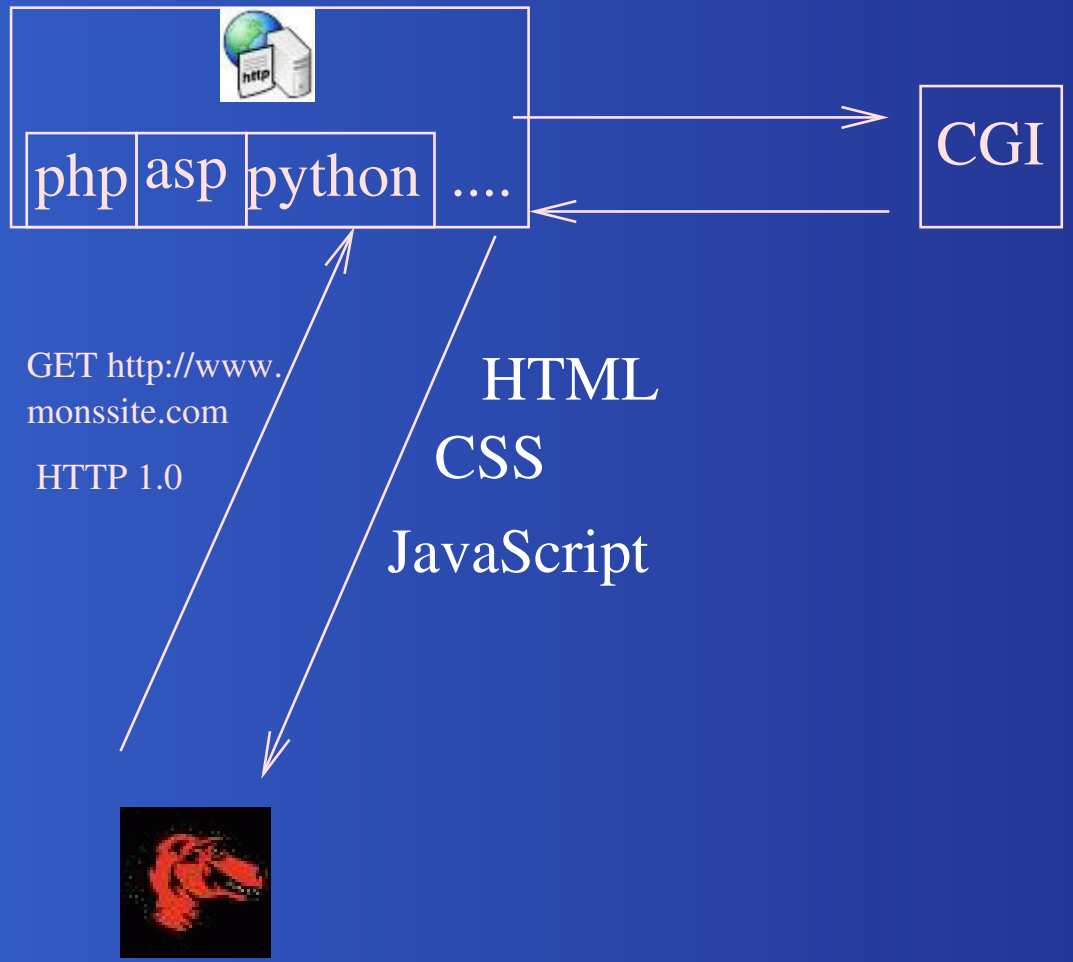
Place du HTML



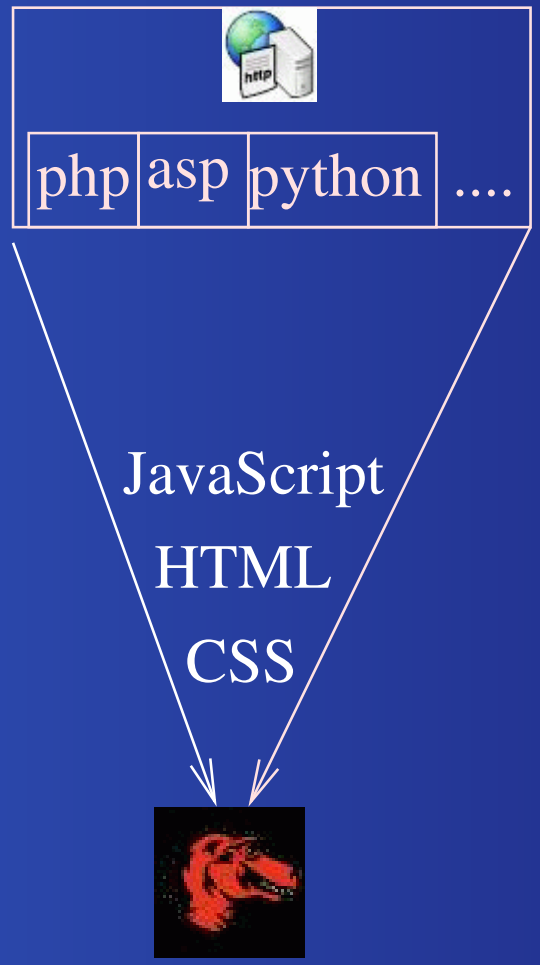
Place du HTML



Place du HTML



Intérêt du Cours



HTML

- Langage à balises : `<b>toto< /b>`→**toto** en gras.
- Forte parenté avec XML (HTML+XML→ XHTML) :
  - Toute balise ouverte doit être fermée,
  - Une balise ne délimitant rien se ferme elle même (exemple : `<br/ >`)

Structure d'un document HTML

Deux sections :

- head : Définitions générales sur le document : Informations non affichées,
- body : corps du document.
  - textes,
  - tableaux,
  - listes (à puce, numérotées),
  - images. . .

La section head

```
<head>
<title>Un exemple de section head</title>
<link rel="stylesheet" type="text/css"
href="mon_css.css"/>
<meta http-equiv="Content-Type"
content="text/html; charset=ISO-8859-1"/>
<meta name="keywords" content="html,head"/>
<meta name="author" content="Luc Brun"/>
<meta name="description"
content="description de la section head"/>
</head>
```

Les textes

- Jeu de caractère par défaut : ASCII. Codage des accents par des séquences de caractères (é $\leftrightarrow$ &eacute ;) (Cf. Tab.2.1)
- langage non wyswig $\Rightarrow$ (blanc :   ; retour à la ligne : <br/ >).

> ⚠ Éviter de coder la mise en forme dans la structure du texte.

Modification de polices

- Explicite :
  - `<b>` : gras ;
  - `<i>` : italique
  - `<u>` : souligné
- Logique :
  - `<big>`, `<small>` : tailles,
  - `<strong>`, `<em>`(emphase) : mise en évidence.

Titres, lignes

- Titres : <hx> avec $x \in \{1, \ldots, 6\}$
  - <h1> titre de niveau 1,
  - <h6> titre de niveau 6,
- Lignes horizontales : <hr/> ———————

Paragraphes

Balise `<p>` avec comme principal attribut `align`$\in \{left, center, right\}$.


Utilisez des pagraphes plutôt que des sauts de lignes (`<br/ >`).

Divisions

Balise `<div>` permet de regrouper un ensemble de textes/balises. Permet d'affecter des propriétés à un ensemble de balises :

```
<div align="right">
   <p>un premier paragraphe</p>
   <p> et un second</p>
   <p>tous align&eacute;s
   &agrave; droite</p>
</div>
```

Tous alignés à droite : Concept d'héritage.

Les listes

- liste non numérotées (balise `<ul>`) ou numérotées (balise `<ol>`) :
  - item : `<li>contenu< /li>`
  - type : type de liste (tables 2.2 et 2.3)
- liste de description : balise `<dl>` (description list)
  - titre item `<dt></dt>`
  - description de l'item `<dd></dd>`

Les tableaux

Trois balises :

- table : délimite une table,
- tr : délimite une ligne,
- td : délimite une cellule.

Possibilités de cellules multi-lignes, multi-colonnes, de sélection de fond de tables, de lignes, de cellules. Tables de taille fixes/variables.

Les liens

Change l'url de la page.
`<a href="url"> lien</a>` affiche lien et change l'url courante pour celle spécifiée lors d'un clic sur le texte. L'url peut être un lien vers :

- un serveur : www.ismra.fr
- une page www.ismra.fr/annuaire.html
- une application CGI www.ismra.fr/toto.cgi (voir plus loin)
- une adresse mail : mailto :toto@ismra.fr

La zone délimitée par `<a></a>` peut être du texte, une image. . .

Les ancres

Permet de positionner une marque (une ancre) dans une page HTML. Des liens peuvent être positionnés sur cette ancre.

Syntaxe :

```
<a href="identifiant"/>
```

Liens sur l'ancre :

```
<a href="page.html#identifiant">
Lien
</a>
```

Les images

Inclus une image : Syntaxe :

```
<img src="logo_ensicaen.jpg"
width="2cm" height="3cm"
align="left" hspace="5mm"
alt="Logo de L EnsiCaen"/>
```

Possibilité de spécifier la largeur (width), hauteur(height), l'alignement, l'espacement (hspace, vspace), la bordure (border) et le texte avant chargement (alt) (Cf. Tab. 2.6).

Les images clicables (1/2)

Permet de spécifier un lien sur certaines parties d'une image. Syntaxe :

```
<map name="id map">
<area shape="nom forme"
coords="liste coords" href="lien"/>
.
.
</map>
```

shape$\in \{rect, circle, polygon\}$

Les images clicables (2/2)

Valeurs de coords :

- rect (x_1, y_1, x_2, y_2) coins haut et bas,
- circle (x, y, R),
- polygon $(x_1, y_1, \ldots, x_n, y_n)$ liste des points du polygone.

Utilisation :

```
<img src="url" usemap="id map"/>
```

Les cadres

Découpe la fenêtre du navigateur en sous fenêtres (frames).
Exemple :

```
<frameset cols="200,*">
<frame src="menu.html" name="menu"/>
<frame src="main.html" name="main"/>
</frameset>
```

Liens dans les frames :

```
<a href="loisirs.html" target="main">
Mes loisirs</a>
```

De moins en moins utilisé.

Les formulaires

Permet à l'utilisateur de rentrer des informations par le biais de balises spécifiques. Trois attributs :

- action : nom du CGI ou de programme devant traiter les informations,
- method : méthode de passage des paramètres (GET ou POST)
- enctype : codage du document.

Les balises input

```
<input type="type" name="nom">
```

type : type d'input
name : nom de variable
value : valeur par défaut
checked : sélection par défaut
size : nb de caractères
maxlength : nb max. de caractères

type$\in$ {text, password, image, checkbox, radio, submit, reset}

La balise select

Code un menu à options :
Syntaxe :

```
<select name="un_menu">
        <option> 1er choix.</option>
        <option> 2eme choix.</option>
</select>
```

Attributs : name, size (nb d'éléments simultanéments affichés), multiple (sélection de plusieurs éléments).

La balise textarea

Saisie d'une zone de texte.
Syntaxe :

```
<textarea name="saison"
rows="10" cols="40">
L'hivers :
L'été :
</textarea>
```

Attributs : name, rows (nb ligne), cols (nb colonnes)

Les CSS

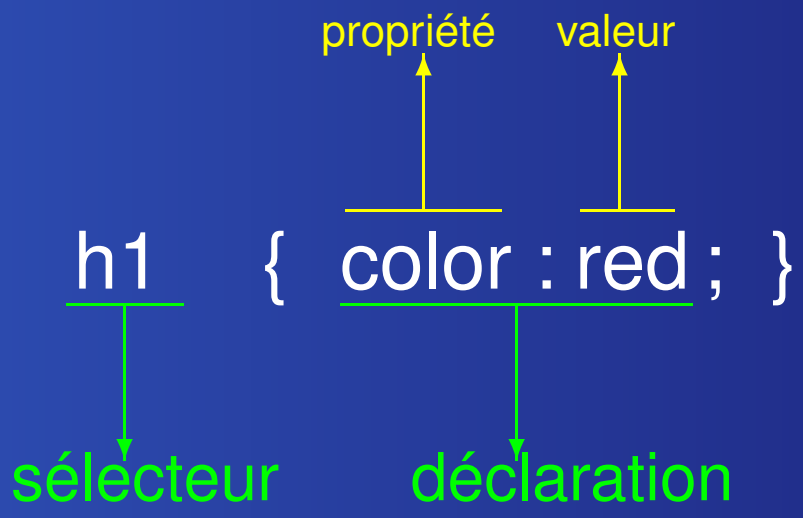
Cascading Style Sheets : Permet de modifier l'apparence de balises
Permet une claire différenciation entre :

- le contenu,
- la structure,
- la présentation.

 Fondamental !

Feuille de style

Feuille de style : ensemble de règles spécifiant la mise en forme de certaines balises.
Structure d'une règle :



Les sélecteurs

Spécifie la ou les balises qui vont utiliser les déclarations.
Sélecteur simple : nom de balise.

```
h1
{
    color: red;
    font-weight: bold;
    font-size: xx-large;
    text-align: center;
}
```

Les sélecteurs à évènements

Les déclarations s'appliquent sur un type de balise lors de certains évènements ou état.
Exemple : `a :hover {color : red; }`

Principaux évènements :

:focus	prise de focus par un texinput ou textarea
:hover	passage de la souris sur un lien
:link	lien pas encore visité
:visited	lien cliqué

Les différents types de sélecteurs

- simple : `a{ } ,`
- à évènement : `a:hover { } ,`
- contextuel : `p a{ } ,`
- groupés : `a,p,h1 { }`
- de classe : `.intro { }`
- d'identifiant : `#titre { }`

Voir pages HTML

Liaison HTML-CSS

- Les feuilles internes

```
<head>
<style type="text/css">
<!--
p { text-align: justify; }
--!>
</style>
</head>
```

Les balises `<!- -!>` permettent aux vieux navigateurs d'ignorer les feuilles de style.

Liaison HTML-CSS

- Les feuilles internes

```
<head>
<link rel="stylesheet"
type="text/css" href="ma_feuille.css"/>
</head>
```

ou

```
<head><style type="text/css">
@import url("feuille.css")
</style></head>
```

S'utilise en complément des feuilles externes.

Liaison HTML-CSS

- Les feuilles locales

```
<body>
<p style="text-align: justify;
color: red;"> Mon paragraphe</p>
</body>
```

Permet de rajouter localement une déclaration.

Liaison HTML-CSS

- Les feuilles utilisateur

L'utilisateur peut imposer des styles pour certaines balises (généralement menu préférences). Utile pour les mal voyants.

Les feuilles en cascades

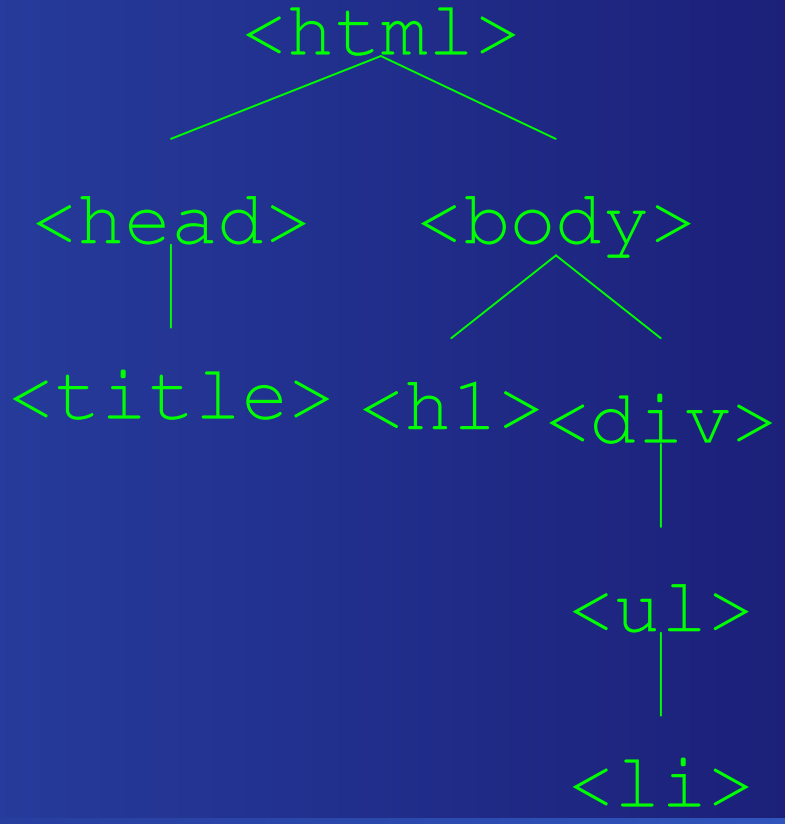
Conflits entre plusieurs règles :*La dernière règle lue a la priorité*

1. Utilisateur,
2. locales,
3. internes/externes : dernière règle lue (positionnement de balises),
4. conflit dans une même feuille : dernière règle lue.

Héritage

Document HTML→ arbre d'inclusion. Les déclarations s'héritent de père en fils.

```
<html><head>
<title> ma page</title>
</head><body>
<h1> Ma page </h1>
<div class="menu">
<ul>
<li> un item</li></ul>
</div></body></html>
```



Les tailles

Tailles de boîtes, de polices, d'images...

- Unités absolues :

mm	minimètres	**pc**	pica (1pica=12pt)
cm	centimètres	**pt**	points
in	inches	**px**	pixel

- Unité relative : %
  - font-size : 50% : dernière taille de police,
  - width : 10% : largeur de la boîte englobante (ex fenêtre)

Tailles de polices

- Unités relatives

xx-large	large
x-large	large
large	large
medium	normal
small	small
x-small	+ small
xx-small	encore + small

em 1em taille de la police précédente

Les couleurs

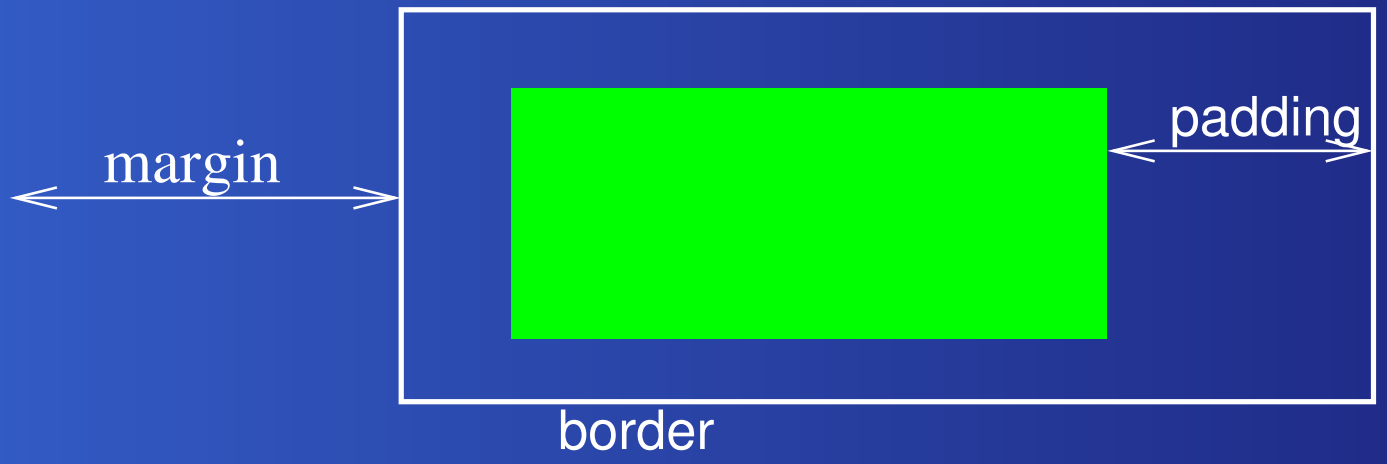
propriété concernées (color, background).

- Spécification du nom (aqua, black. . .) Voir Table 3.5
- Spécification de la valeur (R,G,B) par #rgb avec $r, g, b \in [0, f]$.

Alignement de texte

text-indent	Indentation de paragraphe.
text-align	justify, align,left,right et none
margin-left	marge à gauche
margin-right	marge à droite
margin-top	marge du haut
margin-bottom	marge du bas
margin	top right bottom left

Formatage de boîtes



- border : border-width, border-style, border-color ou border width, style color.
- padding : haut droite bas gauche.
- margin : idem que précédemment.

voir page HTML

Positionnement de boîtes

propriétés impliqués : position, left, top, right, bottom.
Valeurs de position :

- static : par défaut (les uns sous les autres),
- absolute : coordonnées fenêtres,
- fixed : idem absolute, insensible au scroll,
- relative : positionnement relatif.

Voir page HTML

Boîtes flottantes

Alignement à gauche (float : left ;) ou à droite (float :right ;) du bloc parent.
Voir page Web.

Gestion de la profondeur

propriété position$\Rightarrow$ possibilité de Recouvrement.

Gestion des recouvrements : propriété z-index.

En cas de recouvrement la boîte de z-index le plus élevé est affichée.

Les listes

Essentiellement trois propriétés :

1. `list-style-type` : type de liste que l'on compte manipuler (disc, circle, decimal...)
2. `list-style-image` : spécifie une image à la place des puces

```
ul {list-style-image: url(mon_item.jp
list-style-type: circle;}
```

3. `list-style-position` $\in$ {`inside`,`outside`}

JavaScript

- Avantage :
  - Exécuté sur le navigateur du client
- Inconvénients :
  - Exécuté sur le navigateur du client
  - Varie beaucoup en fonction des navigateurs des versions

Surtout utilisé pour de petits effets (apparition/disparition/déplacement de blocs) et la vérification des formulaires.

Variables

- Déclaration explicite de variable : `var i`.
  - Si en dehors de fonction : variable globale,
  - sinon variable locale.
- Déclaration implicite `toto="titi";` : variable globale.

L'instruction if

- `if (condition) {instruction}` ou
- `if (condition) {instruction} else {instruction}`

On peut étendre à : `if (condition) {instruction} else if (condition) {instruction}...`

L'instruction switch

```
switch (val) {
    case 'val1':
        instruction1
        break;
    case val2:
        instruction2
        break;
        . . .
    default:
        instruction par defaut.
        break;
}
```

Instruction 1 est exécuté si val=val1, Instruction 2 si val=val2. . ..
Si aucun test n'est vrai le bloc d'instruction par défaut est exécuté.

Les boucles (1/4)

- Boucle for : `for(init;cond;incr)`
  exemple

```
var sum=0;
for(i=0;i<10<i++)
sum+=i;
```

Les boucles (2/4)

- Boucle While :

```
while(cond){instruction}
```

exemple :

```
sum=0;
i=0;
while(i<10)
{
     sum+=i;
     i++;
}
```

Les boucles (3/4)

- Boucle Do-while :

```
do {instruction} while(cond);
```

exemple :

```
sum=0;
i=0;
do
{
     sum+=i;
     i++;
}
while(i<10);
```

Les boucles (4/4)

- Les séquences d'échapement

return	:	sort de la boucle et de la fonction.
break	:	sort de la boucle
continue	:	passe à l'itération suivante

Les tableaux

```
var tab_vide=new Array();
var tab_10 =new Array(10);
var tab_init=new Array('val1','val2');
```

Tableaux 5×5 :

```
var mat=new Array(5);
var i,j;
for(i=0;i<mat.length;i++)
   mat[i]=new Array(5);
```

Tableau associatif :

```
tab['email']="toto@titi.fr";
```

Manipulation de tableaux (1/2)

Concat	concaténe plusieurs tableaux.
Tableau.join()	conversion en chaîne
Tableau.pop()	supprime dernier élément
Tableau.push	ajoute un ou plusieurs éléments
Tableau.reverse()	inverse l'ordre des éléments du tableau.
Tableau.shift()	supprime le premier élément du tableau.

Manipulation de tableaux (2/2)

Tableau.slice()	partie d'un tableau.
Tableau.sort()	trie
Tableau.unshift	ajoute en tête de tableau
Tableau.toString()	concatène les objets de l'array en une String.
Tableau.valueOf	retourne la valeur de l'objet Array .

Tableau.splice(i,nb,[val1,]) substitue les éléments en position i à i+nb du tableau par les val_i. Renvoi les éléments supprimés.

Les fonctions

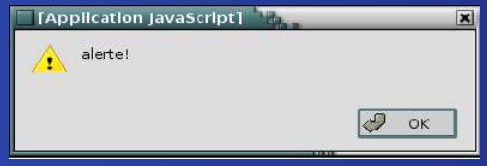
```
function nom(parametres)
{
instructions
[return valeur]
}
```

- Passage des arguments par valeur,
- Return non obligatoire (procédure)

Les boîtes de dialogue

- confirm('Message') : Afficher un message/une information,

- alert('Message') : Afficher un message d'alerte,



- val=prompt('Message',defaut) : Afficher un message et saisir une valeur.

Les classes (1/2)

```
var toto=new fonction([params]);
```

fonction : constructeur de l'objet.

```
function aire()
{  return this.largeur*this.hauteur;}
function Rectangle(x,y)
{// creation du champ largeur
    this.largeur=x;
// creation du champ hauteur
    this.hauteur=y;
// creation de la methode aire
    this.aire=aire;
}
```

Les classes (2/2)

Utilisation des classes :

```
mon_rect=new Rectangle(10,10);
document.write("largeur, hauteur, aire"+
mon_rect.largeur+mon_rect.hauteur+
mon_rect.aire());
```

Ajout dynamique de méthodes :

```
classe.prototype.methode=fonction
```

exemple :

```
Rectangle.prototype.perimetre=perimetre
```

La classe String

```
var toto="titi";
var titi=new String(); // chaine vide
```

concaténation : +

Liens avec la classe Math :

eval()	:	eval("4+5")$\longrightarrow$ 9.
parseInt()	:	conversion en entier
parseFloat()	:	conversion en float
toString()	:	conversion en String

La classe image

```
var mon_img=new Image();
mon_img.src='Images/mon_image.gif';
```

- Préchargée au chargement de la page.
- affichage : propriété src des balises <img>.

  `img.src=mon_img.src`

  Implique le changement dynamique de l'image.

La classe Date

```
ma_date = new Date();
```

Possibilité de spécifier une date ou de prendre la date système (Table 4.6)

getDate()	jour du mois
getDay()	jour de la semaine
getFullYear()	année sur 4 chiffres
getHours(),**getMinutes()**	heure, minutes
getMonth()	mois

Autres fonctions (Table 4.7) et exemples (pages HTML)

Insertion de code Javascript (1/2)

- Syntaxe :
  Balise `<script></script>`.
  Deux syntaxes :

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

  ou

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

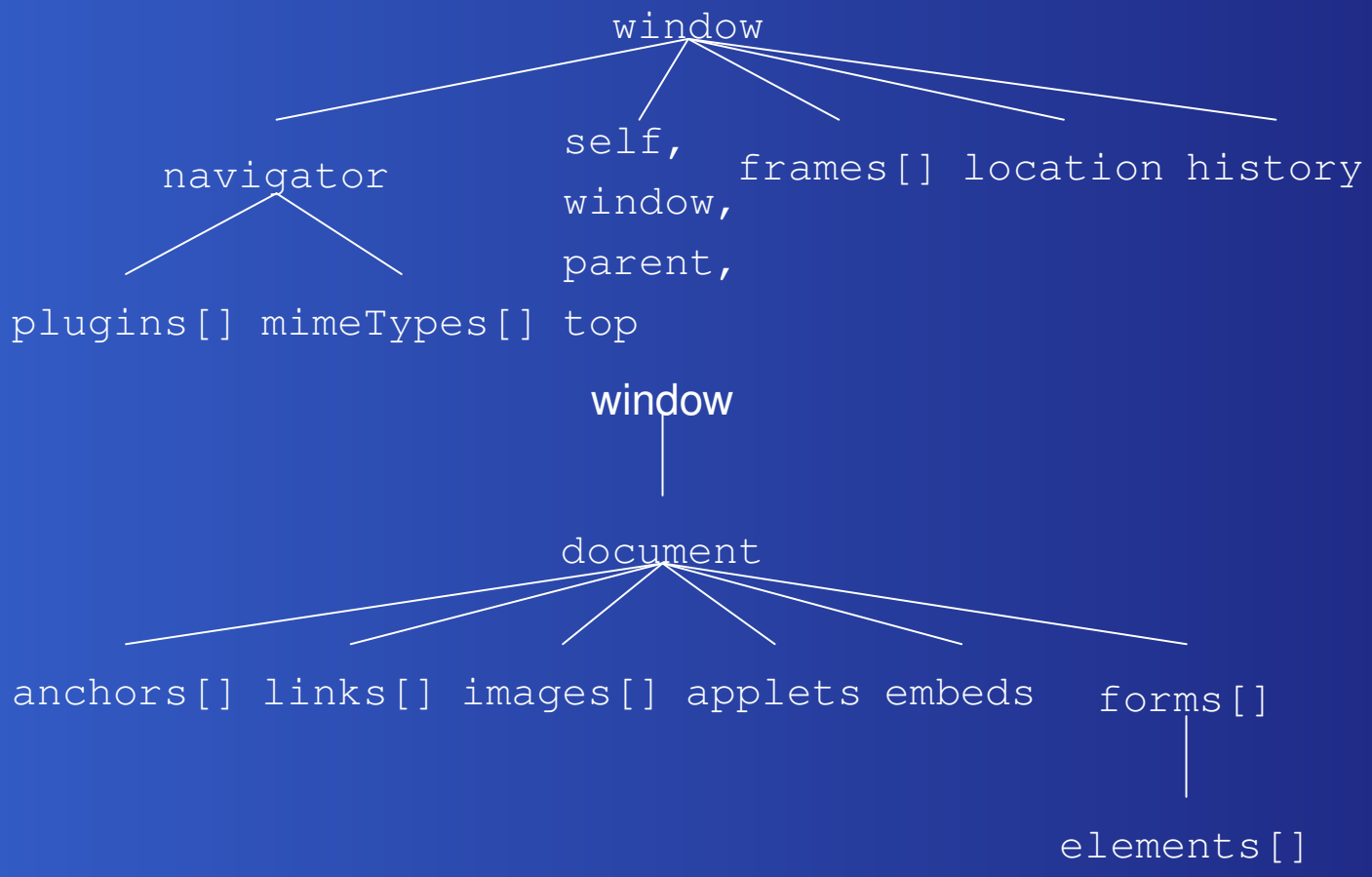
  Possibilité de combiner

Insertion de code JavaScript (2/2)

- Lieu d'insertion
  - Insertion pour exécution différée : Dans la balise `<head>`. Code exécuté sur demande (appel de fonction, instanciation de classe)
  - Insertion pour exécution directe Dans la balise `<body>`. Exécuté au chargement de la page.
  - Réaction à évènements (onclick, onblur. . .Table 4.9)

Possibilité de combiner

Hiérarchie des objets



Recherche par Id

Fonction : `document.getElementById`
Prise en compte des navigateurs :

```
function getobj(id)
{
   if(document.layers)
    return document.id; // Netscape 4.x
// Netscape 6.x IE 5.x
   if(document.getElementById)
    return document.getElementById(id)
   if(document.all)
    return id; // IE 4.x
}
```

Recherche par nom et balise

- Recherche par nom (attribut name) :
```
document.getElementsByName("nom");
```
- Recherche par nom de balise :
```
document.getElementsByTagName("balise"
```

Recherche par classe

```
function getElementbyClass(classname)
{
  var inc=0;
  var elt=new Array();
  var alltags=document.all;
  if(! alltags)
   alltags=document.getElementsByTagName("*");
  for (i=0; i<alltags.length; i++)
    if (alltags[i].className==classname)
      elt.unshift(alltags[i]);
  return elt;
}
```

Les CGI

Common Gateway Interface :

- Protocole de communication entre le serveur Web et des applications.
- De plus en plus supplanté par PHP, Applet Java.
- Garde tout son intérêt pour de grosses applications.

Récupération des paramètres

Paramètres : liste d'affectation variable=valeur. Chaîne d'affectations.

```
var1=val1&var2=val2. . .
```

- méthode POST : Le serveur envoi la chaîne d'affectations sur l'entré standard de l'application.
- méthode GET : Positionnement de la chaîne d'affectation dans la variable d'environnement `QUERY_STRING`.

Principales variables

- `REQUEST_METHOD` : type de passage de la chaîne de paramètres (GET, POST),
- `QUERY_STRING` : valeur de la chaîne d'affectations (mode GET),
- `CONTENT_LENGTH` : longueur de la chaîne d'affectation.

Décodage des affectations

- Un exemple en shell.

```
ifs=$IFS
IFS="&"
set $QUERY_STRING
IFS=$ifs
for egal in $*
do
        echo "$egal <br/>"
done
```

Envoi de données

```
Content-type: type mime

contenu du document
```

type mime a priori quelconque généralement :

- text/plain : ascii ou
- text/html : html.

Attention à la ligne vide entre Content-type et le contenu.

Les Cookies

Affectations de variables stockées chez l'utilisateur. Permet :

- De stocker des informations de pages en pages sur un même site,
- d'identifier un utilisateur,
- d'effectuer des statistiques sur les accès des utilisateurs.

Positionnement des Cookies

```
Content-type: text/html
Set-Cookie: var=val;

contenu du document
```

Possibilité de répéter la ligne `Set-Cookie` pour positionner plusieurs variables. La ligne vide marque le début du document.

Options :

- date d'expiration (`expires`),
- serveurs ayant accès au Cookie (`domain`),
- Zones du serveur ayant accès au Cookie (`path`),

Lecture des Cookies

Variable `HTTP_COOKIE`.

`var1=val1;va2=val2;. . .`

Lecture similaire à celle de `QUERY_STRING`.