

Plan

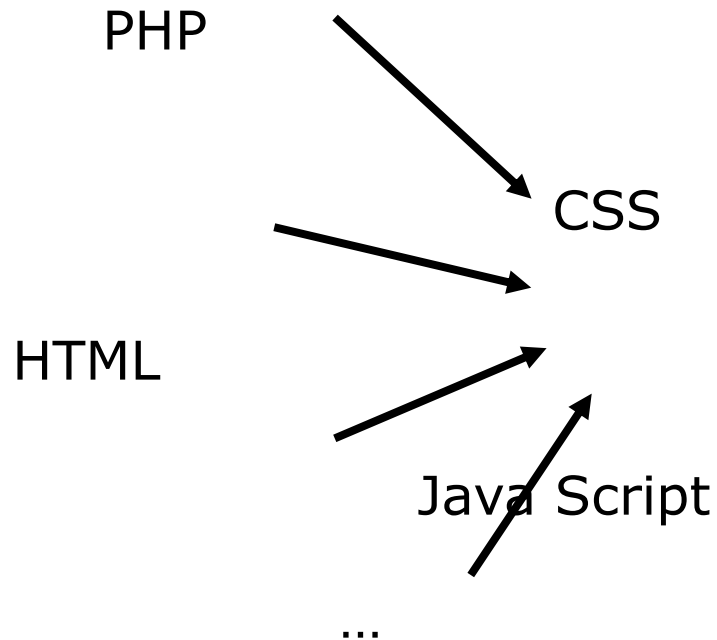
- **HTML**

- Les balises classiques
- Types de balises : Block et Inline
- Les modèles de boîtes (Box Model)
- Les balises DIV et SPAN
- Les attributs class et id

- **CSS**

- Style incorporé dans le document HTML
- Style séparé : lié ou importé
- Syntaxe des règles CSS
- Sélecteurs
- Les pseudo-classes
- Les super-propriétés
- Les modes de positionnement (Les flux).

Qu'est ce qu'une page ?



Page web



Qu'est qu'une balise

- Balisage
 - une **balise** s'écrit avec **<** et **>** deux caractères spéciaux de début et fin de balise et un identificateur(nom) prédéfini
 - le balisage structure le source en **éléments**
 - le balisage associe une balise ouvrante **<** **>** et une fermante **</** **>** par paire (non obligatoire).
 - le texte entre une balise ouvrante et une fermante est le **contenu** de l'élément
 - le balisage est insensible à la casse (préférer les minuscules, obligatoires en XHTML)
- Attributs de balises
 - une balise peut comporter 0 ou plusieurs paires
 - attribut = "valeur"**
 - <table border="1" align="center">**
 - sans ou avec (préférer) quotes doubles **" "** ou simples **' '**

La genèse

2011: HTML 5.0

- 1989 - 1992 : CERN
 - SGML et Hyperliens - HTML 1.0
 - Tim Berners-Lee et Dan Connally
- 1993 : MOSAIC
 - img et form - HTML 1.0
- 1994 : NETCAPE
 - CSS et DOM
- 1995-96 : Web
 - 1995-96: HTML 2.0
 - **1997: HTML 3.2. et 4.0**
- 2000-2006: XHTML
 - **abandon de XHTML 2.0**



L'impossibilité pour le W3C de trouver un consensus entre les éditeurs de navigateurs et les créateurs de moteurs de recherche pour faire évoluer **XHTML** a conduit un groupe indépendant, le **WHATWG** (*Web Hypertext Application Technology Working Group*) dirigé par Ian Hickson (aujourd'hui chez Google !), à entamer le développement de HTML 5

Structure d'une page HTML

```
<!DOCTYPE html>
```

```
<html>
```

- Une entête

```
<head>
```

```
...
```

```
<title>Images de mathématiques...</title>
```

```
...
```

```
</head>
```

- Un corps

```
<body>
```

```
...
```

```
<p>DiS MAMAN (.....</p>
```

```
...
```

```
</body>
```

```
</html>
```



Les balises

Les balises peuvent être de **deux natures différentes**

Les balises « block » : Elles englobent le contenu. Elles contiennent des paragraphes, des titres, etc.

HTML

```
< p > Ceci est un paragraphe < /p >
```

Les balises « inline » : Elles sont **dans le texte**.
(mettre en gras, souligner, etc...)

HTML

```
< p > On va écrire un texte <strong> en gras </strong> !!!</p>
```

L'entête

`<html>`

`<head>`

`</head>`

Placé entre les balises
<head> et termine par
</head>

de la page
permettent de mettre des
options

Cette balise permet de dire que le
contenu est écrit avec la **norme iso-
8859-1**
(avec les accents é-à-è-ë-...)

`<body>`

...
`<p>Bonjour !!!</p>`

`</body>`

`</html>`

Le corps

```
<!DOCTYPE html>  
<html>
```

```
<head>
```

```
</head>
```

```
<body>
```

```
<!-- ceci est un commentaire -->
```

```
</body>
```

```
</html>
```

Placé entre les balises
<body> et termine par
</body>

**C'est là que va figurer
tout le contenu de la page**

Script PHP dans une page HTML

```
<?php ①
// Placez ici du code PHP
?>
<!DOCTYPE html>
<html lang="fr">
<head>
<meta http-equiv="Content-type" content="text/html; charset=UTF-8" />
<title> HTML 5 et CSS 3 </title>
</head>
<body>
<!-- Le corps de la page -->
<h1>Le corps de la page</h1>
<?php ②
echo "<h2> Placez ici du code PHP créant un titre</h2>";
?>
</body>
</html>
```

Script JavaScript et balise style dans HTML

```
<!DOCTYPE html>
<html lang="fr">

<head>
  <title>Titre de la page</title>
  <meta http-equiv="Content-type" content="text/html; charset=UTF-8" />
  <link rel="shortcut icon" type="images/x-icon" href="../images/favicon.ico" />
  <meta name="Author" content="Jean ENGELS" />
  <script type="text/javascript">
    <!-- Scripts JavaScript -->
  </script>
  <style type="text/css">
    <!-- Styles CSS -->
  </style>
</head>
<body>
  <!-- Le corps de la page -->
  <h1>Le corps de la page</h1>

</body>
```

Les types de blocs

```
<h3>Titre</h3><p>ceci est le premier paragraphe. <em>Cette phrase peut être très importante</em>Mais <strong>celle-ci l'est plus encore</strong></p><h3>Autre titre</h3><h4>titre (encore)</h4><h4>titre (toujours)</h4>
```

Titre

Ceci est le premier paragraphe

Cette phrase peut être très importante

Mais

Celle-ci l'est plus encore

Autre titre

Titre (encore)

Titre (toujours)

Deux types d'éléments

- Les éléments « bloc »

- Les éléments blocs s'empilent les uns sur les autres
- Un élément bloc peut contenir d'autres éléments blocs
- Un élément bloc peut contenir des éléments en ligne
- *exemples* : p, h1, h2, h3, h4, h5, h6, div

- Les éléments « en ligne »

- Les éléments en ligne se placent les uns à côté des autres
- Un élément en ligne peut contenir d'autres éléments en ligne
- Un élément en ligne ne peut pas contenir d'éléments bloc.
- *exemples* : em, strong, b, i, span

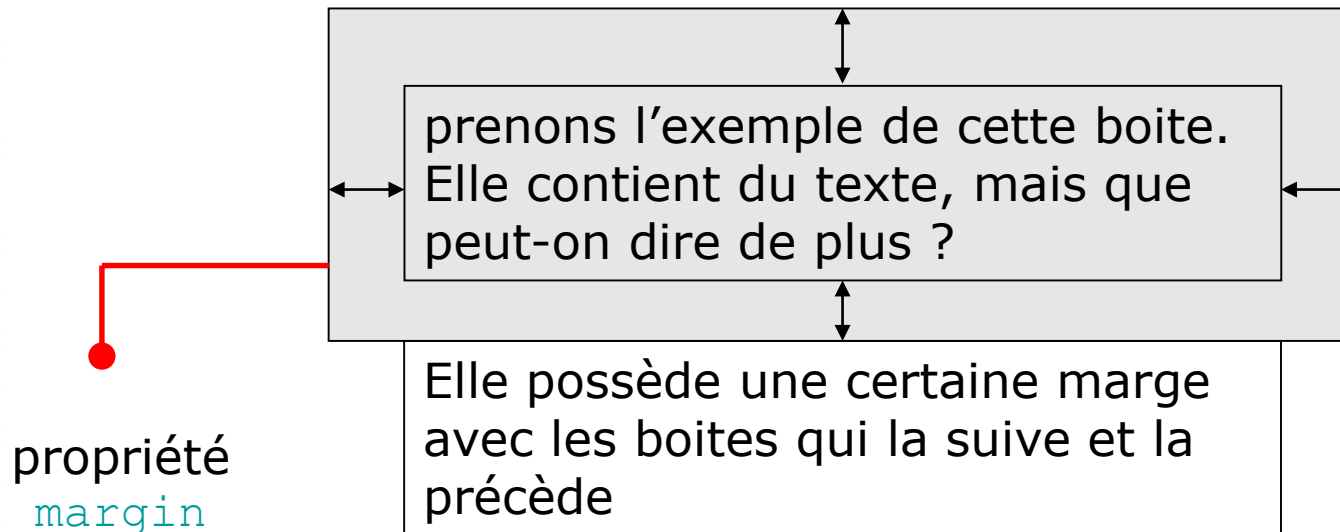
DIV, SPAN

- Containers génériques.
- `<div>` permet de structurer le document
- `` permet de changer la mise en page de quelques mots dans un texte
- `<div>` est très utilisé pour la mise en page

les propriétés des boites

prenons l'exemple de cette boite.
Elle contient du texte, mais que
peut-on dire de plus ?

les propriétés des boîtes



les propriétés des boîtes

propriété

border



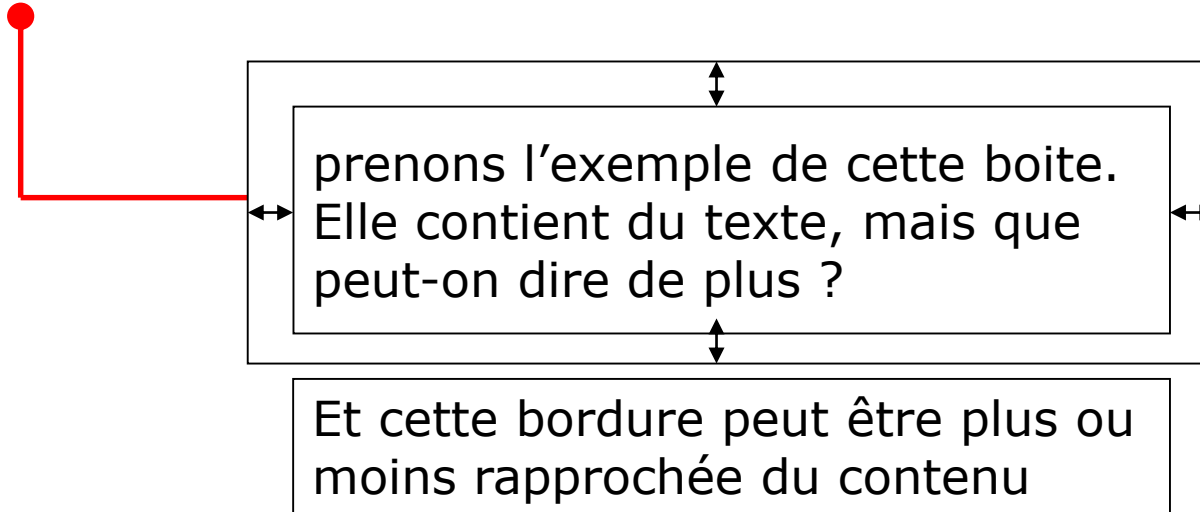
prenons l'exemple de cette boîte.
Elle contient du texte, mais que
peut-on dire de plus ?

Elle possède une bordure blanche,
mais qui aurait aussi bien pu être
verte, en pointillé ou invisible

les propriétés des boîtes

propriété

`padding`



Les paragraphes

Block

```
<body>
```

```
<p>
```

Thomas Andrews (7 février 1873 - 15 avril 1912) est un architecte naval britannique qui a travaillé à la construction de plusieurs paquebots. Il grandit au sein d'une famille aisée, puis devient apprenti dans les chantiers navals Harland & Wolff de Belfast dont le président, Lord Pirrie, est son oncle. Pendant quelques années, Andrews gravit progressivement les échelons avant d'être nommé directeur général de l'entreprise. La réparation du paquebot de la White Star Line Suevic, à laquelle il participe, fait de lui un des architectes navals les plus renommés de son époque.

```
</p>
```

```
</body>
```

Par défaut, les éléments de type **bloc** (Block) sont affichés par le navigateur avec un saut de ligne au début et à la fin.

Les paragraphes

`
` : Retour à la ligne

Inline

Les éléments de type **inline** se placent normalement l'un à côté de l'autre (pas de saut de ligne).

HTML

```
<body>
```

```
    <p>
```

Ligne 1 `
`

ligne 2

`
` Ligne 3

toujours ligne 3

```
    </p>
```

```
</body>
```

Les titres

`<h1> ... </h1>` Titre TRES important
`<h2> ... </h2>` Titre important
`<h3> ... </h3>` Titre moyen
`<h4> ... </h4>` Titre pas important
`<h5> ... </h5>` Titre pas du tout important

Block

```
<body>
  <h1> Chapitre 1</h1>
    <h2> Section 1 </h2>
      <p> Bla bla bla </p>

    <h2> Section 2 </h2>
      <p> camion </p>

  <h1> Chapitre 2 </h1>
    <p> Pouet pouet </p>
</body>
```

Les images

Inline

```
<img ... />
```

```
<body>
```

```
<p>
```

Je suis content !!

```

```

```
</p>
```

```
</body>
```

On peut par exemple mettre le titre de l'image...

enregistre la page) Ex : **image/photo_identite.jpg**

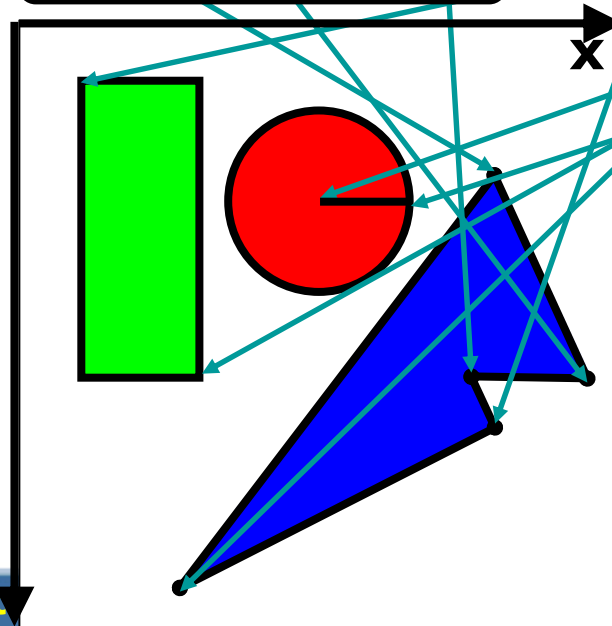
- Soit un **chemin absolue** de l'image (commençant par

La balise **img** est de type *inline*, c'est-à-dire qu'elle doit se trouver à l'intérieure de balises *block* !

Elle est considérée comme une lettre !

Les images cliquables

```
  
<map name="carte">  
  <area href="..." shape="poly"  
    coords="200,65,240,150,190,150,200,170,65,240">  
  <area href="..." shape="rect" coords="25,25,75,150">  
  <area href="..." shape="circle" coords="125,75,40">  
</map>
```



Les liens

Inline

`<a>...` : *Faire un lien vers une image ou un fichier.*

`texte à cliquer`

HTML

`<body>`

`<p>`

Pour plus d'informations,

``

cliquez ici !

``

`</p>`

`</body>`

Lien vers un mail

`Ecrivez-moi !`

Mise en évidence du texte

`...` : *important (en gras).*

`...` : *Mise en évidence (italic).*

Inline

Inline

HTML

```
<body>
```

```
  <p>
```

Ceci est une **chose importante**,

Ceci est une *mise en évidence*.

```
  </p>
```

```
</body>
```


Les listes

`...` : Début et fin d'une liste

Block

`...` : Début et fin d'un des points de la liste

Block

HTML

```
<body>
```

```
  <p>Pour être un bon thésard, il faut : </a>
```

```
  <ul>
```

```
    <li>aimer les ordinateurs</li>
```

```
    <li>être poète (surtout la 3ème année)</li>
```

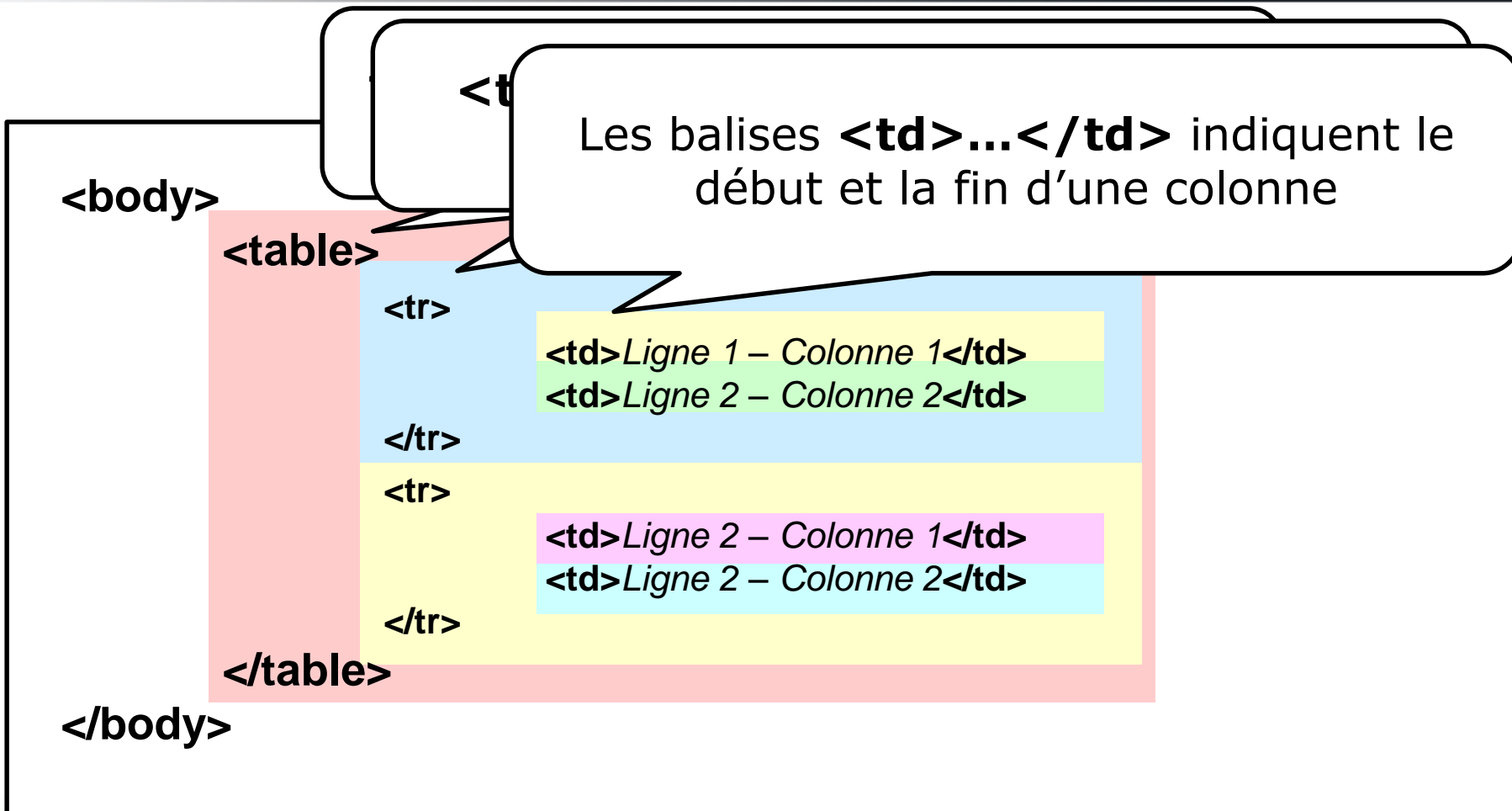
```
    <li>aimer les voyages</li>
```

```
    <li>...</li>
```

```
  </ul>
```

```
</body>
```

Les tableaux

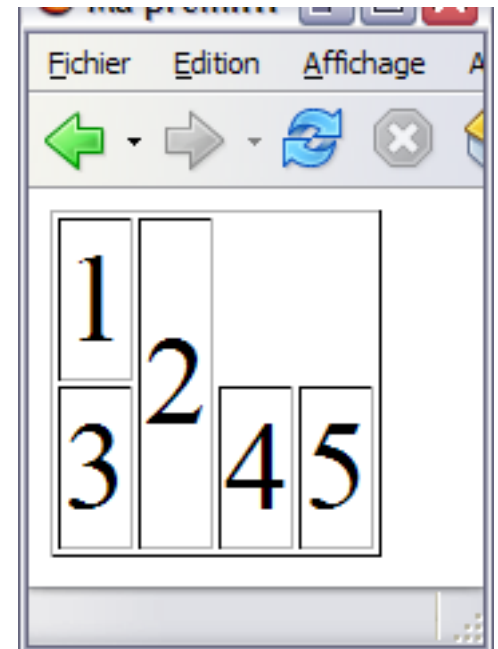


Les tableaux « avancés »

- Extension de cellule sur L lignes :
 - `<td rowspan="L">`

```
<!DOCTYPE html>

<HTML>
  <HEAD>
    <meta charset="utf-8" />
    <TITLE>Essai de script cgi</TITLE>
  </HEAD>
  <body bgcolor="white">
    <table border="2">
      <tr> <td>1</td> <td rowspan="2">2</td> </tr>
      <tr> <td>3</td> <td>4</td><td>5</td> </tr>
    </table>
  </body>
</HTML>
```



Les balises à tout faire

`<div>...</div>` : Balise de type « block ».

Block

`<body>`

`<div>`

*Les chaussettes de l'achi-duchesse
sont-elles sèches, archi sèches ?*

`</div>`

`</body>`

`...` : Balise de type « inline ».

Inline

HTML

`<body>`

`<p>`

*Les chaussettes de l'achi-duchesse
sont-elles sèches, `` archi sèches `` ?*

`</p>`

`</body>`


Ces balises ne servent à priori à rien.

En fait, elles serviront plus tard, lorsqu'on voudra les mettre en forme (souligner, mettre en rouge, etc...)

Les ancres

- Une **ancre** est une sorte de point de repère

```
<body bgcolor="white">
  <h1>Ma grande page</h1>
  <p>
    Aller directement à la partie traitant de :<br />
    <a href="#cuisine">La cuisine</a><br />
    <a href="#rollers">Les rollers</a><br />
    <a href="#arc">Le tir à l'arc</a><br />
  </p>
  <h2 id="cuisine">La cuisine</h2>
  <p>... (beaucoup de texte) ...</p>
  <h2 id="rollers">Les rollers</h2>
  <p>... (beaucoup de texte) ...</p>
  <h2 id="arc">Le tir à l'arc</h2>
  <p>... (beaucoup de texte) ...</p>
</body>
```



Les formulaires

- La présence d'éléments de formulaires HTML fait distinguer une application web d'un site web.
- Un formulaire HTML est une partie de page web qui accepte des entrées de l'utilisateur.

Name:

Food:

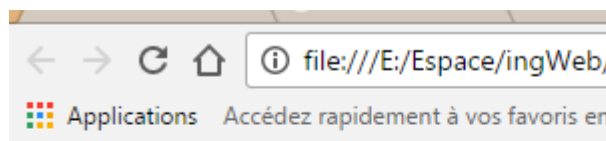
Meat? ☐

Les formulaires

```
<!DOCTYPE html>
<HTML>
  <HEAD>
    <TITLE>RADIO-2</TITLE>
  </HEAD>
  <BODY BGCOLOR="white" TEXT="black">
    <H3>Formulaire pour "sélection d'un choix"</H3>
    <FORM ACTION="http://localhost/test1.php" METHOD="get">
      <P>Faites votre choix:</P>
      <INPUT TYPE="radio" NAME="Saison" VALUE="1"> Printemps</INPUT>
      <INPUT TYPE="radio" NAME="Saison" VALUE="2" CHECKED> Eté</INPUT>
      <INPUT TYPE="radio" NAME="Saison" VALUE="3"> Automne</INPUT>
      <INPUT TYPE="radio" NAME="Saison" VALUE="4"> Hiver</INPUT>
    </FORM>
  </BODY>
</HTML>
```

L'attribut **action** de la balise **form** indique l'adresse url où seront envoyées les données saisies

L'attribut **method** de la balise **form** indique sous quelle forme seront envoyées les données
GET: codées dans le URL
POST: dans le corps de la requête



Formulaire pour "sélection d'un choix"

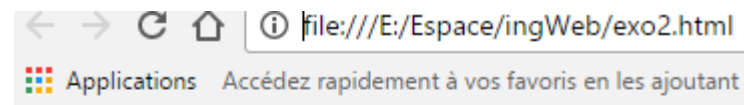
Faites votre choix: ☐ Printemps ☒ Eté ☐ Automne ☐ Hiver

Les formulaires

```
<HTML>
<HEAD>
<TITLE>INPUT -1</TITLE>
</HEAD>
<BODY BGCOLOR="white" TEXT="black">

<H3>Formulaire pour entrer le nom
<FORM ACTION="input_2.htm"
<P>Prénom:
<INPUT NAME="prénom" TYPE="text" SIZE="25" MAXLENGTH="40">
<P>Nom de famille:
<INPUT NAME="nom" TYPE="text" SIZE="25" MAXLENGTH="40">
</FORM>
</BODY>
</HTML>
```

L'attribut **Type** de la balise INPUT



Formulaire pour entrer le nom

Prénom:

Nom de famille:

Les formulaires

```
<!DOCTYPE html>
<HTML>
  <HEAD>
    <TITLE>Submit</TITLE>
  </HEAD>
  <BODY >
    <H3>Formulaire pour "LISTE déroulante"</H3>
    <FORM action="test.php" method="post">
      Faites votre choix:<BR/>
      <select name = "liste" size = 3>
        <option > Element1 </option>
        <option> Element2 </option>
        <option> Element3 </option>
        <option> Element4 </option>
        <option selected> Element5 </option>
        <option> Element6 </option>
      </select>
      <BR/>
      <BR/>
      <BR/>
      <select name = "liste2" size = 1>
        <option> Element1 </option>
        <option> Element2 </option>
        <option> Element3 </option>
        <option> Element4 </option>
        <option selected> Element5 </option>
        <option> Element6 </option>
      </select>
      <BR/> <BR/><BR/><BR/>
      <input type="submit" value="soumettre"/>
    </FORM>
  </BODY>
</HTML>
```

Applications Accédez rapidement à vos favoris en les a

Formulaire pour "LISTE déroulante"

Faites votre choix:

Element3 ▲
Element4 ▬
Element5 ▼

Element5 ▼

soumettre

Classe, identification et description

- Affecter une balise à une classe
 - Attribut **class**
 - `<p class="ma_classe">`
- Donner un identifiant à une balise
 - Attribut **id**
 - Identifiant doit être unique (charge du concepteur)
 - `<p id="mon_id_unique">`
- Décrire un élément
 - Attribut **title** pour la plupart des éléments HTML
 - Texte affiché sous forme d'info-bulle (tooltip)

Sous-groupe de l'ensemble des balises `<p>`

Une balise `<p>` repérée de façon unique

Utilisation et utilité des balises dans `<body>`

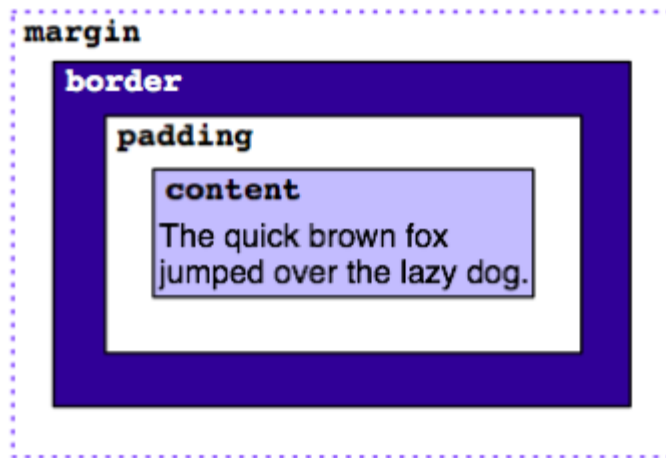
- Initialement : mise en page du document
 - Police, couleur, taille, alignement du texte, ...
- Maintenant : structuration logique du document
 - Titres de différents niveaux
 - Paragraphes, citation, exemple, code, ...
 - Listes
 - Tableaux, ...
- Si le document est correctement structuré, chaque élément structurel pourra est mis en forme via une feuille de style CSS



Le modèle des boîtes (Box Model)

- D'après le "box model", lorsqu'on attribue une taille à un élément de type block (à l'aide d'un width ou d'un height), les marges viennent s'ajouter à cette taille.
- Le modèle de boîte CSS est une collection de propriétés CSS qui s'appliquent à tous les éléments HTML visuels.

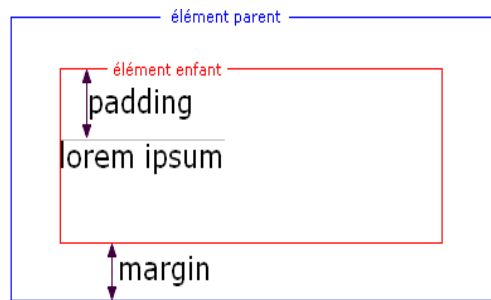
<p>The quick brown fox jumped over the lazy dog. </p>



Le contenu est niché à l'intérieur de 3 boîtes : La marge englobe la bordure, qui entoure le rembourrage, qui enferme le contenu.

Le modèle des boîtes (Box Model)

- Pour chaque élément html on peut donc définir l'espacement qui le séparera des autres éléments (margin) et les espacements intérieurs dont il peut bénéficier (padding).



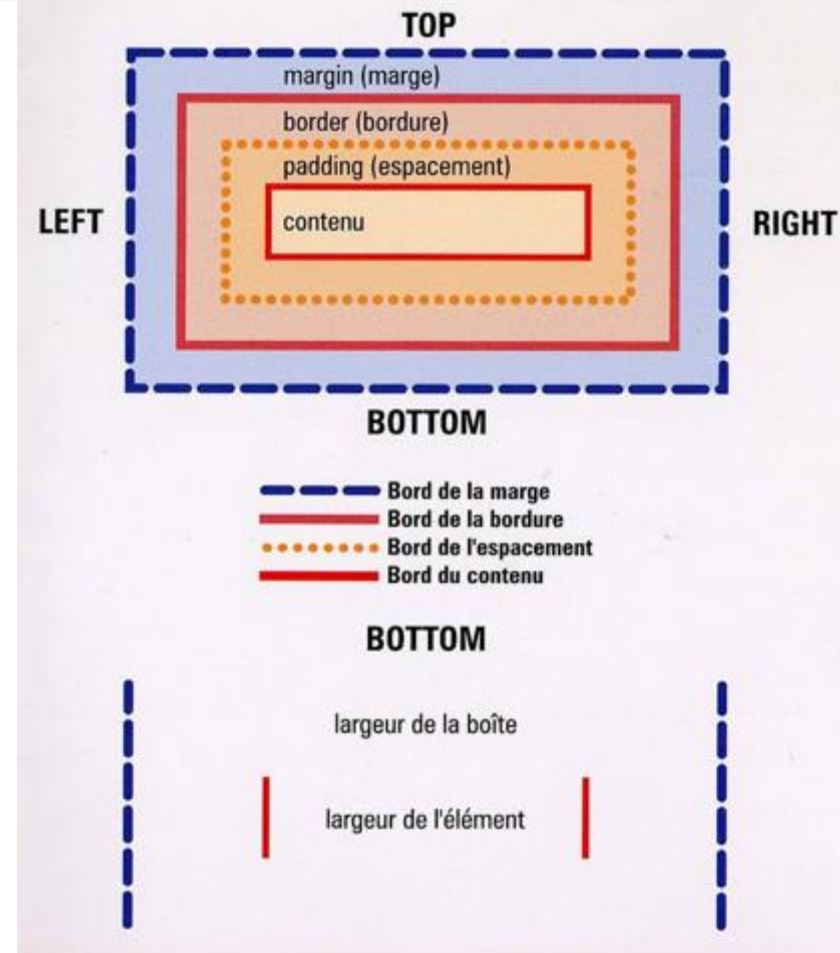
```
<blockquote>  
<p>
```

 Lorem ipsum dolor sit amet,
 consectetur adipiscing elit. Duis eget ligula quis libero
 mollis dapibus.

 Suspendisse potenti. Nullam facilisis neque et sem.
 Proin placerat adipiscing urna. Aenean sollicitudin.

```
</p>  
</blockquote>
```

blockquote est l'**élément parent**,
p est l'**élément enfant**.



Feuilles de style, le rôle

- Le World Wide Web a introduit les feuilles de styles en cascade pour compléter le langage HTML.
 - Les CSS permettent de gérer l'apparence des documents.
 - Les feuilles indiquent aux balises HTML leur comportement ou style.
 - Un style rassemble tous les attributs que l'on peut appliquer à des types de textes similaires.
 - Attribut : taille,format
 - Textes similaires : titres, en-têtes, pied de page
 - Les styles donnent un nom commun à des groupes d'attributs

CSS comment faire?

- 2 façons de faire :
 - Soit balise par balise
 - attribut `style="..."` dans la balise d'un élément
 - Soit toutes les balises d'un coup !
- 2 façons d'ajouter du code CSS
 - Directement dans le document HTML
 - liste des règles CSS dans balise `<style...></style>`
 - placée dans la partie `<head>`
 - Dans un fichier séparé, qui sera inclus au document HTML
 - appel à ce fichier externe avec une balise `<link ... />` ou,
 - appel à ce fichier externe avec `@import`

CSS balise par balise

Le CSS se met à l'intérieur de la balise ouvrante, en ajoutant l'**attribut** *style*.

HTML

```
<p> Mon texte </p>
```



HTML

```
<p style="..." > Mon texte </p>
```

Les propriétés CSS se mettent à la place des
« ... »

CSS balise par balise

Les propriétés CSS se mettent toujours sous la forme :

Propriété : valeurs ;



(n'oubliez pas le « **:** » au milieu, et surtout le « **;** » à la fin !)

CSS balise par balise

Exemple :

HTML

```
<p style="color : red; " > Mon texte </p>
```

Si vous voulez appliquer plusieurs propriétés CSS il suffit de les mettre les unes à la suite des autres (sans oublier les « ; » entre chaque !)

HTML

```
<p style="color : red; margin : 0px ; " > Mon texte </p>
```

Liste des propriétés CSS usuelles

- **color** : donne la couleur du texte (et des traits).

Valeurs : le nom de la couleur en anglais, ou le code hexadécimal de la couleur précédé d'un « # » (#F4A42E)

HTML

```
<p style=" color : red; " > Mon texte </p>
```

- **font-size** : donne taille du texte.

Valeurs : la taille en pixel (suivie de « px »), ou certains mots clés comme « small », « medium », « large », « x-large »

HTML

```
<p style=" font-size : 50px; " > Mon texte </p>
```

- **text-align** : donne l'alignement du texte.

Valeurs : left, right, center, justify.

HTML

```
<p style=" text-align : center; " > Mon texte </p>
```

Liste des propriétés CSS usuelles

- **font-weight** : épaisseur du trait (gras ou pas gras).

Valeurs : « bold » (gras) ou « normal »

HTML

```
<p>Ceci est <span style=" font-weight : bold; " > Important</span> !</p>
```

- **font-style** : Style de trait (italic)

Valeurs : « italic », « oblique » ou « normal ».

HTML

```
<p style=" font-style : italic; " > Texte italic </p>
```

- **text-decoration** : « décore » le texte.

Valeurs : « underline » (souligné), « line-through » (barré), « overline » (ligne au-dessus), « blink » (clignotant), « none » (normal).

HTML

```
<p style="text-decoration : blink; " > C'EST LES SOLDES !!!!</p>
```

Liste des propriétés CSS usuelles



Mettez des guillemets pour les noms composés

- **font-family** : *donne la police du texte.*

Valeurs : le nom de la police : arial, "Arial Black", Georgia, ...

HTML

```
<p style=" font-family : Arial; " > Mon texte </p>
```



Tous les ordinateurs ne connaissent pas toutes les polices. Pensez à mettre des polices connues ! La plus connue est « Serif ».



Au cas où, il est possible de mettre plusieurs polices par ordre de préférence, séparées par des virgules

HTML

```
<p style=" font-family : Verdana, Arial, Serif; " > Mon texte </p>
```

Liste des propriétés CSS usuelles

- **height** : Définit la hauteur de la balise.
- **width** : Définit la largeur de la balise

Valeurs : la taille en pixel (suivie de « px »), ou pourcentage (suivie de « % ») de la largeur de la balise englobante.

HTML

```
<p style=" width : 200 px; " > Mon texte </p>
```

Le paragraphe fera 200px de large

HTML

```
<body>  
  <p style=" width : 80%; " > Mon texte </p>  
</body>
```

Le paragraphe fera 80% de la largeur de la page (body)



Ces propriétés ne fonctionnent que pour les balise de type « block »

Liste des propriétés CSS usuelles

- **border** : *Définit plein de choses sur les bordures.*

Valeurs : - Le type (none, solid, dotted, ...)
- L'épaisseur (en px)
- la couleur (red, blue, #AA34F2,...)

HTML

```
<p style=" border : solid 2px green; " > Mon texte </p>
```

- **background-color** : *Définit la couleur l'arrière plan.*

Valeurs : La couleur (red, ...) ou « transparent »

HTML

```
<p style=" background-color : #AAAAFF; " > Mon texte </p>
```

- **background** : *Image d'arrière-plan*

Valeurs : - l'emplacement de l'image de la forme url('fichier_image')
- des propriété sur l'affichage (alignement, répétition, ...)

HTML

```
<p style=" background : url('image/fond.jpg') right no-repeat " > Mon texte </p>
```

Le fichier CSS séparé

Si les balises de même types ont toutes la même mise en forme comme ci-dessous...

HTML

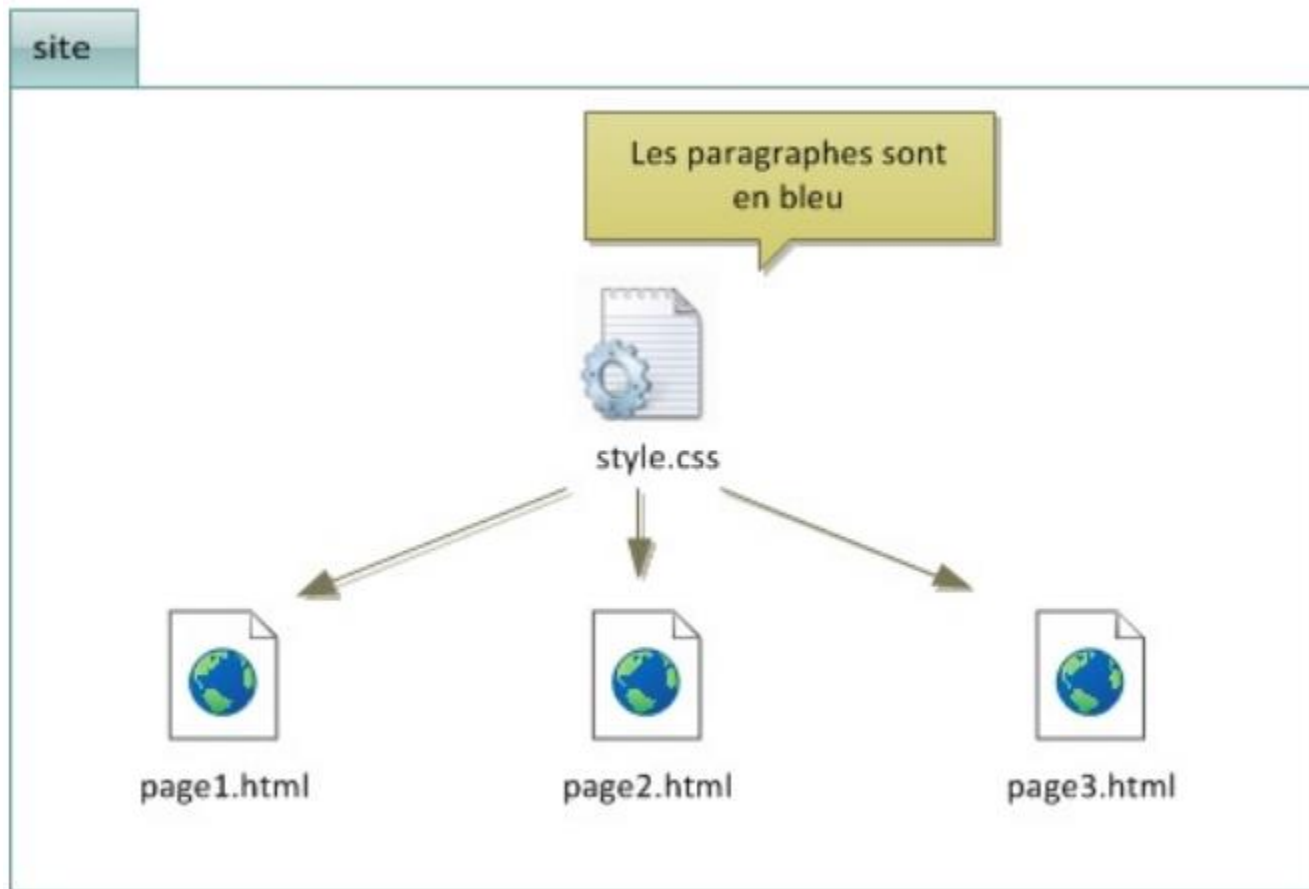
```
<h2 style="text-decoration:underline ; color:red;" > Chapitre I </h2>
<p style="font-style:italic;" >blablabla</p>

<h2 style="text-decoration:underline ; color:red;" > Chapitre II </h2>
<p style="font-style:italic;" >patati patata</p>

<h2 style="text-decoration : underline ; color : red;" > Conclusion</h2>
<p style="font-style:italic;" >gnagnagna</p>
```

... plutôt que de les réécrire à chaque fois pour chaque balise,
**on peut écrire toutes ces propriétés CSS dans un
fichier, une fois pour toute !**

Le fichier CSS



Le fichier CSS

Exemple :

HTML

```
<h2> Chapitre I </h2>
<p>blablabla</p>

<h2> Chapitre II </h2>
<p>patati patata</p>

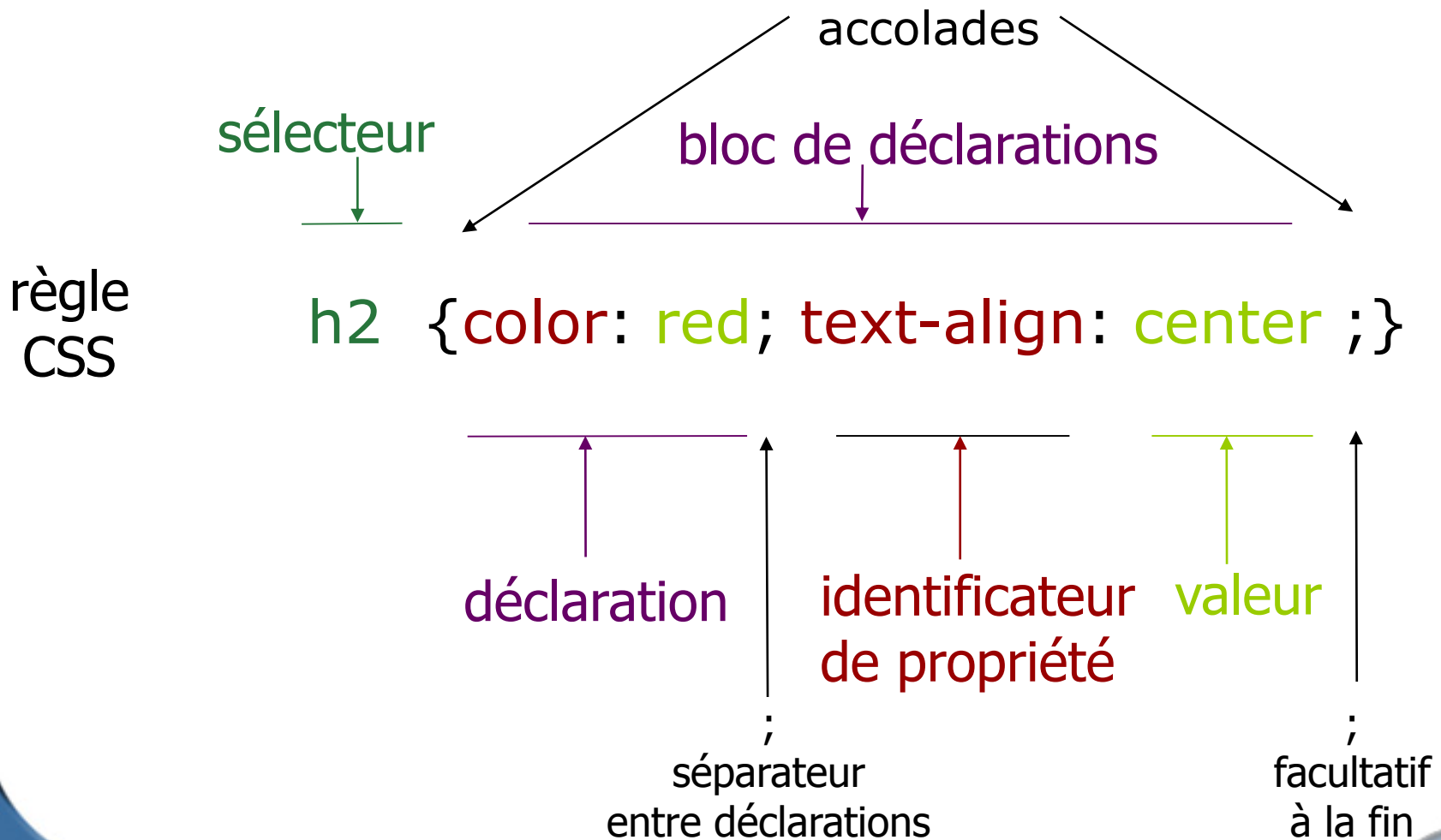
<h2> Conclusion</h2>
<p>gnagnagna</p>
```

CSS

```
h2 {
    text-decoration : underline ;
    color : red ;
}

p {
    font-style : italic ;
}
```

Règle CSS : syntaxe



Le fichier CSS

<html>

<head>

<title>Bienvenue sur mon site !</title>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />

</head>

<body>

<h2> Chapitre I </h2>

<p>blablabla</p>

<h2> Chapitre II </h2>

<p>patati patata</p>

<h2> Conclusion</h2>

<p>gnagnagna</p>

</body>

</html>

Par contre : il faudra dire à la page html d'aller chercher le fichier CSS

→ **On l'indique dans l'entête de la page**

Hierarchie des balises

On peut restreindre la portée des propriétés CSS à **certaines balises contenues dans d'autres balises**.

Pour cela, on met les balises dans **l'ordre hiérarchique, séparées par des espaces**.

CSS

```
h1 a  
{text-decoration : none ;  
color : black ;}
```

Signifie : les balises **liens** (« **a** ») qui se trouvent dans les **titres 1** (« **h1** ») doivent être **noir** et **non-soulignés**.

Qu'est-ce que l'héritage ?

- L'héritage est une notion centrale et fondamentale en CSS.
 - La notion d'héritage signifie que tout élément HTML enfant va hériter, « en cascades », des styles de ses parents.
 - C'est par ailleurs de là que vient le nom du CSS : Cascading StyleSheets, ou Feuilles de Style en Cascades.
 - Par exemple, tous les éléments à l'intérieur de l'élément body sont des enfants de cet élément. Si l'on applique un style à l'élément body, ses enfants en hériteront automatiquement.

Qu'est-ce que l'héritage ?

```
<!DOCTYPE html>
<html>
  <head>
    <title>L'héritage en CSS</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

  <body>
    <h1>Un titre de niveau 1</h1>
    <p>Un paragraphe avec du <strong>texte important</strong></p>
    <p>Un autre paragraphe</p>
  </body>
</html>
```

```
body{
  color: purple;
}
```

Un titre de niveau 1

Un premier paragraphe

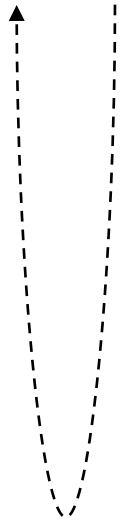
Un deuxième paragraphe

Un troisième paragraphe

En CSS, si il y a un conflit, c'est-à-dire si un élément reçoit plusieurs fois une même propriété avec des valeurs différentes, le style qui va être prioritaire va être le style le plus proche de l'élément.

héritage « sous classe » indépendante

- `.nav li{color:red;}`



li sera enfant
de la classe .nav

```
<ul class="nav">  
  <li>nom</li>  
  <li>prenom</li>  
  <li>age</li>  
</ul>
```

Cela va affecter tous les
éléments li dans la classe .nav

héritage « sous classe » dépendante

- `li.nav{color:red;}`



```
<ul>  
  <li class="nav">nom</li>  
  <li class="nav">prenom</li>  
  <li class="nav">age</li>  
</ul>
```

Sélecteur contextuel

A B : une balise contenue dans une autre

Code : CSS

```
h3 em  
{ }
```

A + B : une balise qui en suit une autre

Code : CSS

```
h3 + p  
{ }
```

A[attribut] : une balise qui possède un attribut

Code : CSS

```
a[title]  
{ }
```

A[attribut="Valeur"] : une balise, un attribut et une valeur exacte

Code : CSS

```
a[title="Cliquez ici"]  
{ }
```

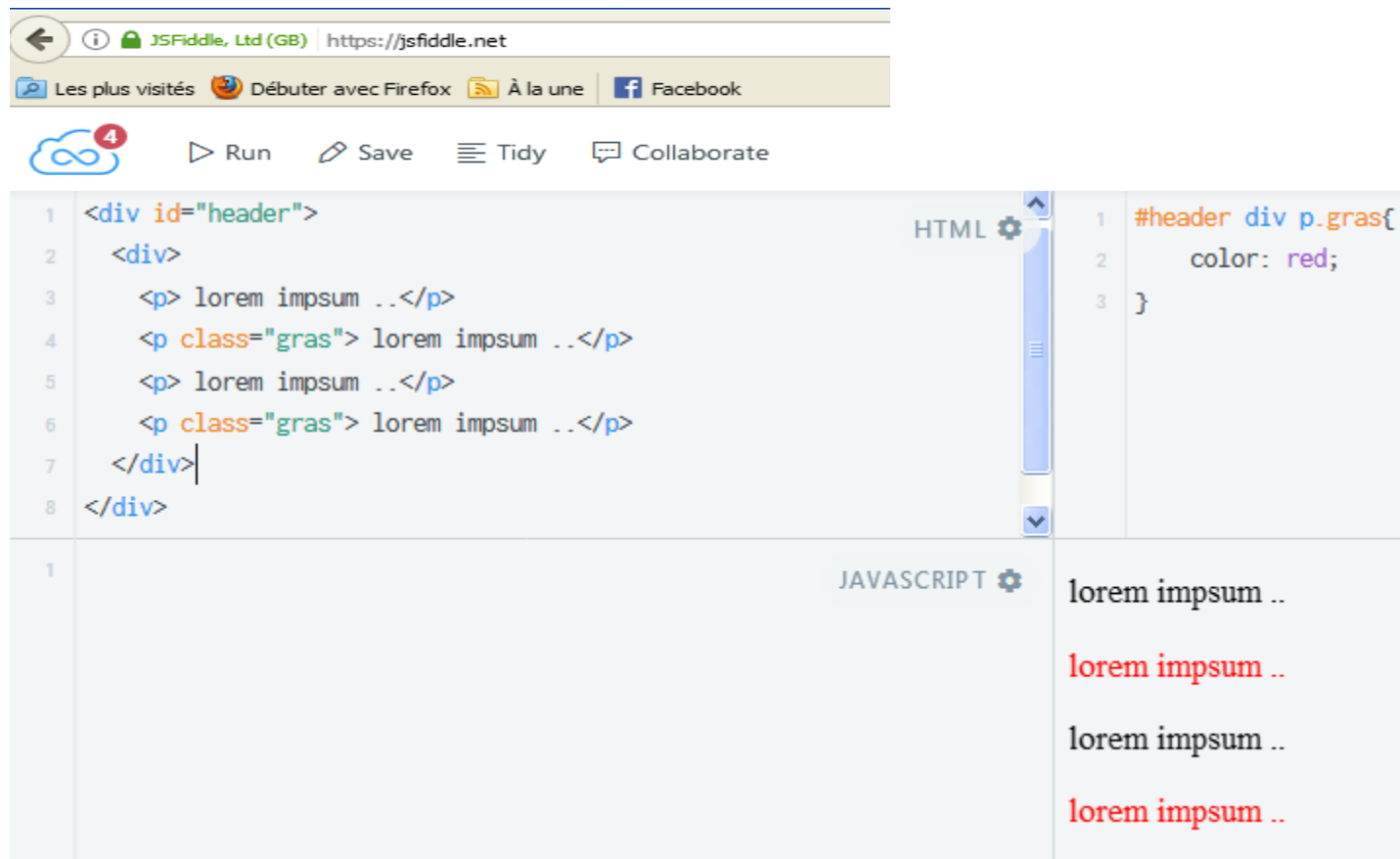
A > B : B fils de A

Code : CSS

```
h3 em  
{ }
```

Le sélecteur *contextuel* permet de cibler les éléments de la page à partir d'un motif de recherche.

Sélecteur contextuel



The screenshot shows the JSFiddle web application interface. The browser's address bar displays 'https://jsfiddle.net'. Below the browser window, there are buttons for 'Run', 'Save', 'Tidy', and 'Collaborate'. The main workspace is divided into two panels: 'HTML' and 'JAVASCRIPT'. The 'HTML' panel contains the following code:

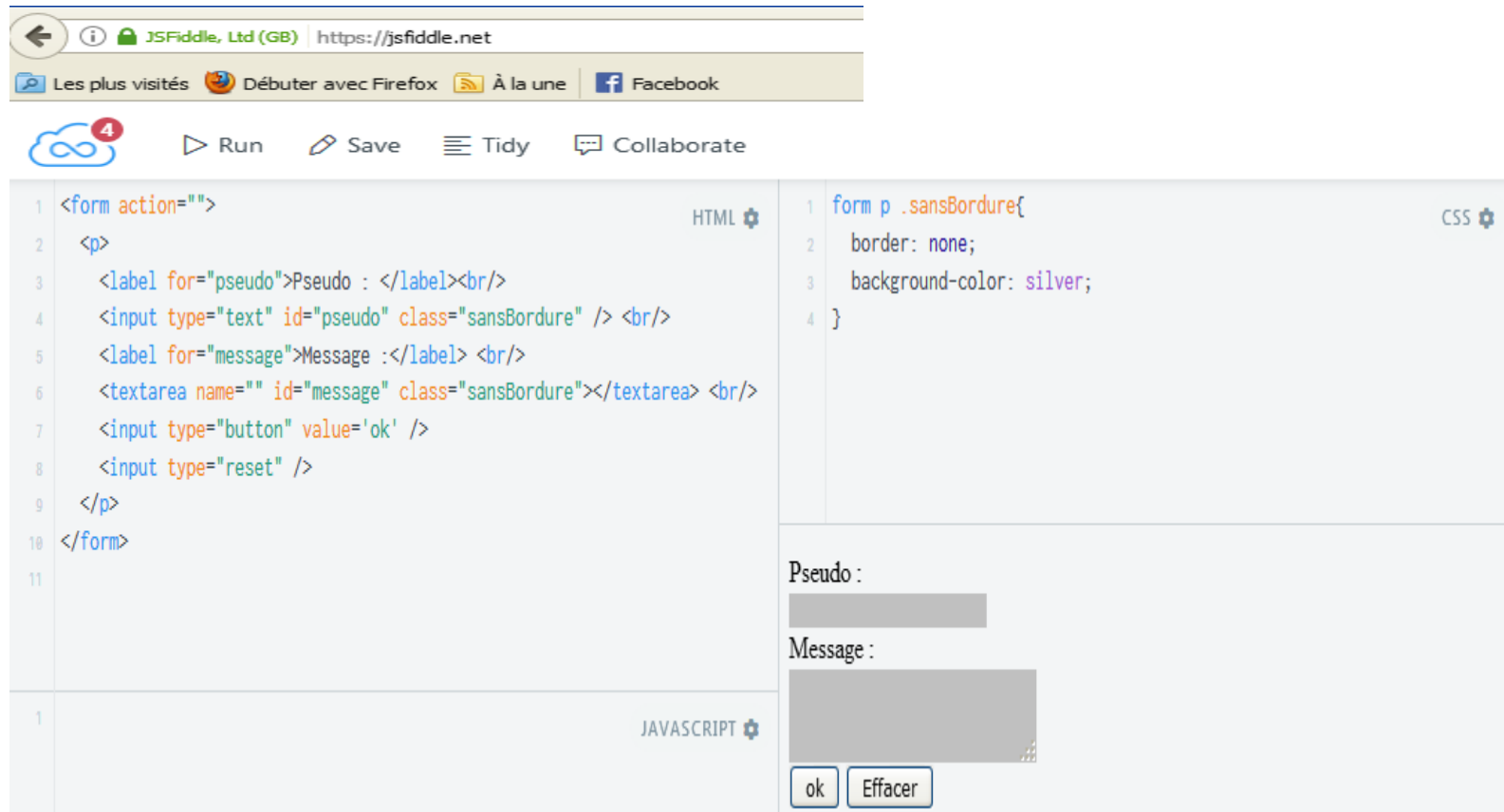
```
1 <div id="header">
2   <div>
3     <p> lorem ipsum ..</p>
4     <p class="gras"> lorem ipsum ..</p>
5     <p> lorem ipsum ..</p>
6     <p class="gras"> lorem ipsum ..</p>
7   </div>
8 </div>
```

The 'JAVASCRIPT' panel is currently empty. On the right side, the CSS panel shows a context selector rule:

```
1 #header div p.gras{
2   color: red;
3 }
```

Below the CSS panel, the rendered output is shown. It consists of four lines of text: 'lorem ipsum ..', 'lorem ipsum ..', 'lorem ipsum ..', and 'lorem ipsum ..'. The second and fourth lines are highlighted in red, demonstrating the effect of the CSS selector.

Sélecteur contextuel



The screenshot shows the JSFiddle web application interface. The top bar includes the JSFiddle logo, a back button, and the URL <https://jsfiddle.net>. Below the bar are links for 'Les plus visités', 'Débuter avec Firefox', 'À la une', and 'Facebook'. The main interface has a top navigation bar with 'Run', 'Save', 'Tidy', and 'Collaborate' buttons. The central area is divided into three panels: HTML, CSS, and a live preview.

HTML Panel:

```
1 <form action="">
2   <p>
3     <label for="pseudo">Pseudo : </label><br/>
4     <input type="text" id="pseudo" class="sansBordure" /> <br/>
5     <label for="message">Message :</label> <br/>
6     <textarea name="" id="message" class="sansBordure"></textarea> <br/>
7     <input type="button" value='ok' />
8     <input type="reset" />
9   </p>
10 </form>
11
```

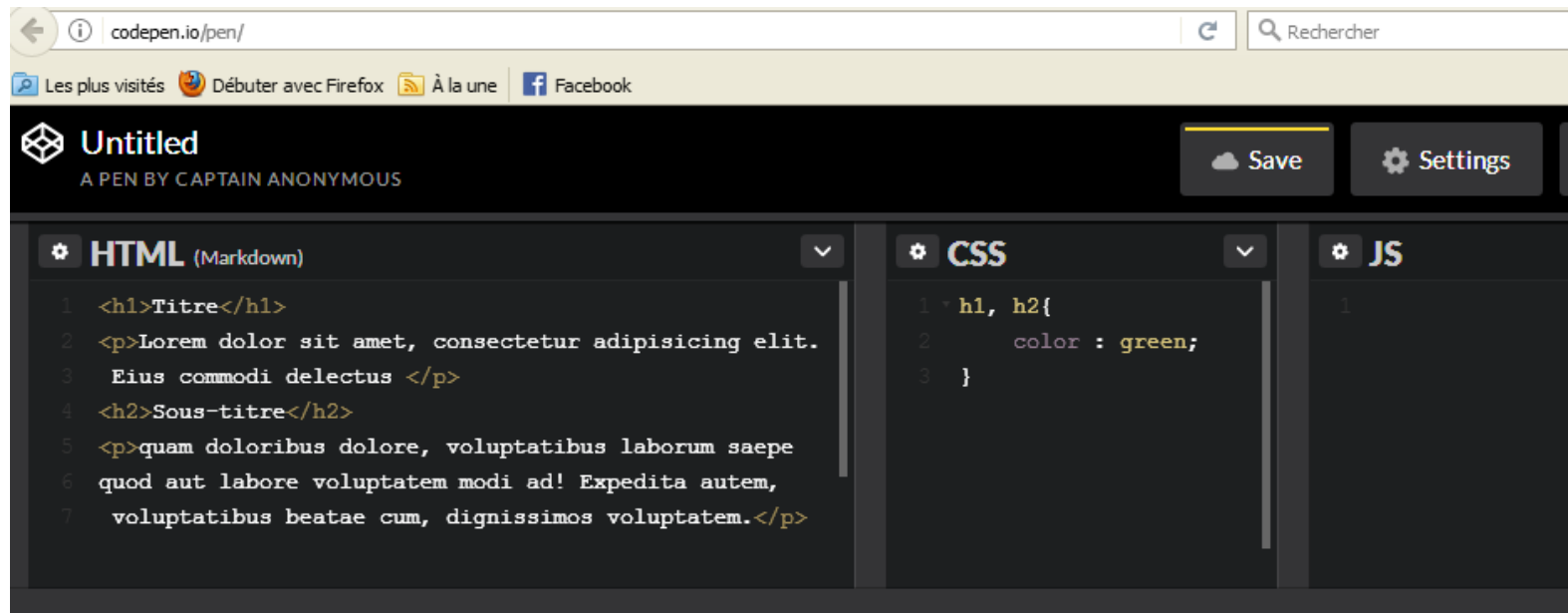
CSS Panel:

```
1 form p .sansBordure{
2   border: none;
3   background-color: silver;
4 }
```

Live Preview:

The preview shows the rendered HTML. It features a form with two labels: 'Pseudo :' and 'Message :'. The 'Pseudo :' label is followed by a text input field. The 'Message :' label is followed by a text area. Both the text input and the text area have a silver background color, demonstrating the effect of the CSS context selector `form p .sansBordure`. Below the text area are two buttons: 'ok' and 'Effacer'.

Sélecteur de regroupement



The screenshot shows a web browser window with the address bar at `codepen.io/pen/`. Below the browser, there's a CodePen editor interface. The editor has a dark theme and three tabs: HTML (Markdown), CSS, and JS. The HTML tab is active, showing the following code:

```
1 <h1>Titre</h1>
2 <p>Lorem dolor sit amet, consectetur adipisicing elit.
3   Eius commodi delectus </p>
4 <h2>Sous-titre</h2>
5 <p>quam doloribus dolore, voluptatibus laborum saepe
6   quod aut labore voluptatem modi ad! Expedita autem,
7   voluptatibus beatae cum, dignissimos voluptatem.</p>
```

The CSS tab is also visible, showing the following code:

```
1 h1, h2{
2   color : green;
3 }
```

Titre

Lorem dolor sit amet, consectetur adipisicing elit. Eius commodi delectus

Sous-titre

quam doloribus dolore, voluptatibus laborum saepe quod aut labore voluptatem modi ad! Expedita autem, voluptatibus beatae cum, dignissimos voluptatem.

Sélecteur d'attribut

The screenshot shows the JSFiddle web application interface. The browser address bar displays 'https://jsfiddle.net'. The main toolbar includes icons for 'Run', 'Save', 'Tidy', and 'Collaborate'. The left sidebar contains 'Fiddle Meta' (Untitled fiddle, No description), 'Add title to make the fiddle public.', 'External Resources', and 'AJAX Requests'. The main workspace is divided into three panels: HTML, CSS, and a preview. The HTML panel contains the following code:

```
1 <label>Pseudo</label>
2 <input type="text">
3 <input value="ok" type="button">
4
```

The CSS panel contains the following code:

```
1 input[type=button]{
2     background-color: orange;
3 }
```

The preview panel shows the rendered HTML with the label 'Pseudo' and two input fields. The button input field is highlighted in orange, demonstrating the effect of the CSS selector.

Sélecteurs, la priorité

1	<code><p style="color:blue"> lorem ipsum </p></code>	HTML ⚙	1	<code>p{</code>	CSS ⚙
2			2	<code>color: red ;</code>	
			3	<code>}</code>	
1		JAVASCRIPT ⚙		<code>lorem ipsum</code>	

<code><p style="color:blue"> lorem ipsum </p></code>	HTML ⚙	1	<code>p{</code>	
		2	<code>color: red !important;</code>	
		3	<code>}</code>	
	JAVASCRIPT ⚙		<code>lorem ipsum</code>	CSS ⚙

Sélecteur d'enfants adjacents

<pre><h1> Titre </h1> <p> Lorem ipsum dolor ..</p> <p> Lorem ipsum dolor..</p></pre>	HTML ⚙	<pre>1 h1+p{ 2 color: magenta; 3 }</pre>
JAVASCRIPT ⚙		<p>Titre</p> <p> Lorem ipsum dolor ..</p> <p> Lorem ipsum dolor..</p>

<pre><p> Lorem ipsum dolor ..</p> <h1> Titre </h1> <p> Lorem ipsum dolor ..</p> <p> Lorem ipsum dolor ..</p></pre>	HTML ⚙	<pre>1 h1~p{ 2 color: red; 3 }</pre>
JAVASCRIPT ⚙		<p> Lorem ipsum dolor ..</p> <p>Titre</p> <p> Lorem ipsum dolor ..</p> <p> Lorem ipsum dolor ..</p>

Sélecteur d'id

```
<div id="Titre"> Grand Titre  
  <div id="sousTitre">sous titre avec l'attribut "id" </div>  
</div>
```

HTML ⚙️

```
1 div#Titre {  
2   position: absolute;  
3   width: 50%;  
4   left: 100px;  
5   top: 40px;  
6   background-color: #CCFFFF;  
7   border: 15px solid #00FF00;  
8   text-align: center;  
9   color: red;  
10  text-transform: uppercase;  
11  font-size: xx-large;  
12  font-weight: bold;  
13 }  
14 div#sousTitre {  
15   text-transform: capitalize;  
16   font-size: x-large;  
17   font-weight: normal;  
18 }  
19
```

JAVASCRIPT ⚙️

GRAND TITRE
Sous Titre Avec L'attribut "Id"

Sélecteur universel

HTML	CSS
<pre>1 <h1>Sélecteurs CSS avancés</h1> 2 <p>Un paragraphe</p> 3 <p>Un paragraphe important</p> 4</pre>	<pre>1 *{ 2 border: 2px solid red; 3 padding: 5px; 4 margin: 5px; 5 }</pre>
JAVASCRIPT	
	

Sélecteur d'enfants

```
<!DOCTYPE html>
<html>
  <head>
    <title>Sélecteurs avancés</title>
    <meta charset= "utf-8">
  </head>

  <body>
    <div class="page-container">
      <h1>Sélecteurs CSS avancés</h1>

      <div class="test">
        <p>Un premier paragraphe dans un div</p>
        <p>Un <strong>paragraphe important</strong>
      </div>

      <div>
        <p>Un paragraphe dans un deuxième div</p>
      </div>

      <p>Un paragraphe hors div</p>
    </div>
  </body>
</html>
```

```
CSS
body > div{
  border: 5px solid green;
}
```

Output

Run with JS

Auto-run JS ☒

Sélecteurs CSS avancés

Un premier paragraphe dans un div

Un **paragraphe important**

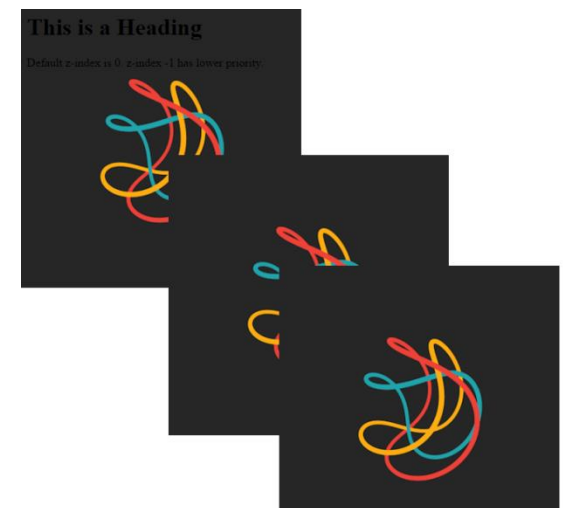
Un paragraphe dans un deuxième div

Un paragraphe hors div

Bin info
just now

Propriété z-index

```
1 <html>
2 <head> <style type="text/css">
3   img {position:absolute;}
4   img.x1{left:350px; top:350px; z-index:-1}
5   img.x2{left:200px; top:200px; z-index:-2}
6   img.x3{left:0px; top:0px; z-index:-3}
7 </style></head>
8 <body>
9   <h1>This is a Heading</h1>
10  
11  
12  
13 </body>
14 </html>
```



Propriété z-index

```
<!DOCTYPE html>
<html>
<head>
<style>
img {
  position: absolute;
  left: 0px;
  top: 0px;
  z-index: -1;
}
</style>
</head>
<body>

<h1>This is a heading</h1>

<p>Because the image has a z-index of -1, it will be placed behind the
text.</p>

</body>
</html>
```



Propriété opacity

```
<!DOCTYPE html>
<html>
<head>
<style>
.container {
  position: relative;
}

.center {
  position: absolute;
  left: 0;
  top: 50%;
  width: 100%;
  text-align: center;
  font-size: 18px;
  border:solid 3px magenta;
}

img {
  width: 100%;
  height: auto;
  opacity: 0.3;
}
</style>
</head>
<body>

<h2>Image Text</h2>
<p>Center text in image:</p>

<div class="container">
  
  <div class="center">Centered</div>
</div>
```

Image Text

Center text in image:



Propriété display

- La propriété CSS **display** permet de modifier le type de rendu d'un élément:
 - Lorsque l'on donne la valeur **inline** à la propriété **display**, l'élément ciblé va se comporter comme un élément de type inline et donc n'occuper que la largeur qui lui est strictement nécessaire.
 - En attribuant la valeur **block** à la propriété **display**, les éléments ciblés vont se comporter comme des éléments HTML de type block et ainsi prendre toute la largeur disponible.
 - La valeur **inline-block** va nous permettre d'emprunter certaines propriétés des éléments de type block et certaines propriétés des éléments de type inline.
 - Finalement, la valeur **none** va nous permettre tout simplement de ne pas afficher un élément.

Les « super-propriétés »

Code : CSS

```
body
{
background-image: url("soleil.png");
background-attachment: fixed; /* Le fond restera fixe */
background-repeat: no-repeat; /* Le fond ne sera pas répété */
background-position: top right; /* Le fond sera placé en haut à droite */
}
```

On peut tout simplement écrire :

Code : CSS

```
body
{
background: url("soleil.png") fixed no-repeat top right;
}
```


Les exceptions : class et id

Si vous avez fait un fichier CSS, mais que vous souhaitez faire exception avec quelques balises, 2 solutions :

→ Décrire directement dans les balises (comme avant)

HTML

```
<h2> Chapitre I </h2>
<p>blablabla</p>

<h2> Chapitre II </h2>
<p style="color:red;font-weight:bold"> patati patata</p>

<h2> Conclusion</h2>
<p>gnagnagna</p>
```

CSS

```
h2
{text-decoration : underline ;
color : red ;}

p
{font-style : italic ;}
```

Les exceptions : class et id

Si vous avez fait un fichier CSS, mais que vous souhaitez faire exception avec quelques balises, 2 solutions :

→ Donner un nom à vos balises !

HTML

```
<h2> Chapitre I </h2>
<p>blablabla</p>

<h2> Chapitre II </h2>
<p class="important">patati patata</p>

<h2> Conclusion</h2>
<p>gnagnagna</p>
```

CSS

```
h2
{text-decoration : underline ;
color : red ;}

p
{font-style : italic ;}

.important
{color : red;
font-weight:bold}
```

Les exceptions : class et id

Si vous avez fait un fichier CSS, mais que vous souhaitez faire exception avec quelques balises, 2 solutions :

→ Donner un nom à vos balises !

HTML

```
<h2> Chapitre I </h2>
<p>blablabla</p>

<h2> Chapitre II </h2>
<p id="important">patati patata</p>

<h2> Conclusion</h2>
<p>gnagnagna</p>
```

CSS

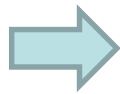
```
h2
{text-decoration : underline ;
color : red ;}

p
{font-style : italic ;}

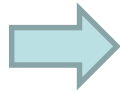
#important
{color : red;
font-weight:bold}
```

Les exceptions : class et id

Quelle différence entre « class » et « id » ?



« **class** » peut être réutilisé avec le même nom pour **plusieurs balises**.



Pour un nom donné, « **id** » ne peut être utilisé **que pour une balise unique**.

Par exemple s'il n'y a qu'un seul menu sur le coté de votre page, on peut définir son style en le nommant :

HTML

```
<div id="menu">
  <a href="lien1.html">lien 1</a>
  <a href="lien2.html">lien 2</a>
  <a href="lien3.html">lien 3</a>
</div>
```

id : Utilisé avec Javascript pour repérer et traiter un élément particulier.

CSS

```
#menu
{width:100px;
float:left;}

#menu a
{display:block;
width:90px;
background-color:blue;}
```

Les pseudo classes

HTML ▾

```
<!DOCTYPE html>
<html>
  <head>
    <title>Pseudo classes</title>
    <meta charset= "utf-8">
  </head>

  <body>
    <h1>Pseudo classes CSS</h1>

    <div class="test">
      <p>Un <a href="http://www.pierre-giraud.</p>
      <p>Un paragraphe</p>
    </div>

    <div>
      <p>Un <a href="http://pierre-giraud.com"
    </div>
    <p>Un paragraphe hors div</p>
  </body>
</html>
```

CSS ▾

```
h1:hover{
  color:orange;
  font-family:verdana, sans-serif;
}

a:link {
  color:blue;
  text-decoration:underline;
}

a:visited {
  color: green;
  font-weight: bold;
  text-decoration:none;
}
```

Output

Run with JS

Auto

Pseudo classes CSS

Un [lien](#) non visité

Un paragraphe

Un [lien](#) déjà visité

Un paragraphe hors div

Les pseudo classes

HTML ▾

```
<!DOCTYPE html>
<html>
  <head>
    <title>Pseudo classes</title>
    <meta charset= "utf-8">
  </head>

  <body>
    <h1>Pseudo classes CSS</h1>

    <div class="test">
      <p>Un premier paragraphe</p>
      <p>Un deuxième paragraphe</p>
      <p>Un troisième paragraphe</p>
    </div>
    <p>Un paragraphe hors div</p>
  </body>
</html>
```

CSS ▾

```
p:first-child{
  color: orange;
}

p:last-child{
  color: green;
}
```

Output

Pseudo classes CSS

Un premier paragraphe

Un deuxième paragraphe

Un troisième paragraphe

Un paragraphe hors div

Chrome (ctrl + shift + j)

The screenshot shows a Google Chrome browser window with the address bar displaying `file:///E:/Espace/ingWeb/exo2.html`. The page content consists of four paragraphs, each with a different border style and color:

- Paragraph 1: "1st. The man-mountain shall not depart from our dominions, without our license under our great seal." (Black border)
- Paragraph 2: "2d, He shall not presume to come into our metropolis, without our express order; at which time, the inhabitants shall have two hours warning to keep within doors." (Blue border)
- Paragraph 3: "3d, The said man-mountain shall confine his walks to our principal high roads, and not offer to walk, or lie down, in a meadow or field of corn." (Green border)
- Paragraph 4: "4th, As he walks the said roads, he shall take the utmost care not to trample upon the bodies of any of our loving subjects, their horses, or carriages, nor take any of our subjects into his hands without their own consent." (Dashed purple border)

Below the paragraphs, there is a link labeled "Mon texte" and three links labeled "lien 1", "lien 2", and "lien 3".

The Chrome Developer Console is open, showing the "Elements" tab. The selected element is a paragraph with the class `empurpledash`. The "Styles" pane shows the following CSS rules:

```
element.style {  
}  
.empurpledash {  
  border-width: 0.5em;  
  border-color: #800080;  
  border-style: dashed;  
}  
p {  
  display: block;  
  -webkit-margin-before: 1em;  
  -webkit-margin-after: 1em;  
  -webkit-margin-start: 0px;  
  -webkit-margin-end: 0px;  
}
```

The "Properties" pane shows the following dimensions for the selected element:

- margin: 16px
- border: 8px
- padding: 8px
- width: 857 x 36px
- height: 8px
- bottom: 16px

Jsfiddle : Débugger et partager du code HTML/CSS/JavaScript

The screenshot displays the Jsfiddle web application interface. At the top, the browser address bar shows the URL <https://jsfiddle.net/fgr1ncx4/1/>. Below the browser, the Jsfiddle toolbar includes buttons for 'Run', 'Update', 'Fork', 'Tidy', 'Collaborate', and 'Embed'. The main workspace is divided into three panels: 'Fiddle Meta' on the left, 'HTML' in the center, and 'CSS' on the right. The 'Fiddle Meta' panel shows the title 'Untitled fiddle' and a file path 'file:///E:/Espace/ingWeb/ex2.css'. The 'HTML' panel contains the following code:

```
1 <body>
2   <p class="thickline">
3     "1st, The man-mountain shall not depart from our dominions, without our
4     license under our great seal.
5   </p>
6   <p class="thinblueline">
7     2d, He shall not presume to come into our metropolis, without our express
8     order; at which time, the inhabitants shall have two hours warning to keep
9     within doors.
10  </p>
11  <p class="greenridge">
12    3d, The said man-mountain shall confine his walks to our principal high
13    roads, and not offer to walk, or lie down, in a meadow or field of corn.
14  </p>
15  <p class="empurpledash">
16    4th, As he walks the said roads, he shall take the utmost care not to
17    trample upon the bodies of any of our loving subjects, their horses, or
18    carriages, nor take any of our subjects into his hands without their own
19    consent.
20  </p>
21  <p style=" background-color : #AAAAFF; "> Mon texte </p>
```

The 'CSS' panel contains the following code:

```
1 #menu
2 {
3   width:200px;
4   Float:left;
5   Background-color:white;
6 }
7
8 #menu a
9 {
10  display:block;
11  width:90px;
12  Background-color:black;
13 }
14
15 .thickline {
16
17   2d, He shall not presume to come into our metropolis
18   order; at which time, the inhabitants shall have two h
19   within doors.
```

Below the code panels, the 'JAVASCRIPT' panel is visible, showing the following code:

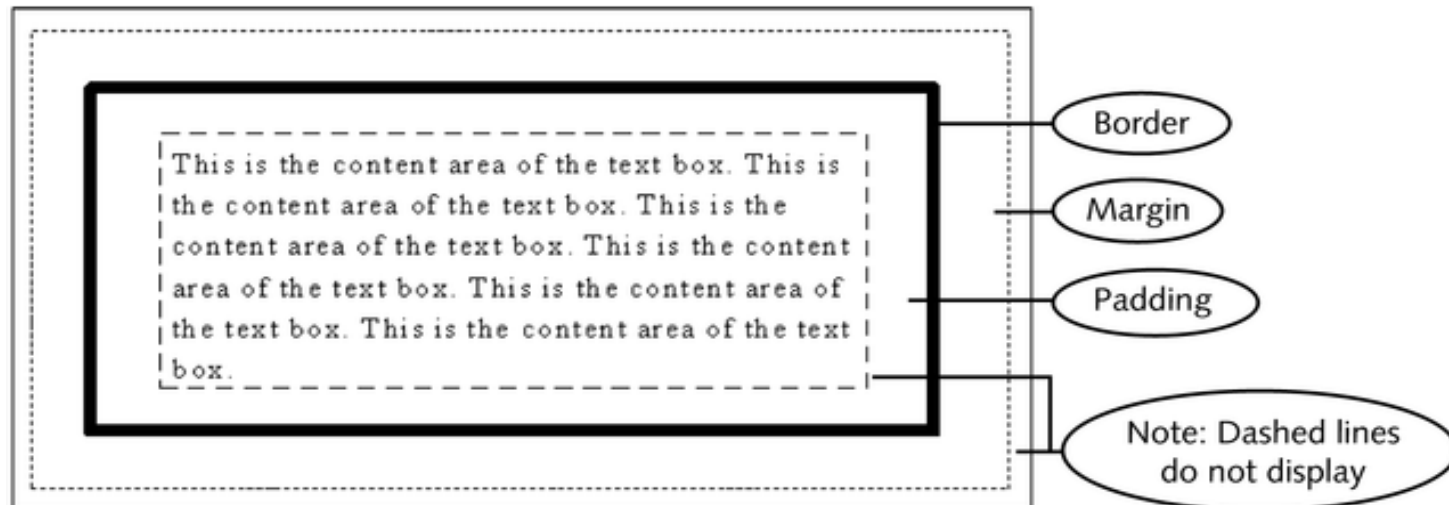
```
1
2
3 3d, The said man-mountain shall confine his walks to our pri
4 to walk, or lie down, in a meadow or field of corn.
5
6 4th, As he walks the said roads, he shall take the utmost care
7 bodies of any of our loving subjects, their horses, or carriage
8 into his hands without their own consent.
9
10 Mon texte
11 lien 1
12 lien 2
13 lien 3
```

At the bottom of the interface, the URL <https://jsfiddle.net/fgr1ncx4/1/> is displayed. The bottom of the slide features a blue banner with the text 'UE: Programmation réseaux et système' and a large number '79' in the bottom right corner.

Les modes de positionnement

- Largeur totale de l'élément ici bas est de 300px:

```
div {  
  width:250px;  
  padding:10px;  
  border:5px solid gray;  
  margin:10px;  
}
```

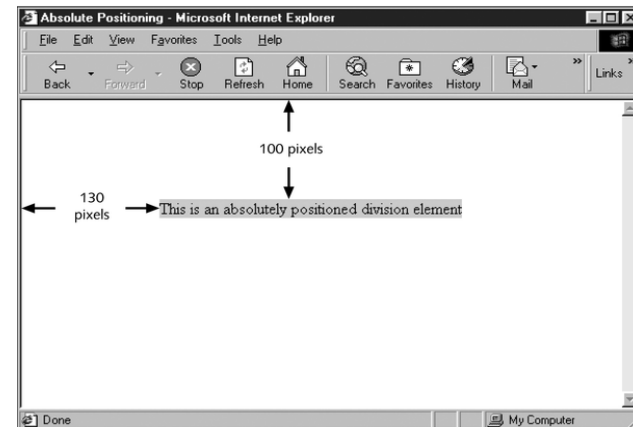


Les propriétés de positionnement

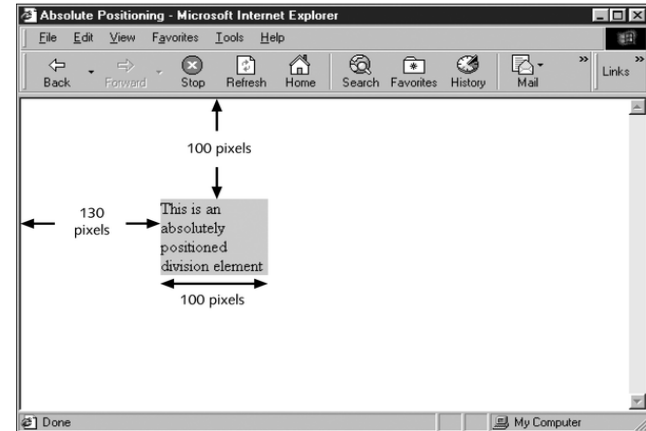
- **position: type position size**
 - **type = static | relative | absolute | fixed**
 - **position = top | left | bottom | right**
 - **size = height | width**
 - **overflow: visible | hidden | scroll | auto**
 - **clip: rect(top, right, bottom, left)**
 - **visibility: visible | hidden**

Les propriétés de positionnement

- `div {position:absolute; left:130px; top:100px;}`



- `div {position:absolute; left:130px; top:100px; width: 100px;}`



Les propriétés de positionnement

- **width: # | %**
 - Largeur de la boîte de contenu
- **height: # | %**
 - Hauteur de la boîte de contenu
- **float: left | right | none**
 - Mettre l'élément à droite/gauche de l'élément parent
- **clear: none | left | right | both**
 - Contrôle le flot de texte autour d'un élément flottant

Les modes de positionnement

- Flux : ordre d'affichage des éléments
- 4 méthodes de positionnement
 - normal
 - float
 - relatif
 - absolu
- Quels sont les différences ?

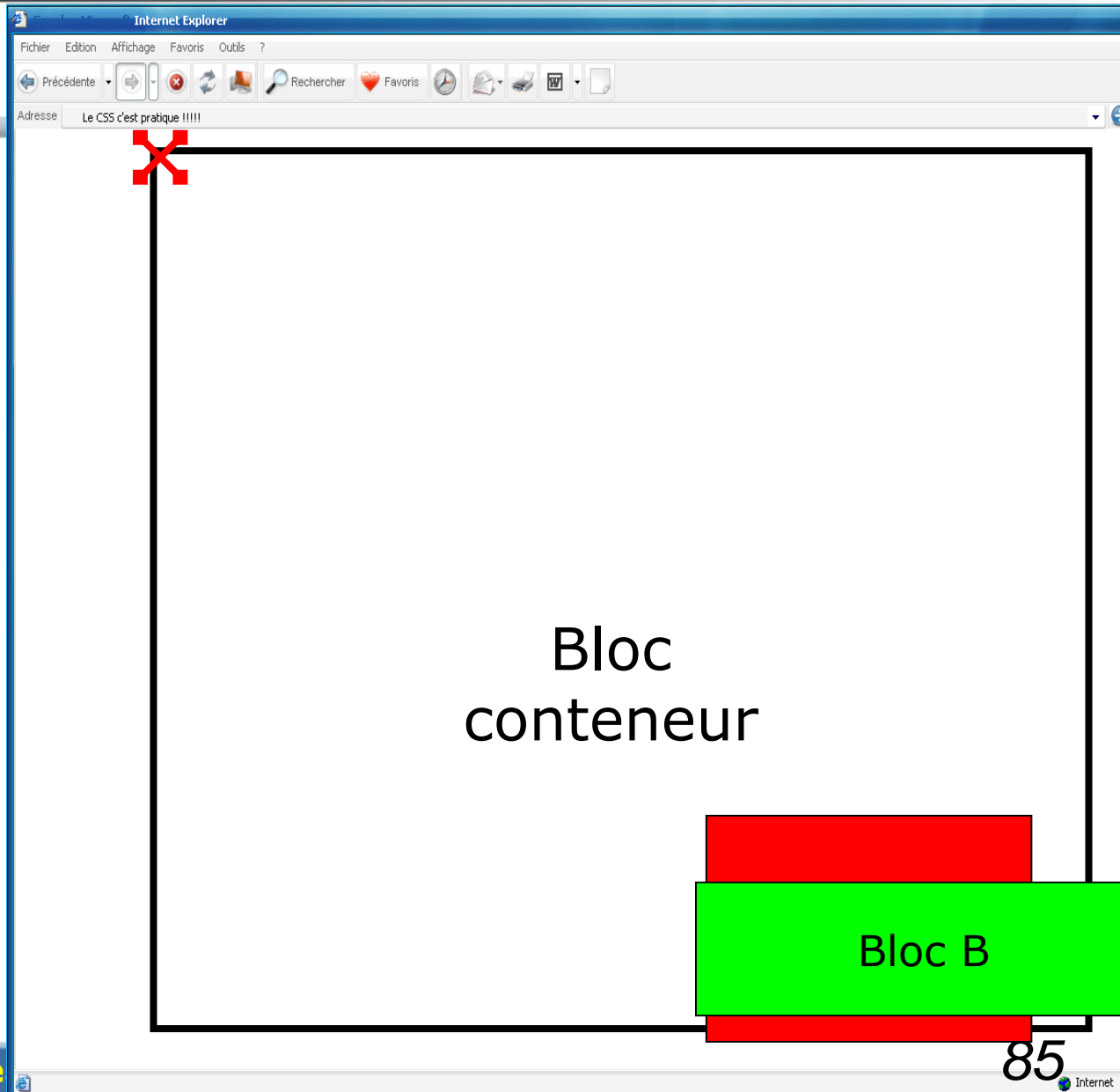
Le Flux normal : en bloc

<BODY>

```
<div class="conteneur">
  <div class="normalA">
    Bloc A
  </div>
  <div class="normalB">
    Bloc B
  </div>
</div>
</BODY>
```

```
.normalA {
width:150px;
height:150px;
background-color:red;
border:1px solid black;
}
```

```
.normalB {
width:250px ;
height:100px ;
background-color:green;
border:1px solid black;
}
```



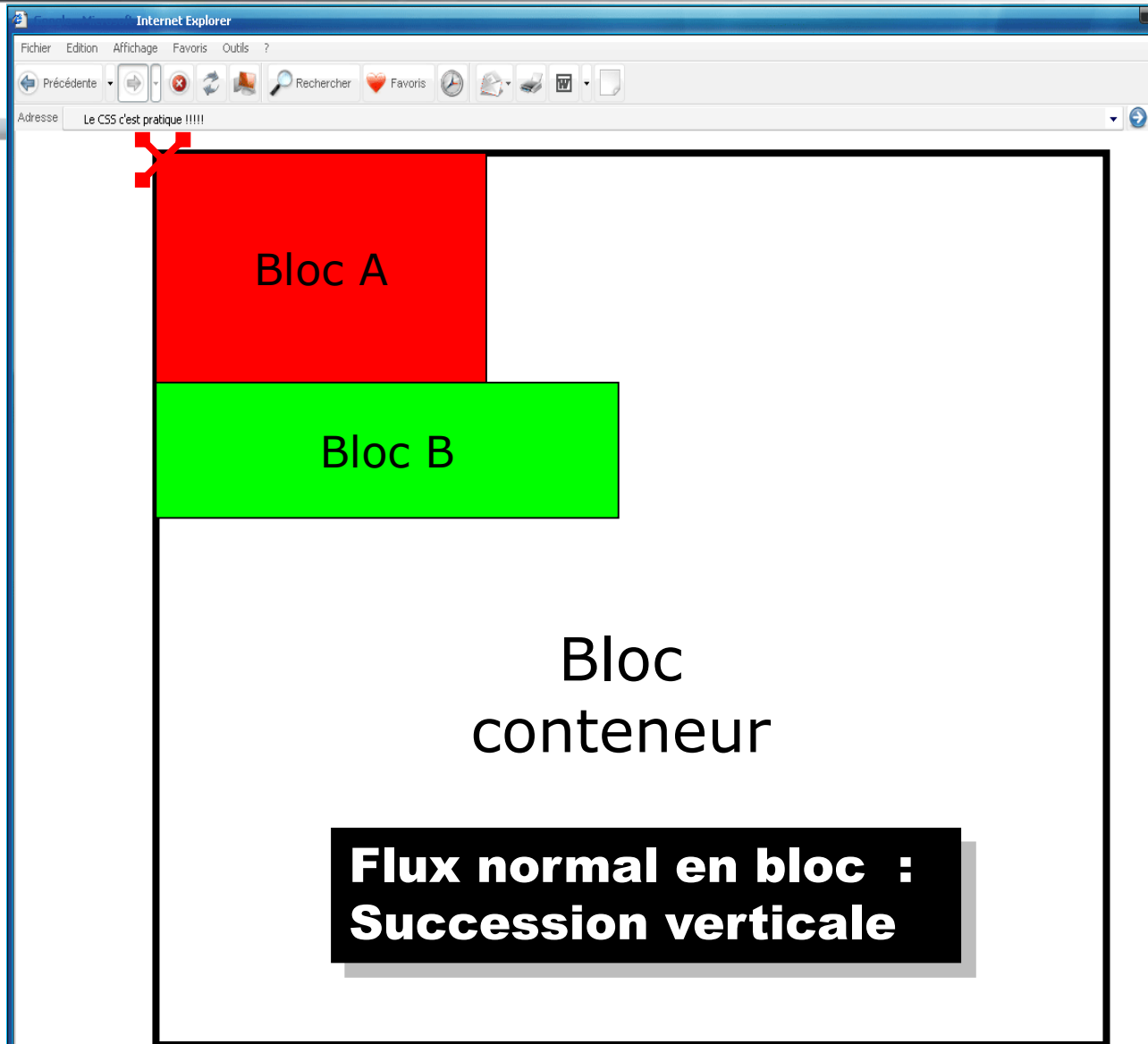
Le Flux normal : en bloc

<BODY>

```
<div class="conteneur">
  <div class="normalA">
    Bloc A
  </div>
  <div class="normalB">
    Bloc B
  </div>
</div>
</BODY>
```

```
.normalA {
width:150px;
height:150px;
background-color:red;
border:1px solid black;
}
```

```
.normalB {
width:250px ;
height:100px ;
background-color:green;
border:1px solid black;
}
```



**Flux normal en bloc :
Succession verticale**

Le Flux normal : en ligne

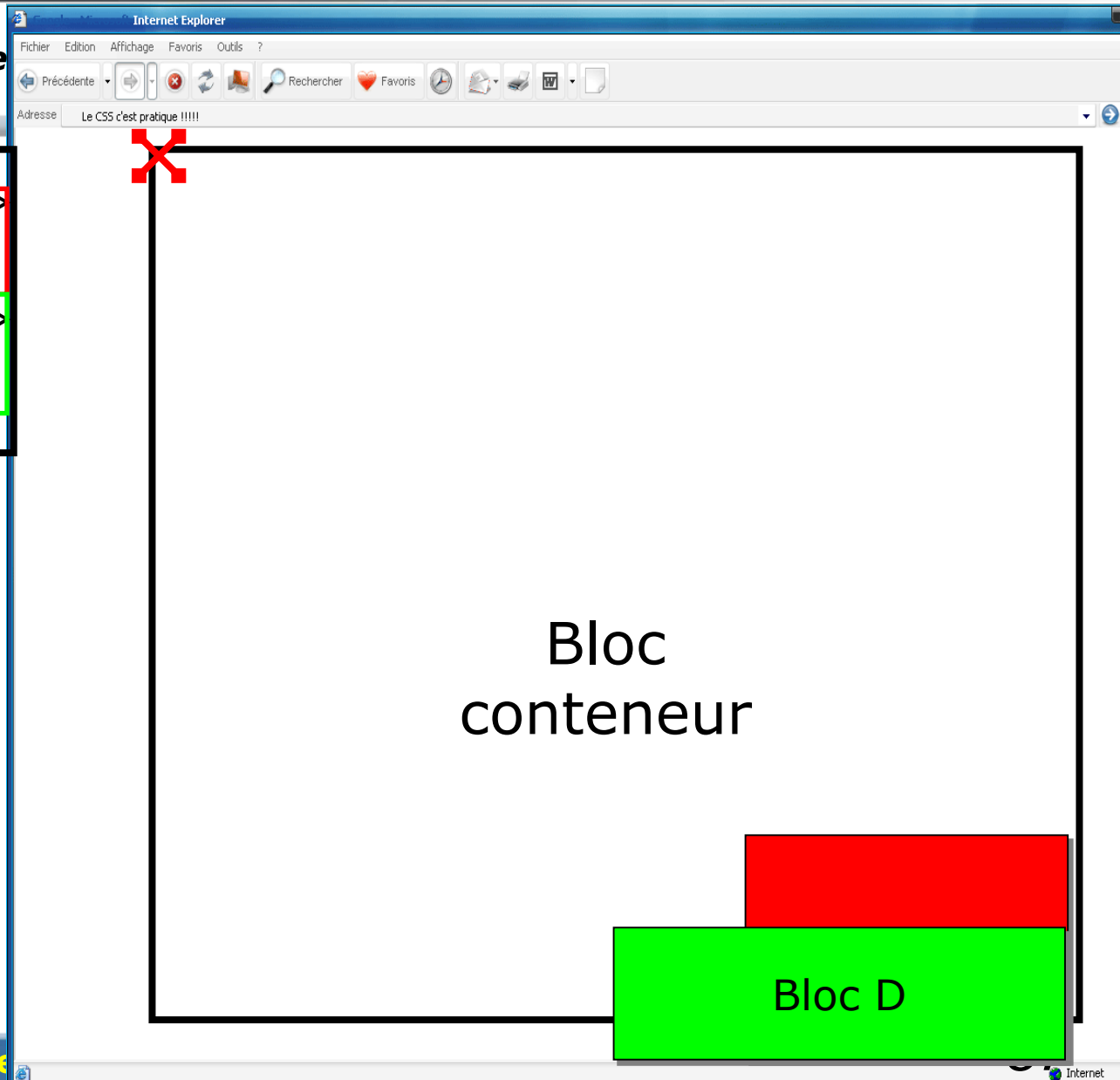
<BODY>

```
<div class="conteneur">  
  <span class="normalC">  
    Bloc C  
  </span>  
  <span class="normalD">  
    Bloc D  
  </span>  
</div>
```

</BODY>

```
.normalC {  
width:150px;  
height:150px;  
background-color:red;  
border:1px solid black;  
}
```

```
.normalD {  
width:250px ;  
height:100px ;  
background-color:green;  
border:1px solid black;  
}
```



Le Flux normal : en ligne

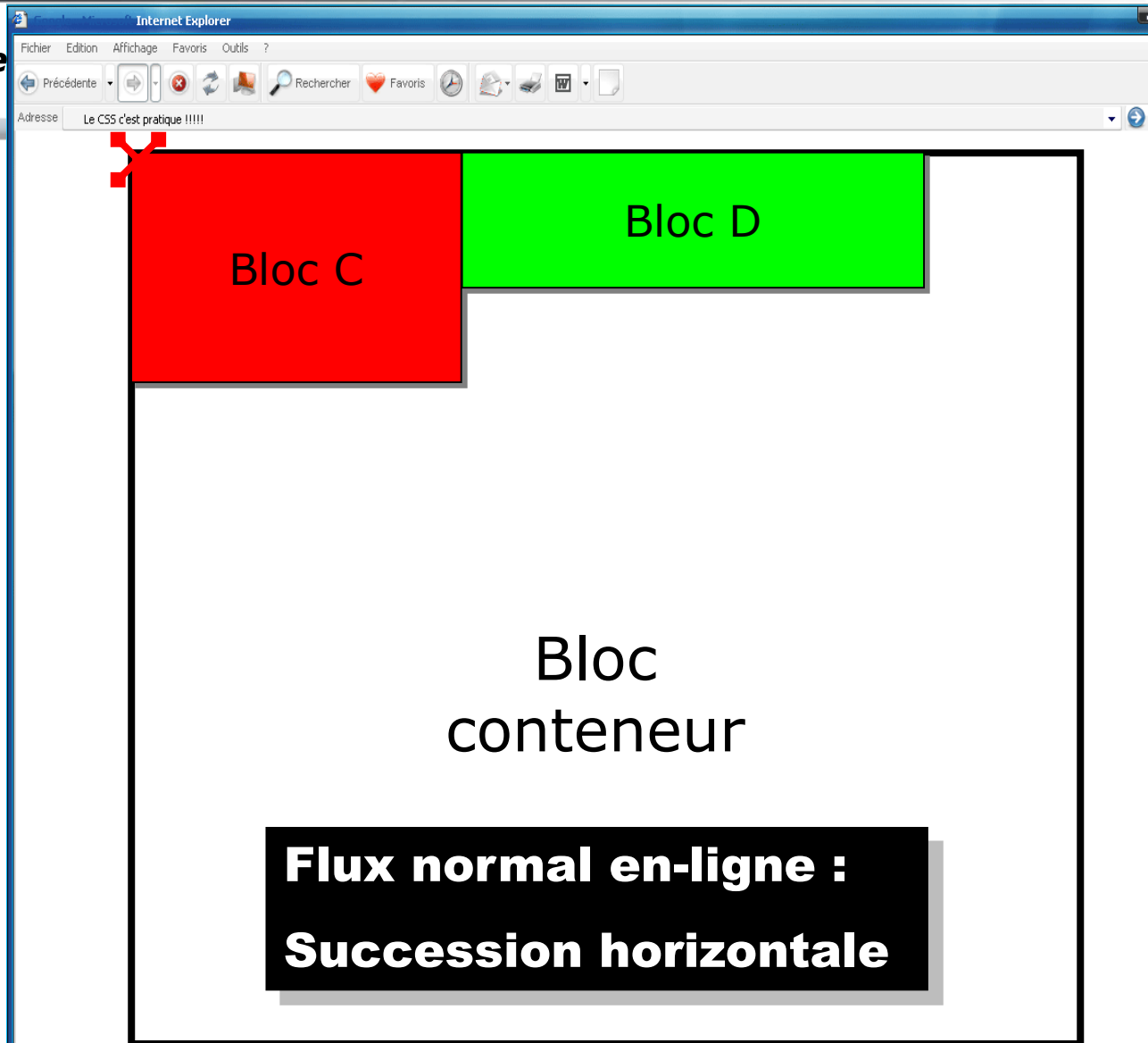
<BODY>

```
<div class="conteneur">  
  <span class="normalC">  
    Bloc C  
  </span>  
  <span class="normalD">  
    Bloc D  
  </span>  
</div>
```

</BODY>

```
.normalC {  
width:150px;  
height:150px;  
background-color:red;  
border:1px solid black;  
}
```

```
.normalD {  
width:250px ;  
height:100px ;  
background-color:green;  
border:1px solid black;  
}
```

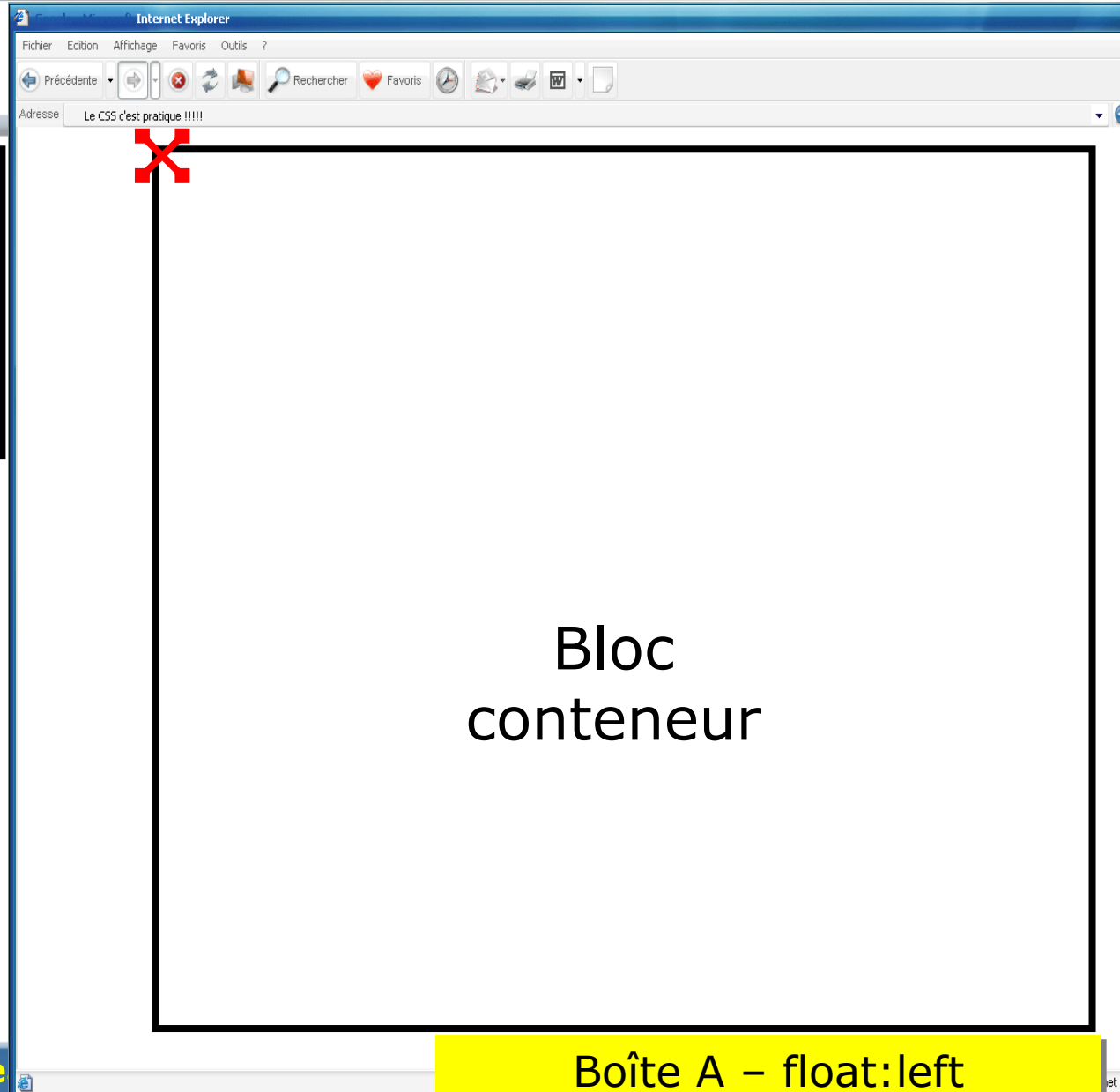


Le Flux flottant

<BODY>

```
<div class="conteneur">  
  <div class="flotteA">  
    Boîte A  
  </div>  
  <p>  
    Texte...blabla ...  
  </p>  
</div>  
</BODY>
```

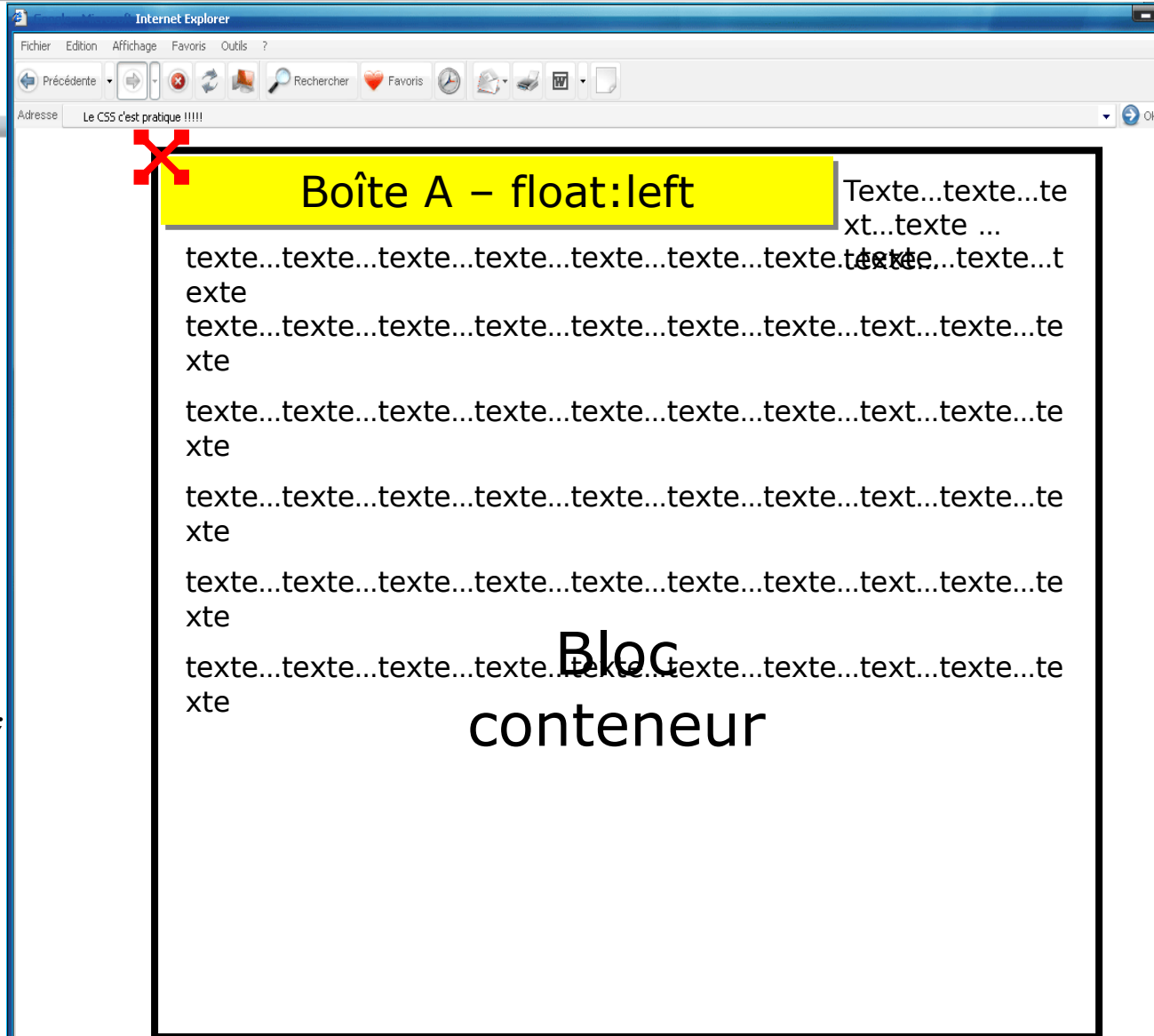
```
.flotteA {  
float:left;  
width:500px;  
background-color:yellow;  
border:1px solid black;  
}
```



Le Flux flottant

```
<BODY>
<div class="conteneur">
  <div class="flotteA">
    Boîte A
  </div>
  <p>
    Texte...texte ...
  </p>
</div>
</BODY>
```

```
.flotteA {
float:left;
width:650px;
background-color:yellow;
border:1px solid black;
}
```



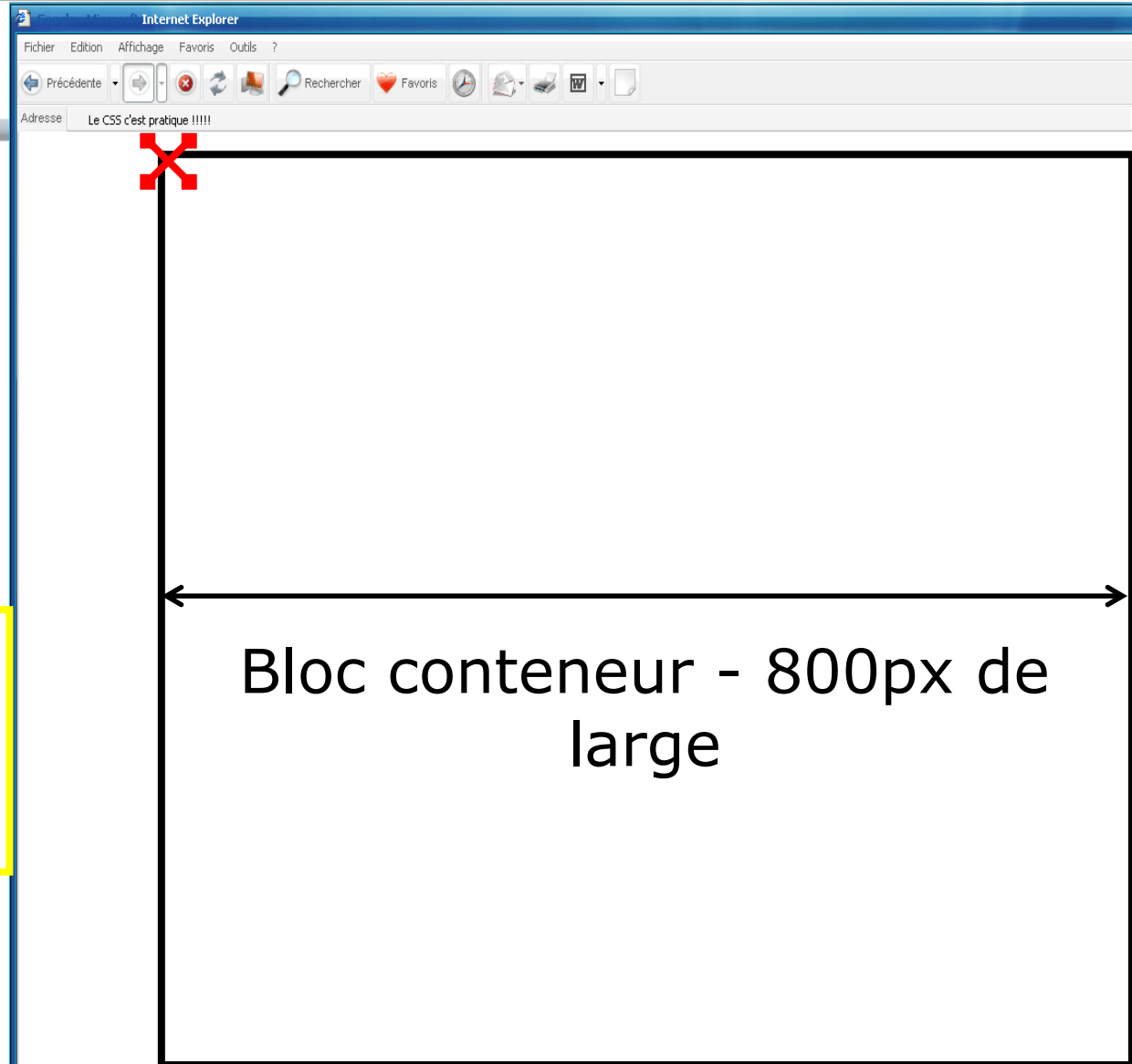
CSS

```
<BODY>  
<div class="conteneur">  
  <div class="flotteA">  
    Boîte A  
  </div>  
  <div class="flotteB">  
    Boîte B  
  </div>  
</div>  
</BODY>
```

```
.conteneur {  
width:800px;  
border:1px solid black;  
}
```

```
.flotteA {  
float:left;  
width:650px;  
background-color:yellow;  
border:1px solid black;  
}
```

```
.flotteB {  
float:left;  
width:100px;  
background-color:blue;  
border:1px solid black;  
}
```



CSS

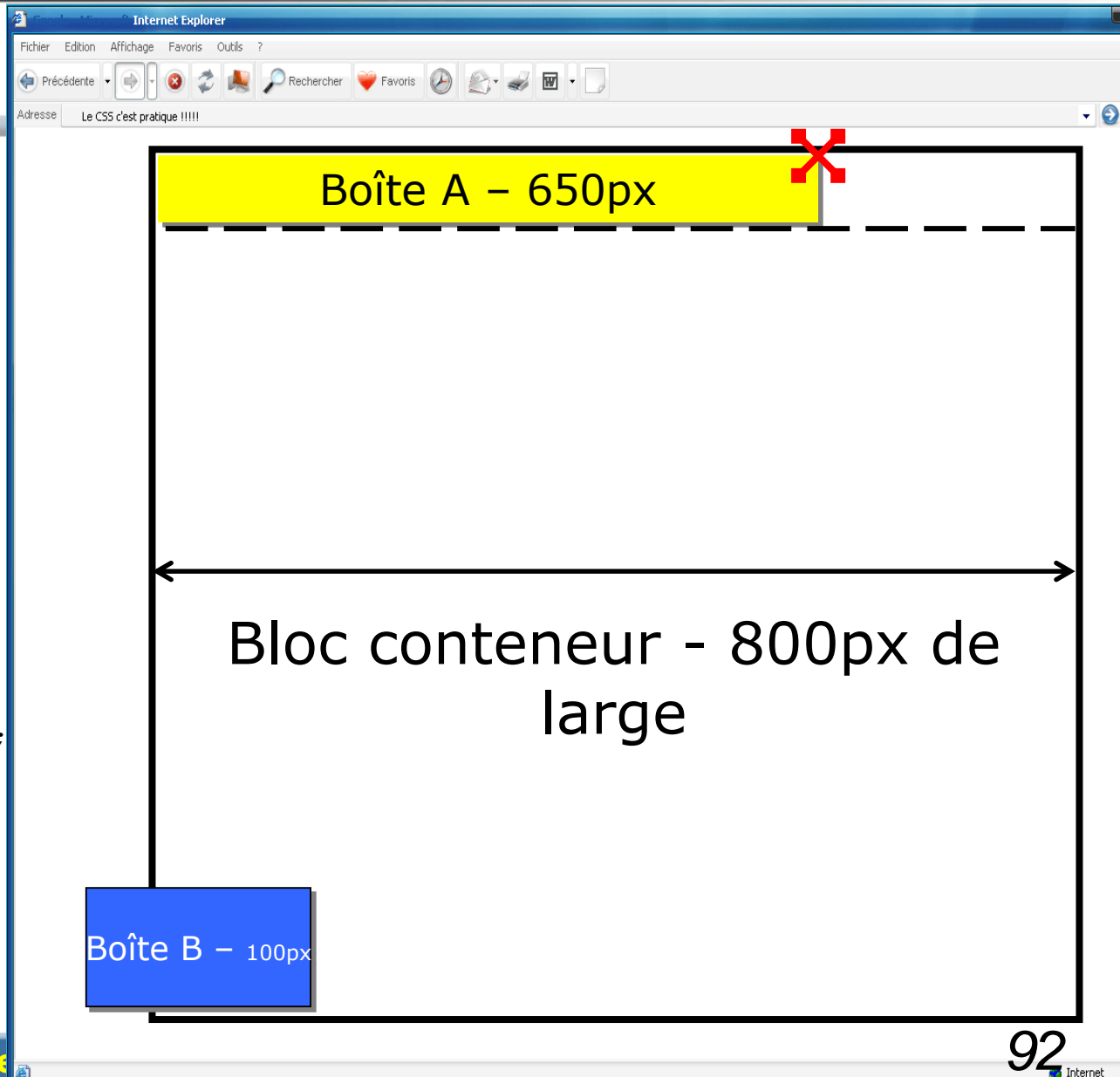
```
<BODY>
<div class="conteneur">
  <div class="flotteA">
    Boîte A
  </div>
  <div class="flotteB">
    Boîte B
  </div>
</div>
</BODY>
```

```
.conteneur {
width:800px;
border:1px solid black;
}

.flotteA {
float:left;
width:650px;
background-color:yellow;
border:1px solid black;
}
```

```
.flotteB {
float:left;
width:100px;
background-color:blue;
border:1px solid black;
}
```

UE: Programmation rése



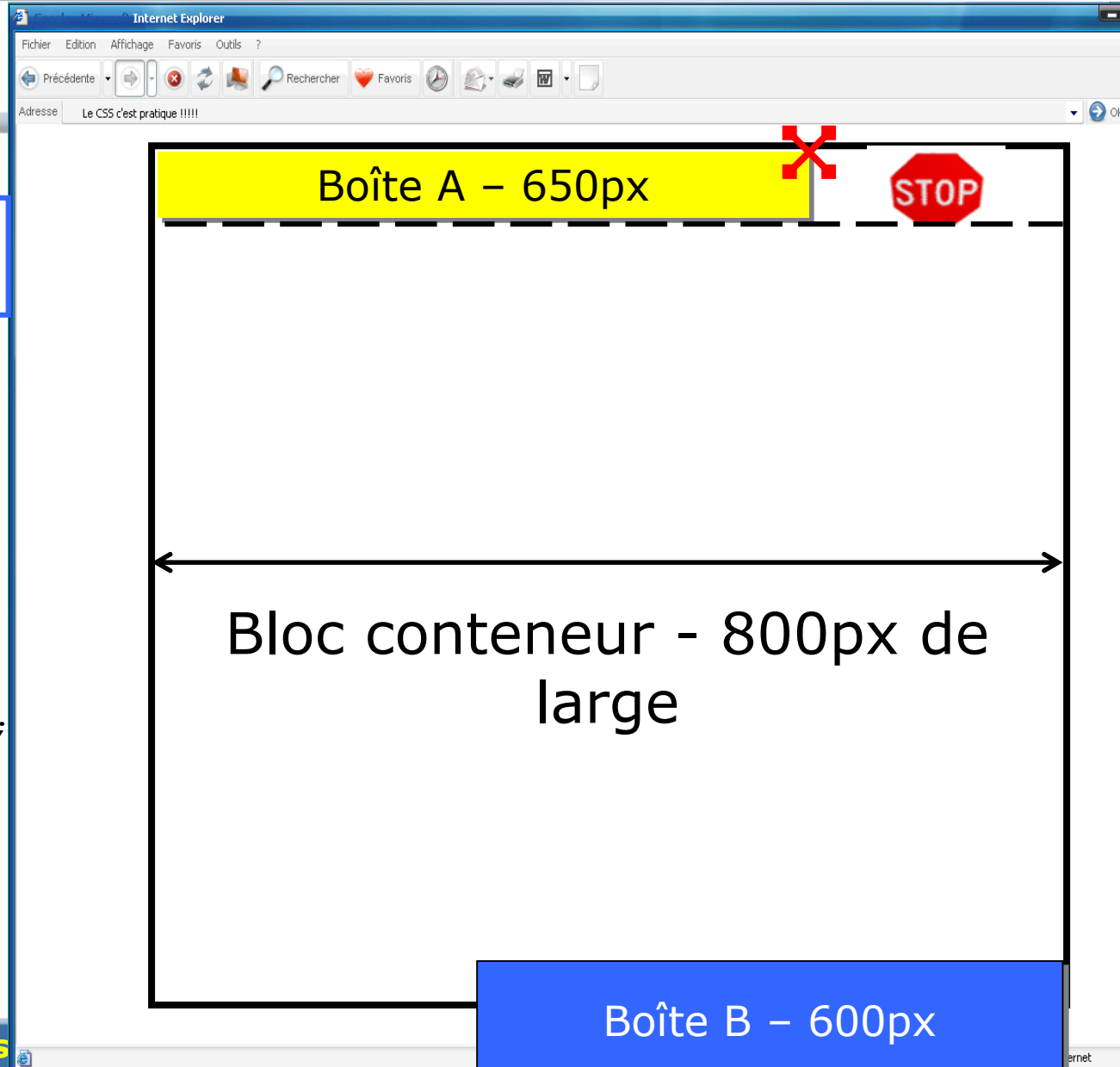
CSS

```
<BODY>
<div class="conteneur">
  <div class="flotteA">
    Boîte A
  </div>
  <div class="flotteB">
    Boîte B
  </div>
</div>
</BODY>
```

```
.conteneur {
width:800px;
border:1px solid black;
}

.flotteA {
float:left;
width:650px;
background-color:yellow;
border:1px solid black;
}
```

```
.flotteB {
float:left;
width:600px;
background-color:blue;
border:1px solid black;
}
UE: Programmation rés
```

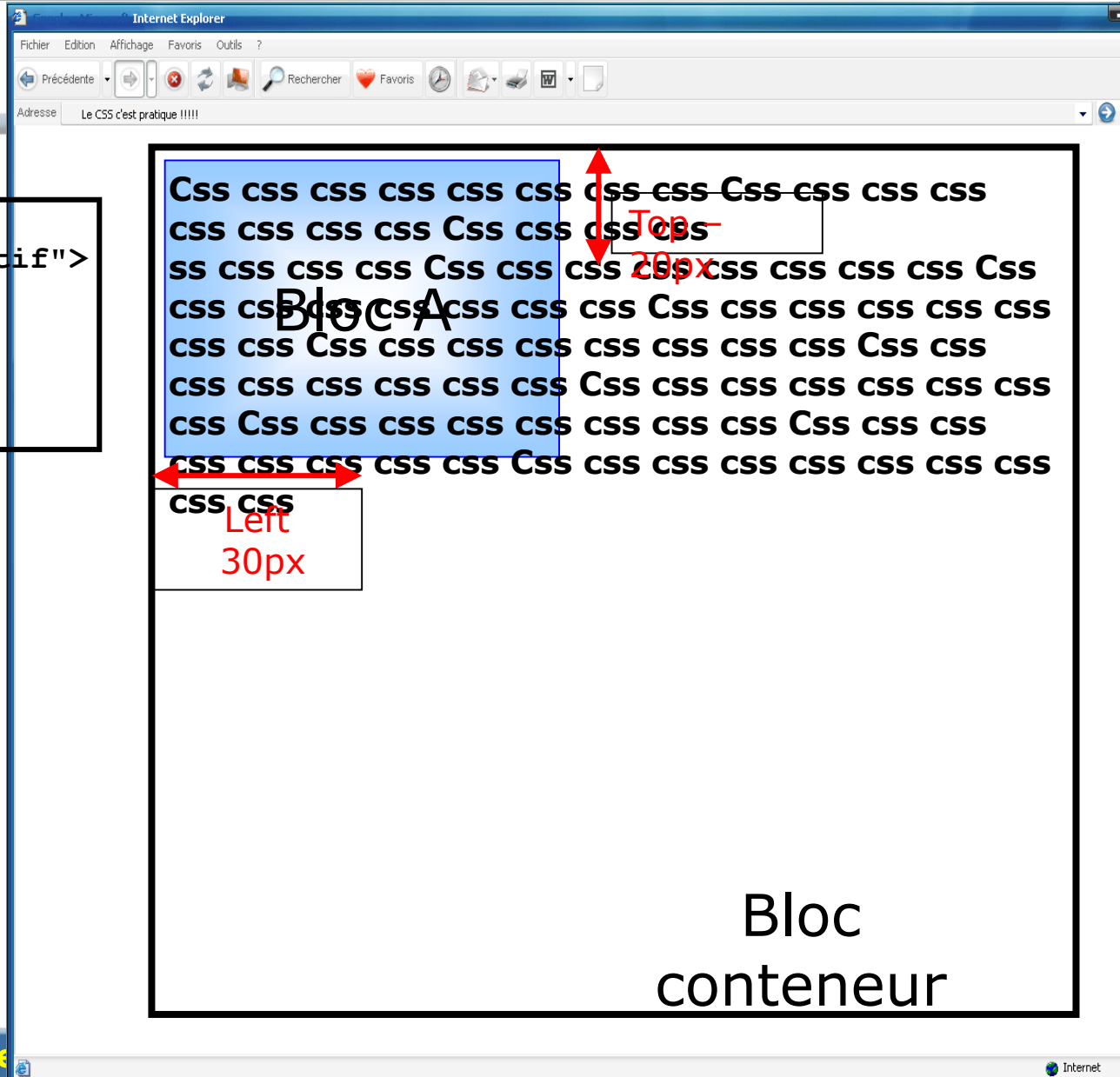


Le Flux relatif

```
<BODY>
<div class="conteneur">
  <div class="bloc_relatif">
    Bloc A
  </div>
  CSS CSS CSS CSS...
</div>
</BODY>
```

```
.conteneur {
width:800px;
border:1px solid black;
}
```

```
.bloc_relatif {
position:relative;
width:300px;
margin-top:20px;
margin-left:30px;
border:1px solid black;
}
```



```
<BODY>
<div class="conteneur">
  <div class="bloc_relatif">
    Bloc A
  </div>
  CSS CSS CSS CSS...
</div>
</BODY>
```

```
.conteneur {
width:800px;
border:1px solid black;
}
```

```
.bloc_relatif {
position: relative;
width: 300px;
margin-top: 20px;
margin-left: 30px;
border: 1px solid black;
}
```



Flux absolu

```
<BODY>
<div class="conteneur">
  <div class="bloc_absolu">
    Bloc A
  </div>

  CSS CSS CSS CSS CSS...

</div>
</BODY>

.conteneur {
width:800px;
border:1px solid black;
}

.bloc_absolu {
position:absolute;
width:300px;
margin-top:20px;
margin-left:30px;
border:1px solid black;
}
```

