

Les Frames HTML :

Nous avons vu jusqu'à présent que, avec HTML, il est possible de créer des tableaux complexes, dont chaque cellule peut contenir indifféremment du texte, un formulaire, une image et bien d'autres encore ... Nous allons voir que les frames, introduites par certains browsers, et maintenant disponibles sur les principaux browsers du marché, apportent des fonctionnalités voisines des tableaux, encore plus puissantes. Le concept est simple. Comme avec les tableaux, il consiste à diviser l'écran du browser en plusieurs zones, appelées **frames**. Chacune d'elles est capable d'afficher du code HTML, comme c'est le cas dans les cellules d'un tableau. Il existe cependant une différence majeure. Dans un tableau, toutes les cellules font partie du même document HTML. Elles sont toutes issues d'un même fichier ou d'un même **script CGI**. Dans le cas des frames, chaque zone de l'écran est un document HTML propre. On voit donc simultanément, sur la même fenêtre d'un browser, la mise en page correspondant à plusieurs fichiers HTML différents. Un lien hypertexte peut alors mettre à jour certaines zones du browser sans modifier les autres. Chaque zone possédant un nom unique, il est simple d'indiquer quelle zone un lien hypertexte doit être modifié.

La notion de frame supprime l'une des limitations du Web. Auparavant, chaque nouveau document venait écraser le précédent, et obligerait bien souvent l'utilisateur à user et abuser de la commande "précédent" des browsers (back en anglais). Désormais, il est possible de dédier certaines zones du browser à des tâches bien précises. On peut imaginer, par exemple, une zone qui contienne une barre de menus et qui ne soit presque jamais rechargée. Ou encore, une autre zone utilisée pour afficher une aide contextuelle, sans que l'utilisateur perde la page sur laquelle il travaille.

Pour mieux comprendre ce nouveau concept, examinons un exemple. La page proposée est composée de trois zones (on parlera indifféremment dans cette leçon de zone ou de frame). La zone supérieure contient un formulaire de recherche. Celle de gauche affiche les résultats de la recherche. Quant à celle de droite, elle permet d'obtenir plus d'informations concernant un élément dans la liste des résultats. La soumission du formulaire de la zone supérieure met à jour la zone de gauche sous forme d'une liste de valeurs dont chaque élément est un lien hypertexte. Chaque lien hypertexte de la liste des résultats modifie la zone de droite. Cet écran permet donc de gérer simultanément trois documents HTML.

Pour donner un premier aperçu de la syntaxe utilisée, voici le code HTML à l'origine de cet exemple :

Exemple d'utilisation d'une page intitulée "[Apercu_1_de_Frame.html](#)", dont le code HTML ci-dessous :

<HTML>

<HEAD>

```

<TITLE> Exemple d'un Annuaire </TITLE>

</HEAD>

    <FRAMESET ROWS="235, *">

        <FRAME          SRC="Annuaire.html"          MARGINWIDTH="25"
MARGINHEIGHT="25" NORESIZE>

            <FRAMESET COLS="48%, 50%">

                <FRAME          SRC="images/Livre.png"          NAME="LISTE"
MARGINWIDTH="35" MARGINHEIGHT="35">

                    <FRAME          SRC="images/Livre.png"          NAME="DETAIL "
MARGINWIDTH="35" MARGINHEIGHT="35">

                        </FRAMESET>

                    <NOFRAMES>

                        Ce document contient des frames et votre browser est incapable de
les g&eacute;rer ...

                    </NOFRAMES>

                </FRAMESET>

            </FRAMESET>

</HTML>

```

Il est intéressant de noter que ce document ne contient aucune des informations qui vont être présentées à l'utilisateur. En fait, il ne contient que des informations destinées au browser. Avec ce document, le browser sait en effet quelles sont les **frames** qu'il doit créer et avec quels documents il doit les remplir. Si l'on regarde de plus près, par exemple, la définition de la zone ou frame supérieure, on note qu'elle doit recevoir le code HTML du document "[Annuaire.html](#)". Ce document est un document HTML classique, qui contient l'information à afficher à l'utilisateur. En voici le détail ci-après :

Exemple d'utilisation d'une page intitulée "[Annuaire.html](#)", dont le code HTML ci-dessous faisant parti du même document HTML, en [Exemple 1] ci-dessus :

```

<HTML>

    <HEAD>

        <TITLE> Annuaire </TITLE>

        <STYLE TYPE="text/css"> /* Style interne */

```


On remarque toutefois un nouvel attribut pour la balise **<FORM>**, l'attribut **TARGET**, qui indique tout simplement le nom de la zone dans laquelle sera affiché le résultat de la soumission du formulaire.

A noter : Si vous voulez changer les noms des pages HTML, il faudra également envisager de changer les noms dans le code HTML.

Création de frames :

En fait, seules trois nouvelles balises ont été ajoutées au langage HTML afin d'apporter cette nouvelle fonctionnalité. Cependant, la structure générale d'un document divisé en plusieurs frames diffère de celle d'un document classique. Dans un document classique, le corps est inséré dans les balises **<BODY>** ... **</BODY>**. Dans le cas des frames, le corps du document est systématiquement inséré dans les balises **<FRAMESET>** ... **</FRAMESET>**, **<FRAME>** et **<NOFRAMES>**.

Nous attirons votre attention que si vous mettez une balise **<BODY>** dans une frame par exemple, avec une balise **<FRAMESET>**, cela ne pourra pas fonctionner.

Exemple de document classique contenant l'information à afficher à l'utilisateur (texte, images, formulaires ...)

```
<HTML>

  <HEAD>

    <TITLE> ..... </TITLE>

    .....

  </HEAD>

  <BODY>

    .....

  </BODY>

</HTML>
```

Exemple de document de définition de frames ne contenant aucune information destinée à l'utilisateur, mais uniquement des informations destinées au browser (géométrie des différentes frames et document HTML associé).

```
<HTML>

  <HEAD>

    <TITLE> ..... </TITLE>
```

.....

</HEAD>

<FRAMESET ... >

<FRAMESET ... >

...

</FRAMESET ... >

...

<FRAMESET ... >

<FRAME ... >

</FRAMESET ... >

<FRAME ... >

</FRAMESET ... >

</HTML>

<FRAMESET > :

<FRAMESET > est la balise d'ouverture permettant de diviser une zone en sous-zone, soit verticalement, soit horizontalement. Si aucune zone n'est encore définie, les divisions s'appliquent à la zone initiale formée par l'ensemble de la fenêtre du browser. Cette balise possède les attributs suivants :

ROWS est utilisé pour diviser la zone en sous-zones horizontales. La syntaxe est la suivante :

ROWS="hauteur_zone_1, hauteur_zone_2, hauteur_zone_n"

ROWS est une liste de valeurs entières séparées par des virgules. Le nombre d'éléments de la liste correspond au nombre de sous-zones horizontales à créer. Chacune des valeurs de la liste peut être donné selon l'un des trois formats suivants, dans lesquels **n** est un nombre entier :

- **n** : la valeur de **n** indique la hauteur de la sous-zone en nombre de pixels ;
- **n%** : **n** indique ici la hauteur de la sous-zone exprimée en pourcentage de la taille de la zone mère ;
- **n*** : ici, **n** est optionnel ; le caractère * indique au browser qu'il doit donner à la sous-zone la place encore disponible. Lorsque plusieurs éléments de la liste

utilisent ce format, la place sera encore disponible est répartie entre les sous-zones correspondantes. **n** pourra alors être utilisé pour indiquer au browser qu'une sous-zone doit recevoir une proportion **n** fois supérieure de la place encore disponible.

COLS est utilisé pour diviser la zone en sous-zones verticales. La syntaxe est en tous points identique à celle de l'attribut **ROWS** :

```
COLS="largeur_zone_1, largeur_zone_2, largeur_zone_n"
```

Voici par exemple deux manières identiques de diviser une zone en trois. Les deux sous-zones supérieures sont de taille identique, égale à la moitié de la zone inférieure :



```
<FRAMESET ROWS="25%, 25%, 50%"> ... </FRAMESET>
```

Ou encore

```
<FRAMESET ROWS="*, *, 2*"> ... </FRAMESET>
```

FRAMEBORDER est utilisé pour indiquer si la frontière entre deux frames doit avoir un effet 3D (un effet d'ombre). La valeur par défaut est "**yes**", si bien que les bordures de frames possèdent en général un effet 3D. La syntaxe est la suivante :

```
FRAMEBORDER="NO/YES"
```

BORDER indique, dans le cas où **FRAMEBORDER** vaut "**no**", la largeur de la bordure entre deux frames. La valeur 0 supprime les bordures de frames. La syntaxe est la suivante :

```
MARGINWIDTH="n"
```

BORDERCOLOR indique, dans le cas où **FRAMEBORDER** vaut "**yes**", la couleur des bordures de frames. La syntaxe est la suivante :

```
BORDERCOLOR="couleur"
```

Insérées entre les balises **<FRAMESET> . . . </FRAMESET>**, on ne peut trouver que les trois balises suivantes :

- **<FRAMESET> . . . </FRAMESET>**, qui permet de rediviser une sous-zone ;
- **<FRAME . . . >** qui est utilisée pour caractériser une sous-zone ;

- Et enfin, `<NOFRAME> . . . </NOFRAME>`, qui permet **d'afficher un texte à la place des frames si le browser n'est pas capable de les gérer.**

Aucune autre balise HTML ne peut être acceptée.

<FRAME> :

<FRAME> est la balise utilisée pour caractériser les sous-zones définies à l'aide de la balise **<FRAMESET>**. Cette caractérisation est réalisée à l'aide de six attributs :

SRC indique l'URL du document qui doit être affiché dans cette zone. Si cet attribut n'est pas précisé, la zone reste vide. La syntaxe est la suivante :

`SRC="URL"`

NAME est utilisé pour nommer la zone afin qu'elle puisse devenir la cible de n'importe quel lien hypertexte. Sinon, cela ne peut fonctionner. Les noms prédéfinis, **_blank**, **_top**, **_parent**, **_self** ainsi que tous les noms commençant par le "caractère de suite" **'_'** ne peuvent pas être utilisés pour renseigner cet attribut. La syntaxe de l'attribut **NAME** est ci-après :

`NAME="nom_de_la_zone"`

MARGINWIDTH est utilisé pour préciser le nombre de pixels entre les frontières gauche et droite de la zone et le document HTML à afficher. La syntaxe est la suivante :

`MARGINWIDTH="n"`

MARGINHEIGHT est utilisé pour préciser le nombre de pixels entre les frontières haute et basse de la zone et le document HTML à afficher. La syntaxe est la suivante :

`MARGINHEIGHT="n"`

Exemple :

The diagram illustrates a two-column frame layout. The left frame contains search results for 'Pierre Durand' at 'Rue-de-la-goutte-vesinette', with links to 'Détail sur la loi d'Ohm', 'Détail sur la loi de Joule', and 'Détail sur les Photos'. The right frame contains the title 'EXEMPLE DE LA LOI D'OHM', a link 'REVENIR AU DEBUT', and text explaining capacitor discharge. Margins are indicated as 20 pixels on all sides.

SCROLLING indique si la zone doit posséder une barre de défilement (**SCROLLING="yes"**) ou non (**SCROLLING="no"**), ou encore si ce choix est laissé au browser (**SCROLLING="auto"**). La syntaxe est la suivante :

SCROLLING="yes/no/auto"

NORESIZE n'a pas de valeur. Sa présence indique uniquement au browser qu'il doit empêcher toute modification de la taille d'une zone. Lorsque cet attribut n'est pas précisé, il est toujours possible de déformer une zone, en déplaçant sa frontière à l'aide du pointeur de la souris.

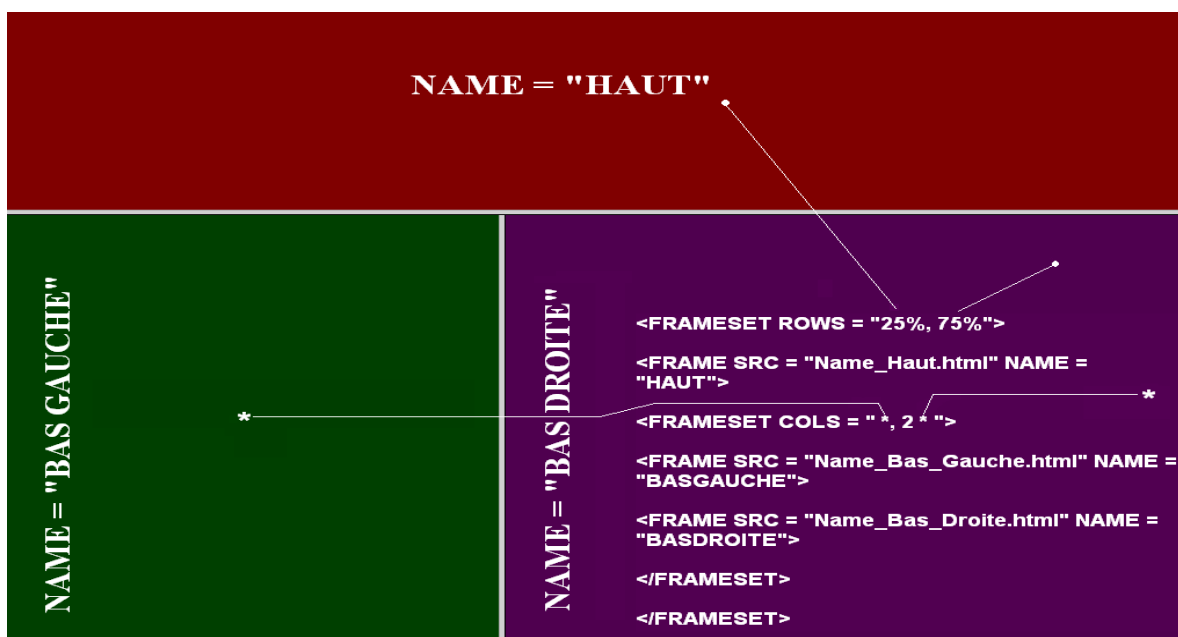
<NOFRAMES> :

<NOFRAMES> est une balise qui indique à tout browser ne soit incapable de gérer les frames le texte qu'il doit présenter à l'utilisateur à la place de la frame. ce texte est bien sûr ignoré par un browser capable de gérer les frames. En fait, le fonctionnement de cette balise est simple. Les browsers Web ne tiennent compte que des balises qu'ils connaissent. Un browser qui ne sait pas gérer les frames ignorera donc les balises <FRAMESET>, </FRAMESET>, <FRAME> et également <NOFRAMES> et </NOFRAMES>. Or, si on retire toutes ces balises d'un document contenant des frames, on voit qu'il ne reste plus que le texte qui était compris entre les balises <NOFRAMES> . . . </NOFRAMES>

Au contraire, un browser capable de gérer des frames interprétera les balises <NOFRAMES> . . . </NOFRAMES> et saura qu'il doit ignorer le texte compris entre ces balises.

Voici deux exemples illustrant la facilité d'utilisation des frames. On suppose que le fichier **Test_Frames.html** utilisé pour renseigner l'attribut **SRC** des balises <FRAME> est un fichier HTML classique.

Premier exemple ; ce qui donne ceci :



Exemple d'utilisation d'une page intitulée "Test_Frames_2.html", dont vous trouverez une partie du code HTML ci-dessous, correspondant uniquement aux frames ... :

Deuxième exemple ; ce qui donne ceci :

```
<HTML>
```

```
  <HEAD>
```

```
    <TITLE> 2ème Exemple d'un Test Frame </TITLE>
```

```
  </HEAD>
```

```
    <FRAMESET ROWS = "*, *, *">
```

```
      <FRAME SRC = "Name_Haut_1.html">
```

```
    <FRAMESET COLS = "3 *, *">
```

```
      <FRAME SRC = "Name_Sous_Haut_Gauche.html">
```

```
      <FRAME SRC = "Name_Droite_1.html">
```

```
    </FRAMESET>
```

```
  <FRAMESET COLS = " *, *, 2 *">
```

```
    <FRAME SRC = "Name_Gauche.html">
```

```
    <FRAME SRC = "Name_Sous_Gauche.html">
```

```
    <FRAME SRC = "Name_Droite.html">
```

```
  </FRAMESET>
```

```
  <NOFRAMES>
```

Ce document contient des frames et votre browser est incapable de les gérer ...

```
  </NOFRAMES>
```

```
  </FRAMESET>
```

```
</HTML>
```

Utilisation des frames :

Maintenant que nous avons vu en détail, comment créer des documents divisés en frames, nous allons voir comment créer des liens hypertexte utilisant ces frames.

L'attribut TARGET :

En plus des trois nouvelles balises décrites ci-dessus, un nouvel attribut a été créé. Il concerne les balises caractérisant des liens hypertexte, en particulier les balises **<A>** et **<FORM>**. Il s'agit de l'attribut **TARGET**, qui permet de préciser le nom de la zone qui doit recevoir le document correspondant au lien. Considérons par exemple le lien hypertexte suivant :

```
<A HREF="/dossier.html" TARGET="zonedossier">Voir le dossier</A>
```

L'attribut est là pour indiquer au browser qu'en réponse à un clic, il doit afficher le document **dossier.html** dans la frame dont le nom est **"zonedossier"** (c'est-à-dire la frame qui a été définie par la balise **<FRAME NAME="zonedossier"... >**). De la même façon, la soumission du formulaire défini par :

```
<FORM ACTION="Apercu_1_de_la_Frame_Gauche.html" METHOD="GET"  
TARGET="LISTE">
```

```
</FORM>
```

Affichera ses résultats dans la frame dont le nom est "LISTE".

On peut isoler trois cas pour décrire les valeurs possibles de l'attribut **TARGET**.

Dans le premier cas, l'attribut **TARGET** a une valeur qui correspond au nom d'une frame existante. Le document associé au lien hypertexte est alors affiché dans la frame en question.

Dans le deuxième cas, l'attribut **TARGET** a une valeur qui ne correspond à aucune des frames existantes. Le browser crée alors une nouvelle fenêtre (un nouveau browser) et y affiche le document. Cette nouvelle fenêtre est considérée comme une frame portant le nom de l'attribut **TARGET** à l'origine de sa création.

Dans le dernier cas, l'attribut **TARGET** prend une de ces quatre valeurs prédéfinies :

- **_blank**, qui est utilisé pour indiquer au browser qu'il doit créer une nouvelle fenêtre (c'est-à-dire un nouveau browser) afin d'afficher le document. Le nouveau browser est une frame qui ne porte pas de nom et ne peut donc pas être la cible d'un autre lien hypertexte ;
- **_self**, qui indique que le document sera chargé dans la même zone que celle dans laquelle se trouve le lien hypertexte. Il s'agit de la valeur par défaut lorsque l'attribut **TARGET** n'est pas renseigné ;

- **_top**, qui indique au browser de supprimer toutes les frames existantes et d'afficher le document de façon classique, en occupant toute la surface du browser ;
- **_parent**, qui indique au browser d'afficher le document, en occupant toute la surface de la zone dans laquelle a été affiché le document contenant le lien. Cette notion n'a de sens que lorsque les différents niveaux de frames proviennent de l'imbrication de documents et non de l'imbrication des balises **<FRAMESET>**. Dans ce cas, un lien situé dans la frame "**basdroite**" par exemple, et ayant un attribut **TARGET="_parent"** provoquera l'affichage du document correspondant dans la frame "**bas**". Le document recouvrera dans les deux frames "**basdroite**" et "**basgauche**". dans ce cas précis, le même résultat aurait été obtenu avec un attribut **TARGET="bas"**.

Afin de bien cerner la signification de ces valeurs prédéfinies (notamment celle de la valeur **_parent**, qui peut paraître un peu obscure à priori), voici un court exemple. Pour le comprendre plus vite, il est conseillé de créer sur votre serveur (ou plus simplement sur la machine où se trouve votre browser), les cinq fichiers qui composent cet exemple, puis d'ouvrir le document "**Plus_Bas.html**" avec un browser capable d'afficher des frames (Internet Explorer, Google Chrome, Opéra et bien d'autres encore).

Voici les cinq fichiers en question :

Descendre.html, **Plus_Haut.html**, **Plus_Bas.html**,
A_Droite.html et **A_Gauche.html**

Le fichier **Descendre.html** est un document constitué de deux frames horizontales de même taille. La frame **supérieure** affiche le document **Plus_Haut.html** et la frame **inférieure**, le document **Plus_Bas.html** :

```
<HTML>
  <HEAD>
    <TITLE> DESCENDRE </TITLE>
  </HEAD>
  <FRAMESET ROWS = "*", *">
    <FRAME SRC = "Plus_Haut.html">
    <FRAME SRC = "Plus_Bas.html">
  </FRAMESET>
</HTML>
```


Le Fichier **A_Gauche.html** est un document constitué de deux zones horizontales de même taille. La frame de haut affiche le document **Plus_Haut.html** et la frame du bas affiche le document **Plus_Bas.html**, en voici le code ci-après :

```
<HTML>

  <HEAD>

    <TITLE> ZONE HAUTE </TITLE>

  </HEAD>

  <FRAMESET ROWS = "*,*">

    <FRAME SRC = "Plus_Haut.html">

    <FRAME SRC = "Plus_Bas.html">

  </FRAMESET>

</HTML>
```

Le dernier fichier, **Plus_Haut.html**, est un peu particulier. Il n'apparaît dans aucun des liens hypertexte des autres fichiers. Il est utilisé uniquement pour remplir une sous-frame du document **Descendre.html** ou **A_droite.html** ou encore **A_Gauche.html**. Il s'agit en quelque sorte d'un sous-document du document **Descendre.html** ou **A_Droite.html** ou encore celui **A_Gauche.html**. Pour un lien du fichier **Plus_Haut.html**, l'attribut **TARGET="_parent"** a alors un sens. Il indique au browser d'utiliser la frame du sur-document (dans notre cas, il s'agira de la frame contenant le document **Descendre.html** ou **A_Droite.html** ou encore **A_Gauche.html**) pour afficher le document correspondant aux liens hypertexte. On remplace ainsi le document **Descendre.html** ou **A_Droite.html** ou celui de **A_Gauche.html** par un autre document.

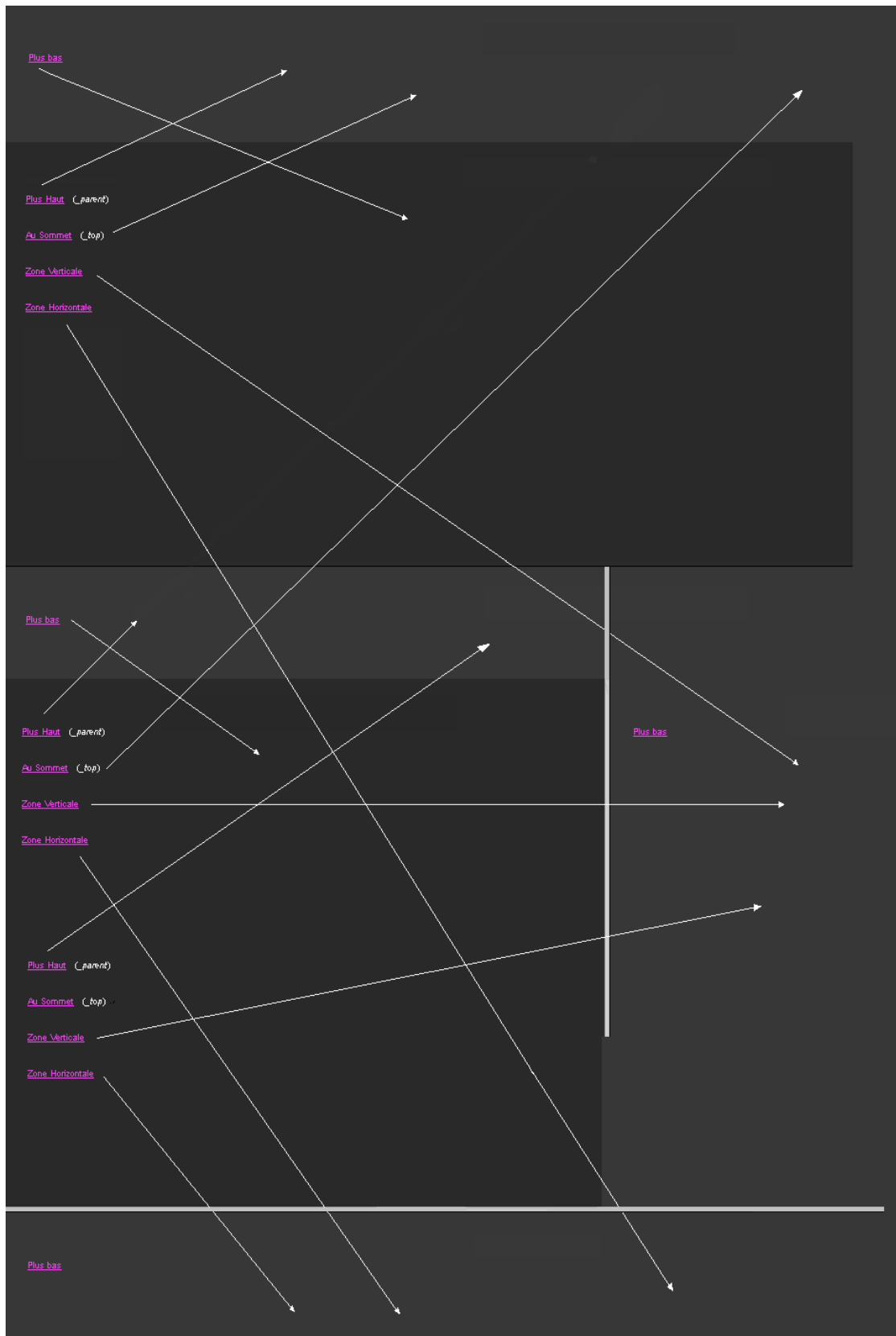
Examinons maintenant en détail les quatre liens du fichier **Plus_Haut.html** comme ci-après :

1°) - Le premier, **Plus Haut**, permet de remonter d'un cran dans la hiérarchie des zones. En effet, il remplace, grâce à l'attribut **TARGET="_parent"**, un document composé de deux frames (**Descendre.html** ou **A_Droite.html** ou encore celui de **A_Gauche.html** par un document classique (**Plus_Bas.html**) ;

2°) - Le deuxième, **Au Sommet**, permet d'afficher le document **Plus_Bas.html**, en occupant tout l'espace de la fenêtre du browser (**TARGET="_top"**). On retombe ainsi dans un affichage classique, sans frame ;

3°) - Le troisième, **Zone Verticale**, permet de remplacer le document **Descendre.html** par le document **A_Droite.html** (toujours à l'aide de l'attribut **TARGET="_parent"**). Les deux documents ayant le même nombre de

Voici quelques enchaînements obtenus à partir de cet exemple. Les flèches indiquent l'effet d'un clic sur un lien hypertexte. Pour plus de lisibilité, toutes les possibilités n'apparaissent pas sur le schéma ci-dessous :



Les frames et JavaScript :

La frame dispose des mêmes caractéristiques que l'**objet window** que l'on verra par la suite de nos leçons (la **Classe Window**).

Exemple : Choisir dans quelle frame l'image apparaîtra

L'exemple suivant montre comment on change l'image d'un point à l'autre, en cliquant sur **Haut** ou sur **Bas**, on change le **TARGET d'un lien** (ici une frame). Lorsque l'on effectue ce choix, rien ne se passe dans les fenêtres ; c'est seulement lorsque l'on clique sur l'**image** que celle-ci apparaît dans la zone choisie.

Le fichier INDEX.HTML (découpage du browser, fichier appelé que l'on cliquera sur ce dernier) :

Le menu est chargé dans la frame de gauche (**Frame_Menu**) et des fichiers vides sont chargés dans les deux autres frames (**Frame_Haut** et **Frame_Bas**) dont nous évoquerons les codes HTML ci-après ainsi que leurs fonctions :

Fichier INDEX.HTML, voici le code suivant :

```
<HTML>
```

```
  <HEAD>
```

```
    <TITLE> D'placement d'une Image </TITLE>
```

```
  </HEAD>
```

```
  <!-- Si une frame n'est pas chargée avec un fichier,
```

il semble que le nommage de la frame ne se fasse pas, car elle ne peut pas être adressée.

En conséquence même, si la frame doit être vide à l'origine, il est nécessaire de la charger avec un document vide (Vide.html) -->

```
  <FRAMESET COLS = " *, 2 * ">
```

```
    <FRAME NAME = "Frame_Menu" SRC = "Menu.html">
```

```
  <FRAMESET ROWS = " *, * ">
```

```
    <FRAME NAME = "Frame_Haut" SRC = "Vide.html">
```

```
    <FRAME NAME = "Frame_Bas" SRC = "Vide.html">
```


</FRAMESET>

</FRAMESET>

</HTML>

Le fichier MENU.HTML (création du menu, appelé par INDEX.HTML) :

<HTML>

<HEAD>

<!-- Cet exemple montre :

1 - l'adressage des liens d'un document, en tableau :

links[0] = Haut

links[1] = Bas

links[2] = L'image

2 - La possibilité de changer le TARGET D'UN LIEN :

a - Si on clique sur l'affichage sans avoir cliqué sur le lien hypertexte Haut

ou sur le lien Bas de l'Image, elle remplace le MENU.

b - Si on clique préalablement sur le lien hypertexte Haut de l'Image, elle vient

dans la frame du Haut à droite.

c - Si on clique préalablement sur le lien hypertexte Bas de l'Image, elle vient

dans la frame du Bas à droite.

3 - L'adressage des frames -->

<TITLE> MENU </TITLE>

<SCRIPT>

function ChangeTarget (NewTarget) {

document.links[2].target=NewTarget;

}

</SCRIPT>

</HEAD>

<BODY BGCOLOR="#292929" LINK="#FF0000" ALINK="#00FFFF" VLINK="#FF40FF">

<CENTER> <i> Où voulez-vous mettre l'image ? </i> <p>

En Haut de la Frame ou

Vers le Bas <p>

 <i> l'image </i>

 </p>

<p> <HR>

 [Recommencer] </p>

</CENTER>

</BODY>

</HTML>

Le fichier IMAGE.HTML (appelé lorsque l'on clique sur " l'image ") :

<HTML>

<HEAD>

<TITLE> IMAGE </TITLE>

</HEAD>

 <CENTER>

<BODY BGCOLOR="#454545">

 [Retour au Sommaire HTML]

 [Revenir en arrière]


```

<BR> <BR> <BR>
```

```

```

```
</CENTER>
```

```
</BODY>
```

```
</HTML>
```

Le fichier VIDE.HTML (appelé par INDEX.HTML) :

```
<HTML>
```

```
<HEAD>
```

```
<TITLE> DOCUMENT VIDE </TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<!-- Si une frame n'est pas charg&eacute;e avec un fichier,
```

```
il semble que le nommage de la frame ne se fasse pas,
```

```
car elle ne peut pas &ecirc;tre adressée -->
```

```
</BODY>
```

```
</HTML>
```

Nous terminons ainsi les exemples des frames, nous allons résumer cette leçon.

Résumé :

Le découpage d'une fenêtre, en frames est réalisé à l'aide d'un document HTML spécial construit à partir de trois nouvelles balises.

<FRAMESET> . . . **</FRAMESET>** va permettre de préciser la géométrie des différentes frames. Il s'agit en quelque sorte d'une balise de déclaration. Elle possède deux attributs :

- **ROWS="hauteur_1, hauteur_2, ... hauteur_N"** définit la hauteur des différentes frames en cas de découpage horizontal ;
- **COLS="largeur_1, largeur_2, ... largeur_N"** définit la largeur des différentes frames en cas de découpage vertical.

<FRAME> est la balise de définition des frames déclarées à l'aide d'une balise **<FRAMESET>**. Elle possède six attributs :

- **SRC="URL"** indique le document à afficher dans la frame ;
- **NAME="nom"** indique le nom de la frame de telle sorte que cette frame puisse être utilisée comme cible d'un lien hypertexte ;
- **MARGINWIDTH="n"** définit l'espace entre le document et les frontières verticales de la frame ;
- **MARGINHEIGHT="n"** définit l'espace entre le document et les frontières horizontales de la frame ;
- **SCROLLING="yes/no/auto"** indique si la frame doit ou non posséder une barre de défilement ;
- **NORESIZE** indique qu'il est impossible à l'utilisateur de redimensionner la frame.

<NOFRAMES> . . . </NOFRAMES> est utilisé pour indiquer le texte que doivent afficher les browsers incapables de gérer les frames.

Enfin, un nouvel attribut a été créé de façon à pouvoir indiquer dans quelle frame doit être affiché le document associé à un lien hypertexte :

- **TARGET="nom_de_la_frame"**

Ceci donnera par exemple :

** . . . **

<FORM ACTION="URL" TARGET="nom_de_la_frame"> . . . </FORM>

<AREA HREF="...">

Exercice HTML : Frames, fenêtres de navigateur, iframes

Thème

Réalisation de frames HTML (partage de la fenêtre du browser Web en sous-fenêtres), et manipulation de fenêtres de navigateur.

La technique des frames (cadres) est principalement utilisée lorsque l'on veut mettre en place des barres de navigation ou lorsque l'on veut réaliser des mises en page complexes.

Remarque : les frames HTML sont supportés à partir des versions de browsers Netscape 2 et IExplorer 3

Attention cependant aux **limites** des frames :

1. l'URL apparaissant au haut de la fenêtre est celui du frameset et reste constant lors de la navigation à travers les frames ; c'est donc celui-là qui est, le cas échéant, mémorisé dans les bookmarks de l'utilisateur
2. si l'utilisateur mémorise, dans ses bookmarks (favoris) l'URL d'un frame (après avoir ouvert ce frame dans une nouvelle fenêtre de browser), en revenant ultérieurement sur ce signet l'utilisateur perdra le contexte (c-à-d. les autres frames) nécessaire à une bonne compréhension du contenu
3. plus grave encore : la plupart des moteurs de recherche refusent d'indexer les frames en raison du point 2 ci-dessus (ils s'arrêtent au niveau du frameset, ou descendent parfois juste dans les pages correspondant aux URLs se trouvant dans le bloc `<NOFRAME>...</NOFRAME>`) ; les parties d'un site construites sur la base de frames resteront donc invisibles du moteur de recherche (non indexées)

Comme alternative aux frames pour faire usage, dans un ensemble de pages Web, de code HTML commun à ces pages (barres de navigation...), on utilisera la technique du **Server Side Include** (SSI) ou l'un des nombreux **systèmes de templating** sous PHP (FastTemplate, PHPLib, Smarty...).

Balises et JavaScript relatifs aux frames

- Début frameset (définition du nombre de frames, structurées en lignes ou en colonnes avec leurs tailles respectives) :

```
<FRAMESET ROWS="pixels ou %, ..." | COLS="pixels ou %, ..."
        BORDER="nb.pixels" FRAMEBORDER="0|1">
```
- Pour chaque frame, spécification du document correspondant au frame :

```
<FRAME SRC="URL" NAME="target name" SCROLLING="YES|NO"
{NORESIZE}
        MARGINWIDTH="pixels"           MARGINHEIGHT="pixels">
```

(l'URL peut être celle d'une page HTML ou d'une image !)
ou, pour browser (ou robot !) ne supportant pas les frames :

```
<NOFRAME> texte à afficher... </NOFRAME>
```
- Fin frameset :

```
</FRAMESET>
```
- Lien provoquant mise à jour d'un frame :

```
<A HREF="URL" TARGET="target name">... </A>
```

(pour mettre à jour plusieurs frames à la fois, voir la technique JavaScript décrite plus bas)
ou formulaire provoquant mise à jour d'un frame :

```
<FORM ACTION="URL appli CGI" TARGET="target name"
METHOD=...>... </A>
```
- Pour définir le frame qui sera utilisé par défaut comme cible par tous les liens définis dans une page, plutôt que de mettre des `TARGET="target name"` dans tous les liens on peut définir, dans l'en-tête de la page (c-à-d. dans le bloc `<HEAD>...</HEAD>`) :

```
<BASE TARGET="target name">
```

Remarques

- on peut imbriquer des framesets les uns dans les autres (c'est du reste ce que l'on doit faire pour obtenir un partitionnement vertical ET horizontal de la fenêtre du browser)
- dans une page HTML de définition de framesets, le bloc `<FRAMESET>...</FRAMESET>` se substitue au bloc `<BODY>...</BODY>` qui n'existe pas
- pour l'attribut `ROWS` ou `COLS`, on spécifie la hauteur ou la largeur (en pixels ou %) des différents frames (il doit y avoir autant de valeurs, séparées par des virgules, que de `FRAME` définis dans la structure) ; on peut aussi mettre, en guise de valeur, le caractère `*` pour l'un des frames, ce qui indique au browser d'allouer la place restante

- l'attribut **FRAMEBORDER** sert à spécifier si l'on veut des bordures en relief (1 = défaut) ou non (0)
- les attributs **MARGINWIDTH** et **MARGINHEIGHT** permettent de définir des marges horizontales (gauche/droite) et verticales (haut/bas)
- l'attribut **TARGET** sert à définir le frame dans lequel doit s'afficher la page (par défaut dans le frame courant) ; avec la valeur "**_top**", affichage dans l'entier de la fenêtre du browser (suppression des frames) ; avec la valeur "**_blank**" (ou un nom de frame qui n'est pas connu du browser), affichage dans une nouvelle fenêtre de browser

Usage de JavaScript dans la manipulation des frames de fenêtres

A) Pour mettre à jour **plusieurs frames** à la fois lorsque l'on active un lien, on est obligé de recourir à JavaScript. Dans la page où est défini le lien :

1) définir le JavaScript suivant (dans le bloc `<HEAD>...</HEAD>`) :

```
<SCRIPT LANGUAGE="JavaScript">
  <!--
      function TwoFrames(URL1,Fr1Name,URL2,Fr2Name) {
          parent.frames[Fr1Name].location.href=URL1;
          parent.frames[Fr2Name].location.href=URL2;
      }
  //-->
</SCRIPT>
```

2) puis définir ainsi le lien qui devra charger les pages "page1.html" et "page2.html" respectivement dans les frames définis par les cibles "frame1" et "frame2" :

```
<A
HREF="javascript:TwoFrames('page1.html','frame1','page2.html',
'frame2');">lien</A>
```

B) Pour ouvrir de **nouvelles fenêtres** de navigateur en jouant sur les **propriétés** de celles-ci (avec ou sans barre d'icônes, de menus, d'ascenseurs, etc...) ou les **refermer**, on doit également faire appel à JavaScript

1) si l'on définit les fonctions JavaScript suivantes (dans le bloc `<HEAD>...</HEAD>`) :

```
<SCRIPT LANGUAGE="JavaScript">
  <!--
      function ouvrir_fenetre(url,target_name,param) {
          nomFenetre = window.open(url,target_name,param) ;
      }
  </SCRIPT>
```

```

function fermer_fenetre() {
    if (nomFenetre) { nomFenetre.close() ; }
}
//-->
</SCRIPT>

```

2) on peut alors faire ouvrir une fenêtre sans aucun attribut (palette de navigation) à partir du lien hypertexte suivant (attention à bien englober tous les paramètres, depuis **toolbar** jusqu'à **screenY** entre deux apostrophes !):

```

<A HREF="Javascript:ouvrir_fenetre('URL','palette_navig',
'toolbar=no,location=no,directories=no,status=no,menubar=no,sc
rollbars=no,
resizable=no,copyhistory=no,width=250,height=400,screenX=20,sc
reenY=100' )">
    ouvrir </A>

```

3) et l'on peut faire refermer cette fenêtre à partir du lien hypertexte suivant :

```

<A HREF="Javascript:fermer_fenetre()"> fermer </A>

```

Les frames en-ligne (IFRAME)

Standardisée sous **HTML 4.01**, la notion de "frame en-ligne" (inline frame, iframe) est applicable depuis les versions de navigateur Mozilla 1, Netscape 6, IE 4. Un peu à la façon de la balise **OBJECT** pour l'insertion d'objets en-ligne dans une page, cette technique implémentée *coté client* permet d'inclure une ressource Web (page HTML statique ou dynamique...) dans un bloc à l'intérieur d'une page (analogie à la technique *coté serveur* **SSI**, Server Side Include).

- **<IFRAME**

```

SRC="URL"      WIDTH="largeur"      HEIGHT="hauteur"
MARGINWIDTH="marge horizontale"  MARGINHEIGHT="marge
verticale"
ALIGN="top|middle|bottom|left|right"
SCROLLING="yes|no|auto"      FRAMEBORDER="0|1">
code alternatif que verront les navigateurs n'implémentant pas les
iframes...
</IFRAME>

```

Références

- Netscape "**Frames Syntax**"

Corrigé de l'exercice

Fichier "frameset.html"

```
<HTML>
<HEAD>
  <TITLE>Exercice HTML : Définition d'un frameset</TITLE>
</HEAD>
<FRAMESET ROWS="*,30" BORDER="2">

  <FRAMESET COLS="160,*">
    <FRAME SRC="frame_navig.html" NAME="navig">
    <FRAME SRC="frame_page.html" NAME="page">
  </FRAMESET>

  <FRAME SRC="frame_pied.html" NORESIZE SCROLLING="no"
MARGINHEIGHT="1" MARGINWIDTH="1"
  NAME="pied">

  <NOFRAME>
    <B>Attention</B> : ce document contient des "frames" que
votre navigateur WWW n'est
    pas capable d'interpréter. Utilisez Netscape Navigator
(vers. 2 ou sup.) ou
    Microsoft Internet Explorer (vers. 3 ou sup.) ; sinon,
    <A HREF="frame_navig.html"><B>suivez ce lien</B></A>.
  </NOFRAME>

</FRAMESET>
</HTML>
```

Fichier "frame_navig.html"

```
<HTML>
<HEAD>
  <TITLE>Exercice HTML : frame de navigation</TITLE>
</HEAD>
<BODY BGCOLOR="YELLOW">

  <CENTER>
  <B>Frame de navigation</B>
  </CENTER>

  <P><HR WIDTH="60%" SIZE="1"></P>
```

```
<UL>
<LI><A HREF=111.html TARGET="page">111</A>
<LI><A HREF=222.html TARGET="page">222</A>
  <P>
<LI><A HREF=frame_back.html TARGET="_top">back</A>
</UL>
<P><HR WIDTH="60%" SIZE="1"><P>

</BODY>
</HTML>
```

Fichier "frame_page.html"

```
<HTML>
<HEAD>
  <TITLE>Exercice HTML : frame de bienvenue</TITLE>
</HEAD>
<BODY BGCOLOR="WHITE">

<CENTER><B>Frame de bienvenue</B></CENTER>

</BODY>
</HTML>
```

Fichier "frame_pied.html"

```
<HTML>
<HEAD>
  <TITLE>Exercice HTML : frame de pied de page</TITLE>
</HEAD>
<BODY BGCOLOR="GRAY">

<CENTER><B>Pied de page</B></CENTER>

</BODY>
</HTML>
```