



# Formation HTML

Version 0.1

Eric BERTHOMIER

15 mars 2006

# Table des matières

<b>Table des matières</b>	<b>1</b>
<b>1 HTML : Introduction</b>	<b>3</b>
1.1 Environnement	3
1.1.1 Système d'exploitation	3
1.1.2 Navigateur Web	3
1.1.3 Programmation Web	4
1.2 Principe d'une connexion à Internet	4
1.3 Principe de fonctionnement d'une page Web	4
1.3.1 Application Pratique	5
1.3.2 Complément d'information	5
1.4 Ma première page Web	6
1.4.1 Les balises	6
1.4.2 Le squelette	6
1.4.3 Explication des différentes parties	6
1.4.4 Application Pratique : Balises de Titres	7
<b>2 HTML : Première page</b>	<b>8</b>
2.1 Le Texte	8
2.1.1 Une police avec accents	8
2.1.2 Mise en forme simple	8
2.1.3 Le codage des couleurs en HTML	9
2.1.4 Taille des fontes	10
2.2 Mise en forme de la page	10
2.2.1 Retour à la ligne	11
2.3 Mise en forme d'un paragraphe	11
2.4 Mise en forme d'un bloc de texte	12
2.5 Insertion d'images	13
2.6 Astuce	13
<b>3 HTML : Première mise en ligne</b>	<b>14</b>
3.1 Mise en ligne d'une page Web	14
3.1.1 Étape 1 : Fournisseur d'espace	14
3.1.2 Étape 2 : Connexion et transfert	14
<b>4 HTML : Les ancrs</b>	<b>17</b>
4.1 Ancre à l'intérieur d'un même document	17
4.2 Navigation relative et absolue	18
4.3 Ancre entre les documents	18
4.4 Ancre sur un document à une position précise	18

*TABLE DES MATIÈRES*

2

**Listings**

**19**

# Chapitre 1

## HTML : Introduction

### 1.1 Environnement

#### 1.1.1 Système d'exploitation

L'environnement utilisé pour ce cours est un environnement GNU Linux Debian avec une interface graphique Gnome. Petite explication sur ces quelques termes :

**GNU** (Anglais : Gnu's Not Unix). Association de programmeurs pour l'écriture et la diffusion de logiciels libres (pour Linux par exemple).

**LINUX** Linux est une version d'UNIX gratuite et librement diffusable. C'est donc un système d'exploitation comme peut l'être Windows ou Mac OS.

**Système d'Exploitation (OS)** Programme assurant la gestion de l'ordinateur et de ses périphériques.

**Debian** Distribution de Linux. Il en existe une petite centaine, celle-ci est la plus utilisée par les professionnels.

**Gnome** Gnome est, à l'instar de KDE (K Desktop Environment), une interface graphique pour linux et autres unices (Solaris par exemple).

A la grande différence de Windows, il n'existe pas un seul Linux et il n'existe pas une seule interface graphique pour le système Linux ; il a donc été choisi une interface graphique proche de celle connu par la plupart des utilisateurs de l'informatique : Gnome.

Une autre différence importante entre le système Windows et Linux est la possibilité d'avoir plusieurs bureaux. Les petits carrés en bas à droite permettent d'accéder à ces différents bureaux. Un bureau est l'ensemble des éléments d'une interface graphique. Par ce biais, il est donc possible de lire une page web en plein écran sur un bureau et de lire ces mails en plein écran sur un autre et ainsi de suite ... sans à aucun moment à avoir à réduire sa ou ses fenêtres.

Les différences ne s'arrêtent donc pas là mais si vous désirez en savoir plus, n'hésitez pas à pousser la porte d'une LUG (Linux User Group).

#### 1.1.2 Navigateur Web

Le logiciel qui vous permet de surfer sur Internet et que communément nous associons à Internet Explorer est appelé un *navigateur web*. En fait, Internet Explorer, est un navigateur Web, mais pas le seul. Il existe notamment, fonctionnant aussi sous Windows, Mozilla Firefox<sup>1</sup>, c'est ce navigateur que nous utiliserons sous Linux mais vous pourriez aussi utiliser Konqueror, Netscape, Opéra ou bien d'autres navigateurs.

---

<sup>1</sup><http://www.mozilla-europe.org/fr/products/firefox/>

### 1.1.3 Programmation Web

Il n'y a en fait besoin que d'un éditeur de texte (Notepad sous Windows par exemple) pour faire du HTML. Les outils fournis par les éditeurs ne sont là que pour vous aider et masquer les difficultés.

Pour ce cours, nous utiliserons un outil d'aide à la saisie nommé **BlueFish**<sup>2</sup>. Celui-ci n'est pas disponible sous Windows, mais il vous est possible d'utiliser sous Windows **Context**<sup>3</sup> ou encore **Arachnophilia** avec lequel j'avais développé mon premier site Web<sup>4</sup>.

## 1.2 Principe d'une connexion à Internet

Avant toute chose, l'architecture utilisée pour notre cours dans la salle informatique de Renac n'est pas la même que chez vous. Il nous faut donc comprendre les différences et notamment le principe général d'une connexion Internet que ce soit en ADSL ou en RTC.

Le principe de base est donc le suivant, votre modem compose un numéro (RTC) ou utilise directement votre ligne téléphonique (ADSL) pour joindre un des serveurs de votre FAI ( Fournisseur d'Accès Internet).

Une fois contacté, celui-ci vous demande un `login` et un `mot de passe` permettant de vous identifier en tant que membre de leur clientèle. Ces paramètres sont le plus souvent enregistrés une fois et une seule dans votre modem (dans le cas d'un modem routeur ADSL) ou dans votre ordinateur.

Que se passe t'il ensuite ?

Eh bien, votre FAI va vous fournir (à votre ordinateur) une adresse IP. Cette adresse IP vous permet de vous connecter au grand réseau de l'Internet. Vous me direz que vous n'utilisez pas de carte réseau pour vous connecter à Internet, mais c'est du pareil au même, d'ailleurs, si vous regarder dans les paramètres réseaux de Windows, vous pourrez y découvrir une connexion distante dans le cas d'une utilisation par Modem.

Qu'est ce que signifie "être sur le grand réseau de l'Internet" ?

Eh bien, cela signifie que vous pouvez vous adresser au monde entier via l'Internet mais aussi que le monde entier peut s'adresser à vous ... d'où l'utilité d'un Firewall pour cacher sa présence des méchants Black Hat<sup>5</sup>

Et dans le cas de la salle informatique de Renac ?

La salle est équipée d'un modem routeur ADSL qui comme un grand nous authentifie auprès du FAI qui gentiment nous redonne une adresse IP. Nous nous connectons à Internet en passant par ce point d'accès, on appelle ceci une passerelle car c'est elle qui fait la liaison entre notre groupe d'ordinateur et l'Internet. Sur la toile (l'Internet), nous sommes vus comme une seule et unique entité.

## 1.3 Principe de fonctionnement d'une page Web

Pour pouvoir accéder à une page Web, la première chose qu'il est nécessaire d'avoir est un navigateur, sans celui-ci, il est possible de voir certaines choses mais pas de manière très agréable. Une fois notre navigateur Web ouvert, il nous faut taper l'adresse d'un site, par exemple `http://www.google.fr`. Que se passe t'il à partir de ce moment ?

1. L'adresse du site est transformée en une adresse IP (notée @IP par la suite). Ceci se fait par l'intermédiaire d'un DNS (Domain Name Server). Dans notre cas, l'@IP d'un des serveurs correspondant à `http://www.google.fr` est `64.233.183.99`. Cette adresse permet de désigner de manière

---

<sup>2</sup><http://bluefish.openoffice.nl>

<sup>3</sup><http://www.context.cx>

<sup>4</sup><http://www.arachnoid.com/arachnophilia/>

<sup>5</sup>Hacker du côté obscur de la force informatique

unique un ordinateur sur le réseau Internet.

Comment avoir cette adresse ?

Il suffit de taper dans une fenêtre console la commande `ping www.google.fr`

2. Donc ce que nous avons tapé précédemment correspond donc à `http://64.233.283.99`. Le navigateur va donc demander une page Web au serveur. Deux cas se présentent :

**1er cas : Aucun fichier n'est spécifié** Dans ce cas, le serveur va renvoyer à notre navigateur, le fichier par défaut qui lui a été indiqué dans sa configuration, dans la plupart des cas, ce fichier est `index.html`.

**2nd cas : Un fichier est spécifié** Par exemple, j'ai saisi `http://www.google.fr/toto.html`, dans ce cas, le serveur va renvoyer à notre navigateur, le fichier correspondant si celui-ci existe ou une erreur si celui-ci n'existe pas.

3. Une fois que notre navigateur a reçu la page, il se déconnecte du serveur et interprète ce qui lui a été envoyé. Mais qu'a t'il reçu ?

Pour le savoir, il suffit de cliquer `Affichage => Code source de la page`. Ceci est le code source HTML.

### 1.3.1 Application Pratique

- Se connecter sur `http://eric.berthomier.free.fr/courshtml` puis sur `http://eric.berthomier.free.fr/courshtml/index.html`, comparer.
- Une fois la page reçue, afficher le code source de celle-ci

### 1.3.2 Complément d'information

Le fonctionnement décrit ci-dessus est en fait incomplet, nous allons le détailler de manière plus précise. Une fois que votre navigateur a téléchargé le fichier fourni par le serveur, celui-ci le met en cache sur votre disque dur. Par la suite, le navigateur regarde d'abord dans son cache pour savoir si la page est toujours d'actualité, si c'est le cas, il vous affiche ce qu'il a dans son cache, dans le cas contraire, il demande au serveur de lui renvoyer la page. Ainsi, nous économisons un trafic inutile.

Il est possible malgré tout de forcer le navigateur à recharger une page Web, pour se faire, il faut cliquer sur le bouton `Rafraîchir` de celui-ci.

### Sécurité

La mise en cache des pages Web est très pratique mais permet aussi à toute personne passant derrière vous d'aller consulter ce que vous avez été voir. Ce même problème se constate avec l'enregistrement des mots de passe ou l'acceptation des cookies pour la navigation.

Si vous désirez épurer votre connexion, il suffit de supprimer ce cache, pour se faire :

- Sous Mozilla Firefox : `Editions => Préférences` puis le bouton `Effacer tout` ou le bouton `Effacer` pour les éléments concernés.
- Sous Internet Explorer : `Outils => Options Internet` puis les boutons `Supprimer les fichiers` et/ou le bouton `Supprimer les cookies`. Pour les consulter, ouvrir Internet Explorer, Menu Outils, Options Internet :
  - Cliquer sur `Paramètres`
  - `Afficher les fichiers` permettra de voir le cache et les cookies
  - `Afficher les objets` permettra de voir les activeX et autre complément d'IE.

## 1.4 Ma première page Web

Pour nous aider dans la réalisation de notre première page Web, nous allons utiliser l'éditeur Web : BlueFish. Celui-ci va nous permettre de disposer de colorisation syntaxique et de nous insérer les commandes à notre place mais ne fera pas bien sûr notre site à notre place.

### 1.4.1 Les balises

Le principe de base d'une page HTML réside dans l'utilisation des balises, toute balise ouverte doit être refermée (à part quelques exceptions).

Listing 1.1 – balise

```
1 <balise ouverte>
  </balise fermée>
```

Une balise s'ouvre en la mettant entre < et >, par exemple <b>.

Une balise se ferme en fermant le 1er < par un /, par exemple </b>.

Listing 1.2 – balise.html

```
<b>Ce texte est en gras</b>
```

### 1.4.2 Le squelette

À l'image de notre corps, HTML possède un squelette qui lui permet d'héberger les données de manière structurée.

Listing 1.3 – squelette.html

```
4 <html>
  <head>
    <title>
      Titre
    </title>
  </head>
  <body>
    Corps de la page Web
  </body>
9 </html>
```

### 1.4.3 Explication des différentes parties

**html** : Cette section indique que nous allons écrire du HTML, en effet, le HTML n'est pas le seul langage connu par les navigateurs, il existe notamment le JavaScript.

**head** : Cette section définit l'entête du fichier HTML, bien que vide pour l'instant celle-ci est indispensable pour référencer votre site sur un moteur de recherche.

**title** : Le titre (ici `Titre`) qui apparaît dans la barre haute du navigateur, ce n'est pas le titre affiché dans la page mais au niveau du navigateur lui-même.

**body** : Cette section définit le corps du fichier HTML, c'est à dire son contenu. C'est ce qui va apparaître dans le navigateur.

### 1.4.4 Application Pratique : Balises de Titres

Nous y voici enfin, la pratique ... Ouvrez un éditeur de texte ou BlueFish à votre guise. Afin de rendre votre texte plus joli, voici votre première balise de formatage de texte : hX ou X est un numéro entre 1 et 6.

La balise <hX> permet d'afficher un texte sous la forme d'un titre, :

– h1 représentera le plus haut niveau de titre, on peut parler de chapitre.

– h6 représentera le plus bas niveau de titre, on pourrait alors parler de paragraphe.

Voici un petit exemple illustrant ces propos.

Listing 1.4 – titres.html

```
<html>
  <head>
    <title>
      Différents niveaux de titre
5    </title>
    <head>
  <body>
    Voici les différents niveaux de titre possibles en HTML.
10  <h1>H1</h1>
    <h2>H2</h2>
    <h3>H3</h3>
    <h4>H4</h4>
    <h5>H5</h5>
    <h6>H6</h6>
15  </body>
</html>
```

Pour le visualiser, taper ce texte dans votre éditeur et enregistrer sous le nom "titres.html" par exemple. Ouvrez votre navigateur, aller dans le menu Fichier -> Ouvrir un fichier et sélectionner votre fichier.

Si des erreurs apparaissent ou si vous modifier votre code, il sera nécessaire d'appuyer sur le bouton Rafraichir du navigateur pour prendre en charge les modifications.

## Chapitre 2

# HTML : Première page

Maintenant que nous savons créer un squelette de page Web et y insérer différents niveaux de titre, passons à la mise en forme du texte et donc à l'insertion de texte dans la page Web.

### 2.1 Le Texte

Un texte est avant tout représenté par une police de caractères. Cette police de caractères est déclarée comme une fonte (`font` en anglais) et il est possible de changer celle-ci au gré de nos envies. Par contre, l'une des premières contraintes qui va apparaître est que nous sommes français et que la police de caractères que nous utilisons couramment possède des accents.

#### 2.1.1 Une police avec accents

Pour ajouter les accents à notre page Web, il faut indiquer au navigateur que nous utilisons une police en relation avec notre langue. Pour se faire, dans BlueFish, aller dans Document – > Encodage et sélectionner ISO-8859-15.

Ceci devrait (suivant la version) insérer la balise Meta suivante dans la section head :

```
<meta http-equiv="content-type" content="text/html; charset=ISO-8859-15" />
```

Une autre technique consiste à encoder chaque accent en HTML. Par exemple, é s'écrira `&eacute;`.

Vous comprenez aisément que ceci est contraignant mais à contrario, cette méthode permet la création de majuscules accentuées et ainsi É s'écrira de la façon suivante : `&Eacute;`.

#### 2.1.2 Mise en forme simple

Balise	Signification	Exemple
<code>&lt;B&gt;</code>	Bold : gras	<code>&lt;B&gt; Ce texte est en gras &lt;/B&gt;</code>
<code>&lt;I&gt;</code>	Italic : italique	<code>&lt;I&gt; Ce texte est en italique &lt;/I&gt;</code>
<code>&lt;U&gt;</code>	Underline : souligné	<code>&lt;U&gt; Ce texte est souligné &lt;/U&gt;</code>

Ces balises servent à mettre en forme du texte. Il ne faut surtout pas les utiliser (bien que ce soit possible) avec les balises de titres `hX`. En effet, si nous désirons modifier le comportement d'une balise de titre, il suffira de surcharger sa définition comme nous le verrons plus tard (CSS).

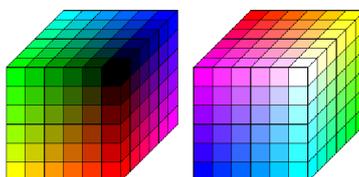
### 2.1.3 Le codage des couleurs en HTML

#### Rappel sur l'hexadécimal

L'hexadécimal ou base 16, est un système arithmétique basé sur 16 valeurs. Ces 16 valeurs sont :

Base 10	Base 16
0 à 9	0 à 9
10	A
11	B
12	C
13	D
14	E
15	F

Source : <http://www.pescadoo.net/html/colors.html>



Les couleurs sont codées sur 3 octets, généralement notés en hexadécimal sous la forme : RRGGBB.

**RR** = valeur hexadécimale de l'octet codant la primaire rouge

**GG** = valeur hexadécimale de l'octet codant la primaire verte

**BB** = valeur hexadécimale de l'octet codant la primaire bleue

**Exemple :** COLOR=#33ffcc -> RR=#33, GG=#ff, BB=#cc.

En principe, ce codage permet de définir 16777216 couleurs (256 x 256 x 256), soit plus que l'oeil humain ne peut en discerner. En pratique, on se limite aux couleurs qui peuvent être représentées sur n'importe quel écran et n'importe quel système.

La syntaxe utilisée pour mettre en couleur une chaîne de caractères est la suivante :

<FONT color='red'>Rouge</FONT>		Rouge
<FONT color=#FF0000>Rouge</FONT>		Rouge

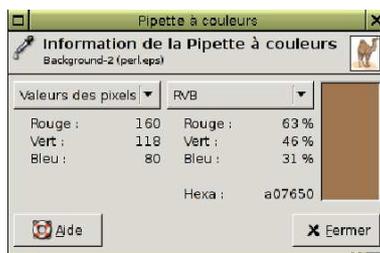
Toutes les couleurs peuvent ainsi être définies par un nom ou par leur encodage RGB en hexadécimal (**Attention à ne pas oublier le #!**).

#### Quelques couleurs

Rouge		red		#FF0000		Blanc		white		#FFFFFF
Vert		green		#00FF00		Noir		black		#000000
Bleu		blue		#0000FF		Gris		gray		#BEBEBE

### Application courante

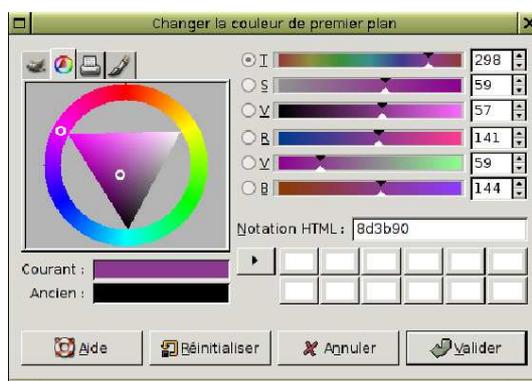
Les couleurs sont souvent appliquées pour rappeler l'image d'un logo ou pour tenir compte d'une charte graphique. Si vous ne connaissez pas les couleurs que vous pouvez utiliser, une manière simple de le savoir consiste à ouvrir votre image sous GIMP (<http://www.gimp-fr.org/>) et d'utiliser alors la pipette sur la couleur désirée.



Une autre manière est d'utiliser GIMP pour vous aider à choisir une couleur. Pour cela, double cliquer sur l'un des deux icônes de la palette de couleurs :



Puis sélectionner votre couleur à l'aide de l'onglet de colorisation montré ci-dessous :



#### 2.1.4 Taille des fontes

Là encore, il n'est pas préférable de changer la taille des fontes, il faudra mieux utiliser un style et l'appliquer. Voici tout de même pour information, la syntaxe utilisable :

```
<FONT size=X>Police de caractère de taille X</FONT>
```

X peut varier de 1 à 7. La valeur par défaut de la taille d'une police est 3.

## 2.2 Mise en forme de la page

Comme vous avez pu le constater, malgré vos retours à la ligne et autre tentative de mise en forme, aucun de vos essais n'a pu aboutir à quelque chose de censé. La raison de cela est simple : les pages Web n'ont pas de taille prédéfinie, donc une page peut accepter 80 caractères comme 120 ou 20 sur une même ligne, tout dépend de la grandeur accordée sur l'écran au navigateur Internet. Il nous faut donc utiliser des balises pour mettre en forme ce texte.

### 2.2.1 Retour à la ligne

La première mise en forme possible est le retour à la ligne, rien de plus simple avec un éditeur classique mais en HTML, il vous faudra taper la balise `<BR>`.

Il est à noter que cette balise ne se ferme pas (pas de `</BR>`) bien qu'il soit possible de l'écrire sans aucune incidence (sauf celle de sauter une seconde ligne) sur notre page.

## 2.3 Mise en forme d'un paragraphe

Il est possible de définir un paragraphe de la façon suivante :

```
<P>
Je commence mon paragraphe<BR>
et le termine
</P>
```

Cette mise en forme nous permet de séparer ce texte de tout le reste en lui adjoignant un espace avant et après. Il est possible en plus de cela d'utiliser l'attribut `align` pour spécifier la position du paragraphe dans la page.

```
<P align=left>
Paragraphe cadré à gauche
</P>
<P align=right>
Paragraphe cadré à droite
</P>
<P align=center>
Paragraphe centré
</P>
```

Il est possible de résumer cette utilisation dans un petit code HTML :

Listing 2.1 – paragraphes.html

```
<html>
  <head>
    <title>Paragraphes</title>
4  </head>
  <body>
    Ceci est un exemple d'utilisation des paragraphes .
    <p>
      Je commence mon paragraphe<BR>
9      et le termine
    </p>
    <P align=left>
      Paragraphe cadré à gauche
14  </P>
    <P align=right>
      Paragraphe cadré à droite
    </P>
    <P align=center>
      Paragraphe centré
19  </P>
  </body>
</html>
```

Ceci est un exemple d'utilisation des paragraphes.

Je commence mon paragraphe  
et le termine

Paragraphe cadré à gauche

Paragraphe cadré à droite

Paragraphe centré

## 2.4 Mise en forme d'un bloc de texte

Un bloc de texte ou une division d'un document se note par la balise `div`. A la différence d'un paragraphe, une division n'est pas séparée du texte qui la précède et la suit. A noter que la balise `<div align='center'>` est équivalente à `<center>`.

Voici un exemple de l'utilisation qui peut en être faite :

Listing 2.2 – divisions.html

```

<html>
  <head>
    <title>Divisions</title>
4  </head>
  <body>
    Ceci est un exemple d'utilisation des divisions
    <div>
      Je commence ma division de texte<BR>
9     et la termine.
    </div>
    <div align=left>
      Division cadrée à gauche
14  </div>
    <div align=right>
      Division cadrée à droite
    </div>
    <div align=center>
      Division centrée
19  </div>
    <center>Mon texte centré</center>
      Fin du texte.
  </body>
</html>

```

Ceci est un exemple d'utilisation des divisions

Je commence ma division de texte  
et la termine.

Division cadrée à gauche

Division centrée  
Mon texte centré

Division cadrée à droite

## 2.5 Insertion d'images

Notre page est bien pâle et nous aimerions la rendre un peu plus jolie. Pour se faire rien de tel qu'une petite image. Pour se faire, il suffit d'enregistrer l'image sur votre répertoire et dire à HTML d'aller la lire.

**Attention!** : l'image n'est pas insérée dans la page mais une balise indique que le navigateur doit aller la chercher et l'afficher.

```
<img src='monimage'></img>
```

Comme vous pouvez aisément le deviner, il est possible de centre l'image de la même façon que les balises précédentes :

```
<img src='tuxprof.png' align='...'></img>
```



## 2.6 Astuce

Nous le constatons de plus en plus souvent, notre page Web grandit de plus en plus et les idées fourmillent sous nos doigts. Une petite astuce pour se rappeler de toutes ces idées est de les mettre en commentaire dans notre page Web, ainsi elles resteront sur le papier et n'apparaîtront pas à l'écran.

```
<!-- Commentaire -->
```

## Chapitre 3

# HTML : Première mise en ligne

Votre première page Web est maintenant créée, aussi pauvre qu'elle soit, c'est un page Web et à ce titre nous devons la mettre sur Internet.

### 3.1 Mise en ligne d'une page Web

#### 3.1.1 Étape 1 : Fournisseur d'espace

Votre FAI (Fournisseur d'Accès Internet) peut vous permettre d'avoir un espace d'hébergement sur ses serveurs. Cet espace vous permettra de déposer vos pages et par ce biais de les offrir à la communauté des internautes. Dans notre cas, c'est Free qui est notre provider ou FAI, et nous demandons donc un espace à notre FAI par le biais de l'Internet. Un compte mail est donc créé : `viederrenac@free.fr` ainsi qu'une demande d'espace associée à ce compte. 24 heures plus tard notre espace est créé et pointe sur `http://viederrenac.free.fr`

#### Correspondance entre Internet et espace disque sur le serveur

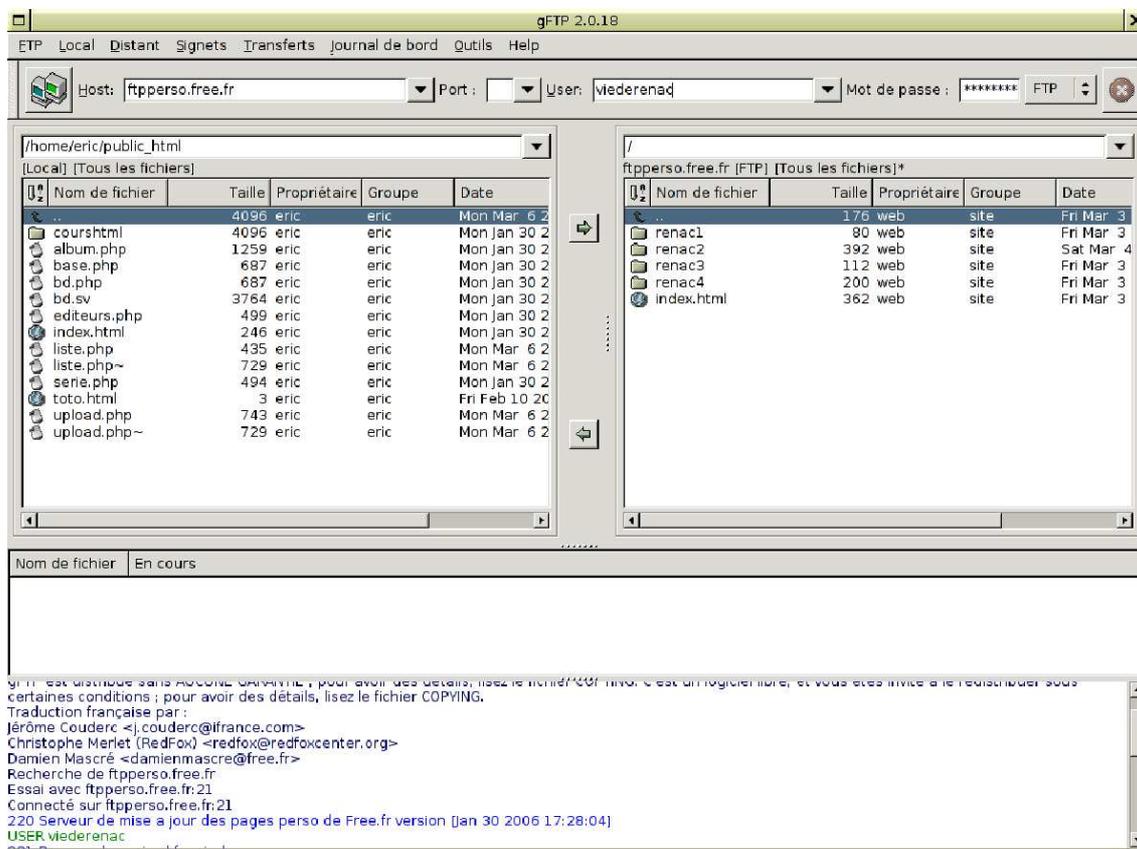
Lorsque vous allez voir une page Web sur l'Internet, votre navigateur demande à accéder à un fichier spécifique (voir chapitre 1) ou dans le cas où celui-ci est absent au fichier `index.html`. Ces fichiers sont en fait de réels fichiers mais déposés sur un serveur. Ce serveur possède une adresse IP sur l'Internet, on l'appelle Serveur Web. Une page Web est donc une simple fenêtre sur un espace disque du serveur. En l'absence d'un fichier `index.html` ou équivalent, le serveur Web présentera une liste de fichiers disponibles.

#### 3.1.2 Étape 2 : Connexion et transfert

Afin de pouvoir transférer les fichiers de nos pages Web sur le serveur distant, il va être nécessaire d'utiliser autre chose que le protocole HTTP (bien que cela soit possible dans certains cas). Cette autre chose est le protocole FTP. Son utilisation est simple et peu se faire de deux façons :

- en ligne de commande (DOS ou Linux)
- à l'aide d'un logiciel spécifique (Filezilla, gFTP par exemple).

Nous nous intéresserons dans ce cours uniquement à l'utilisation de gFTP.



### Description des champs

Sur la partie gauche de l'écran se trouve votre disque dur local, votre compte qui contient vos fichiers HTML que vous désirez transférer.

Sur la partie droite se trouvera l'espace disque du serveur qui vous a été alloué, pour l'instant celui-ci n'est pas visible car nous ne sommes pas connectés. Pour se connecter il va falloir renseigner les champs suivants :

**Host :** Dans notre cas `ftpperso.free.fr`, c'est le nom du serveur qui héberge notre page Web.

**Port :** Il n'est pas nécessaire de renseigner ce champs, par défaut c'est 21.

**User :** Il s'agit de l'utilisateur associé à l'espace disque sur lequel on désire se connecter, dans notre cas `videderenac`.

**Mot de passe :** Le mot de passe défini avec l'utilisateur. Celui-ci est donné par le FAI ou par vous même lors de la création du compte.

Une fois ces renseignements donnés, il ne reste plus qu'à se connecter en appuyant sur le bouton :



Le cadre de gauche se remplit alors.

Pour transférer les fichiers, il suffit de cliquer sur le fichier dans le cadre de gauche et appuyer sur la flèche lui indiquant de transférer.

Une fois les transferts terminée, visualiser le résultat en vous rendant sur la page Web :

`http://viederrenac.free.fr`

. Attention cependant, dans le cas où aucun fichier `index.html` n'a été transféré, la liste des fichiers disponibles apparaîtra ...

## Chapitre 4

# HTML : Les ancres

Il vous faut maintenant naviguer entre vos pages ou à l'intérieur d'une même page. Pour se faire, il vous est possible d'utiliser des ancres, soit des ancres à l'intérieur d'un même document, soit des ancres entre les documents.

### 4.1 Ancre à l'intérieur d'un même document

Listing 4.1 – ancrs1.html

```
2 <html>
  <head>
    <title>
      Gestion des ancrs
    </title>
  </head>
7 <body>
  <a href="#debut">Debut</a><br>
  <a href="#ligne10">Ligne 10</a><br>
  <a href="#ligne20">Ligne 20</a><br>
  <a name="debut">Début de page</a>
12 Ligne 1<br>
  Ligne 2<br>
  Ligne 3<br>
  Ligne 4<br>
  Ligne 5<br>
17 Ligne 6<br>
  Ligne 7<br>
  Ligne 8<br>
  Ligne 9<br>
  <a name="ligne10">Ligne 10</a>
22 Ligne 11<br>
  Ligne 12<br>
  Ligne 13<br>
  Ligne 14<br>
  Ligne 15<br>
27 Ligne 16<br>
  Ligne 17<br>
  Ligne 18<br>
  Ligne 19<br>
```

```
32 <a name=" ligne20 ">Ligne 20</a>  
   Ligne 21<br>  
   Ligne 22<br>  
   Ligne 23<br>  
   Ligne 24<br>  
37   Ligne 25<br>  
   Ligne 26<br>  
   Ligne 27<br>  
   Ligne 28<br>  
   Ligne 29<br>  
42 <h1>Index</h1>  
   <a href="#debut">Debut</a><br>  
   <a href="#ligne10">Ligne 10</a><br>  
   <a href="#ligne20">Ligne 20</a><br>  
</body>  
</html>
```

## 4.2 Navigation relative et absolue

## 4.3 Ancre entre les documents

## 4.4 Ancre sur un document à une position précise

# Listings

1.1	balise . . . . .	6
1.2	balise.html . . . . .	6
1.3	squelette.html . . . . .	6
1.4	titres.html . . . . .	7
2.1	paragraphes.html . . . . .	11
2.2	divisions.html . . . . .	12
4.1	ancre1.html . . . . .	17