

Pour surfer il faut :

1 : Une connexion Internet

= un point d'accès au réseau Internet.

réseau = Connexion physique (des câbles) + protocoles de communication permettent de définir de façon standardisée la manière dont les informations sont échangées entre les équipements du réseau.

Internet = réseau informatique à l'échelle du monde, reposant sur le protocole de communication IP (Internet Protocol), et qui rend accessible au public des services comme le courrier électronique et le web

Fournisseur d'accès = offre un point de connexion à Internet. Fait partie du réseau Internet et étend son réseau jusqu'à vous.

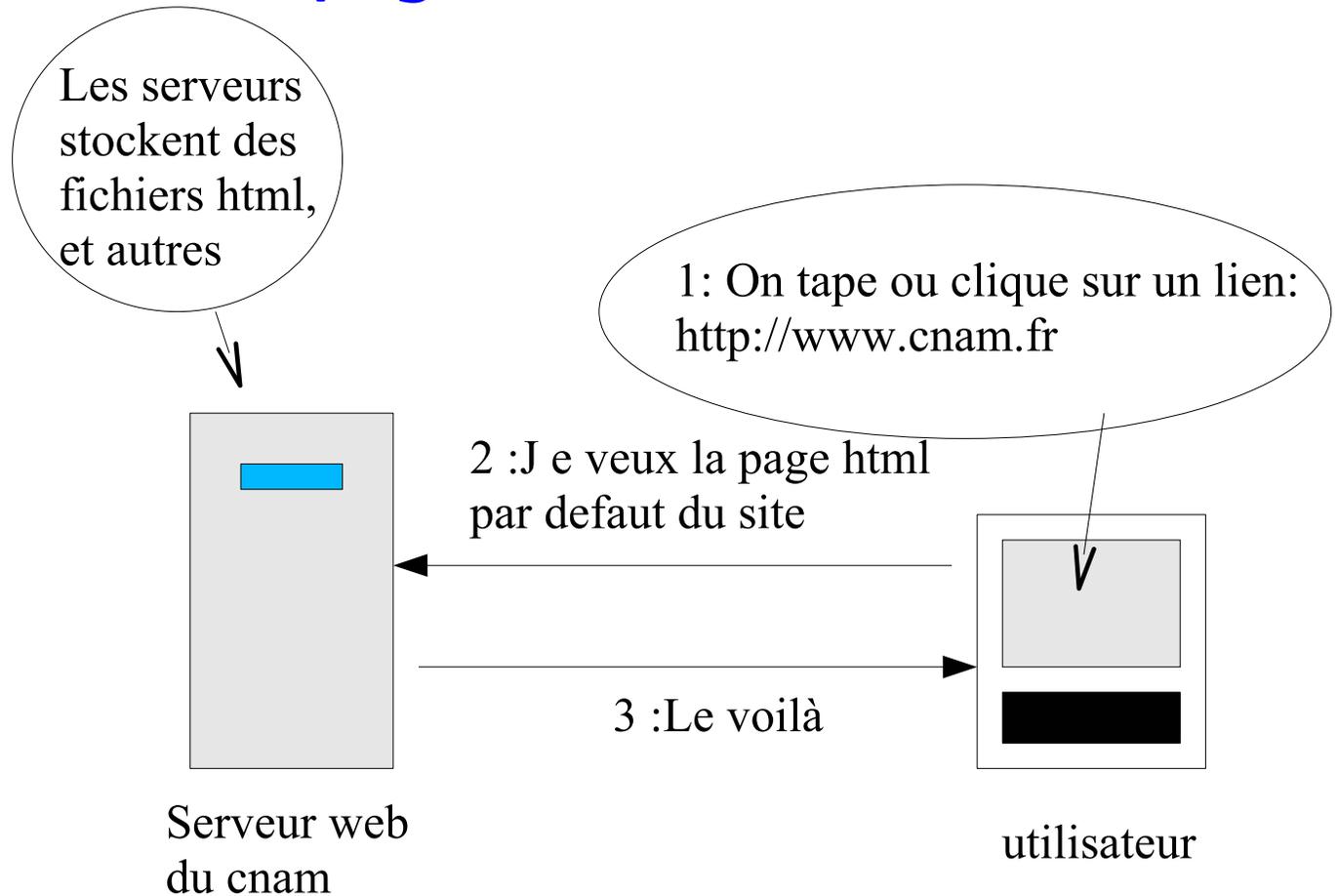
Pour surfer il faut :

2 : Un navigateur Web

Le World Wide Web (le Web, la « toile (d'araignée) mondiale »), est un système hypertexte public fonctionnant sur Internet et qui permet de consulter, avec un navigateur Web, des pages Web mises en ligne dans des sites Web. L'image de la toile vient des hyperliens qui lient les pages Web entre elles.

Un navigateur Web est un logiciel conçu pour consulter le World Wide Web, l'ensemble des sites web. Il est composée d'un espace d'affichage des pages web, d'une barre de menus déroulants, d'une barre d'outils (au moins la barre d'adresse)

Naviguer = demander à un serveur web une page web



`http://www.cnam.fr` = **adresse web** (Uniform Resource Locator) indique l'endroit où se trouve la ressource et le **protocole de communication** à utiliser (ici HTTP).

Autres protocoles : mailto, ftp ...

Affichage d'une page web : exemple



The screenshot shows a Mozilla Firefox browser window with the title 'developpement web - Mozilla Firefox'. The address bar contains the file path 'file:///C:/Documents%20and%20Settings/ft' and the page name 'openweb'. The main content of the page is as follows:

Bienvenue au cours Développement Web



Ce cours a pour objectif de dresser un tableau complet de l'architecture du WEB et des technologies concernées dans la construction d'un site.

horaires et lieu

Le cours a lieu a saint Denis salle 415 , le lundi de 18h15 à 21H15.

telle qu'elle est transmise au navigateur :

```
<html>
  <head>
    <title>developpement web</title>
  </head>
  <body>
    <h1>Bienvenue au cours Developpement Web</h1>
    
    <p>
Ce cours a pour objectif de dresser un
tableau complet de l'architecture du WEB et des
technologies concernées dans la construction d'un s

    <h2>horaires et lieu</h2>
    <p>
Le cours a lieu a saint Denis salle 415 , le lundi c

    </p>
  </body>
</html>
```

- Une page HTML est une page de texte écrite dans le langage HTML = un langage de **balises** indiquant la structure de la page (`<head>`, `<p>` ...)
- Elle est **interprétée** par le navigateur. Les balises lui expliquent où sont les titres, les paragraphes, les images Le navigateur applique ses propres conventions d'affichage.
- Pour voir le code HTML d'une page (ou le récupérer) : dans le navigateur : menu affichage puis code source.
- Pour écrire des pages web : Utiliser un éditeur de texte (bloc note, notepad, texte-dit...) et écrire du texte fidèle à la syntaxe HTML (comme code source)

Debut du htm

```
<html>
```

debut de l'entete

```
<head>
```

titre (de la fenetre navigateur)

```
<title>developpement web</title>
```

fin de l'entete

```
</head>
```

debut du contenu de la page

```
<body>
```

Le texte entre balises est un gros titre

```
<h1>Bienvenue au cours Developpement Web</h1>
```

ensuite vient l'image Nenuphars

```

```

debut de paragraphe

```
<p>
```

contenu du paragraphe

Ce cours a pour objectif de dresser un tableau complet de l'architecture du WEB et des technologies concernées dans la construction d'un site.

(les sauts de ligne ne s'affichent pas)

fin du paragraphe

```
</p>
```

titre moyen

```
<h2>horaires et lieu</h2>
```

debut de paragraphe

```
<p>
```

contenu du paragraphe

Le cours a lieu a saint Denis salle 415 , le lundi de 18h15 à 21H15.

fin paragraphe

```
</p>
```

fin contenu de la page

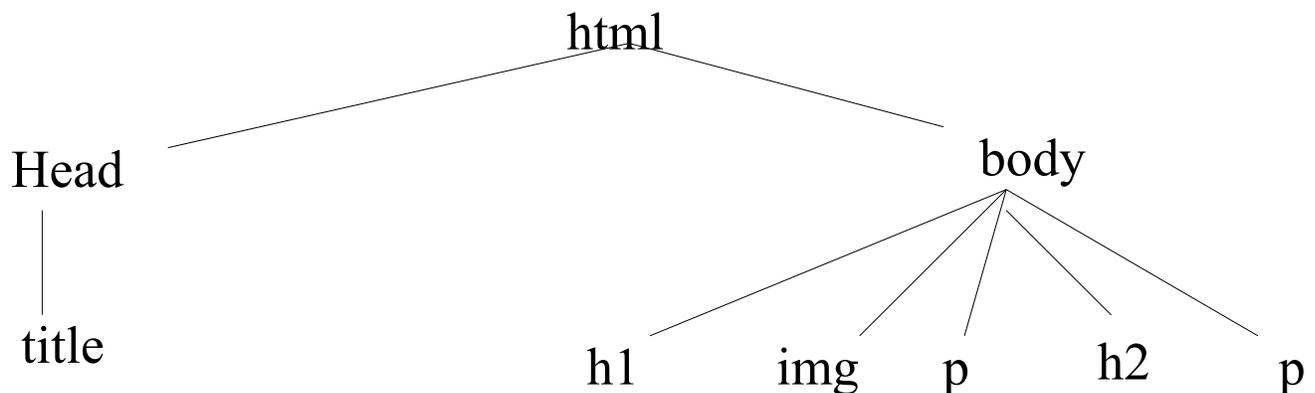
```
</body>
```

fin du HTML

```
</html>
```

conclusion sur HTML

- un élément HTML = balise ouvrante + contenu + balise fermante
- HT = hypertexte = balise <a> qui permet d'écrire des liens vers d'autres page web.
- Une page HTML = un emboitement de balises (un arbre).



- Langage HTML = ensemble de balises (on en apprendra quelques unes)

La présentation : CSS

HTML = description de la structure de la page

présentation = laissée au navigateur (applique ses styles par défaut)

OU

écrite par vos soins dans un autre langage que HTML : CSS (cascading style sheet)

Écrire une feuille de style = écrire un fichier texte d'extension css.

Balise link indique au navigateur d'utiliser la feuille de style externe styleDev.css

```
<html>

<head>

<title>developpement web</title>
<link rel="stylesheet" media="screen" type="text/css"
      title="Design1" href="styleDev.css" />
</head>

<body>

  <h1>Bienvenue au cours Developpement Web</h1>

  <p>
Ce cours a pour objectif de dresser un tableau complet de
l'architecture du WEB et des
technologies concernées dans la construction d'un site.
</p>

  <h2>horaires et lieu</h2>

  <p>
Le cours a lieu a saint Denis salle 415 ,
le lundi de 18h15 à 21H15.
</p>
</body>
</html>
```

La couleur du texte du body sera violet, dans une police Trebuchet. Il y aura un cadre de couleur aqua et une marge de 1,5cm

```
body {  
  color:rgb(183,0,183);  
  font-family:"Trebuchet MS";  
  border: 4px outset aqua;  
  margin:1.5cm;  
}
```

L'element pour lequel on definit le style

```
h1 {  
  color:red;  
  font-size:xx-large;  
  font-family:"Comic Sans MS";  
  text-align:center;  
}
```

Les titres seront rouges, de taille xxlarge d'une police Comic et centré.

la fichier CSS sera transmis au navigateur en meme temps que la page web.

Le navigateur sait aussi interpréter le CSS.

Ce qui a comme rendu final :



Interet des feuilles de style externes

- uniformité du style du site entier
- Changement d'apparence facile.
- adaptabilité : différentes feuilles de styles alternative (ecran, impression, braille etc)

Pages dynamiques (1) : appel de scripts coté serveur

Hypertexte : navigation de page en page.

HTML et CSS : pages écrites une fois pour toutes et stockées sur le serveur (statique).

Insuffisant : le contenu des pages doit souvent être construit au moment de la demande.

exemple : un livre d'or.

Page 1 : choix entre **consulter les textes du livre d'or**, et **écrire un texte dans le livre d'or**

=

page html avec 2 liens : [Consulter](#) et [Ecrire](#).

Scénario utilisateur

Consulter puis Ecrire puis Consulter.

- consultation 1 : le serveur fournit une page html contenant les textes déjà écrits.
- consultation 2 : pas la même page !
Doit contenir un texte de plus !

Impossible à faire avec HTML + CSS!!!!!!

Solution :

- Le serveur doit stocker (fichier, BD) la liste des textes écrits.

- consultation

≠ un lien vers une page HTML (statique)

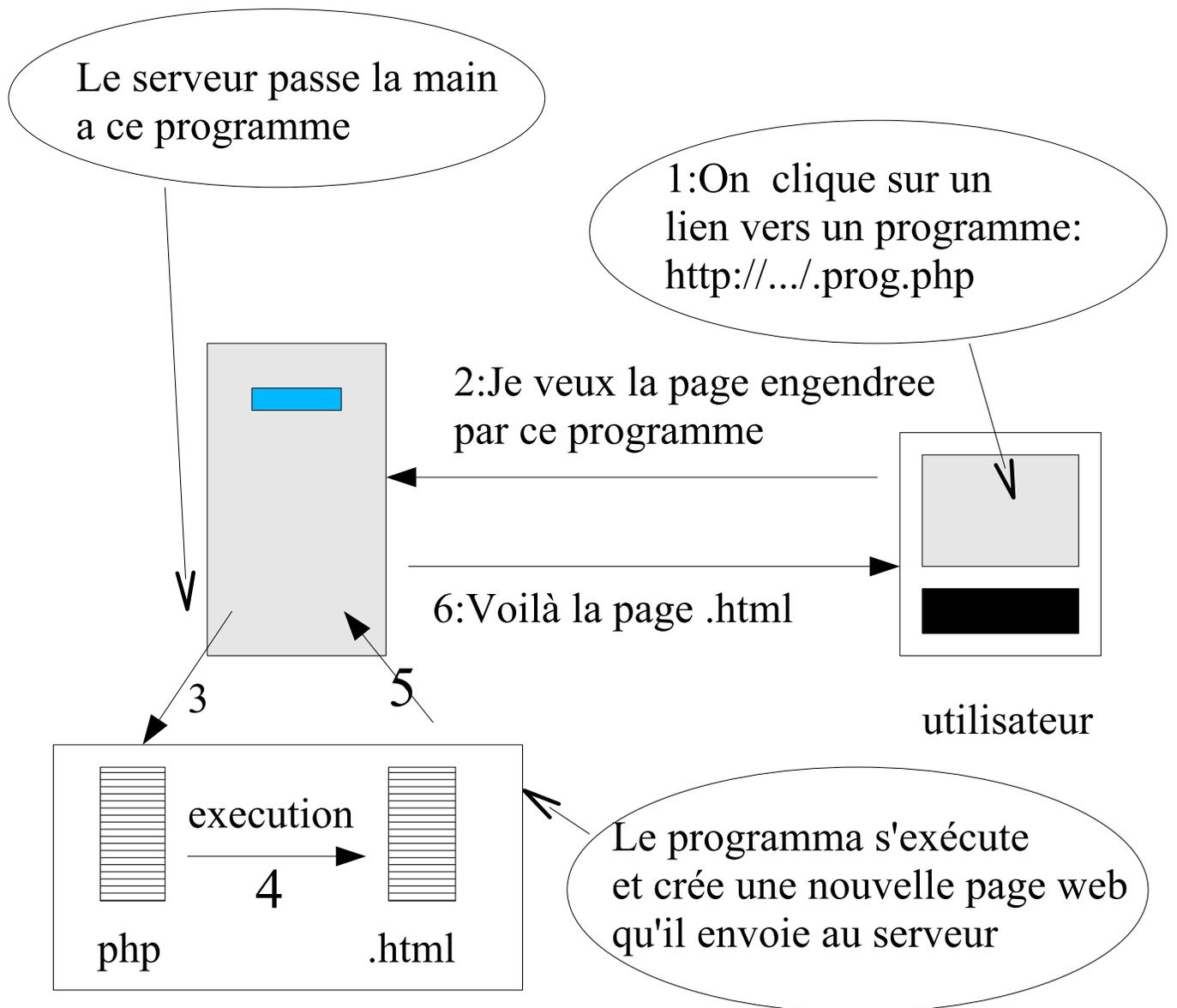
= lien vers quelque chose qui **construit la page** html des textes a partir de l'ensemble des textes stockés au moment de la demande (dynamique)

= lien vers un **programme** qui construit la page HTML en fonction des textes stockés sur le serveur.

programme ≠ HTML

programme = cgi-bin ou PHP ...

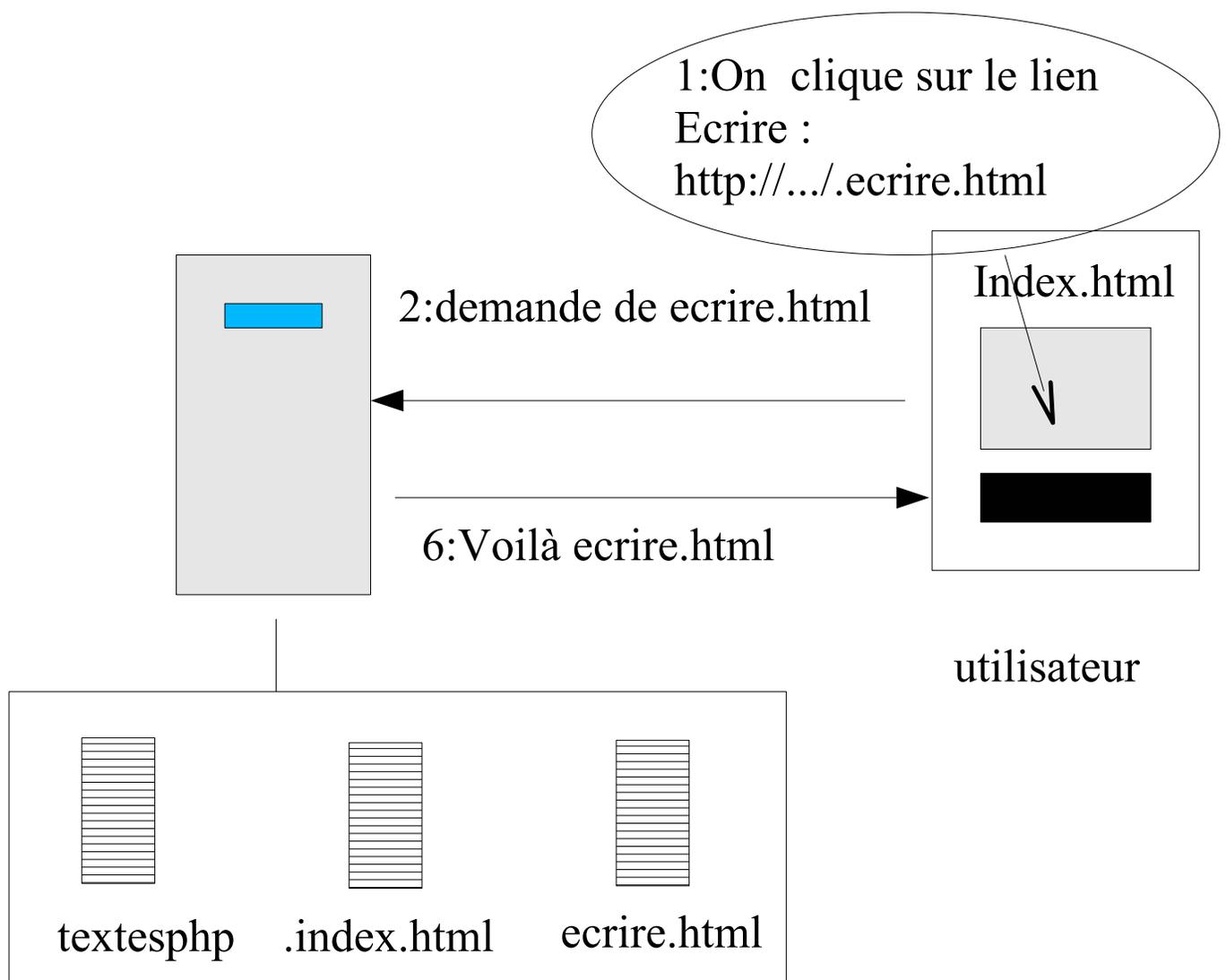
programme = stocké coté serveur et s'exécute coté serveur



Pages dynamiques (2) : Transmettre des données aux scripts serveurs

Ecriture dans livre d'or

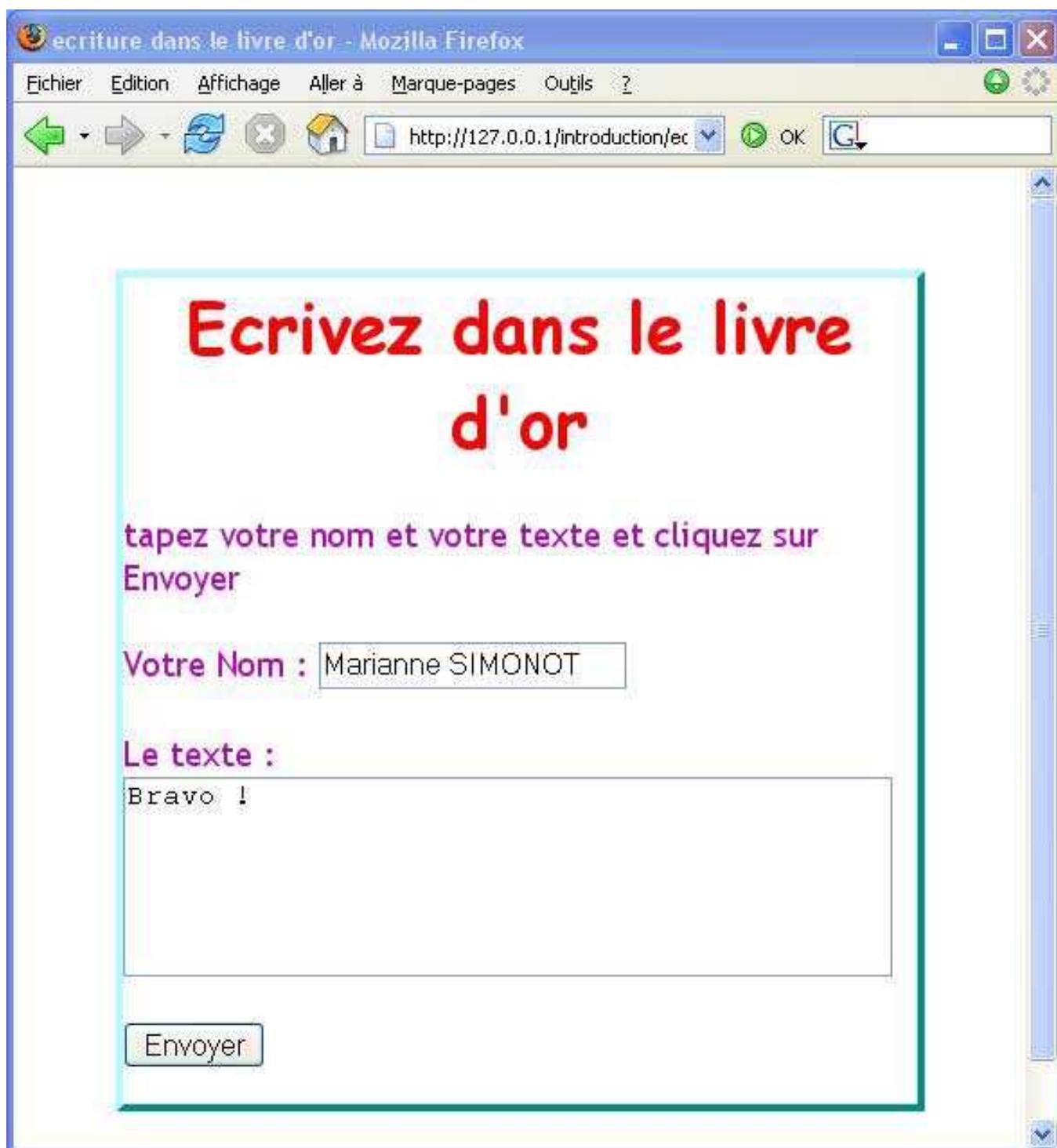
Cliquer sur **Ecrire** = demande d'une page HTML (pas d'un script).



= lien vers une page HTML offrant un champ de saisie à l'utilisateur
= un formulaire HTML
= Element HTML permettant aux visiteurs de communiquer avec le site.

= la donnée saisie par l'utilisateur doit être transmise au serveur qui modifie ses données stockées en fonction (programmes cgi-bin ou php peuvent le faire) et construit la page web à renvoyer.

Voilà comment s'affiche écrire.html (après qu'on ait rempli les champs) :



Après avoir cliqué sur Envoyer, les données du formulaire sont envoyées au Serveur qui renvoie la page suivante :



```

<html>
<head>
<title>écriture dans le livre d'or</title>
<link rel="stylesheet" media="screen" type="text/css"
title="Design 1" href="examRatMar06.css" />
</head>
<body>
<h1>Ecrivez dans le livre d'or</h1>
<form action=traitement.php method="POST">
<p> tapez votre nom et votre texte et cliquez sur Envoyer
<br /><br />
Votre Nom : <input type="text" name="nom" value="" />
<br /><br />
Le texte : <textarea name="com" rows="5" cols="40">
votre texte </textarea><br /><br />
<input type="submit" value="Envoyer" />
</p>
</form>
</body>
</html>

```

Programme serveur qui va traiter le formulaire

le formulair est contenu entre les balises <form > et </form>

Des champs de saisie du formulaire

Le bouton de soumission
Cliqué, il declenche l'action (traitement.php)

Et le programme traitement.php :

C'est du code php

echo signifie : écrire dans la page html à construire .

On écrit donc <html> puis <head> etc..

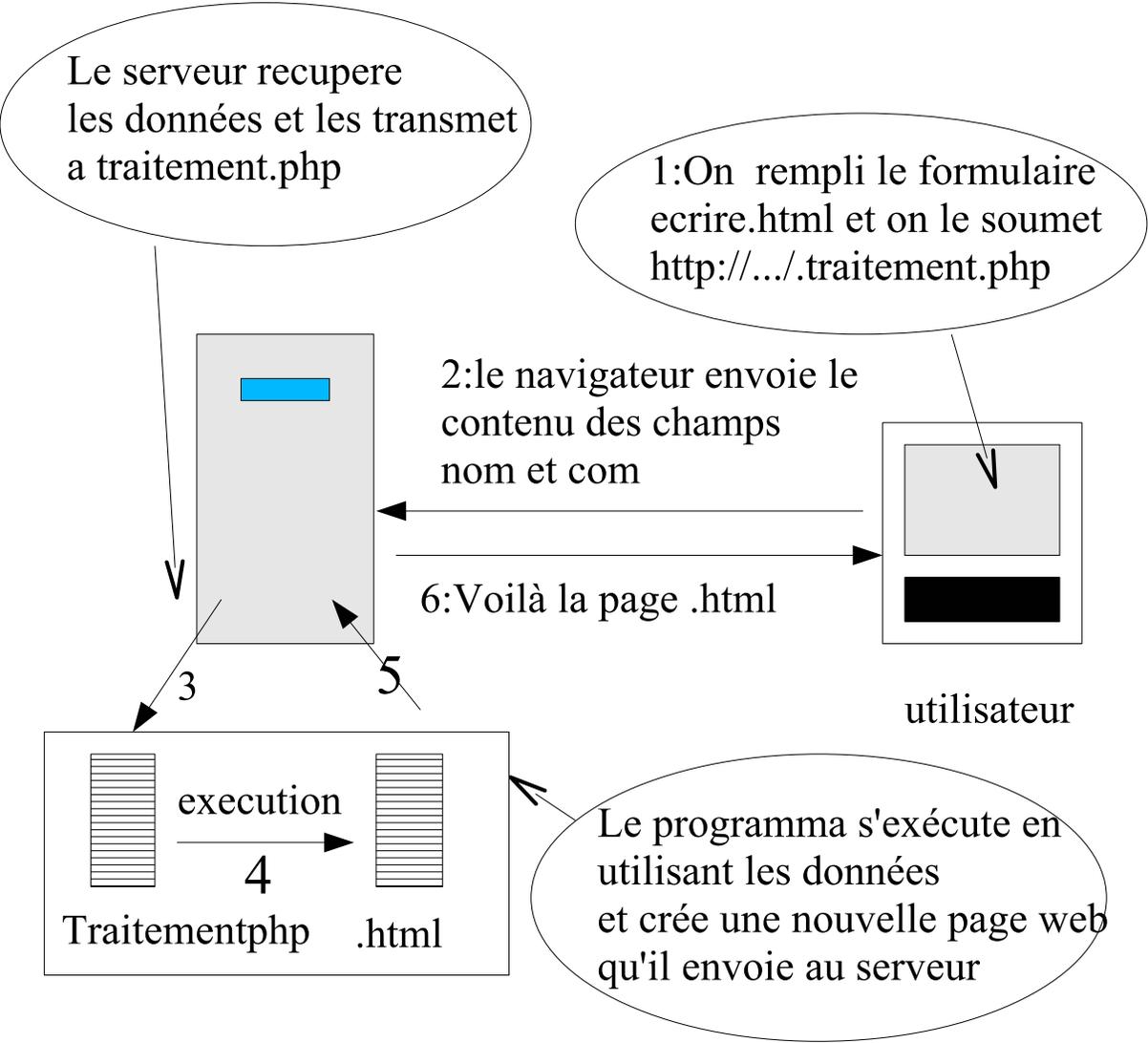
```
<?php
echo "<html>";
echo "<head>";
echo "<title>traitement</title>";
echo "</head>";
echo "<body>";
echo "<p>";
echo "merci </br /> <br />";

echo $_POST['nom'];

echo "<p>pour avoir laissé un message sur le livre
d'or.";
echo "<p></body></html>";
?>
```

Signifie le contenu du champ nom du formulaire que l'on traite : le contenu du formulaire a donc été envoyé au serveur

En résumé :



Dynamisme coté Navigateur

On peut aussi, dans une page HTML, appeler des scripts qui s'exécutent coté Navigateur.

Javascript : langage de programmation rudimentaire.

applets : programme Java stocké sur le serveur et transféré à la demande coté client qui charge l'applet et l'exécute.

scripts coté Navigateur ont accès (lecture et écriture) au contenu de la page qui les appelle.

MAIS n'ont pas accès aux données stockées coté serveur.

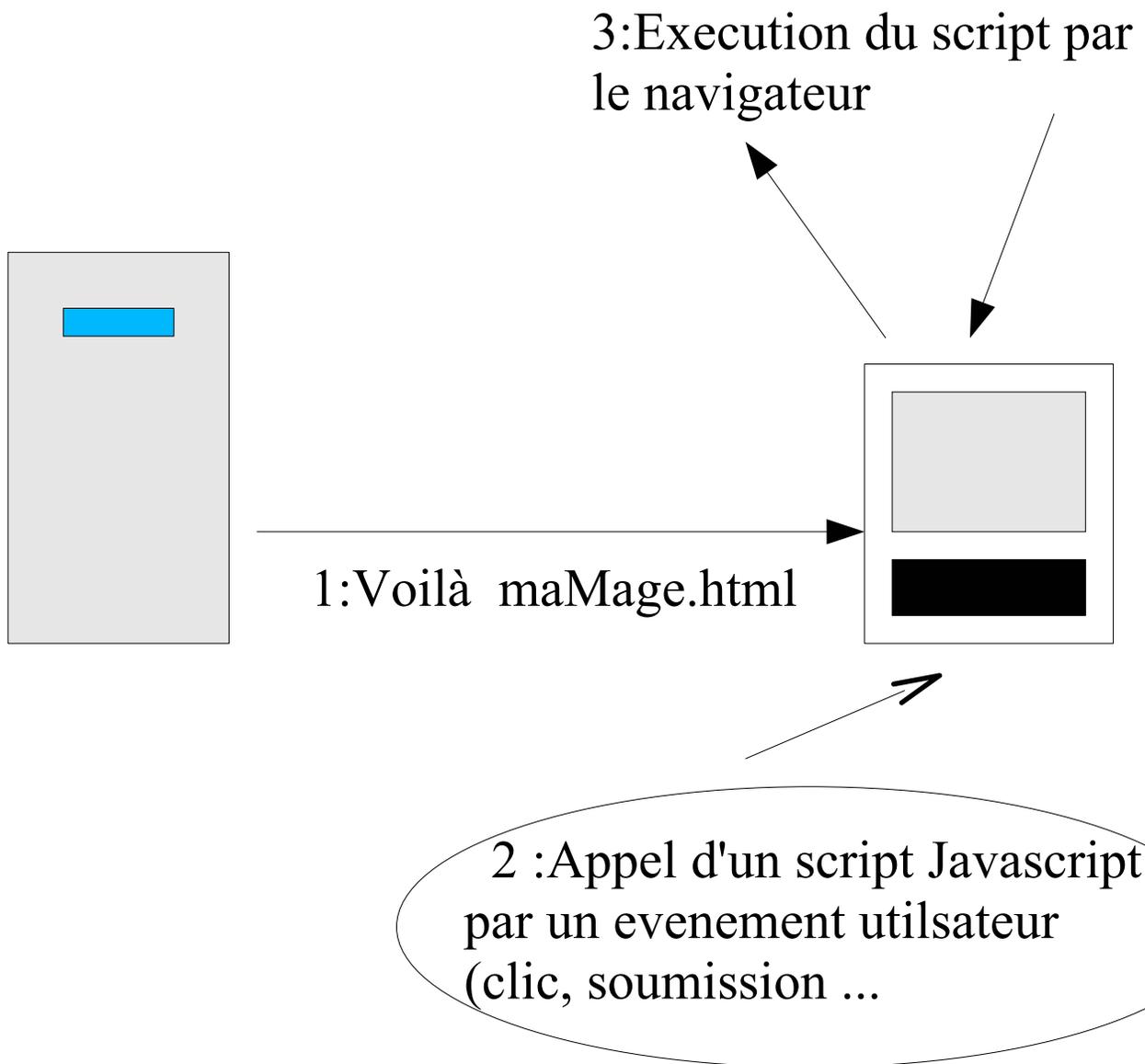
MAIS ont des restrictions de sécurité (ne peuvent pas accéder aux fichiers de la machine locale ...)

On fait essentiellement appel à des programmes exécutés coté navigateur pour **réagir à des**

évènements utilisateurs : click ou passage sur une zone de la page . . . , **quand la réaction demandée ne nécessite rien de ce qui est stocké sur le serveur.**

Exemples : une calculette, un sudoku, vérifications de base du remplissage d'un formulaire...

Appel de script Javascript



Convertisseur Francs euros



Source HTML du convertisseur

```
<html>
<head>
  <title>Page 1 </title>
  <link rel="stylesheet" media="screen" type="text/css"
    title="Design1" href="examRatMar06.css" />
  <script language="javascript">
    fonction cfe(lef) { // ... on le verra tout a l'heure }
  </script>
</head>
<body>
```

En HTML, la balise script permet de définir des fonctions javascripts

Le formulaire avec les deux champs francs et euros.

On remplit un champ, on soumet et la conversion doit de placer dans le champ reste vide.

Il n'y a pas action = ... car on n'a pas besoin du serveur.

Lorsque l'évènement Onclick sur le bouton de soumission se produit, on appelle la fonction JS cfe avec comme argument le formulaire tout entier.

```
<h1> Convertisseur Francs Euros </h1>
```

```
<form class=regle method="get">
```

```
<p>
```

```
Francs
```

```
<input type="text" name="fr" size="20" />
```

```
</p>
```

```
<p>
```

```
Euros
```

```
<input type="text" name="eu" size="20" />
```

```
</p>
```

```
<input type="button" onClick="cfe(this.form)"
  value=" conversion " />
```

```
</form>
```

```
</body>
```

```
</html>
```

Fonction JS du convertisseur

Ceci represente le formulaire

```
<script language="javascript">
function cfe(lef) {
    tdc = 6.55957;
    var ffr = lef.fr.value;
    var eeu = lef.eu.value;
    if (ffr.length>0) {
        eeu = ffr/tdc;
        lef.eu.value = eeu.toFixed(2);
    }
    else {
        ffr = eeu*tdc;
        lef.fr.value = ffr.toFixed(2);
    }
}
</script>
```

La variable ffr contient
la valeur du champ fr
du formulaire.,
la variable eeu celle du champ
eu

plan du cours

Introduction (c'est fait)

1 : web statique : Html / CSS

2 : Les protocoles de communication

3 : Introduction au web dynamique coté serveur : POST et GET , cgibin , php. Application au traitement des formulaires.

3 : web dynamique coté navigateur : Javascript.

4 : Introduction aux applets

5 : points épineux : gestion de session, sécurité, accessibilité.

exam : entre 28 janvier et 11 fevrier