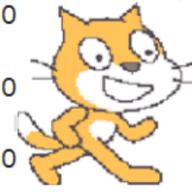


0100100100**1**0100100
101001**1**01001000100
0101**0**0010010010010



Création de sites WEB

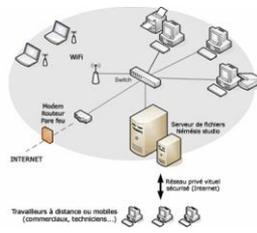
Sommaire :

Introduction	2
Les sites statiques et le HTML	3
<i>Le langage HTML</i>	3
<i>L'architecture client-serveur</i>	5
Les sites interactifs et les langages de script	7
<i>Interactivité coté client avec JavaScript</i>	7
<i>Interactivité coté client avec ActionScript (Flash)</i>	8
<i>Interactivité coté serveur avec PHP</i>	9
Les sites dynamiques et les bases de données	10
<i>Création de modèles dynamiques</i>	10
<i>Pourquoi utiliser une base de données</i>	11
<i>Application dynamique avec PHP-MySQL</i>	12
<i>Application dynamiques avec PHP-MySQL et Flash</i>	13

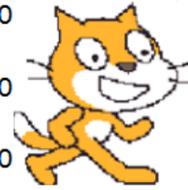
Présentation réalisée grâce à :



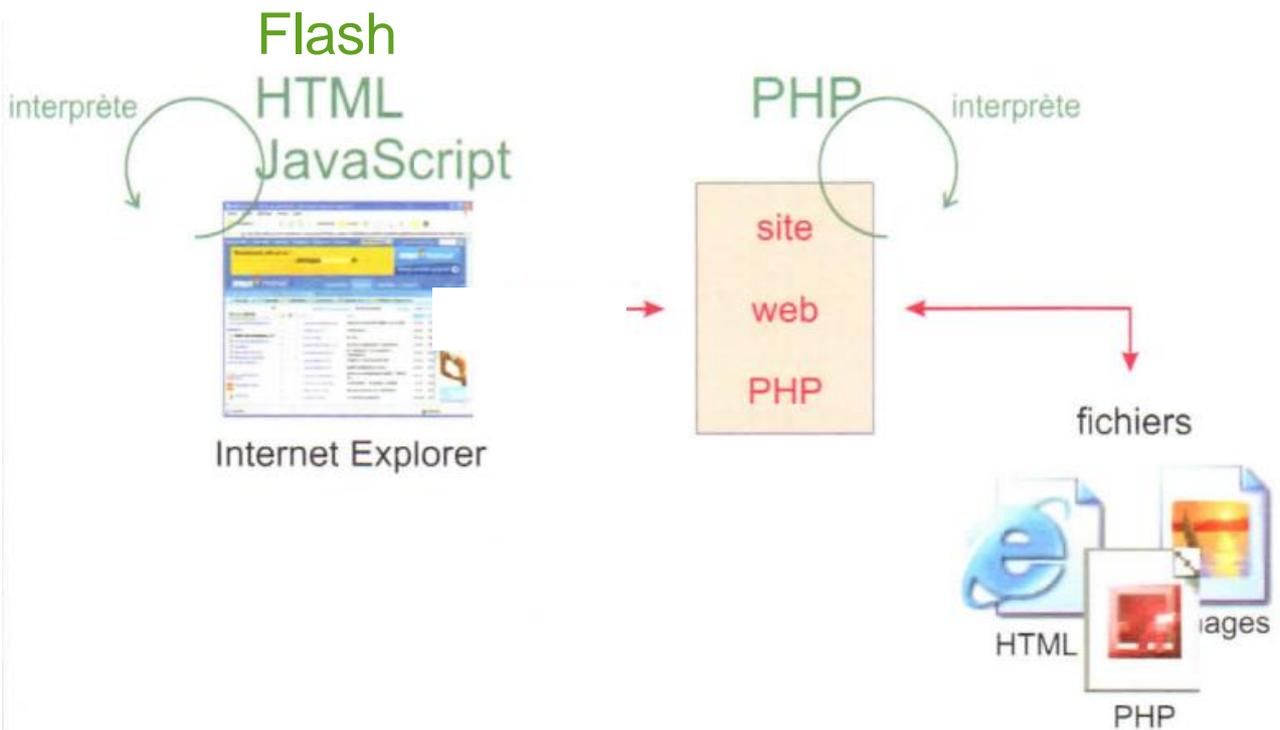
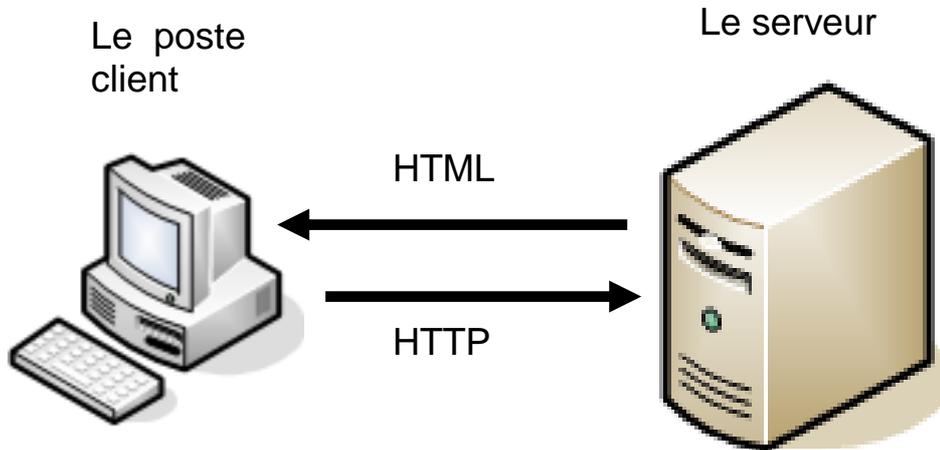
Jean-Marie DEFRANCE, *PHP/MySQL avec Flash MX 2004*, EYROLLES, mars 2005, 709 p.

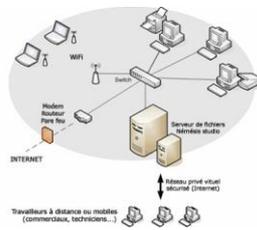


0100100100**1**0100100
101001**1**01001000100
0101**0**0010010010010

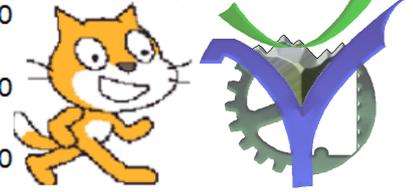


Introduction





0100100100**1**0100100
101001**1**01001000100
0101**0**0010010010010



Les sites statiques et le HTML

Le langage HTML

Avant de présenter les langages utilisés pour la conception de sites dynamiques, rappelons quelques notions de base sur les pages Web statiques. Nous appelons « page Web » toute page pouvant être affichée dans un navigateur (Internet Explorer, Netscape...). Le langage utilisé pour la conception d'une page Web est le Hyper Text Markup Language. Il ne s'agit pas d'un langage de programmation au sens propre, mais d'un langage de description d'une page Web. Le fichier qui contient la description de cette page porte en général l'extension .htm ou .html. Il est constitué du texte et des liens aux images à afficher, répartis entre des balises (par exemple : <p>...</p>) qui déterminent la façon dont ces éléments seront présentés dans le navigateur. Certaines de ces balises permettent également de transformer un texte ou une image en lien hypertexte (<a>...). Ces hyperliens (les liens hypertextes) sont très importants dans une page Web, puisqu'ils permettent d'organiser la navigation dans un site en reliant les pages entre elles. Les internautes peuvent passer d'une page à l'autre grâce à un simple clic sur ces liens, d'où l'expression « naviguer » ou « surfer » sur le Web (voir figures 1-1 et 1-2).

Figure 1-1
Exemple
de code HTML
d'une page Web.

```

<HTML>

<HEAD><title>Page Html</title></HEAD>

<BODY>
<p>
<font size="7">Html, un langage statique :</font>
</p>
<p>Avant de présenter... pages web statiques</p>
<p></p>
<p>
<a href="page1.htm">page1</a>
<a href="page2.htm">page2</a>
</p>
</BODY>
</HTML>
    
```

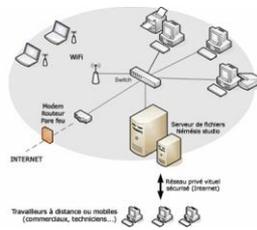
Le code d'une page HTML est constitué de balises de description qui définissent la mise en page des éléments

Selon le type de balise, il sera possible d'afficher un texte ou une image.

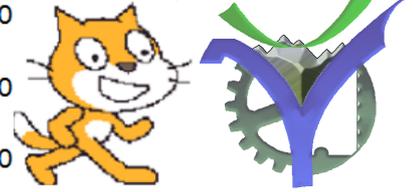
Certaines balises permettent de créer des liens hypertextes avec une autre page HTML.



2nd ICN 2nd ICN



0100100100**1**0100100
101001**1**01001000100
0101**0**0010010010010



Vous pouvez ainsi mettre en forme votre texte et disposer les images à votre convenance dans la page en les reliant entre elles par des liens hypertextes. Cependant, vous ne disposez d'aucune instruction pour réaliser un traitement différent en fonction d'un événement ou d'une condition particulière. C'est pourquoi une page HTML est dite « statique » : elle s'affiche toujours sous la même forme et toutes les pages susceptibles d'être appelées doivent être stockées sur le serveur (voir figure 1-3).

Nous verrons plus loin que d'autres langages, comme PHP (*Personal Home Page*, devenu par la suite *Hypertext Preprocessor*), permettent de créer des pages « dynamiques » qui peuvent être personnalisées selon une requête ou le profil de l'internaute. Ils utilisent pour cela un seul et même fichier modèle, en interaction avec une base de données.

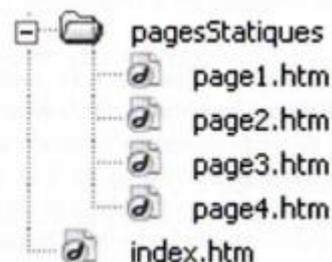
Figure 1-2

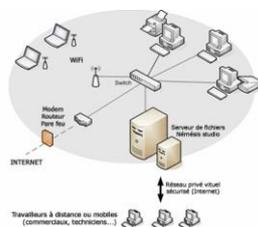
Interprétation et affichage du code de la figure 1-1 dans un navigateur Internet : le navigateur reçoit le code HTML de la page et l'affiche à l'écran en interprétant les différentes balises qu'il contient.



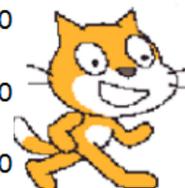
Figure 1-3

Arborescence d'un site statique : toutes les pages du site doivent être présentes sur le serveur.





0100100100**1**0100100
 101001**1**01001000100
 0101**0**0010010010010



L'architecture client-serveur

Nous venons de voir que les sites statiques sont constitués d'un ensemble de pages HTML reliées entre elles par des liens hypertextes qui permettent de naviguer de l'une à l'autre. Le protocole utilisé pour transférer des informations sur Internet s'appelle HTTP (*Hyper Text Transfer Protocol*). Une requête HTTP (par exemple : `http://www.eyrolles.com/page.htm`) est envoyée vers le serveur afin d'accéder à la page désirée et de la visualiser dans le navigateur du poste client (voir étape 1 de la figure 1-4).

Lorsque le serveur Web reçoit cette requête, il recherche la page demandée parmi toutes les pages HTML présentes sur le site concerné et la renvoie ensuite au client (voir étape 2 de la figure 1-4). Le code HTML reçu par le poste client est alors interprété et affiché par le navigateur (voir étape 3 de la figure 1-4). C'est ce qu'on appelle l'architecture client-serveur (je demande, on me sert) : le client

est le navigateur Internet (Internet Explorer, Netscape...) et le serveur est le serveur Web sur lequel est stocké le site Internet.

Ce type de site est très simple à réaliser et on peut s'en contenter dans le cadre de petits projets de quelques dizaines de pages et dont la mise à jour n'est pas fréquente. Cependant, il affiche vite ses limites pour la conception d'applications plus conséquentes ou nécessitant de fréquentes mises à jour. Les sites marchands et autres portails d'informations ne peuvent pas être réalisés sur ce modèle.

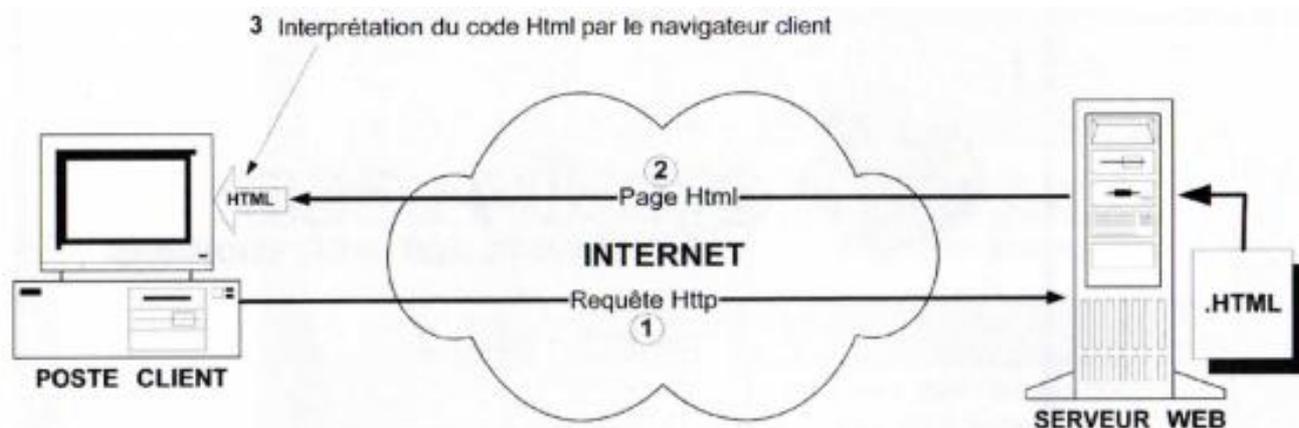
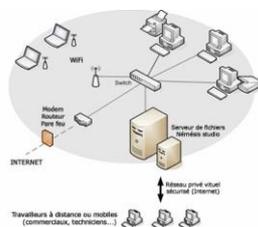


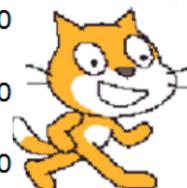
Figure 1-4

L'architecture client-serveur : le poste client envoie au serveur une requête HTTP ; le serveur Web recherche puis fournit au poste client la page demandée, qui est ensuite interprétée par le navigateur.





0100100100**1**0100100
 101001**1**01001000100
 0101**0**0010010010010



Le code HTML est un langage interprété et non compilé comme le sont les différents programmes dédiés à un type d'ordinateur spécifique (PC, Mac...). Pour illustrer ce qu'est un programme compilé, prenons le cas de votre éditeur de texte (Word, par exemple). Lorsque vous l'avez acheté, vous avez dû préciser si vous aviez un PC ou un Mac, car il a été compilé différemment selon le type d'ordinateur auquel il est destiné. Ce programme ne peut fonctionner que sur la plate-forme pour laquelle il a été compilé. Ce n'est pas le cas des langages interprétés, qui ont un code commun à tous les types d'ordinateurs. La raison de cette polyvalence est que le code source est interprété du côté client par le logiciel adapté à la machine (voir figure 1-5). Ce genre de langage est donc bien adapté à Internet où le parc d'ordinateurs est très hétérogène. On peut ainsi envoyer le même code HTML à tous les navigateurs des internautes, quel que soit leur ordinateur. En revanche, cela oblige le serveur à envoyer tout le code source sur le poste client, laissant à quiconque la possibilité de le copier et de l'utiliser comme bon lui semble. Cet inconvénient n'est pas négligeable car, dans ces conditions, il devient difficile de protéger son code source et la confidentialité des informations qu'il pourrait contenir.

Nous verrons plus loin que le langage PHP n'hérite pas de ce défaut, car son code source est préinterprété et transformé en équivalent HTML côté serveur ; seul le code HTML ainsi produit est envoyé au client, ce qui préserve les sources PHP et leur contenu.

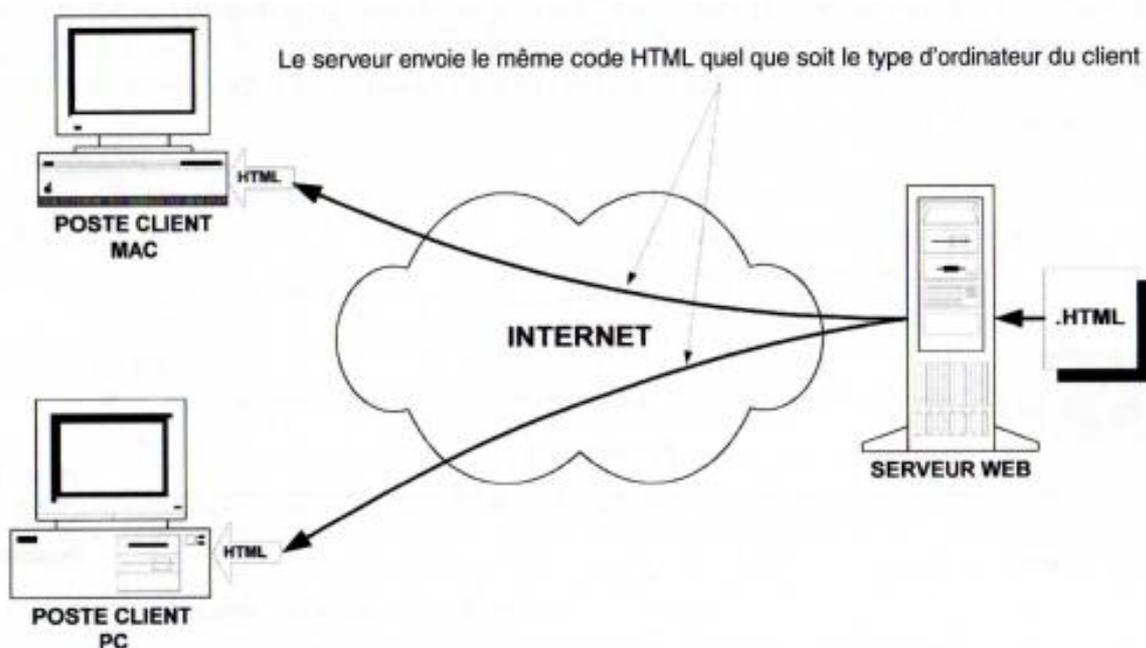
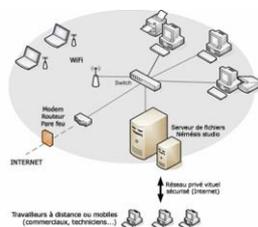


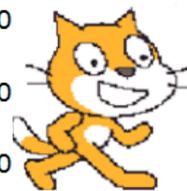
Figure 1-5

L'interprétation du code HTML côté client permet d'envoyer le même code quel que soit le type d'ordinateur de l'internaute. Chaque navigateur étant adapté à la plate-forme sur laquelle il est installé, il interprète le code HTML en l'adaptant aux particularités de l'ordinateur du client.





0100100100**1**0100100
 101001**1**01001000100
 0101**0**0010010010010



Les sites interactifs et les langages de script

Heureusement, l'évolution des techniques Internet permet désormais de développer des pages interactives beaucoup plus intéressantes et attractives pour l'internaute.

Pour créer de l'interactivité sur un site, le concepteur multimédia dispose de plusieurs technologies qui peuvent être exécutées côté client (JavaScript dans une simple page HTML ou ActionScript dans une animation Flash) ou côté serveur (PHP, ASP, JSP, CFML, etc.). Le choix du type de technologie dépend de l'application à mettre en œuvre, de son niveau de sécurité, de la qualité de son interface et de sa rapidité d'exécution.

Interactivité coté client avec JavaScript

JavaScript

JavaScript (à ne pas confondre avec Java) est un langage très largement employé sur Internet côté client, même s'il peut aussi fonctionner côté serveur. Il a été mis au point par Netscape Communications. Ses instructions sont incluses dans le code HTML des pages envoyées sur le poste client et sont traitées directement par le navigateur.

La solution la plus simple pour créer de l'interactivité consiste à intégrer quelques lignes de code ActionScript dans une page HTML. Lorsqu'une requête HTTP appelle la page HTML (voir étape 1

de la figure 1-6), le serveur Web la retourne au poste client afin qu'elle puisse être interprétée comme une page HTML classique (voir étapes 2 et 3 de la figure 1-6). Le script inclus dans la page est ensuite traité par le navigateur dès que l'événement pour lequel il a été programmé survient (voir étape 4 de la figure 1-6).

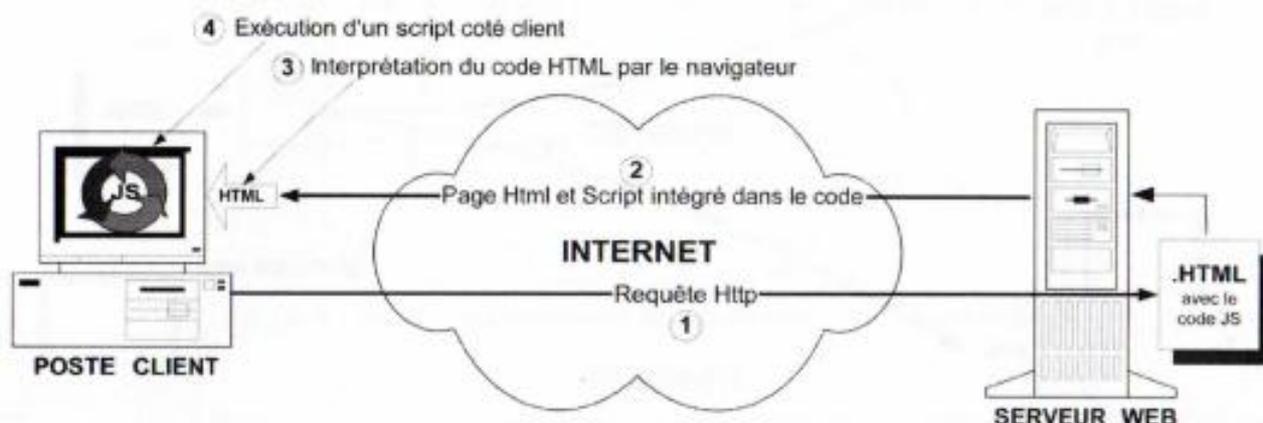
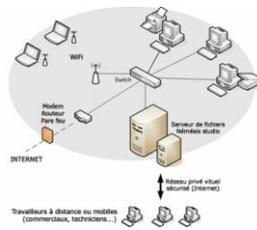


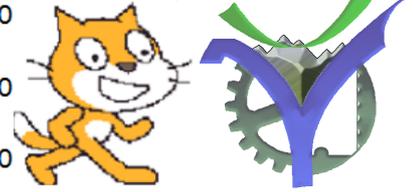
Figure 1-6

Utilisation d'un script côté client avec JavaScript : il existe une dépendance relative au navigateur client mais l'interactivité est rapide.





0100100100**1**0100100
 101001**1**01001000100
 0101**0**0010010010010



Les scripts côté client sont très réactifs car le script s'exécute directement sur le poste client. En revanche, les programmes JavaScript peuvent se comporter différemment selon le type d'ordinateur et la version du navigateur. Par exemple, un script en JavaScript peut parfaitement fonctionner sur Firefox mais poser des problèmes avec Internet Explorer ou créer des erreurs sous IE 5 alors qu'il fonctionne sous IE 6. De même, les résultats peuvent varier selon qu'on utilise un PC ou un Mac. Tout cela impose au concepteur multimédia de réaliser des tests importants s'il désire que sa page interactive fonctionne sur toutes les plates-formes et dans toutes les configurations.

Interactivité coté client avec ActionScript (Flash)

Une autre solution pour mettre en œuvre de l'interactivité côté client consiste à créer une animation Flash dans laquelle sera intégré un programme ActionScript. Le script étant présent sur le poste client, l'application est aussi réactive qu'avec le Javascript, mais la qualité de l'interface et son esthétique expliquent l'engouement toujours croissant des développeurs pour cette solution.

Cependant, même si le plug-in Flash est présent par défaut sur plus de 98 % des navigateurs actuels (source site Macromedia), sa version peut ne pas être compatible avec celle de votre animation Flash. Il faudra vérifier ce paramètre à l'aide d'un détecteur de version (en utilisant le Dispatcher de Macromedia, par exemple) afin d'aiguiller le visiteur vers l'espace de téléchargement du plug-in si sa version n'est pas adaptée à votre animation.

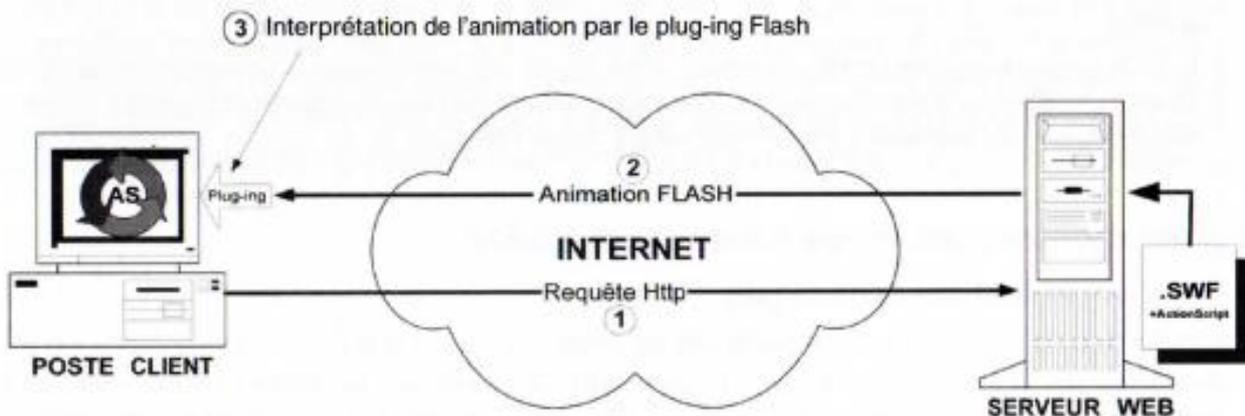
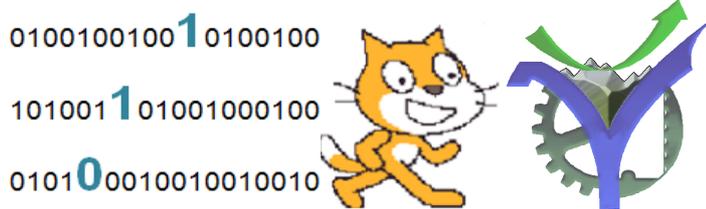
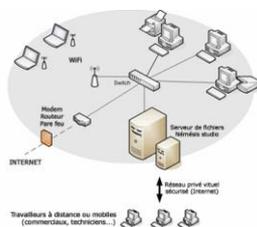


Figure 1-7

Utilisation d'une animation Flash avec ActionScript : l'interactivité est aussi rapide qu'avec JavaScript côté client, mais l'interface est plus esthétique.





Interactivité coté serveur avec PHP

L'interactivité peut être placée côté serveur. Dans ce cas, le serveur Web doit disposer d'un préprocesseur PHP afin de traiter les scripts PHP intégrés dans la page avant de l'envoyer au poste client qui en a fait la demande (voir étapes 1 et 2 de la figure 1-8).

Si on le compare avec un script côté client, la réaction à un événement d'un script côté serveur est beaucoup plus lente car elle nécessite l'envoi d'une requête au serveur (voir figure 1-8, étape 1), son exécution sur le serveur (étape 2), le retour de la réponse par le réseau Internet (étape 3) et le chargement de la page dans le navigateur (étape 4).

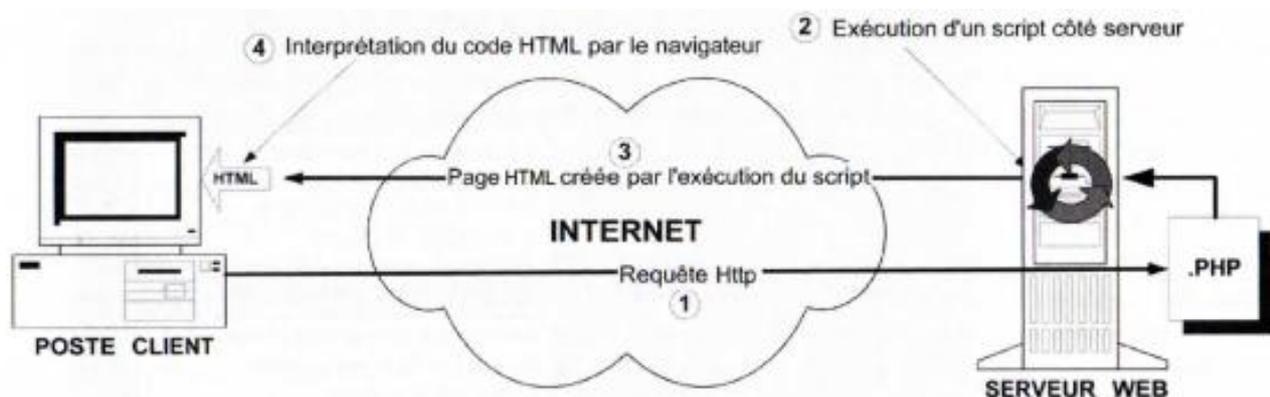


Figure 1-8

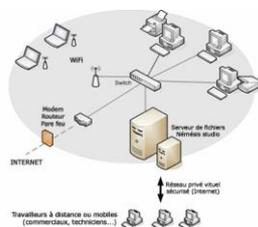
Utilisation d'un script côté serveur : il n'y a pas de dépendance vis-à-vis du navigateur client mais l'interactivité est plus lente.

En revanche, les langages côté serveur sont indépendants de la plate-forme du client ou de la version de son navigateur. En effet, l'interprétation du script est réalisée côté serveur et le code envoyé vers l'ordinateur du client est compatible avec le standard HTML et donc interprété de la même manière par tous.

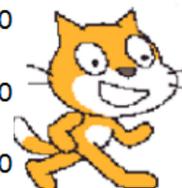
À noter

Parmi les inconvénients des scripts côté serveur, il faut signaler que leur utilisation nécessite la disponibilité d'un serveur adapté. Même si les offres des hébergeurs qui proposent des serveurs intégrant des scripts dynamiques sont de plus en plus accessibles, il faut en tenir compte lors de votre choix.





0100100100**1**0100100
 101001**1**01001000100
 0101**0**0010010010010



Les sites dynamiques et les bases de données

Création de modèles dynamiques

L'exécution du script côté serveur permet de créer une page « à la volée » lors de son exécution par le préprocesseur PHP intégré au serveur. La page ainsi créée contient les mêmes informations qu'une simple page HTML. Elle peut donc être interprétée sans problème par le navigateur côté client (voir figure 1-9). Lors de la création de cette page, les scripts intégrés au fichier dynamique sont exécutés et, si nécessaire, établissent une connexion à un serveur de données. Avec ce processus, la page dynamique devient un modèle de présentation des informations. Ce modèle est personnalisé par des contenus différents selon la requête du client.

Figure 1-9

Exemple de code d'une page dynamique produisant le même affichage que la page de la figure 1-2.

```

<?php include("requeteBase.php") ?>
<HTML>

<HEAD><title>Page Dynamique</title></HEAD>

<BODY>

<p>
<font size="7"><?php echo $titre ?></font>
</p>

<p><?php echo $descriptif ?></p>

<p></p>

<p>
<a href="page.php?id=1">page1 </a>
<a href="page.php?id=2">page2 </a>
</p>

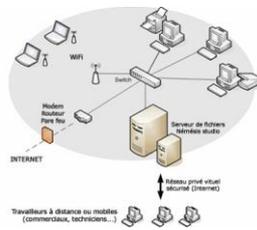
</BODY>
</HTML>
    
```

Si des informations doivent être récupérées d'une base de données, une requête serveur permettra de les intégrer dans la page

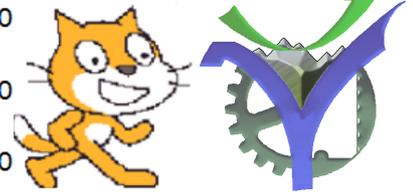
La page dynamique peut être personnalisée selon différentes variables affichées à l'écran.

Les images ou les liens hypertextes pourront aussi être gérés dynamiquement.





010010010010100100
101001101001000100
01010001001001010



Il n'est donc plus nécessaire, par exemple, de créer une page spécifique pour présenter chaque produit d'un catalogue : une seule page dynamique peut être utilisée. Il suffit de lui indiquer l'identifiant du produit demandé grâce à une variable qui lui est transmise en même temps que son appel ; la page renvoyée au client contient toutes les informations et photos relatives au produit concerné. L'arborescence du site est simplifiée puisque cette page dynamique remplace les nombreuses pages statiques correspondant à chaque produit (voir figure 1-10).



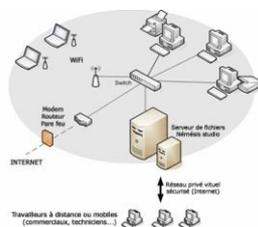
Figure 1-10

L'arborescence d'un site dynamique contient beaucoup moins de fichiers que celle d'un site statique : les fichiers dynamiques faisant office de modèles, il suffit de leur envoyer une variable différente pour qu'ils se personnalisent à la demande.

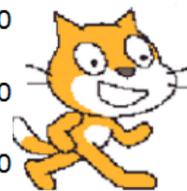
Pourquoi utiliser une base de données

Nous venons d'expliquer que l'utilisation des scripts crée de l'interactivité dans un site. En effet, ils permettent d'intégrer dans une page des instructions conditionnelles pour réaliser des traitements différents en fonction de l'état des variables testées. Les pages créées par ces scripts sont élaborées avec des informations contextuelles fournies par l'internaute lui-même ou issues d'un traitement réalisé à partir de celles-ci. Ces informations sont exploitables uniquement le temps de la session active et ne peuvent pas être mémorisées d'une session à l'autre, ce qui limite considérablement les applications utilisant uniquement des scripts serveur. Une première solution pour conserver ces informations consiste à les enregistrer dans de petits fichiers (cookies) côté client, afin de récupérer le profil de l'internaute lors de sa prochaine visite. Cependant, de nombreux internautes interdisent l'enregistrement d'informations sur leur ordinateur. Une deuxième solution s'appuie sur l'enregistrement de ces informations dans des fichiers de données, mais du côté serveur cette fois. Dans ce cas, la disponibilité de ces fichiers n'est plus tributaire du bon vouloir de l'internaute, mais cette solution manque de souplesse dans l'exploitation des informations et son organisation devient vite ingérable pour des sites conséquents. La troisième solution est d'utiliser une base de données dans laquelle on stocke toutes les informations utiles aux applications du site. Le script côté serveur contient alors les procédures de connexion à la base de données et des instructions spécifiques pour lire, ajouter, modifier ou créer des enregistrements. Même si cette solution nécessite la présence d'une base de données et le développement de scripts de gestion de ses enregistrements, c'est de loin la plus efficace et la plus souple dans une grande majorité d'applications. Elle est actuellement employée sur la plupart des sites professionnels définis comme dynamiques. Les sites dynamiques sont donc caractérisés par le fait qu'ils fonctionnent avec des scripts côté serveur et qu'ils exploitent les informations issues d'une base de données.





0100100100**1**0100100
 101001**1**01001000100
 0101**0**0010010010010



Pour pouvoir exploiter une base de données, le système doit être organisé selon une architecture à trois niveaux (dite architecture trois tiers) mettant en relation le client, le serveur Web et la base de données. Même si la base de données est souvent installée sur le même ordinateur que le serveur Web, ce modèle est valable dans la plupart des cas (voir figure 1-11 et figure 1-12).

Application dynamique avec PHP-MySQL

Les étapes de traitement d'un site dynamique classique (sans animation Flash) sont les suivantes :

1. Le poste client envoie une requête HTTP sur un fichier comportant un script PHP (voir étape 1 de la figure 1-11).
2. Le serveur Web localise le fichier dynamique et l'exécute (voir étape 2 de la figure 1-11).
3. Si le script nécessite des informations issues de la base de données, il adresse une requête SQL au serveur de la base de données MySQL qui lui renvoie les informations demandées (voir étape 3 de la figure 1-11).
4. Les informations issues de la base de données sont ensuite intégrées dans la page dynamique (voir étape 4 de la figure 1-11).
5. La page ainsi créée est ensuite envoyée au client et interprétée comme une simple page HTML par le navigateur (voir étapes 5 et 6 de la figure 1-11).

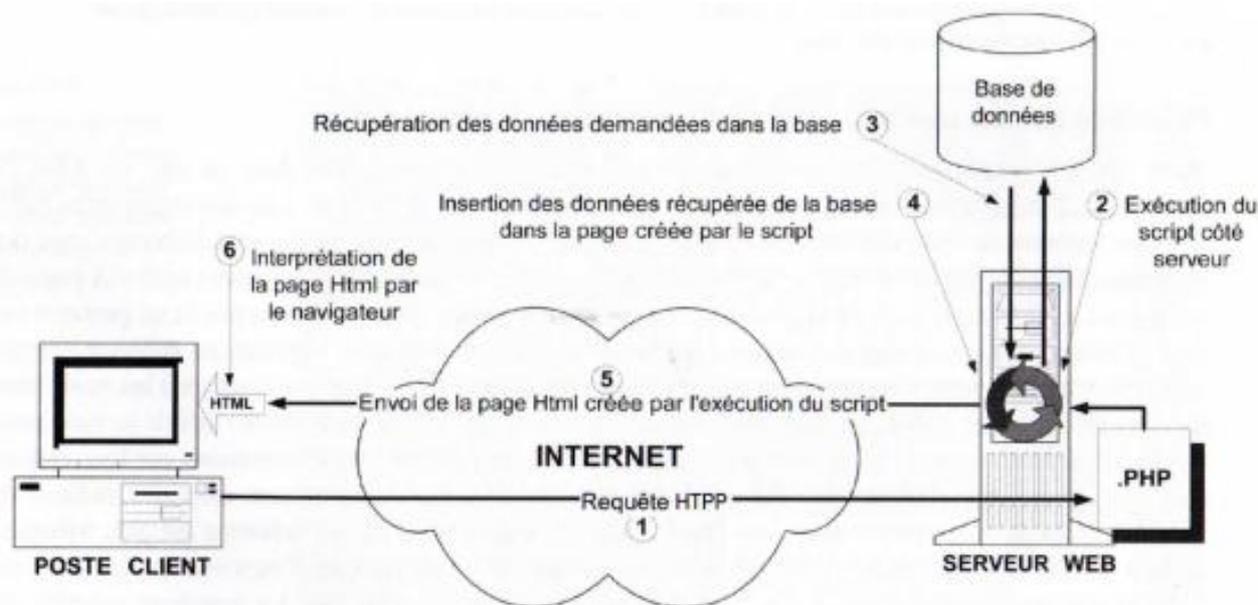
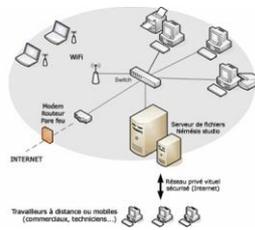


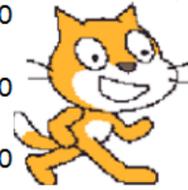
Figure 1-11

Fonctionnement d'un site dynamique classique PHP-MySQL.





010010010010100100
101001101001000100
010100010010010010



Application dynamiques avec PHP-MySQL et Flash

Avec un site dynamique classique, le serveur Web envoie au navigateur une page HTML entière générée à la volée par le préprocesseur PHP. Le fonctionnement est différent dans le cas d'une animation Flash dynamique. En effet, la première étape consiste à charger l'animation Flash dans le navigateur (voir étapes 1, 2 et 3 de la figure 1-12) comme nous l'avons expliqué dans la partie consacrée à l'interactivité côté client avec ActionScript (revoir figure 1-7). Ensuite, lorsqu'un événement nécessitant une information dynamique survient (voir étape 4 de la figure 1-12), une requête est envoyée directement par l'animation Flash au serveur Web (voir étape 5 de la figure 1-12). Cette requête cible cette fois un script PHP

spécifique (voir étape 6 de la figure 1-12) qui, contrairement à celui d'un site dynamique classique, ne génère pas une page HTML dynamiquement mais crée et renvoie à l'animation Flash une structure de données au format couple variable-valeur ou, mieux encore, au format XML (voir étapes 7, 8, 9 et 10 de la figure 1-12). Cette structure de données est ensuite réceptionnée puis traitée par l'animation Flash qui exploite les informations reçues comme de simples variables internes.

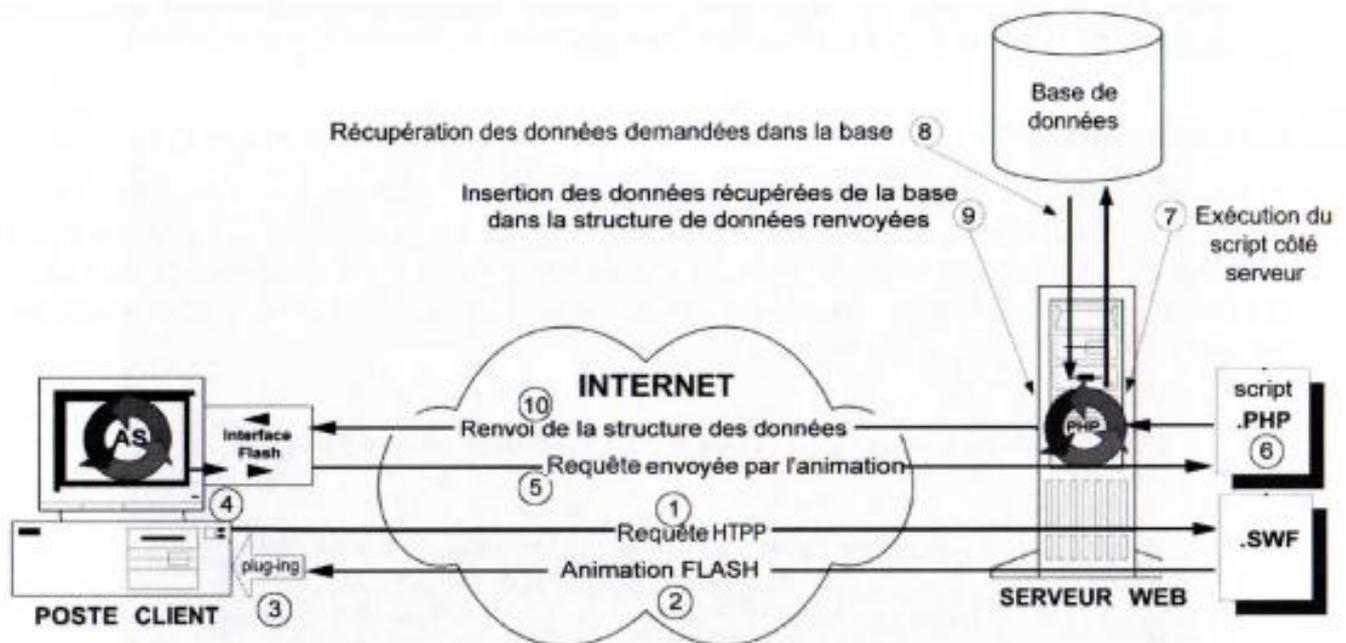


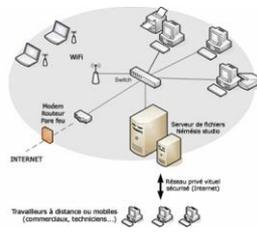
Figure 1-12

Fonctionnement d'un site dynamique PHP-MySQL couplé avec une application client Flash.



2nd ICN

2nd ICN



0100100100**1**0100100
101001**1**01001000100
0101**0**0010010010010

