

COMMENT COMMUNIQUENT LES FORMULAIRES

Lorsqu'un utilisateur appelle une page HTML statique, le serveur la prend dans votre répertoire (dossier) et la lui envoie, avec l'aspect exact qu'elle avait lorsque vous l'avez placée dans votre serveur Web. Cependant, il est possible à toute page envoyée par le serveur d'être générée à la volée lorsqu'elle est appelée par l'utilisateur. Par exemple, imaginons que vous possédez une boutique de livres en ligne. Un champ (select) de votre formulaire permet à l'utilisateur de faire un choix parmi les livres disponibles. Mais, chaque fois que la disponibilité des livres (ou leur nombre) changera, il vous faudra ouvrir la page dans GoLive et procéder vous-même aux modifications. Au lieu de cela, vous pouvez faire en sorte que le serveur fasse appel à un intermédiaire qui lira la page contenant votre formulaire, prendra connaissance des balises et commandes spéciales qu'il contient, vérifiera quoi rechercher dans une base de donnée (une recherche de tous les titres disponibles dans cet exemple), fusionnera les résultats dans une sorte de page modèle et présentera à l'utilisateur une liste de titres mise à jour.

L'intermédiaire qui effectue ce travail sur le serveur est appelé un *server-side processing tool* (un outil de traitement coté serveur). Quel que soit l'outil de traitement considéré, vous utiliserez toujours un formulaire HTML comportant les habituels champs de texte, bouton radio, cases à cocher, etc. Bien sûr, vous utiliserez GoLive pour créer ce formulaire. En plus du HTML standard, vous devrez utiliser certaines commandes et balises spéciales, spécifiques au programme de traitement, qui gèrera le formulaire côté serveur avant de l'envoyer à l'utilisateur.

La page que vous créez dans GoLive est celle qui contient ces commandes (balises) spéciales, spécifiques à la technologie employée sur le serveur. La page envoyée à l'utilisateur est une combinaison de votre page et du résultat de la requête qui sera insérée dans ces balises. Si l'utilisateur regarde la source de votre page, il ne verra que des codes HTML standards, car ils auront été insérés durant le processus.

Les sections suivantes décrivent quelques programmes utilisés par les serveurs, et ce qu'ils font.

Les formulaires peuvent faire bien plus que simplement recueillir l'information. Ils peuvent rendre une page plus vivante en y ajoutant de l'interactivité. Lorsque vous créez une page Web standard, y plaçant texte, images, animations, etc., vous ne faites que créer une page statique — c'est-à-dire une page qui reste toujours identique à elle-même. Cependant, un formulaire donne une opportunité de communication entre l'utilisateur et le site Web. Chaque fois que vous utiliserez ce qu'aura entré l'utilisateur dans un formulaire, pour aider à la formation de la page qu'il recevra, vous aurez créé une page Web *dynamique*.



Une page Web dynamique est une page personnalisée pour chaque utilisateur ou type d'utilisateur. Cette personnalisation se fonde sur les informations directement entrées par l'utilisateur, comme les réponses entrées dans un formulaire, ou sur celles entrées indirectement par ce dernier, comme son environnement informatique, par exemple. La page qui est renvoyée à l'utilisateur est souvent construite à partir de données sélectionnées dans une base de données.

En conjonction avec une base de données finale, les formulaires peuvent permettre à l'utilisateur de rechercher une information spécifique. La Figure C.1 donne un exemple de base de données finale. La base de donnée Holli Boyd contient des cartes postales fusionnées en un formulaire généré à partir des informations entrées par l'utilisateur, pour créer une page à la volée. Le système envoie la carte au destinataire. Les forums de discussion utilisent aussi des formulaires pour sortir des données d'une base de données. Le panier électronique et les sites d'enchères sont une autre utilisation des formulaires.

Il est important de réaliser que GoLive, bien qu'il vous permette de créer des formulaires sur une page (comme décrit au Chapitre 16), ne fournit pas les moyens de communiquer avec les actions du serveur qui sont requises pour donner l'interactivité. Lorsque l'utilisateur clique sur le bouton d'envoi, la commande est envoyée au serveur Web, mais ce dernier ne fait que servir des pages et ne pratique pas de recherche dans des bases de données. De ce fait, un programme intermédiaire, appelé parfois *middleware*, devient nécessaire. Les commandes appelant ce programme sont cachées dans le formulaire qui envoie la requête à la base de données. (Vous insérez ces balises cachées, dans le formulaire, à l'aide de GoLive).

The screenshot shows a web browser window titled "HSB Designs Virtual Postcards". The address bar shows "http://www.hsbdesigns.com/postcards/cards.html". The page content includes a "New Home" header with a small image of a house. Below the header, the form is structured as follows:

- STEP #2: NAME AND E-MAIL INFORMATION**
In the boxes below, please enter both your's and the recipient's name and e-mail addresses.
- PLEASE BE CERTAIN OF RECIPIENT'S E-MAIL ADDRESS**
- Your Name:
- Your E-Mail Address:
- Recipient's Name:
- Recipient's E-Mail Address:
- STEP #3: ENTER YOUR TITLE & MESSAGE**
In the box below, please enter your title and message.
- Card Title:
- Your Message:
- STEP #4: SIGN YOUR CARD**
Please fill in how you would like to sign your card. Examples would be:
• Love, Your Sweetheart

Figure C.1 : La carte postale électronique d'Holli Boyd (à www.hsbdesigns.com/postcards/cards.html). Un formulaire recueille les informations, crée une page Web et envoie ensuite un e-mail informant le destinataire.

Le serveur reconnaît ces balises cachées comme appartenant à une autre application, et envoie les instructions au programme intermédiaire (ou *agent*) auquel elles sont adressées. Ce programme reconnaît ces instructions, car elles sont écrites dans un langage spécial qui lui est propre et agit en conséquence. Son travail peut être d'envoyer des instructions à une base de donnée et d'attendre la réponse, ou encore d'envoyer une demande d'autorisation de carte de crédit et d'attendre la réponse. Quelle que soit sa tâche, il attend une réponse et crée une page dynamique, dans laquelle il incorpore les informations reçues. Tout ce qui se passe sur le serveur, en dehors de ses actions normales, est appelé *traitement côté serveur*.

Plusieurs types d'agents sont disponibles pour faire du traitement côté serveur. Comprendre ce qu'ils font peut vous aider dans le choix d'un hébergeur et/ou d'un serveur. après un bref survol de cette communication côté serveur, vous comprendrez comment créer votre formulaire.

CGI — Common Gateway Interface

La méthode la plus commune de communication entre un formulaire et le traitement qui survient après l'envoi de la requête à un programme est appelée Interface de Passerelle Commune (CGI). Ce script fournit les règles par lesquelles le programme communique avec le serveur Web. Tout comme il existe différents programmes de traitement de texte ou d'images, il existe de nombreux scripts CGI. Tout comme chaque programme de traitement de

texte réalise différentes fonctions particulières, qui lui sont propres, au-delà des tâches générales de création de document de texte, chaque script CGI réalise des fonctions différentes, alors que la tâche générale — la communication vers et depuis le serveur — demeure la même.

Les scripts CGI varient considérablement. Ce peut être un simple document texte contenant quelques commandes conçues pour des tâches spécifiques, ou un programme de service complet. Vous pouvez créer vos propres scripts CGI ou en acheter un sur le commerce.

La manière dont se comporte chaque CGI varie d'une plate-forme à une autre. L'information envoyée au serveur, par l'envoi de la requête du formulaire, comporte une *action* qui contient le nom du CGI, d'une manière ou d'une autre. Elle contient aussi certaines informations ou commandes que vous aurez incorporées au formulaire. Chaque programme CGI sait comment prendre l'information qui lui a été donnée, comment la traiter et la retourner au serveur.

Comment fonctionne le CGI

Voici comment un programme CGI fonctionne (vous pouvez suivre la méthode en vous référant à la Figure C.2).

1. Lorsque l'utilisateur clique sur le bouton d'envoi, le navigateur regroupe les informations du formulaire et les envoie au serveur Web. Le navigateur transporte toujours l'information au serveur. Dans ce cas, il en transporte simplement un peu plus. L'élément supplémentaire qu'il transporte est l'*Action* du formulaire.
2. Le serveur reçoit l'information et remarque que les instructions ne sont pas celles d'un formulaire normal. Il reconnaît l'instruction Action du formulaire qui comporte le nom du CGI (ou d'un autre programme similaire) et sait ainsi où le rediriger.
3. Le CGI interprète la requête et exécute la commande incorporée. Par exemple, si la commande consiste à rechercher dans une base de données tous les enregistrements contenant un mot ou un événement particulier, le programme traduit la commande dans un langage que comprendra ladite base de données et attendra le résultat de la recherche. Lorsqu'il l'aura acquis, le CGI emballera le résultat de la recherche, afin que le serveur Web puisse la renvoyer.
4. Après avoir réalisé ce travail spécifique, le CGI renvoie son information au serveur Web.
5. Le serveur renvoie l'information au navigateur.
6. Le navigateur affiche le résultat complet.



Figure C.2 : Une description visuelle de la manière dont un formulaire communique avec une base de données et renvoie la réponse.

Réinvention de la roue ?

De nombreux CGI sont disponibles. Certains ont été spécifiquement conçus pour communiquer avec un certain type de base de données. Dans ce cas, vous pouvez permettre à vos utilisateurs d'envoyer des informations à la base de données, ainsi que cela se pratique avec les livres d'or. Une autre utilisation commune de CGI est le traitement d'un formulaire et l'envoi d'un e-mail de confirmation. Une autre encore permet de créer un panier électronique.

Le choix portant sur le CGI à utiliser dépend de plusieurs facteurs. Le premier d'entre eux est le service que vous souhaitez proposer et la plate-forme qu'utilise le serveur. (La fonctionnalité du CGI doit être écrite pour s'exécuter sur la plate-forme sur laquelle tourne le serveur, comme Windows, UNIX ou Mac.) Votre fournisseur Internet ou les paramètres de l'hébergeur sont un autre souci majeur. Vous pouvez avoir l'autorisation d'utiliser n'importe quel CGI, ou bien, être limité à certains, bien spécifiques. Cela peut être gratuit ou non. Si vous devez utiliser une base de données pour recueillir et distribuer l'information, c'est encore un autre cas de figure, car un programme CGI doit être capable de communiquer avec ce type de base de données.

Vous pouvez écrire votre propre script CGI, si vous vous en sentez capable ou en utiliser un du commerce. Si vous n'avez besoin que d'un échange limité d'informations, il est préférable d'écrire votre propre script ou d'en emprunter un à quelqu'un. (Il est habituel que les gens s'entraident en partageant des ressources sur la mailing liste de GoLive Talk. Reportez-vous au Chapitre 2 pour en savoir plus.) Si vous échangez beaucoup de données ou que vous ayez à faire à de nombreuses variables, un script CGI du commerce peut être un meilleur choix. Les paniers électroniques, par exemple, sont des scripts complexes, car ils doivent garder la trace des achats de l'utilisateur et effectuer ensuite la méthode de paiement. Plusieurs paniers électroniques, encore appelés solution e-commerce, sont aujourd'hui disponibles. Certains sont des produits commerciaux, alors que d'autres sont librement partagés. Si vous envisagez d'en acheter un ou d'en implémenter un nouveau en partage libre, menez une investigation minutieuse. Les scripts CGI varient en qualité. Un script mal écrit peut ralentir, voire planter le serveur, tandis qu'un script bien écrit est performant tant en rapidité qu'en qualité.

Pour déterminer quel script CGI vous pouvez utiliser, commencez par vérifier auprès de votre fournisseur d'accès ou auprès de votre hébergeur pour répondre aux questions suivantes :

- ◆ Quel est sa politique concernant les CGI?
- ◆ En fournit-il avec mon compte?
- ◆ Si oui, que fait chacun d'entre eux?
- ◆ En coûte-t-il pour son utilisation ou existe-t-il d'autres facteurs de contrainte que vous devriez connaître?

Plug-in — plus proche que l'intégration CGI

Vous êtes probablement familier avec le concept de plug-in, car Photoshop et d'autres programmes similaires les utilisent depuis longtemps. La clé du plug-in est de se brancher (plug in) vers un programme associé, plutôt que d'agir en tant que programme totalement indépendant. Lorsque vous utilisez Photoshop, par exemple, vous pouvez accéder à un plug-in depuis l'application, plutôt que d'en lancer un séparément.

De même, un serveur Web peut disposer de plug-ins ou modules s'ils sont souvent appelés. Cela élimine une étape dans le processus de communication, qui devrait, sinon, prendre place entre le serveur Web et le CGI. En fait, cela élimine deux étapes — l'une quand la requête arrive, l'autre quand les informations sont renvoyées.

Les plug-ins ou modules sont considérablement plus rapides que les CGI. Cependant, ils sont plus compliqués à écrire, car ils ont besoin d'être codés par l'interface de programmation de l'application (API) qui comporte la propriété de le lier au serveur. Un module écrit pour un serveur ne sera pas compatible avec un autre serveur. Aussi, comme il est lié au serveur, il doit être installé, et le serveur doit être relancé (car un module doit être chargé en mémoire pour être fonctionnel).

La plupart des serveurs Web comportent une interface de plug-in ou de module type. Les serveurs Apache possèdent des modules. Netscape a le NSAPI. Microsoft a le ISAPI, qui est très populaire dans le monde NT. Votre hébergeur peut très bien faire tourner des plug-ins ou des modules, dont vous pouvez tirer parti. Si vous en connaissez un, vous pouvez même le demander spécifiquement.

En raison de tout ce que cela implique, il est préférable que vous n'écriviez pas de plug-ins. C'est le domaine des développeurs de logiciels. C'est la raison pour laquelle les CGI demeurent populaires.



L'un des modules du serveur Apache est nommé "mod_perl". C'est un interpréteur Perl imbriqué dans le serveur. Il vous permet de faire tourner des scripts en Perl, directement dans le serveur. C'est plus rapide que faire tourner ces scripts via CGI (langage dans lequel sont écrits les scripts CGI).

Du point de vue de la création de formulaire, il n'existe aucune différence entre un plug-in et un CGI, et j'utiliserai donc le terme CGI dans le reste de cet appendice. (Un CGI est, en réalité, un script ou un programme, mais, comme j'ai pris la liberté d'écrire simplement CGI au lieu de script CGI ou programme CGI, je continuerai ainsi.)

Les autres technologies côté serveur

Plusieurs outils plus évolués existent pour créer des pages dynamiques, comme ColdFusion, Active Server Page (ASP), Personal Home Page (PHP) et Java ServerPages (JSP). Certains sont des programmes propriétaires, ce qui signifie qu'ils doivent être utilisés sur le serveur spécifique du fabricant. Vous ne les utiliserez que si vous avez l'intention définitive d'héberger votre site sur ce type particulier de serveur. Les autres outils peuvent être utilisés sur une variété de serveurs. Tout comme pour CGI, c'est hors de propos de ce livre. Cependant, je vous présente ici certaines des technologies côté serveur les plus évoluées et les plus populaires.

ColdFusion

C'est un produit de chez Allaire. En utilisant ColdFusion, les programmeurs créent des gabarits, qui sont des fichiers contenant des balises HTML standards et des balises ColdFusion. Vous pouvez créer ces gabarits dans GoLive, et les garder dans votre Fenêtre de Site avec le reste de vos fichiers.

Les balises ColdFusion ont la forme `<CFIenomdelacommande> xyz </CFIenomdela commande>`, et permettent des requêtes à une base de données, l'appel à des objets COM et plus encore. La force de ColdFusion réside dans le fait qu'il ne requiert pas la connaissance d'un langage de programmation traditionnel, car ces balises ressemblent à celles de HTML. En réalité, Allaire appelle ses balises CFML (ColdFusion Markup Language). Alors que les premières versions de ColdFusion fonctionnaient via CGI, les versions récentes utilisent un API — un protocole de programmation fourni par de nombreux serveurs Web, comme Apache. ColdFusion tourne en tant qu'application séparée (appelée Serveur ColdFusion), mais, à cause de l'API, il est fermement connecté au logiciel serveur. Il tourne sur les plates-formes Windows NT et Solaris, et bientôt sur des machines Linux.

GoLive ne reconnaît pas les balises ColdFusion (CF) comme des balises valides. Deux manières garantissent que GoLive respectera néanmoins ces codes : en les entrant dans la configuration de la base de données Web ou en les imbriquant dans le conteneur `<noedit></noedit>`.

Dans ColdFusion, vous pouvez avoir un gabarit (une page) appelé `choisirLivre.cfm` qui pourrait ressembler à ceci :

```
<HTML>
<HEAD>
<CFQUERY NAME="Books" DATASOURCE="BooksDatabase">
select title from BooksTable where InStock=' True'
</CFQUERY>
</HEAD>
<BODY>
<FORM ACTION="BuyBook.cfm" METHOD="POST">
<CFSELECT QUERY="Books"></CFSELECT>
<INPUT TYPE="Submit" VALUE="Proceed to checkout">
</FORM>
</BODY>
<HTML>
```

Lorsque les utilisateurs parcourent la page, ils voient une page HTML standard, avec un champ de sélection dont les options affichent les titres des livres. S'ils regardent la source, ils ne verront que du pur code HTML et aucune balise ColdFusion comme `CFSELECT`, car le serveur aura exécuté la requête et remplacé les balises `CF` par des données et du HTML. L'ASP et le PHP fonctionnent de manière similaire.

ASP — Active Server Pages

Cet outil est une offre Microsoft au monde des pages Web dynamiques. Il est conçu pour tourner exclusivement sur des serveurs Microsoft IIS, sous Windows NT et 2000. Cependant, un programme nommé Chili!Soft ASP permet aux serveurs UNIX et à d'autres plates-formes de faire tourner ASP. (Rien n'existe pour Mac au moment de l'écriture de ces lignes.)

En utilisant ASP, les programmeurs créent des pages à l'aide de HTML et de scripts écrits en VBScript, JScript ou Perl. Les balises ASP ont la forme `<% xyz %>`. Ces scripts indiquent quoi faire au serveur. (Ils n'auront aucune utilité sur les autres serveurs, car ces balises sont des codes propriétaires à la technologie ASP.)

Vous pouvez utiliser Les liens Dynamiques de GoLive (Dynamic Link) pour créer des pages ASP.



Reportez-vous au Chapitre 24 pour en savoir plus sur Dynamic Link.

PHP (officiellement "Hypertext Preprocessor")

PHP est un langage de script imbriqué dans HTML côté serveur. Il est libre et demeure une alternative aux autres programmes intermédiaires du commerce.



La gestion des codes PHP dans le code HTML s'est beaucoup améliorée avec GoLive 5. Les problèmes rencontrés sous GoLive 4 avec les citations et les balises de fin, dans le code PHP, sont maintenant résolus.

Au lieu d'écrire un programme avec de nombreuses lignes de commandes, pour afficher une page HTML, vous écrivez une page HTML avec du code inclus à l'intérieur, afin de réaliser une action précise (dans ce cas-là, afficher du texte). Le code PHP est inclus entre une balise de début et une balise de fin, qui permettent au navigateur de passer en "mode PHP". Considérez l'exemple suivant :

```
<html>
<head>
<title>Example</title>
</head>
<body>
<?php echo "Hi, I'm a PHP script!"; ?>
```

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

Le code PHP est entre une balise spéciale de début et une balise de fin, qui vous permettent d'entrer et sortir du mode PHP. Il tourne sous les plates-formes Windows NT, Windows 2000, UNIX, Mac OS X et Linux. Il peut fonctionner via CGI ou comme partie du serveur Web Apache. Plus d'un million de serveurs, aux quatre coins de la planète, ont installé PHP. (Vous devez contacter votre hébergeur pour savoir si le serveur gère PHP.) PHP s'adapte très facilement, dans la mesure où il fournit quatre balises différentes que vous pouvez imbriquer dans votre code HTML.

Ecrire une base de données compatible avec une page Web en PHP est incroyablement simple. En effet, PHP gère les bases de données d'Adabas D, d'InterBase, de Solid, de dBase, de mSQL, de Sybase, d'Empress, de MySQL, de Velocis, de FilePro, d'Oracle, d'Unix, de dbm, d'Informix et de PostgreSQL. D'autres systèmes de base de données, comme Microsoft SQL, sont gérés via ODBC. PHP gère aussi d'autres services de communication, en utilisant les protocoles comme IMAP, SNMP, NNTP, POP3 ou même HTTP. Vous pouvez également ouvrir des connexions et interagir en utilisant d'autres protocoles.

PHP peut être utilisé pour de nombreuses applications sur votre page Web, comme le panier électronique, le système de gestion de contenu, la création de pages dynamiques, la personnalisation de logos, la sécurité et plus encore.

L'exemple précédent utilisé avec ColdFusion peut prendre cet aspect avec PHP :

```
<HTML>
<HEAD>
<?php mysql_connect("user", "pass", "bookstore")
$query = "select title from BooksTable where InStock=' True' " ;
$result = mysql_query("BooksDatabase", "$query"); ?>
</HEAD>
<BODY>
<FORM ACTION="BuyBook.php" METHOD="POST">

<SELECT name="selectName" size="1">
<?php while($row = mysql_fetch_array($result)) {
print '<OPTION value="" ;
echo $row["ISBN"];
print '>';
echo $row["title"];
print '</OPTION>'; ?>
</SELECT>

<INPUT TYPE="Submit" VALUE="Proceed to checkout">
</FORM>
</BODY>
<HTML>
```

Lorsque les utilisateurs parcourent la page, ils voient une page HTML standard avec un champ de sélection, dont les options affichent les titres des livres. S'ils regardent la source, ils ne verront que du pur code HTML et aucune balise PHP, car le serveur aura exécuté la requête et remplacé les balises PHP par des données et du HTML.

Vous pouvez tout apprendre sur PHP à www.php.net. Vous y trouverez de nombreux exemples et une documentation en français. Le manuel PHP en français est téléchargeable gratuitement depuis ce site.

Choisir une technologie

Avant de choisir parmi ces différentes technologies, vous devez considérer plusieurs facteurs, dont les suivants :

- ◆ **La plate-forme serveur.** C'est-à-dire, la machine et le système d'exploitation. Par exemple, un Pentium III tournant sous Windows NT, un G4 tournant sur Mac OS X Server ou un serveur Sun Enterprise tournant avec Solaris.
- ◆ **Le logiciel du serveur Web.** Celui qui traite les requêtes HTTP. Par exemple, Apache, IIS, Zeus, Xitami, AOLServer ou WebSTAR.
- ◆ **Le coût.** De nombreux facteurs peuvent influencer le coût d'une technologie particulière : le coût de l'hébergement, le temps de développement, le prix de la licence et celui de la maintenance, s'il s'agit d'un site important et que vous faites appel à une tierce partie.
- ◆ **La facilité de programmation.** Vous devez considérer vos acquis et la courbe d'apprentissage qu'impliquera chaque technologie (ColdFusion, par exemple, sera plus facile si vous connaissez HTML, alors que PHP sera un meilleur choix si vous pouvez déjà programmer en JavaScript, Perl ou C). Considérez aussi bien la construction du site que sa maintenance. Considérez encore la disponibilité des programmeurs dans votre région si vous devez faire appel à eux.
- ◆ **La portabilité.** Demandez-vous si vous pouvez facilement transporter votre site d'un serveur à l'autre, et s'il est facile de trouver un autre serveur qui gère la technologie que vous utilisez.
- ◆ **Echelle.** Si votre site génère beaucoup de passages, la technologie sera-t-elle assez puissante pour gérer ce flux?
- ◆ **Support.** Un support technique pour cette technologie est-il disponible? Existe-t-il une grande communauté d'utilisateurs avec laquelle vous pourriez interagir si vous rencontrez des problèmes?

Utiliser une technologie

Si vous créez un formulaire qui utilise un traitement côté serveur et requiert des balises propriétaires, je vous recommande de construire votre page et votre formulaire d'abord. Ensuite, une fois tous les éléments visuels placés (ainsi que le HTML standard), passez en mode Source et ajoutez les codes spécifiques à la technologie employée.

Reportez-vous à la section "Négocier avec les balises HTML non standards" du Chapitre 16, qui détaille les éléments de formulaire.

Comprendre les problèmes de GoLive avec des balises non-HTML.

Souvenez-vous que, parfois, GoLive peut avoir des problèmes avec quelques codes. En fait, aucun programme de création de page ne peut connaître, identifier et rendre facile le travail de chaque CGI et de chaque application de traitement côté serveur, et ce parce qu'il en existe trop pour que l'éditeur HTML puisse s'adapter à chacun d'entre eux. Le langage primaire du Web étant HTML, la priorité doit lui être donnée. Le XML (Extensible Markup Language) devient de plus en plus populaire et offre une création de page évoluée, et sa prise charge par GoLive est un excellent ajout. Mais lorsqu'une application doit gérer autre chose que du code de la famille HTML, la gestion de ce code est une décision arbitraire.

Comme tous les autres programmes, GoLive peut choisir de prendre en charge tel ou tel code, mais il est impossible qu'il puisse les gérer tous. Les doubles crochets (<< ou >>) sont problématiques car ils ne font partie de la syntaxe HTML. (Ils sont aussi problématiques dans Dreamweaver.)

Comme GoLive ne peut pas tout fournir à tout le monde en mode Mise en page, son onglet Source, qui est un programme d'édition de HTML complet, peut être utilisé pour écrire tout code pour le Web.

Cela dit, la gestion de GoLive 5 des balises et du code non-HTML a grandement progressé depuis GoLive 4. La nouvelle fonctionnalité Code 360 signifie que GoLive aura des problèmes avec du code non-HTML qu'à de très rares occasions.