

Nachdem wir nun erfolgreich eine statische Website aufgebaut haben, wollen wir uns dem Aufbau dynamischer Seiten widmen. Zunächst beschäftigen wir uns mit einigen Grundlagen.

18 Dynamische Websites – Einführung

Das Wort Dynamik bezieht sich in der Webentwicklung auf das Generieren der Inhalte von (X)HTML-Dokumenten aus Datenquellen aller Art. Um Websites zu dynamisieren, gibt es zwei grundlegende Vorgehensweisen:

1. Die Daten werden in Echtzeit beim Aufruf der Website durch eine Skriptsprache als CGI aus der Datenbank gelesen, aufbereitet und *dynamisch in (X)HTML-Dokumenten ausgegeben*. Diese Methode hat den Vorteil, dass die Website sehr aktuell ist. Änderungen in der Datenbank werden sofort auf der Website dargestellt. Der Nachteil ist, dass bei hoher Besucherfrequenz die Serverlast enorm zunimmt und Suchmaschinen immer wieder Probleme mit aus Datenbanken generierten Websites haben. Dennoch bietet die Methode viele Vorteile, auf die wir gleich noch ausführlich eingehen werden.
2. Die Website wird, sozusagen auf Knopfdruck, aus einer Datenbank generiert und in Form von *statischen HTML-Dokumenten auf dem Server abgelegt*. Bei Firmenwebsites, die sich nicht ständig ändern, kann man über diese Methode nachdenken. Suchmaschinen können diese Seiten meist problemlos indizieren. Mit dieser Methode ist auch die Ausgabe von XML-Dateien möglich.

18.1 Vorteile dynamischer Sites

Der Folgeaufwand in der Programmierung und Administration ist geringer als bei rein statischen Websites. Und auch die weiteren Vorteile liegen auf der Hand: Wenn es bislang so war, dass Firmen beim Ändern der Website ihre Internetagentur anrufen mussten, können sie Änderungen am Seiteninhalt mit einem Content-Management-System einfach selbst vornehmen. Neue Inhalte werden automatisch online verfügbar gemacht. Nach einem etwas höheren Initialaufwand beim Programmieren der Website sind die Folgekosten geringer.

18.1.1 Statische versus dynamische Sites

In den letzten Jahren haben wir 80 % der neuen Websites dynamisch aufgebaut. Der Trend dahin ist unverkennbar. Es gibt sicherlich viele Gründe dafür, die auch in der breiten Verfügbarkeit und einfachen Einsetzbarkeit von Skriptsprachen wie PHP zu suchen sind.

PHP und MySQL

PHP ist einfach zu programmieren. Mit ein wenig Übung ist eine einfache dynamische Website ebenso schnell erstellt wie eine statische, da viele Skriptelemente (entsprechende Programmierung vorausgesetzt) wiederverwendbar sind.

Noch immer herrscht in vielen Agenturen andächtiges Schweigen, wenn von dynamischen Websites die Rede ist, und massive Kosten werden prognostiziert. Es gibt keinen Grund dafür. PHP und MySQL erfordern in der Regel keine hohen Entwicklerkosten, und es gibt massenhaft Open-Source-Projekte, die Ihnen alles an Mustern und Tools zur Verfügung stellen, was Sie benötigen. Außerdem stehen mit Typo3, Joomla! und ähnlichen CMS-Lösungen mittlerweile auch im Open-Source-Bereich Enterprise-Lösungen zur Verfügung, die den Vergleich mit kommerziellen und sehr teuren Produkten nicht scheuen müssen. Inzwischen stellt sich nicht mehr die Frage, ob es für eine Anforderung ein Open-Source-Produkt gibt, sondern, welches der vielen man verwendet.

18.1.2 Dynamische Sites und Suchmaschinen

Dass Suchmaschinen Websites mit Datenbankverbindungen nicht indizieren, ist ein Gerücht, das nicht stimmt. Die Nachteile sind etwas anders gelagert. Dynamische Websites haben unter Umständen einen sehr kurzen Lebenszyklus. Inhalte von heute sind morgen bereits wieder offline oder unter anderen Pfaden zu finden. Das führt unter Umständen dazu, dass in Suchmaschinen Seiten indiziert sind, die es so gar nicht mehr gibt.



Rewrite-Engine (`mod_rewrite`)

Ein häufiges Argument gegen dynamische Websites sind auch deren seltsame URL-Angebaben. In der Tat sieht die URL einer dynamischen Website anders aus als die einer statischen, und Suchmaschinen finden nicht die Begriffe, die zur korrekten und guten Indizierung notwendig sind. Hier kann mit dem Apache-Modul `mod_rewrite` Abhilfe geschaffen werden. Dieses Modul erlaubt die nahezu beliebige Manipulation der URL (im Positiven wie Negativen). Weitere Informationen dazu finden Sie zum Beispiel unter: <http://www.modrewrite.de/>

18.1.3 Dynamische Sites und Sessions

Bei dynamischen Websites wird häufig mit Sessions gearbeitet. Das sind kleine Dateien, die so lange existieren, wie Sie auf eine Site zugreifen. Häufig werden damit zum Beispiel Warenkörbe bei Shop-Systemen realisiert.

Session-IDs

Beim Zugriff auf eine Seite wird eine Session-ID erzeugt, die in einem Cookie gespeichert oder einfach an die URL angehängt wird. Dabei entstehen in der Regel ellenlange Nummern in der Browser-Adressleiste. Eine Session-ID existiert nur für eine bestimmte Zeit, danach wird sie gelöscht. Greift nun eine Suchmaschine auf eine Site zu, wird ebenfalls eine Session erzeugt. Die Seite wird dann inklusive der Session-ID indiziert. Greift die Suchmaschine dann ein weiteres Mal auf die Seite zu, gibt es diese ID nicht mehr, und die Suchmaschine erhält eine Fehlermeldung. Die Seite fliegt dann entweder aus dem Index, oder es wird gleich ihre ganze URL gelöscht.

Seiten nicht indizieren

Seiten, die mit Sessions arbeiten, sollten Sie daher gegebenenfalls von der Indizierung durch Suchmaschinen ausschließen. Das ist nicht schwierig: Starten Sie Sessions einfach nie schon auf der ersten Seite, geben Sie auf dieser in den Metaangaben `<nofollow>` ein, und lassen Sie dann die erste Seite indizieren. Eine andere, allerdings aufwendigere Methode ist, gleich auf dem Webserver bei einem Request abzufragen, ob eine Suchmaschine zugreift. Suchmaschinen können dann umgeleitet werden, z. B. auf eine spezielle Seite, deren Inhalt mit vielen Metaangaben und Schlagwörtern einer guten Platzierung dient.

Websites ohne Session und mit beständigen Kennungen werden, auch wenn sie dynamisch sind, sehr gut indiziert.

18.2 Typen dynamischer Websites

Es gibt viele Einsatzbereiche für dynamische Websites. Auch wenn sich diese in der Anwendung stark voneinander unterscheiden, funktionieren sie technisch gesehen alle ähnlich. Die wichtigsten sind im Folgenden aufgeführt.

18.2.1 Web-Content-Management-Systeme (WCMS)

WCMS dienen der Pflege und Verwaltung von Website-Inhalten über das Internet. Nicht nur die Website wird dafür online gestellt, sondern auch die Adminis-

trationsoberfläche, mit der man Inhalte anlegt und verändert. Im einfachsten Fall geschieht dies per E-Mail. Aufwendigere Systeme benutzen auch online verfügbare WYSIWYG-Editoren für die grafische Aufbereitung von Inhalten. Hinzu kommt die Benutzerverwaltung, die dafür sorgt, dass Mitarbeiter nur auf die ihnen zugedachten Bereiche zugreifen können.

WCMS gibt es als kostenlose Open-Source-Tools und als ausgefeilte, zu lizenzierende Systeme für Preise von bis zu mehreren 100.000 €, wobei man erstaunt feststellen wird, dass die extrem teuren Lösungen keineswegs die besten sind.

In den nächsten Kapiteln werden wir ein kleines WCMS aufbauen. Es ist allerdings recht einfach gehalten, ansonsten würde es den Umfang dieses Buches sprengen.

18.2.2 Content-Management-Systeme (CMS)

Der Unterschied zu den WCMS ist, dass hier keine eindeutige Ausrichtung auf den Zugriff über das Web besteht, sondern in abgeschlossenen Netzwerken gearbeitet wird. Die meisten CMS haben allerdings eine WCMS-Erweiterung.

18.2.3 Redaktionssysteme

Ein Redaktionssystem ist ein CMS mit besonderer Ausrichtung auf redaktionelle Arbeit. Ein Journalist beispielsweise befindet sich in seiner Redaktion oder irgendwo anders auf der Welt und schreibt seinen Artikel in das Redaktionssystem. Dieser wird dann von einem Redakteur gegengelesen und freigeschaltet. Redaktionssysteme sind für diesen Arbeitsablauf optimiert.

Ein weiterer Einsatzzweck, auf den man dank XML immer öfter trifft, ist Cross-media Publishing. Inhalte für Print-, Onlinemedien, Handys usw. werden aus einer einzigen Datenquelle generiert und auf Knopfdruck in allen Formaten publiziert.

18.2.4 Shopsysteme

Sie alle kennen so etwas – denn jeder von uns hat schon einmal online eingekauft. Das besondere Kennzeichen eines Shopsystems ist die Bestellmöglichkeit mit Warenkorb. Die angezeigten Inhalte, die im Prinzip genau wie bei einem WCMS generiert werden, lassen sich vom User sammeln und als Bestellung abschicken. Die Sammelfunktion, der Warenkorb, ist das Ausschlaggebende bei einem Shopsystem. Es gibt auch Shopsysteme mit kompletter Fakturierung und Rechnungsstellung.

Fertige Shopsysteme

Der Markt für Shopsysteme ist nahezu unüberschaubar. Ein passendes System für einen Kunden zu finden ist eine anspruchsvolle Aufgabe. Mittlerweile bieten einige Provider Shopsysteme zur Miete an (T-Online, 1&1 usw.). Für kleine Hobbyshops sind diese Systeme durchaus ausreichend. Leider kann das Layout dieser Systeme meist nicht hinreichend beeinflusst werden, und eine vollständige Integration in eine Website ist kaum möglich.

18.2.5 Foren

Auch ein Forum ist vom Prinzip her ein WCMS – mit dem Unterschied, dass praktisch jeder Benutzer Rechte hat, Artikel zu posten, und auf die Artikel der anderen Benutzer antworten kann. Ein sehr verbreitetes und durchweg zu empfehlendes Open-Source-Forensystem ist phpBB.

Ein Forum lebt von der Moderation der Beiträge, weniger durch ausgefeilte Technik. Ein wirklich gut geführtes Forum ist daher keine Nebenbeschäftigung, sondern bedarf der exakten Planung sowie redaktioneller Vor- und Nachbereitung. So geführt kann es ein Besuchermagnet und ein Know-how-Beweis sein. Foren können idealerweise auch den Kundendienst unterstützen und eine Nutzergemeinschaft auf hohem Niveau bilden. Aus Marketingsicht kann ein gutes Forum unmittelbaren Einblick in Kundenverhalten und -wünsche geben.

18.2.6 Portale

Portale sind eine Mischung aus WCMS und umfangreichen Linklisten. Ein mit der Thematik CMS verbundenes Portal ist z. B. <http://www.contentmanager.de>. Der Gedanke hinter einem Portal ist, Usern zu einem speziellen Thema alle verfügbaren Informationen, Links, Artikel und Foren auf einer Website gesammelt zu präsentieren und den Betrieb in der Regel über Werbeeinnahmen zu finanzieren.

Während der Zeit des Internet-Hypes wurden Portale oft als Gelddruckmaschinen angesehen. Heute hat sich dies relativiert. Der tatsächliche Nutzen ist außer in branchenspezifischen Portalen eher gering. Der administrative Aufwand jedoch ist enorm. Sehr gute Portale sind beispielsweise die Website der Stadt Hamburg (<http://www.hamburg.de>) und das Portal der Bundesverwaltung (<http://www.bund.de>). Im Zusammenspiel mit Portalen bieten sich auch personalisierte Inhalte an.

18.2.7 Personalisierte Websites

Ein ganz wichtiger Aspekt dynamischer Websites ist die Möglichkeit, einem User speziell auf ihn zugeschnittene Informationen zu präsentieren und damit die Attraktivität des Angebots massiv zu steigern.

Wenn sich ein Benutzer durch ein Login zu erkennen gibt und er an anderer Stelle sein Interesse an bestimmten Inhalten bekundet hat, können diese individuell aus der Datenbank generiert werden. Ein Einsatzbereich hierfür sind beispielsweise Onlinezeitschriften. Wenn jemand möchte, dass er als Startseite die Sportnachrichten sieht, dann bekommt er diese vor allen anderen Inhalten angezeigt. Bei anderen Portalen und fast allen Free-Mailern (GMX, Web.de, Firemail usw.) müssen Sie zum Registrieren ein Interessenprofil erstellen. Anhand dieses Profils werden dann die Werbeinhalte zusammengestellt.

Cookies

Fragwürdig sind anhand von User-Tracking personalisierte Websites. Damit wird Ihr Weg durch eine Website verfolgt und protokolliert. Wenn Sie dann zwanzigmal die neuesten Turnschuhe angeklickt haben, können Sie sicher sein, beim nächsten Login alle Turnschuhe gleich auf der Startseite präsentiert zu bekommen.

Einige Anbieter installieren Cookies bei Ihnen, die sogar Ihre Wege im ganzen Web verfolgen. Gelangen Sie dann wieder auf die Website eines an dieses System angeschlossenen Anbieters, werden diese Cookies ausgewertet und Ihnen entsprechende Inhalte angezeigt. Um dagegen vorzugehen, installieren Sie am besten SpyBot oder Ad-Aware. Sie werden Augen machen, wie verbreitet diese Methode ist. 150 Cookies auf einem System nur für das User-Tracking sind keine Seltenheit.

18.2.8 Weblogs

Es gibt verschiedene Weblog-Typen. In den einfachsten Fällen ist das ein Online-tagebuch. Zu fast allen wichtigen und unwichtigen Themen gibt es mittlerweile Weblogs. Innerhalb kürzester Zeit ist ein regelrechter Hype um das »Bloggen« entstanden. Ein Kennzeichen vieler Weblogs ist die Querverlinkung zu anderen Weblogs. Diese so entstandene Community wird auch als *Blogosphäre* bezeichnet.

Blogs sind einfachste WCMS. Sie ermöglichen das schnelle Aktualisieren von Website-Inhalten ohne HTML-Kenntnisse, erreichen aber nicht die Leistungsfähigkeit eines vollwertigen WCMS.

Wenn Sie den kompletten Teil III dieses Buches über dynamische Sites durchgearbeitet haben, werden Sie sicherlich in der Lage sein, einen einfachen Weblog zu erstellen. Wir können Ihnen dies als Übung sehr empfehlen. Eine Schritt-für-Schritt-Anleitung haben wir dafür allerdings nicht erstellt, da das zugrunde liegende Prinzip eines Weblogs der von uns erstellten Struktur sehr ähnlich ist.

Dies ist nur eine kleine Auswahl der wichtigsten Applikationen, die man mit dynamischen Websites entwickeln kann. Es gibt noch viele andere Anwendungen, so etwa Webmailer und andere Tools, die eine Interaktion des Benutzers mit einer Datenbank benötigen.

18.2.9 Social Networks

In aller Munde sind sogenannte *Social Networks*. Allen gemeinsam ist, dass der Benutzer bzw. die Gemeinschaft der Benutzer hier die Inhalte erstellt, austauscht und verlinkt. Mittlerweile ist es durchaus wichtig, mit seinen eigenen Seiten in diesen Netzwerken gefunden zu werden oder Verlinkungen zu diesen Netzwerken bilden zu können. In manchen Branchen und spezifischen Zielgruppen ist es sogar erfolgsentscheidend, möglichst überall »im Gespräch« zu sein.

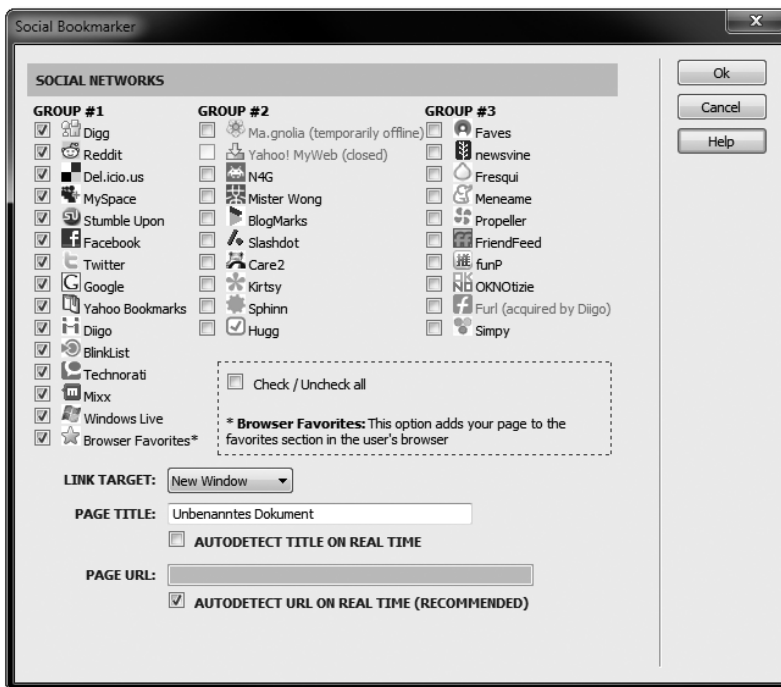


Abbildung 18.1 Erweiterung: Social Bookmarker

Hier hilft eine kleine, leider kostenpflichtige Erweiterung: der Social Bookmarker (<http://www.topdreamweaverextensions.com>). Sie ermöglicht die automatische Verlinkung zu den Social Networks auf einer Website, da sie für jedes ausgewählte Netzwerk ein Icon in das XHTML-Dokument einfügt (siehe Abbildung 18.1).

18.3 Client- und serverseitiges Skripting

Um die folgenden Kapitel über die Erstellung einer dynamischen Site in Dreamweaver zu verstehen, ist zunächst ein Exkurs in die technischen Abläufe bei dynamischen Websites notwendig.

HTTP-Requests

Beim Aufruf einer Website sendet Ihr Rechner, der Client, eine Anfrage (HTTP-Request) an den Webserver. Dieser sammelt, vereinfacht ausgedrückt, die von Ihnen angeforderten Daten und sendet diese an Ihren Client zurück. Zu diesen versandten Daten können auch Skripte mit JavaScript usw. gehören.

Das HTML wird sofort in Ihrem Browser dargestellt, JavaScripts werden temporär abgespeichert und erst dann ausgeführt, wenn sie ausgelöst werden. Ihre Ausführung ist dabei von der auf Ihrem Rechner installierten Software abhängig. Die unterschiedlichen Effekte, die dabei auftreten können, sind bekannt. Eine unserer liebsten Aufgaben ist, dafür zu sorgen, dass Websites auf möglichst vielen Plattformen laufen und identisch aussehen.

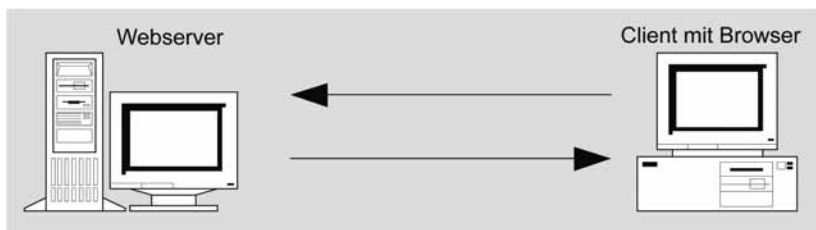


Abbildung 18.2 Clientseitiges Skripting

Clientseitiges Skripting

Diese Art der Verarbeitung nennt man *clientseitiges Skripting* (siehe Abbildung 18.2). Ein Shoppingsystem, das einen Warenkorb mit JavaScript realisiert, funktioniert nur auf Browsern mit aktiviertem JavaScript. Benutzer ohne passende Browser können somit nicht einkaufen. Würde man mit dieser Methode auch Datenbanken abfragen, sähen auch hier nur die User mit aktiviertem JavaScript die Inhalte der Website. Zudem ist eine Datenbankabfrage mit JavaScript nur

schlecht zu realisieren. Würde man etwa eine Personalisierung einbinden, hätten Sie sehr schnell große und völlig unüberschaubare Quelltexte.

Serverseitiges Skripting

Um vom Zielsystem unabhängig zu werden und überhaupt Datenbanken vernünftig abfragen zu können, verwendet man *serverseitiges Skripting* (siehe Abbildung 18.3).

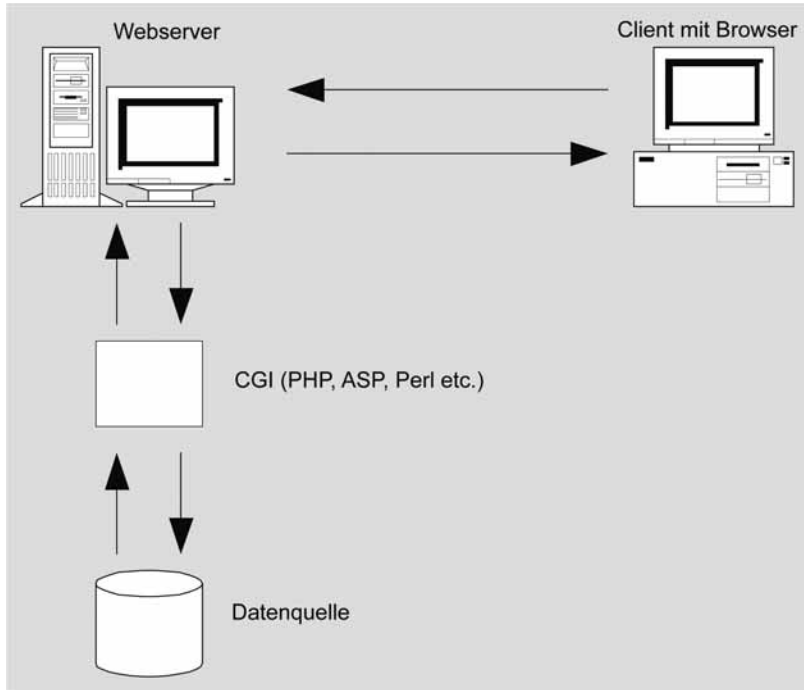


Abbildung 18.3 Serverseitiges Skripting

Hierbei läuft auf dem Webserver eine Skriptsprache. Diese setzt einen installierten Interpreter auf dem Webserver voraus. Serverseitige Skripte funktionieren nach dem folgenden Prinzip:

1. Bei der Anforderung eines entsprechenden Dokuments wird ein CGI-Skript (*Common Gate Interface*), das z. B. in PHP programmiert ist, gestartet.
2. Dieses analysiert nun die Anfrage und stellt eine Verbindung zu der Datenbank her, die die angefragten Daten enthält.
3. Es holt die gewünschten Daten hervor und generiert damit ein fertiges HTML-Dokument.
4. Dieses wird dann vom Webserver an den Client zurückgeschickt.

Von dem ganzen Vorgang bekommt der User nichts mit. Er sieht in seinem Browser das Ergebnis in reinem HTML. Das eigentliche CGI-Skript bleibt vor ihm verborgen. Aus diesem Grund finden Sie bei PHP-Websites im Quelltext keinen einzigen PHP-Befehl.

Ein Beispiel aus dem wirklichen Leben

Um dies noch einmal für alle zu verdeutlichen, die noch nichts mit diesen Technologien zu tun hatten, folgt hier ein Vergleich mit einem Vorgang aus dem wirklichen Leben.

Stellen Sie sich vor, Sie bestellen online oder per Telefon eine Pizza. Stellen Sie sich weiter vor, Sie sind jetzt der Browser oder Clientrechner. Der Webserver ist der Pizza-Bringdienst. Sie haben jetzt eine Pizza (Website) mit bestimmten Inhalten (Salami, Spinat, Brokkoli und viel Käse) angefordert. Auf Serverseite (Pizzaservice) sitzt jemand, der Ihre Bestellung (HTTP-Request) annimmt. Dieser leitet sie weiter an unser CGI, den Koch. Der Koch (CGI) entscheidet anhand der Bestellung (HTTP-Request), welche Zutaten er benötigt, und holt diese aus dem Regal (der Datenbank). Aus diesen Zutaten und Ihren Anforderungen backt er nun Ihre Pizza (das HTML-Dokument). Anschließend schickt er diese mit einem Pizzataxi (dem HTTP-Protokoll) zu Ihnen, dem Client. Am Ende sitzen Sie da und genießen Ihre Pizza.

Was Sie jetzt damit machen – essen, verschenken oder zerteilen – ist dann sozusagen clientseitig und interessiert den Bringdienst nicht. Sie haben weder den Telefondienst noch den Koch oder die Küche gesehen und trotzdem Ihr Abendessen bekommen.

Genau so funktionieren dynamische Websites. Das CGI-Skript, das diesen ganzen Vorgang steuert, läuft auf dem Webserver und ist somit völlig unabhängig von Ihrem Rechner – das generierte HTML-Dokument jedoch nicht. Es unterliegt wie bei statischen Seiten den individuellen Gegebenheiten des jeweiligen Browsers. In skriptgenerierten HTML-Dokumenten können nun wieder JavaScripts enthalten sein, die Ihren Browsertyp erkennen und die passende HTML-Datei anzeigen.

[»] Skriptings mischen

In der Praxis ist eine Vermischung von client- und serverseitigen Skriptings durchaus üblich. Relevante Abfragen und Bedingungen werden serverseitig ausgeführt. Userabhängige Aktionen wie z. B. Rollover-Effekte usw. werden clientseitig programmiert und ausgeführt. Wenn diese dann nicht auf dem Client funktionieren, wird zumindest das Basisdokument mit den Inhalten dargestellt.

18.4 Konzeption dynamischer Sites

Die Konzeption ist bei dynamischen Sites wesentlich bedeutungsvoller als bei statischen Sites. Nicht nur die Gestaltung, sondern auch die technischen Abläufe müssen bis ins Detail entworfen werden. Wir fassen an dieser Stelle die wichtigsten Punkte zusammen.

18.4.1 Besonderheiten für das Layout

Bei statischen Websites haben Sie alles fest in der Hand. Der Kunde gibt Ihnen die Texte und Bilder. Sie bearbeiten diese und binden sie ein. Bei dynamischen Websites wissen Sie nicht, welche genauen Inhalte sich einmal auf der Seite befinden werden. Sie erhalten zwar Beispielinhalte, ob diese später jedoch genau so eingesetzt werden, ist fraglich.

Ihre Site sollte daher kein Layout haben, das mit Textlängen oder Bildgrößen zu restriktiv umgeht. Das Layout muss flexibel genug sein, um lange wie auch kurze Inhalte darstellen zu können. Auf den meisten dynamischen Websites sind daher die Inhalte nach unten hin scrollbar. In den meisten Fällen ist es sinnvoll, so zu verfahren. Es sind aber auch andere Layouts möglich. Diese erfordern allerdings einen höheren technischen Aufwand. Es ist immer abzuwägen, ob ihr Einsatz aus Benutzersicht sinnvoll ist oder nicht.

Textnavigation

Ein weiterer wichtiger Punkt ist die Navigation. Üblicherweise werden Sie eine Navigation mit Grafiken erstellen. Das geht prinzipiell auch bei dynamischen Websites. Dennoch ist es ein Kundenanliegen, schnell einen neuen Navigationspunkt einfügen, entfernen oder an eine andere Stelle verschieben zu können. Grafische Menüs bieten sich daher nur sehr bedingt an. PHP unterstützt zwar das dynamische Erstellen von JPEGs, die Qualität ist aber relativ schlecht. GIFs dürfen aus lizenzrechtlichen Gründen nicht dynamisch generiert werden. Das Generieren von Grafiken lastet einen Server zudem stark aus. Es bleibt als Fazit, dass Navigationen immer am besten mit Textlinks erstellt werden. Dank CSS und Selektoren ist dies auch kein Problem und kann ebenfalls sehr gut aussehen.

Sie sollten ebenfalls bedenken, dass eine Navigation Platz beansprucht und Sie auch für Navigationserweiterungen Raum vorhalten müssen. Wenn Sie Ihrem Kunden die Möglichkeit einräumen, eigenständig Navigationspunkte verändern und hinzufügen zu können, muss das Menü genauso flexibel gehalten werden wie der Rest des Layouts.

Bei der im Buch erstellten Website finden diese eben genannten Aspekte keine Berücksichtigung, da das Anwenden aller im Buch gezeigten Techniken im Vordergrund steht.

18.4.2 Trennung von Layout und Inhalt

Besonders bei dynamisch eingespeisten Inhalten ist es sinnvoll, mit CSS zu arbeiten. Damit wird der Bereich in einer Website mit CSS-Stilen vorformatiert, der später die dynamischen Inhalte enthält. Geschickt programmiert, kann der Kunde oder Redakteur dann einfügen, was er möchte. Der Inhalt wird sich immer Ihren gestalterischen Rahmenbedingungen anpassen. Und wenn Sie dann doch mal etwas daran verändern müssen, können Sie das ohne großen Aufwand in der externen CSS-Datei durchführen.

18.4.3 Konzeption der Administrationsoberfläche

Viele dynamische Websites benötigen eine Administrationsebene, wenn die Inhalte vom Kunden gepflegt werden sollen. Legen Sie das Layout und die Benutzerführung der Administrationsoberfläche wirklich idiotensicher an. Jeder mögliche Bedienfehler wird auch gemacht, da können Sie ganz sicher sein! Es geht vor allem auf Ihre Nerven und Kosten, wenn ein Kunde ständig anruft, weil er mit der Administration nicht zurechtkommt.

Usability (Anwenderfreundlichkeit)

Geben Sie einem Kunden nur die Möglichkeiten, die er wirklich benötigt. Was für Sie als Profi ein tolles Feature ist, kann für den Laien schon eine Schaltfläche zu viel sein. Es gibt eine Menge kaum bedienbarer WCMS. Gerade im hochpreisigen Niveau finden sich einige Vertreter, die wenig benutzerfreundlich sind und einen eigenen Administrator benötigen. Folgen Sie bei der Konzeption und beim Layout der bekannten KISS-Regel: »Keep it simple, stupid.« Testen Sie die Benutzerführung der Administration am besten mit völlig unerfahrenen Usern, und suchen und beheben Sie so die Fehlerquellen.

Zugriffsrechte

Wenn Sie ein umfangreiches CMS planen, achten Sie daher darauf, welche Möglichkeiten Sie Ihren Usern zur Pflege der Daten geben und wie qualifiziert die zu erwartenden Administratoren im Umgang mit Websites sind. Unterteilen Sie am besten die User in Gruppen mit verschiedenen Zugriffsrechten. Jemand, der Texte ändern muss, sollte nicht unbedingt Systemparameter verändern können.

All diese Anforderungen machen die Angelegenheit für Sie nicht gerade einfacher. Im Gegenteil: Je benutzersicherer eine Administrationsebene sein muss, desto mehr müssen Sie sich Gedanken über die Benutzerführung machen. Bei größeren CMS mit tausend und mehr zu pflegenden Dokumenten ist der Aufwand für die Administrationsebene und für eine einwandfreie Benutzerverwaltung in der Regel höher als für die eigentliche Website.

18.4.4 Planung, Planung, Planung ...

Bevor Sie die ersten Abfragen schreiben oder die erste Datenbank anlegen, müssen Sie viel Zeit in eine gute Planung investieren. Die eigentliche Programmierung der Datenbank und der PHP-Skripte ist dann mit einiger Übung schnell erledigt. Unsere ersten dynamischen Websites haben wir bestimmt fünfmal von neuem aufgebaut, bis sie zufriedenstellend funktionierten. Sehen wir uns heute den Quelltext dieser Seiten an, stehen uns die Haare zu Berge.

Je mehr Zeit Sie für eine exakte und gute Planung investieren, desto leichter fällt Ihnen die anschließende Umsetzung. Mit zunehmender Erfahrung und Sicherheit wird diese mit Dreamweaver zur reinen Fleißarbeit.

Wenn man bei einer statischen Website noch einen Anteil von geschätzten 30 bis 40% der gesamten Arbeitszeit für die reine Programmierung veranschlagen sollte, verschiebt sich der Anteil der Programmierung bei einer dynamischen Website auf etwa 20%. Dafür steigt der Anteil der Vorüberlegung und strukturellen Konzeption mit Microsoft Visio oder anderen Tools von etwa 25 auf 60% an.

Zu oft werden hier auch die Fehler in mangelhafter Software und allen anderen möglichen Dingen gesucht. Bei diesem Thema gilt für uns folgender Leitsatz:

Ein guter Handwerker kann auch mit miserablen Werkzeugen gute Arbeit leisten. Ein schlechter Handwerker nicht. Der gute Handwerker wird sich aber ein schlechtes Werkzeug nicht antun!

Dreamweaver ist ein gutes Werkzeug. Kommen wir nun dazu, wie wir damit dynamische Websites entwickeln können.

Um schnell und effizient mit Dreamweaver und dynamischen Websites arbeiten zu können, ist es sinnvoll, einen lokalen Webserver einzurichten. In diesem Kapitel zeigen wir Ihnen an zwei Beispielen, wie das geht.

21 Lokalen Webserver installieren

Da dynamische Websites erst von einem Webserver erstellt werden, benötigen Sie zu Entwicklungszwecken einen lokalen Webserver, auf dem Sie Ihre Skripte testen können.

21.1 Webserver-Grundlagen

Jede Website benötigt einen Webserver, auf dem sie läuft. Hier erläutern wir erst einmal kurz die wichtigsten Begriffe zu diesem Vorgang.

Der Webserver ist dafür verantwortlich, Daten mit dem HTTP-Protokoll zu versenden. Mehr macht er eigentlich nicht. Er wartet auf eine Anforderung, sammelt Daten ein und schickt sie ab.

Für den Einsatz mit Dreamweaver und PHP kommen folgende Webserver in Frage:

► **Apache**

Dies ist der definitiv meistgenutzte Webserver überhaupt. Eigentlich aus der Welt von UNIX kommend, laufen die neueren Versionen auch auf dem PC sehr stabil. Auf dem neuen Betriebssystem Mac OS X ist Apache bereits vorinstalliert. Die wesentlichen Vorteile sind, dass er sehr schnell, klein und leistungsfähig ist und unter Linux äußerst betriebs- und datensicher läuft.

► **Internet Information Server (IIS)**

Der IIS ist der Webserver aus dem Hause Microsoft. Bei den Windows-Versionen XP, Vista und 7 wird er in der Professional Edition mitgeliefert, muss allerdings nachinstalliert werden. Unter Windows 7 kommt der IIS 7.5 zum Einsatz.

Der IIS ist ein kompletter, hochprofessioneller Webserver und für den Einsatz im Web auf Microsoft-Systemen konzipiert. Einer der wichtigsten Vorteile ist,

dass er die Entwicklung mit ASP und ASP.NET ermöglicht. Er stellt auch viele Dienste wie Mail und FTP zur Verfügung. Um den IIS gut und sicher zu konfigurieren, muss man sich allerdings gut mit den Windows-Systemen auskennen.



Buchwebsite

Aufgrund seiner hohen Verbreitung verwenden wir den kostenlos verfügbaren Webserver Apache. Natürlich können Sie auch jeden anderen Webserver betreiben. Für das weitere Vorgehen im Buch wird lediglich vorausgesetzt, dass PHP und MySQL auf ihm installiert und lauffähig sind.



IIS oder Apache

Die in den folgenden Abschnitten behandelten Webserver stellen Alternativen dar. Bei Ihrer Arbeit müssen Sie sich für den IIS oder den Apache entscheiden. Lesen Sie am besten vor der Installation dieses Kapitel, und entscheiden Sie dann, welches System Ihnen am verständlichsten erscheint.

Bei einigen der einfachen Windows-Versionen (HOME etc.) ist der IIS nicht im Lieferumfang enthalten, kann allerdings von der Microsoft-Website heruntergeladen werden.

LAMP und WAMP

Häufig fallen im Zusammenhang mit der Webentwicklung die Begriffe *LAMP* und *WAMP*. Sie bezeichnen Kombinationen verschiedener Systeme auf einem Rechner, bestehend aus Betriebssystem, Webserver, Datenbank und Skriptsprache. *LAMP* ist die Abkürzung für Linux, Apache, MySQL und PHP. Bei *WAMP* ist nur das Betriebssystem ein anderes: Statt Linux wird hier Windows eingesetzt. Eine Implementation von *LAMP* und *WAMP* ist unter dem Namen *XAMPP* bekannt, die wir im Folgenden auch für unseren Testserver einsetzen werden.

WAMP-Entwicklungsserver

Wir werden in diesem Buch ein *WAMP*-System installieren und dieses als Testserver betreiben. Bei Ihrem Provider steht wahrscheinlich ein *LAMP*-System. Das Internet ist eine Linux- und UNIX-Welt. Diese Systeme sind wesentlich zuverlässiger als Windows. Für unsere Arbeit spielt es allerdings kaum eine Rolle, ob wir auf *WAMP* oder *LAMP* entwickeln. Die Technologie dahinter ist für das Funktionieren einer Site ohne Bedeutung, solange der Webserver stabil läuft.

Es ist wichtig zu wissen, dass Apache ursprünglich nicht für Windows-Systeme geschrieben wurde, sondern für Linux. Er funktioniert zwar auch auf Windows-Servern, für reale Bedingungen im Web ist diese Kombination aber nicht ideal. Als Entwicklungsserver ist *WAMP* aber vollkommen ausreichend.

PHP, MySQL und Apache haben zunächst nichts miteinander zu tun. Nur durch unsere Programmierung nutzen sie sich gegenseitig. PHP und MySQL können völlig unabhängig voneinander laufen. Üblicherweise startet man allerdings die Dienste gemeinsam.

Webserver auf dem MAC

[«]

Apache, PHP und MySQL laufen auch auf einem Mac, auf den neueren Systemen von Apple sogar hervorragend. Lesen Sie dazu die Installationsanleitungen auf <http://www.apple.de> oder auf den Websites von Apache, PHP und MySQL. Es gibt auch einige Foren, die sich mit dieser Thematik befassen. Bei Mac OS X ist standardmäßig bereits Apache usw. auf dem System installiert.

21.2 Vorbereitung: Firewall einstellen oder ausschalten

Nun genug der langen Rede. Im Folgenden installieren wir einen lokalen Testserver und sehen uns die Sache am lebenden Objekt an.

Auf der dem Buch beigelegten DVD finden Sie alle notwendigen Programme zur Installation im Verzeichnis *Webserver*. Wir beschreiben hier die Installation von Apache und eines Internet Information Systems unter Windows XP sowie des IIS 7 unter Vista Business.

[o]

MAMP

[«]

Für den Mac steht auch ein Paket zur Verfügung: Es nennt sich MAMP, Sie finden es ebenfalls auf der DVD. Weitere Informationen gibt es unter <http://www.mamp.info>. XAMPP läuft unter Mac nur mit einer Intel-CPU.

Desktop-Firewall konfigurieren

Wenn Sie auf dem Rechner für den Testserver eine Desktop-Firewall installiert haben, kann es zu Problemen mit einem Webserver kommen. Falls nach den folgenden Schritten einer der Webserver wider Erwarten nicht funktioniert, überprüfen Sie die Einstellungen Ihrer Firewall. Vereinfacht ausgedrückt sprechen Sie Ihren Rechner beim Webserverbetrieb in einer Art Rückkopplung an. Für eine Firewall sieht das so aus, als würde ein fremder Rechner einen Zugriffsversuch starten. Falls möglich, betreiben Sie Ihren Rechner an einem Router, und aktivieren Sie auf diesem die Firewall mit den gewünschten Einstellungen. Oder noch besser: Schalten Sie einfach einen Rechner mit Linux als Firewall und Router dazwischen. Die geschilderten Probleme treten dann nicht auf.



Probleme von Apache und IIS mit der Firewall Outpost

Schwerwiegende Probleme traten bei uns mehrfach mit der freien Version der Firewall Outpost von Agnitum auf. Die Zugriffsrechte lassen sich zwar einfach konfigurieren, jedoch bringen sowohl der Apache als auch der IIS in Kombination mit dieser ansonsten sehr guten Desktop-Firewall das System zum Totalabsturz. Das Problem tritt bei uns grundsätzlich in Kombination mit Hyper-Trading-Prozessoren auf. Nach einigen Zugriffen auf die lokale Site startet der Rechner einfach neu.

Nachdem wir anfänglich der Meinung waren, dass ein Virus daran schuld ist, sind wir mittlerweile sicher, dass es an Outpost liegt. Falls Ihnen dieses Problem bekannt ist und Sie eine Lösung kennen, lassen Sie es uns wissen.

Interne Firewall

Windows XP, Vista und 7 benutzen eine interne Firewall. Auch hier müssen Sie je nach Systemkonfiguration Einstellungen vornehmen oder die Firewall deaktivieren.

Weitere Vorbereitungen für die Installation eines Webserver sind nicht erforderlich. Für den IIS benötigen Sie nur den Original-Datenträger von Windows XP, Vista oder 7. Ohne die Original-CD oder -DVD ist eine Installation nicht möglich.

21.3 WAMP installieren

Einen vollwertigen und sicheren Webserver für den Produktivbetrieb zu installieren ist keineswegs trivial. Für unser Testsystem wählen wir daher den einfachen Weg und verwenden die bereits erwähnte Instant-Installation eines WAMP-Systems namens XAMPP.



Nicht als realer Webserver geeignet

Beachten Sie bitte, dass keine der von uns vorgeschlagenen Konfigurationen zum Betrieb eines »echten« Webserver im Internet geeignet ist. Die Systeme sind ausschließlich zum lokalen Gebrauch bei der Entwicklungsarbeit gedacht.

Komplettes Installationspaket

XAMPP ist ein Komplettsystem mit allen erdenklichen Modulen und Bibliotheken. Auf der Website <http://www.apachefriends.org> finden Sie neben einer Downloadmöglichkeit eine vollständige Dokumentation sowie einige Erweiterungen für dieses Tool.

XAMPP ist mittlerweile in der Version 1.7.3 verfügbar (Stand Juni 2010). Diese Version nutzt die neueste MySQL-Version 5.1. MySQL ab der Version 5 liegt in einer Community-Version vor, die auch für private und Open-Source-Projekte kostenlos nutzbar ist.

XAMPP installieren

Im Verzeichnis *Webserver/XAMPP* auf der DVD zum Buch finden Sie die Datei *xampp-win32-1.7.3.exe*. Kopieren Sie diese Datei auf Ihren Desktop, und starten Sie die Installationsroutine durch einen Doppelklick. [o]

XAMPP unter Windows 7

XAMPP funktioniert in der von uns verwendeten Version auch unter Windows 7 hervorragend, wenn man einige Kleinigkeiten beachtet. Während der Installation kommen einige Sicherheitsmeldungen, die Sie ignorieren können. XAMPP wird standardmäßig im Verzeichnis *C:/XAMPP/* installiert, Sie können aber auch ein eigenes Verzeichnis angeben und sollten das unter Windows 7 auch machen.

In unseren Beispielen installieren wir XAMPP auf dem Laufwerk C:

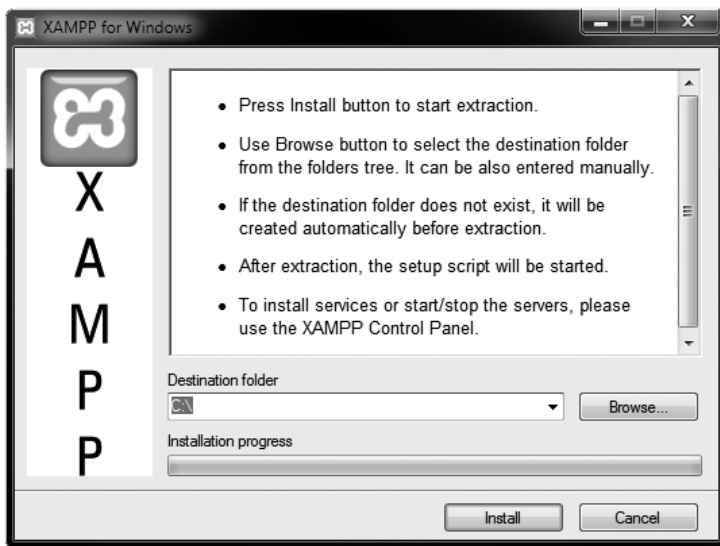


Abbildung 21.1 Installationsverzeichnis unter Windows 7

Zum Abschluss der Installation müssen Sie in einigen Dialogen die jeweiligen Optionen mit *y* oder *n* bestätigen. Wir empfehlen Ihnen, hier immer mit den Vorgaben zu arbeiten.

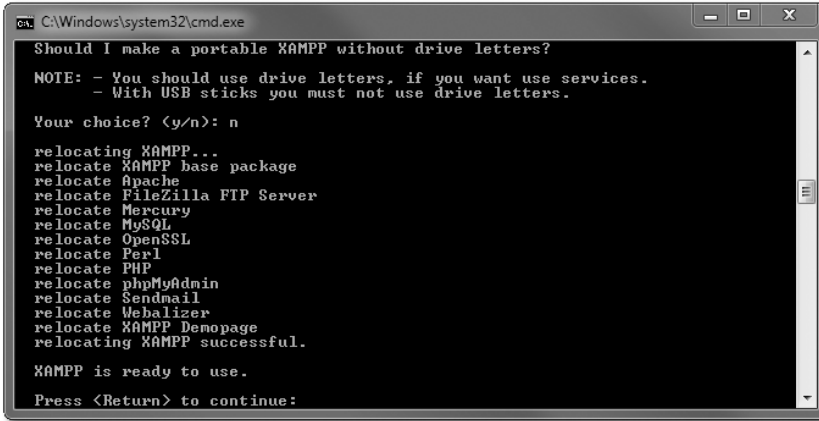


Abbildung 21.2 Abschluss der Installation

Nach der Installation finden Sie unter ALLE PROGRAMME • XAMPP FOR WINDOWS die in Abbildung 21.3 gezeigten Einträge.

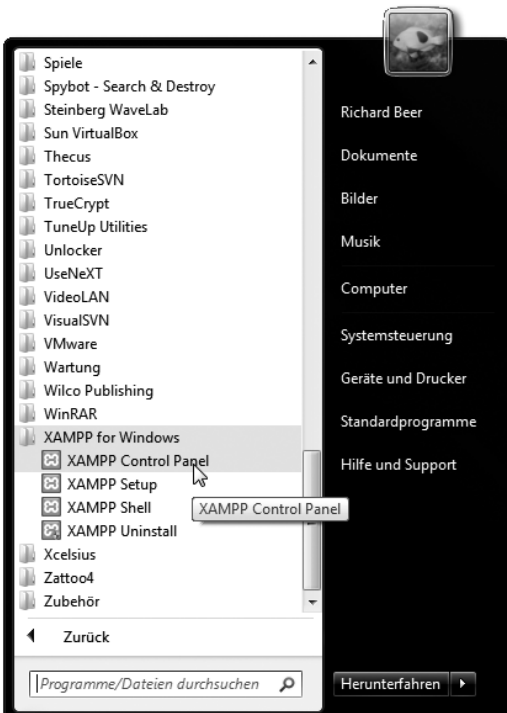


Abbildung 21.3 Startmenü für XAMPP

21.3.1 XAMPP testen und konfigurieren

Klicken Sie auf PROGRAMME • XAMPP FOR WINDOWS • XAMPP CONTROL PANEL.

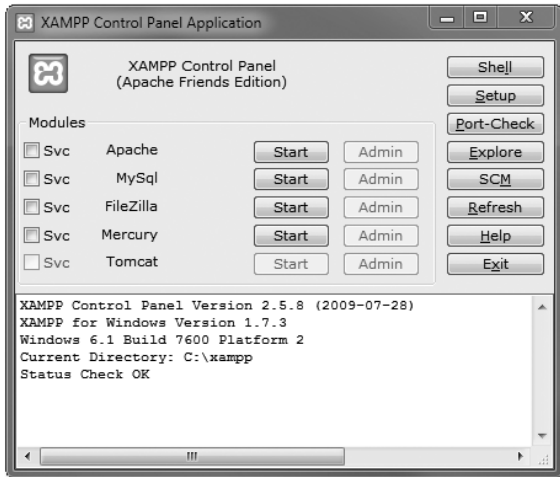


Abbildung 21.4 XAMPP Control Panel

Im folgenden Fenster (siehe Abbildung 21.4) werden die einzelnen Dienste gestartet und auch wieder beendet. Klicken Sie zum Testen von XAMPP bei *Apache* und *MySQL* auf START.

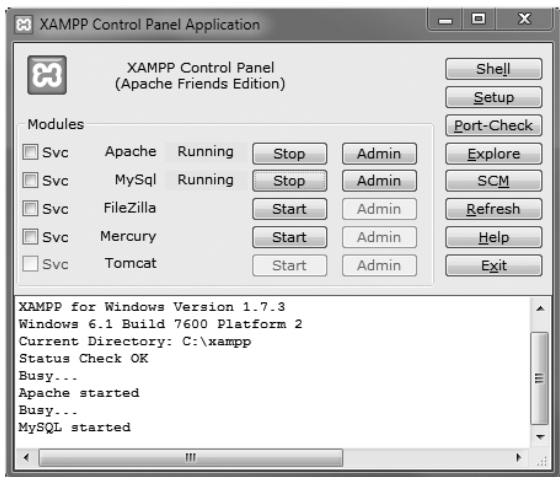


Abbildung 21.5 XAMPP Control Panel mit gestarteten Apache und MySQL

Öffnen Sie einen Browser, um die Funktionen zu überprüfen. Der lokale Webserver hat den Namen *localhost* und die IP 127.0.0.1.

Geben Sie `http://localhost` in die Adressleiste des Browsers ein.

Wenn jetzt – nach einer eventuellen Sprachauswahl – der Bildschirm aus Abbildung 21.6 erscheint, war Ihre Installation erfolgreich.

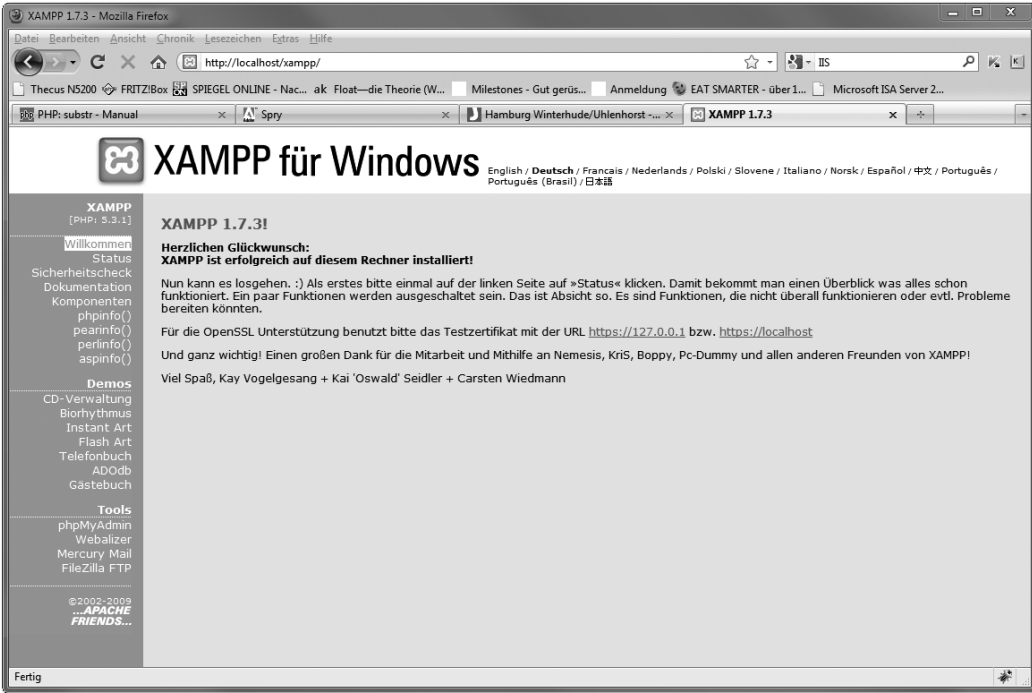


Abbildung 21.6 Startbildschirm von XAMPP unter `http://localhost/xampp`

Ablageort für dynamische Websites

Betrachten wir nun die Verzeichnisse auf Ihrer Festplatte. Auf Laufwerk C: befindet sich ein neuer Ordner mit dem Namen `xampp/htdocs/`. Dieser Ordner ist die sogenannte *Documentroot*.

In diesen Ordner müssen Sie alle dynamischen Websites ablegen, die Sie auf Ihrem System entwickeln möchten. Nur dort werden die Dokumente ausgeführt. Zunächst liegen hier jedoch die Dokumente für die XAMPP-Website und einige Testdokumente.

Löschen Sie den kompletten Ordnerinhalt von `htdocs`, und geben Sie im Browserfenster `http://localhost` ein. Sie sehen jetzt die Rootverzeichnisstruktur im Browserfenster. Da noch keine Verzeichnisse angelegt wurden, sieht das Browserfenster zunächst aus, wie in Abbildung 21.8 dargestellt. Werden Ihnen andere, fehlerhafte Inhalte angezeigt, müssen Sie eventuell XAMPP neu starten.

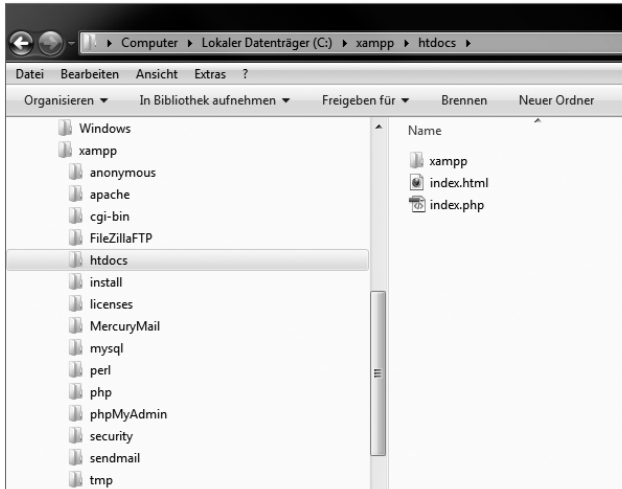


Abbildung 21.7 Inhalt des Ordners »htdocs«

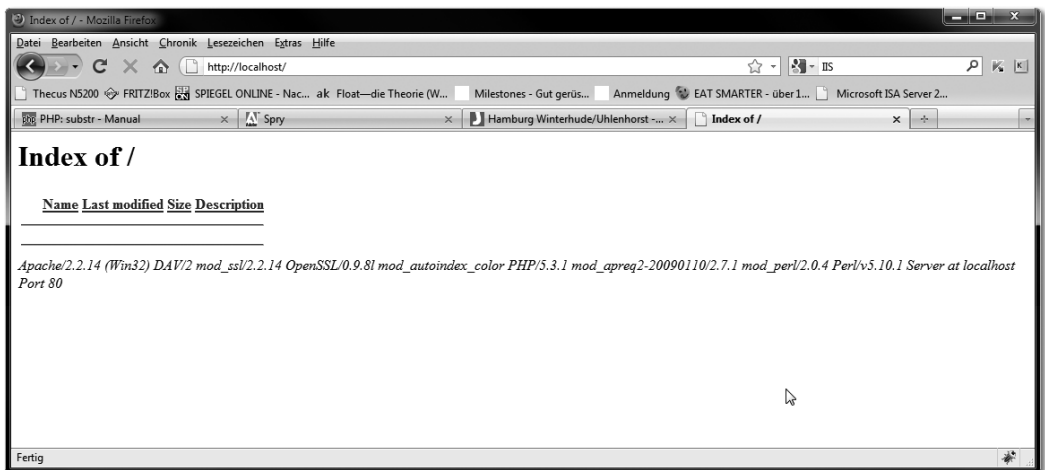


Abbildung 21.8 Anzeige im Browser bei Serverroot ohne Dokumente

21.3.2 Eigene Documentroot angeben

Sie müssen nicht zwingend mit der von XAMPP voreingestellten Serverroot arbeiten. Die Datenhaltung für wichtige Websites auf Laufwerk C: unter Windows ist sehr kritisch. Bei einer Neuinstallation von Windows wären alle vorhandenen Daten verloren. Aus diesem Grund empfiehlt es sich, die Serverroot auf eine eigens dafür angelegte Partition zu legen. XAMPP löscht bei der Deinstallation die Documentroot, wenn Sie diese im Originalverzeichnis belassen. Auch das ist ein Grund, die Documentroot auf ein anderes Laufwerk zu legen.

Im weiteren Verlauf des Buches arbeiten wir, wie in der vorherigen Installation beschrieben. Im folgenden Beispiel zeigen wir Ihnen, wie Sie das Laufwerk *E*: als Documentroot angeben. Das kann je nach Konfiguration Ihres Rechners auch ein anderes Laufwerk sein.

!! Sicherungskopie anlegen

Legen Sie, bevor Sie eine eigene Documentroot definieren, unbedingt eine Sicherungskopie der zu verändernden Dateien an. So können Sie bei einem Misserfolg die Originaleinstellungen wiederherstellen.

■ Schritt-für-Schritt: Eigenes Rootverzeichnis angeben

1 Konfigurationsdatei öffnen

Öffnen Sie mit einem Texteditor die Datei *httpd.conf* aus dem Ordner *C://xampp/apache/conf/*, und suchen Sie die folgenden Zeilen:

```
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory,
# but symbolic links and aliases may be used to point to other
# locations.
#
DocumentRoot "C:/xampp/htdocs"
```

2 DocumentRoot ändern

Ändern Sie die DocumentRoot auf Ihr gewünschtes Verzeichnis. Bei uns ist die DocumentRoot auf

```
DocumentRoot "E:/"
```

eingestellt.

Suchen Sie einige Zeilen weiter den Eintrag

```
# This should be changed to whatever you set DocumentRoot to.
#
<Directory "C:/Programme/xampp/htdocs">
```

Hier müssen Sie unter *Directory* den gleichen Eintrag wiederholen. Wir stellen auch diesen auf

```
Directory "E:/"
```

3 Server neu starten

Starten Sie jetzt mit XAMPP den Apache neu. ■

21.3.3 phpMyAdmin in die Documentroot legen

phpMyAdmin ist eine Benutzeroberfläche für die Administration von MySQL. Damit Sie ohne die von uns gelöschte XAMPP-Website Zugriff auf phpMyAdmin haben, müssen Sie das gesamte Verzeichnis `c:/xampp/phpmyadmin/` in die von Ihnen gewünschte Documentroot verschieben. In unserem Beispiel ist es das Laufwerk E:

Browser starten

Nachdem Sie dies getan haben, starten Sie den Browser. In der Liste müssten Sie jetzt den Eintrag PHPMYADMIN sehen. Klicken Sie auf den Eintrag, dann sehen Sie die Oberfläche von phpMyAdmin. Unter Windows 7 kann es vorkommen, dass die Verzeichnisse im Browser nicht angezeigt werden. Geben Sie dann `http://localhost/phpMyAdmin/` direkt in die Adressleiste Ihres Browsers ein, um phpMyAdmin zu starten.

Falls phpMyAdmin nicht aufgerufen wird, überprüfen Sie alle Schritte, und achten Sie darauf, dass Apache und MySQL gestartet sind.

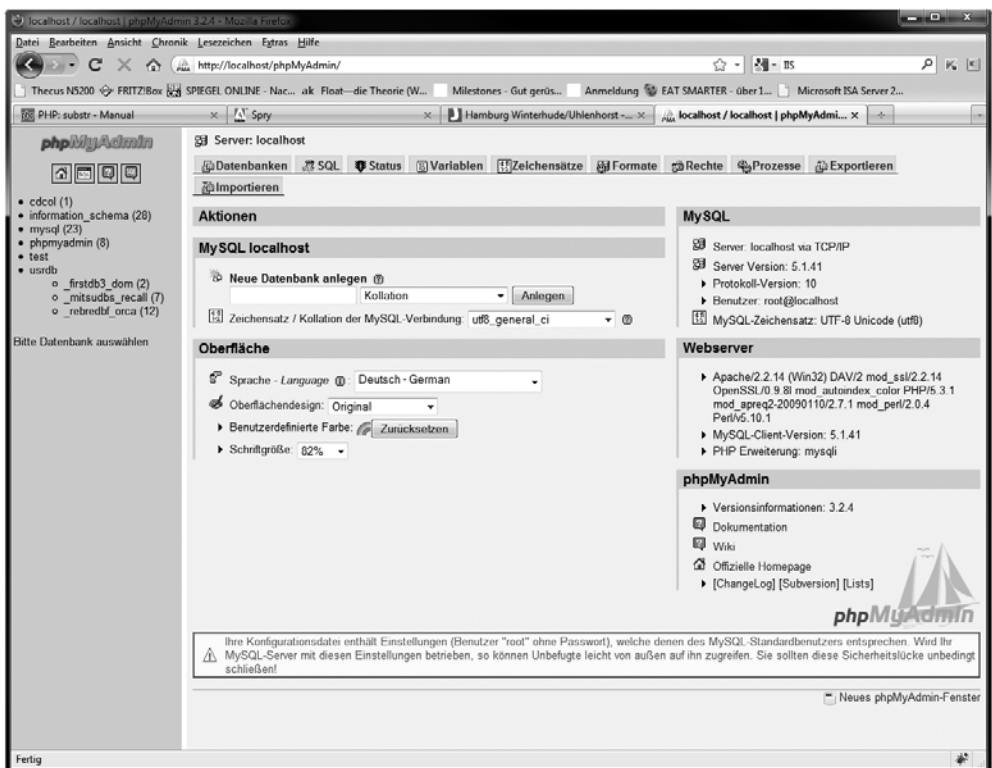


Abbildung 21.9 Die phpMyAdmin-Oberfläche



Vorsicht beim Deinstallieren

Sichern Sie unbedingt alle Dokumente und Datenbanken, wenn Sie XAMPP einmal deinstallieren möchten. Man kann bei der Deinstallation zwar angeben, dass die Verzeichnisse und Daten erhalten bleiben sollen, jedoch übersieht man den Dialog sehr schnell.

21.3.4 MySQL

Beim Start von phpMyAdmin wird automatisch auf MySQL zugegriffen. Wenn Sie also keine Fehlermeldungen erhalten haben, können Sie davon ausgehen, dass MySQL korrekt installiert ist.

MySQL verwalten

- MySQL selbst besitzt keine grafische Benutzeroberfläche. Bei dem Basissystem müssten Sie eigentlich alle Befehle direkt auf der Kommandozeile eingeben – wie in alten MS DOS-Zeiten. Dies wollen wir Ihnen aber ersparen. Wesentlich komfortabler lässt sich MySQL mit phpMyAdmin oder mit dem Tool MySQL Workbench verwalten. Beide Programme haben wir Ihnen ebenfalls auf der Buch-DVD im Ordner *Webserver* beigelegt.

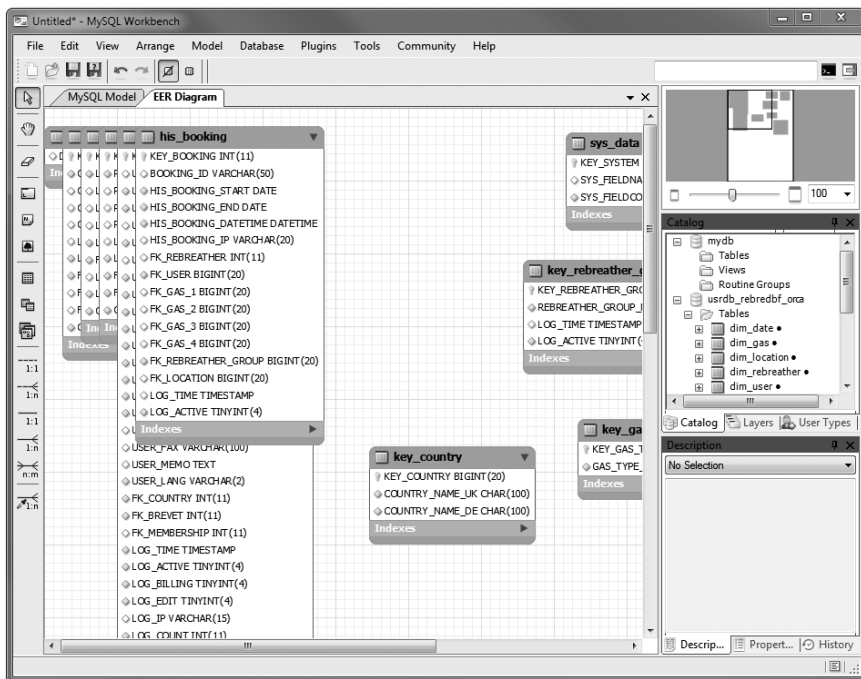


Abbildung 21.10 MySQL Workbench

Nachdem wir nun einen lokalen Webserver installiert haben, bereiten Sie Dreamweaver für die Arbeit mit dynamischen Websites vor. In diesem Kapitel erfahren Sie, wie das geht.

22 Dynamische Sites in Dreamweaver einrichten

Vor allen weiteren Arbeiten mit dynamischen Websites muss Dreamweaver für den Testserverbetrieb eingerichtet werden. Wichtig ist dabei, dass sich der lokale Stammordner Ihrer Site im Dokumentverzeichnis Ihres zuvor eingerichteten Webserverns befindet und natürlich der Testserver, wie in den vorherigen Kapiteln beschrieben, installiert ist. Einen ersten Test sollte er auch bestanden haben.

22.1 Testserver-Einstellungen

Die Einstellungen für Dreamweaver werden in der Site-Verwaltung vorgenommen. Legen Sie dort am besten unter SITE – NEUE SITE... eine neue Site an.

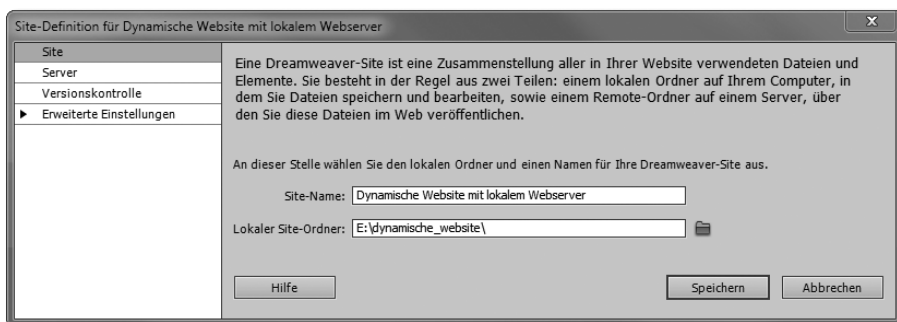


Abbildung 22.1 Stammordner in der Documentroot

Dann wechseln Sie in der SITE-DEFINITION zum Menüpunkt SERVER. Klicken Sie auf das Plusymbol **+** (siehe Abbildung 22.2), um eine neue Serverkonfiguration hinzuzufügen. Natürlich können Sie die folgenden Schritte auch mit jeder bestehenden Site durchführen.

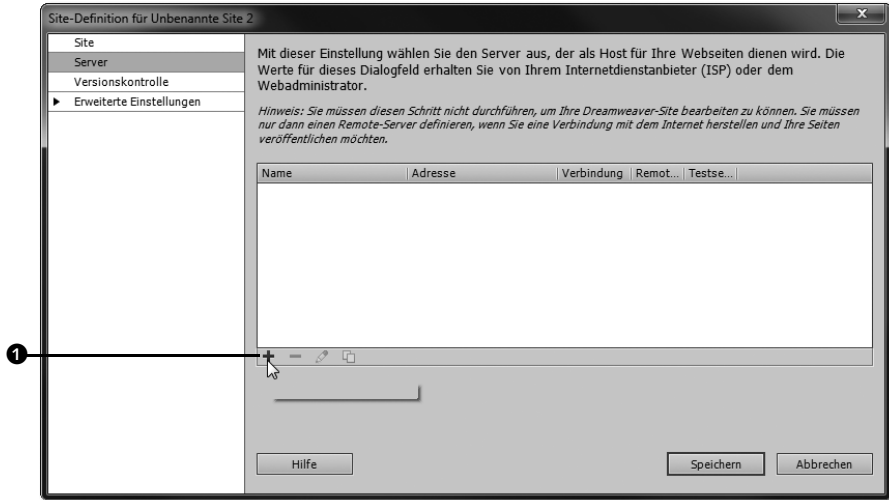


Abbildung 22.2 Serverkonfiguration hinzufügen

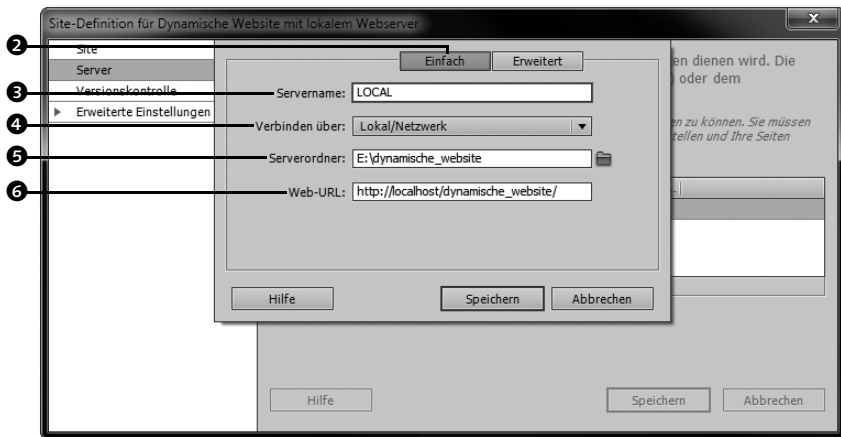


Abbildung 22.3 Menüpunkt »Server« in der Site-Definition

Achten Sie bitte darauf, dass im Menü die Schaltfläche EINFACH 2 aktiv ist. Zunächst müssen Sie hier dem Server einen Namen geben 3. Dieser Name hat keinen Einfluss auf die Funktion, sondern dient ausschließlich Ihrer eigenen Verwaltung. In Dreamweaver können mehrere verschiedene Testserver konfiguriert werden, daher die notwendige Benennung, doch dazu später mehr.

Zugriff auf den Server

Als Nächstes muss Dreamweaver erfahren, wie der Zugriff auf den Server erfolgen soll. In den meisten Fällen – wie auch in unserem – wird das die Einstellung

LOKAL/NETZWERK sein. Im Dropdown-Feld VERBINDEN ÜBER ④ können Sie diese auswählen.

Sie können mit Dreamweaver auch direkt auf einem *echten* Webserver im Internet arbeiten. Wenn Sie dies möchten, wählen Sie unter VERBINDEN ÜBER einfach FTP aus und geben dann die Zugangsdaten ein.

Testserver-Ordner

Der SERVERORDNER ⑤ ist in unserem Fall identisch mit dem lokalen Stammordner und muss entsprechend ausgewählt werden. Dreamweaver nimmt Ihnen die Arbeit ab, die Dateien immer auf den Server zu kopieren. Wenn Sie FTP als Zugriff angegeben haben, wird beim Arbeiten die aktuelle Datei automatisch auf den Server kopiert und die Vorschau über die URL des Webserver generiert. Dieses Verfahren dauert jedoch deutlich länger als bei einem lokal installierten Testserver.

Adresse der Website

Die WEB-URL ⑥ ist der Pfad bzw. die HTTP-Adresse, die Sie im Browser eingeben müssen, um die Website anzuzeigen. In der Regel trägt Dreamweaver hier nichts ein, zur Funktion ist die Angabe jedoch zwingend notwendig. Nur wenn die WEB-URL korrekt angegeben wurde, kann eine Website durch Dreamweaver ausgeführt werden. Die meisten Fehler beim Anlegen eines Testservers treten genau an dieser Stelle auf. Überprüfen Sie daher den Eintrag auf seine Korrektheit.

Die Web-URL setzt sich aus der lokalen Adresse des Servers – das ist in den meisten Fällen `http://localhost` oder die IP 127.0.0.1 – und dem Ordernamen Ihrer Website zusammen. Abbildung 22.3 zeigt, wie die Bezeichnungen eingetragen werden müssen.

Servermodell angeben

Im Menüpunkt ERWEITERT ⑦ (siehe Abbildung 22.4) wird das Testsystem ausgewählt. In unserem Fall ist es ein PHP-MySQL-System ⑧. Welcher Server (IIS oder Apache) tatsächlich im Hintergrund agiert, spielt dabei keine Rolle. Wichtig ist nur die Auswahl des Systems, um die spätere Syntax für die Abfragen usw. festzulegen.

Anschließend wird der Server im Hauptbildschirm des Dialogfensters noch als Testserver aktiviert ⑨ (siehe Abbildung 22.5). (Ein Testserver kann übrigens auch als FTP-Server konfiguriert werden.) Ebenso können Sie mehrere Testserver anlegen und nach Bedarf umschalten. So könnten Sie einen firmeninternen Testserver nutzen, um die Websites zu entwickeln – nichts anderes machen wir hier –, und einen weiteren Live-Server, der im Intranet zugänglich ist.

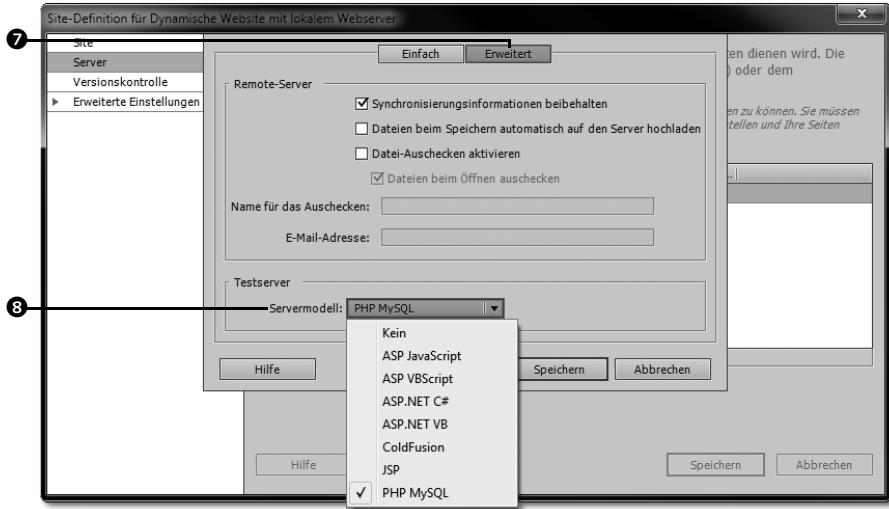


Abbildung 22.4 Erweiterte Servereinstellungen

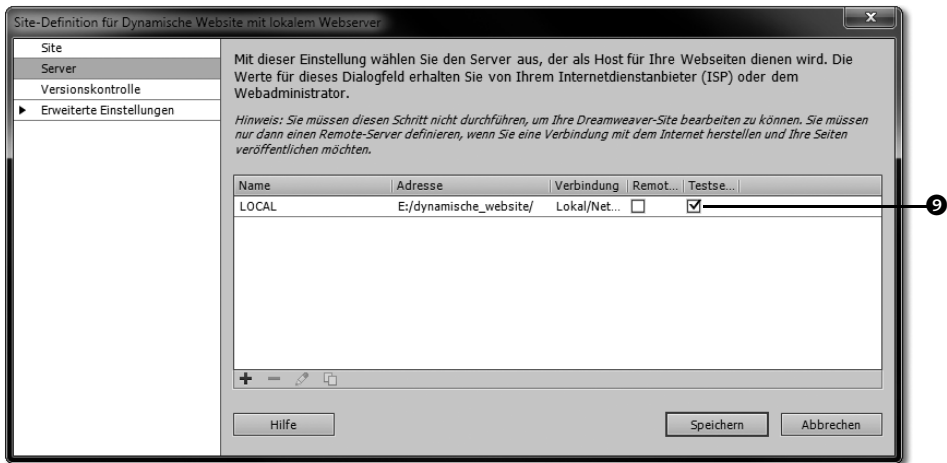


Abbildung 22.5 Server aktivieren

22.2 Site-Einstellungen testen

Nach dem Anlegen des Servers in Dreamweaver sollte er getestet werden. Erstellen Sie dazu am besten ein neues Dokument *index.php*. Wichtig ist die Endung *.php*: Der Webserver erkennt sonst nicht, dass es sich bei dieser Datei um ein PHP-Skript handelt. Löschen Sie den gesamten Quelltext der Datei, und fügen Sie folgende Befehlszeile ein:


```
<?php
phpinfo()
?>
```



Abbildung 22.6 Aufruf der Funktion phpinfo() im Quelltext

Mit diesem Befehl können Sie die PHP-Grundeinstellungen anzeigen lassen. Wechseln Sie anschließend wieder in die Layoutansicht.

Vorschau dynamischer Sites

In der Layoutansicht (siehe Abbildung 22.7) können Sie die Einstellungen durch einen Klick auf die LIVE-ANSICHT  testen. Im Dokumentfenster wird das PHP-Skript direkt ausgeführt, und Sie sehen im Layout die automatisch von PHP generierten Inhalte.

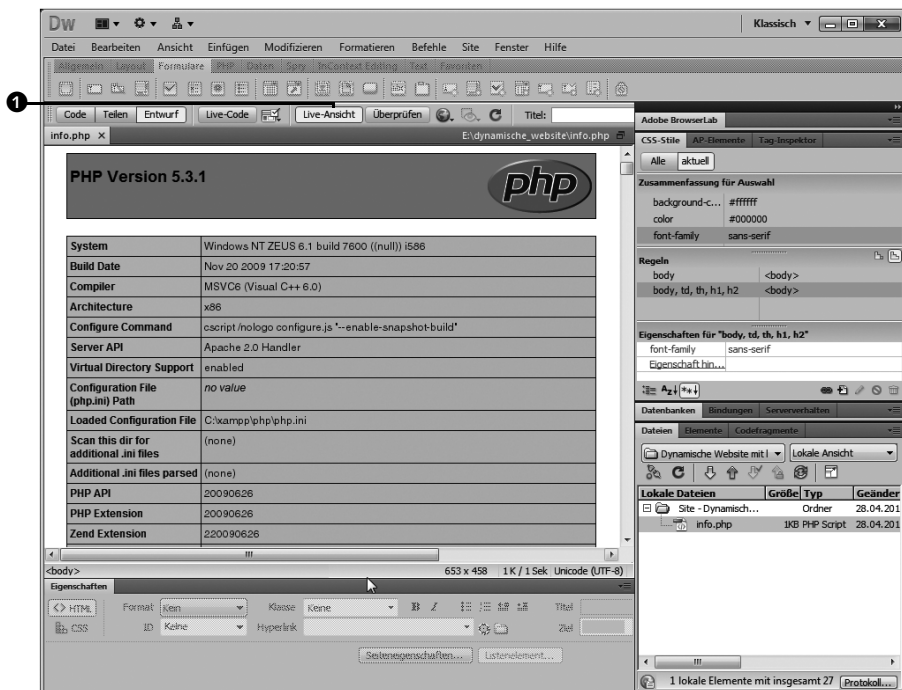


Abbildung 22.7 Den Testserver aktivieren

Vorschau in der Live-Code-Ansicht

In Dreamweaver CS5 können Sie neben der LIVE-ANSICHT auch eine LIVE-CODE-Ansicht aktivieren **1**. Im Layoutfenster zeigt Dreamweaver dann den von PHP erzeugten Quelltext direkt an.

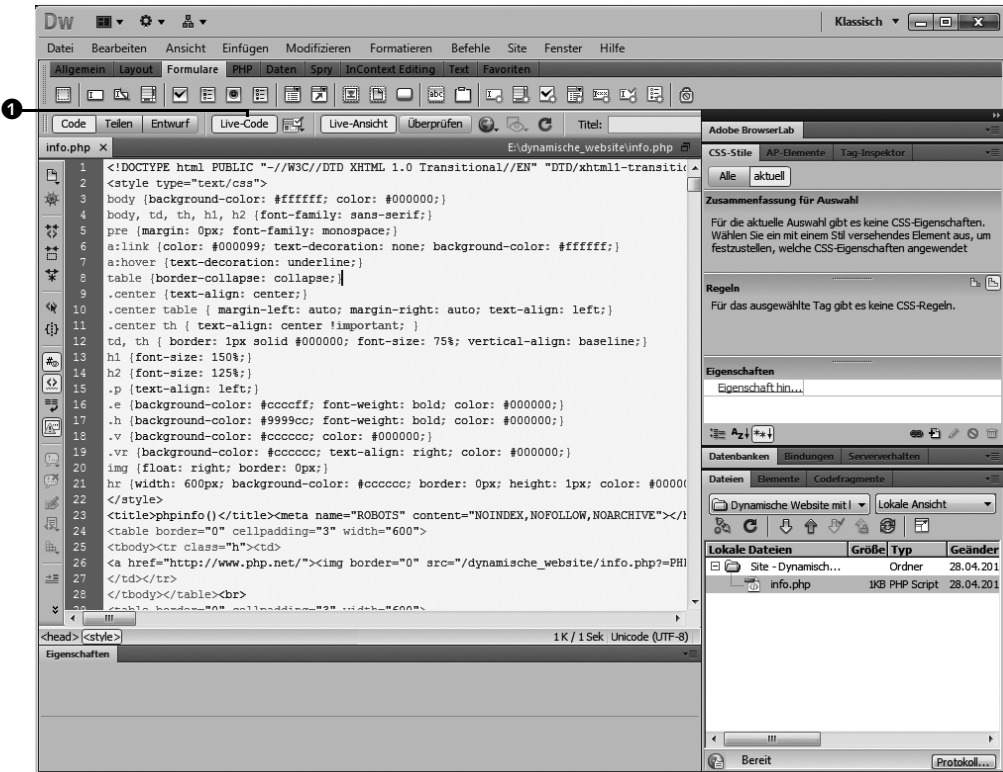


Abbildung 22.8 Live-Code-Ansicht

Wenn die Live-Ansichten nicht funktionieren (Fehlersuche)

Sollte dies wider Erwarten nicht funktionieren, gehen Sie am besten folgende Liste durch:

- ▶ Überprüfen Sie, ob der Webserver aktiv ist.
- ▶ Haben Sie die Site korrekt definiert? Falls nicht, erscheint entweder das Icon zur Live-Ansicht nicht, oder Dreamweaver gibt eine Fehlermeldung aus.
- ▶ Stellen Sie sicher, dass Ihre Dateien wirklich in der Documentroot liegen. In 90 Prozent aller Fehlerfälle in Schulungen sind die Pfade nicht korrekt eingegeben worden, oder die Web-URL ist falsch.
- ▶ Überprüfen Sie der Reihe nach alle Einstellungen.

Keine Frame-Vorschau dynamischer Sites

[«]

Dynamische Sites können Sie lokal nicht über ein Frameset aus Dreamweaver heraus aufrufen. Starten Sie immer ein Inhaltsdokument, um eine Vorschau zu erhalten, oder rufen Sie das Frameset direkt aus dem Browser heraus auf, indem Sie den Pfad zum Frameset eingeben.

Testserver-Ansicht

In der Site-Ansicht steht Ihnen jetzt auch die Option TESTSERVER ② zur Verfügung. Aktivieren Sie Ihren Testserver durch einen Klick darauf. Die Ordner auf der linken Seite ③ werden jetzt rot dargestellt.

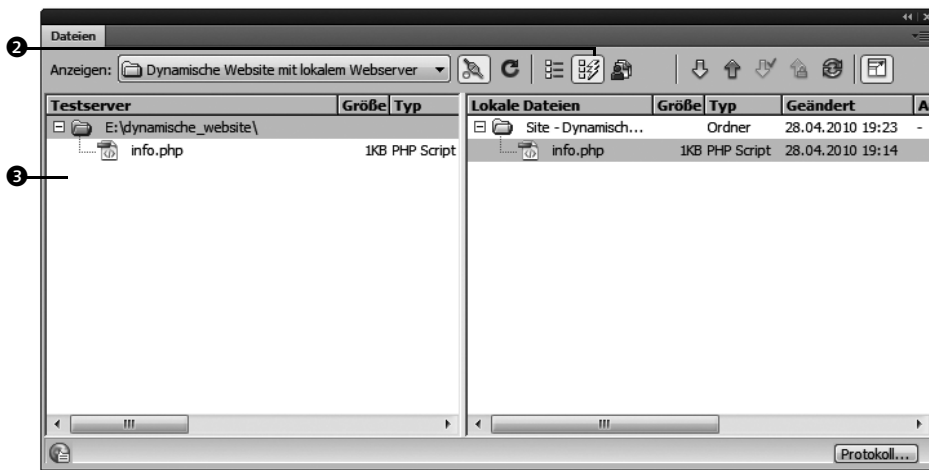


Abbildung 22.9 Testserver-Ansicht in der Site-Verwaltung

Wenn Sie Ihren Testserver lokal betreiben, ist diese Ansicht etwas verwirrend, zumal ja eigentlich beide Seiten gleich sind. Für einen Testbetrieb spielt es aber keine Rolle, wo sich der Testserver befindet. Auch Ihr eigener Rechner wird über eine IP-Adresse angesprochen (127.0.0.1).

Dreamweaver ist jetzt für den Betrieb mit einem Testserver vollständig eingerichtet.

Daten auf dem Testserver sichern!

[!]

Wenn Sie auf einem Windows-System arbeiten, werden Apache und IIS auf Ihrem Laufwerk C: installiert. Dieses wird in den meisten Fällen auch Ihr Systemlaufwerk sein und ist sehr anfällig für Systemfehler. Bei einer Neuinstallation Ihres Systems sind im schlimmsten Fall alle Daten für immer verloren. Sorgen Sie daher dafür, dass die Daten im Rootverzeichnis des Webservers regelmäßig auf einer anderen Partition oder CD gesichert werden.

22.3 Buchwebsite für dynamische Inhalte einrichten



Buchwebsite

Wir gehen jetzt davon aus, dass Sie bereits einige Übung im Umgang mit Dreamweaver und dessen Layoutfunktionen haben. Wir sparen uns daher in allen weiteren Schritten beim Aufbau der Buchwebsite diese Erklärungen. Wenn es an der einen oder anderen Stelle hakt, lesen Sie bitte in den entsprechenden Kapiteln nach.

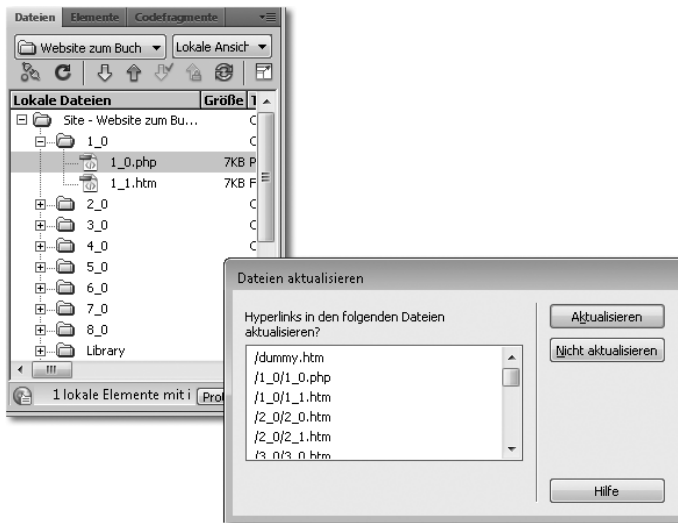
Ebenso ist es erforderlich, die einleitenden Kapitel zum Aufbau des dynamischen Teils zu lesen. Ohne das Wissen über die Funktionen ist es sehr schwer, die Arbeitsschritte nachzuvollziehen.

[] Schritt-für-Schritt: Website für dynamische Inhalte einrichten

1 Dateiendungen ändern

Um auch weiter mit der von Ihnen erstellten Buchwebsite arbeiten zu können, müssen einige Änderungen an den Site-Einstellungen erfolgen.

Ändern Sie die Dateiendung aller HTML-Dokumente in *.php*. Nehmen Sie dies bitte unbedingt im Dateifenster von Dreamweaver vor, und lassen Sie alle Verlinkungen automatisch aktualisieren, wenn Dreamweaver dies vorschlägt.



2 Lokalen Stammordner kopieren

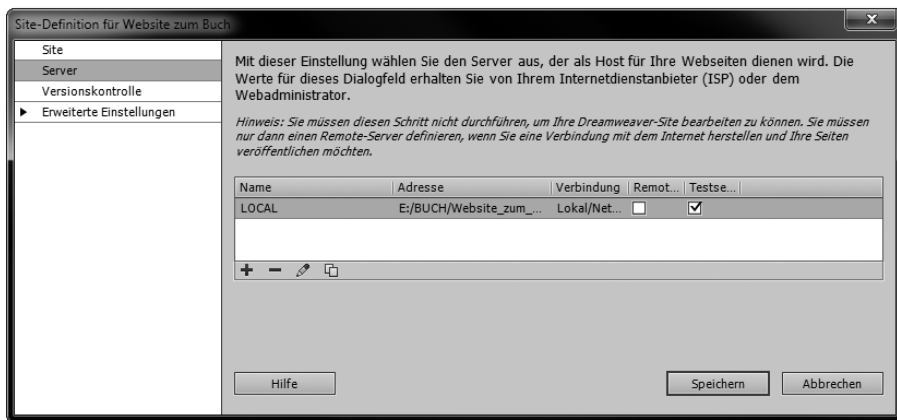
Kopieren Sie anschließend den gesamten bisherigen lokalen Stammordner Ihrer Übungswebsite in das Dokumentverzeichnis Ihres lokalen Webservers.

3 Site-Einstellungen anpassen

Ändern Sie die Site-Einstellungen Ihrer Übungswebsite so, dass der in die DokumentRoot kopierte Ordner als lokales Stammverzeichnis angegeben wird, oder erstellen Sie eine neue Site mit den soeben genannten Vorgaben.

4 Einstellungen für den Testserver

Legen Sie anschließend, wie in der folgenden Abbildung gezeigt, die Einstellungen für den Testserver fest. Sie müssen natürlich Ihre lokalen Parameter eintragen.



5 Geänderte Einstellungen testen

Öffnen Sie nun das Dokument `1_0.php`, und starten Sie mit `[F12]` bzw. `[Alt]+[F12]` am Mac eine Browservorschau. Wenn Sie alles richtig angelegt haben, wird dieses Dokument genauso angezeigt wie bisher.

Eine Live-Ansicht können Sie erst starten, wenn sich im Dokument PHP-Skripte befinden. Dies ist zum jetzigen Zeitpunkt noch nicht der Fall.

Wenn Sie nun alles beendet haben, steht einer erfolgreichen Arbeit nichts mehr im Wege. ■

Richard Beer, Susann Gailus

Adobe Dreamweaver CS5

Das umfassende Handbuch