

# Gratis Webseiten- vorlage für Musiker

## Schritt-Für-Schritt-Anleitung

Quelle: [www.bandologie.de/webseitenvorlage](http://www.bandologie.de/webseitenvorlage)

**TEIL EINS** → Grundverständnis / Einblick für angehende Webmaster (Seite 2 bis 8)

**TEIL ZWEI** → Schritt-Für-Schritt-Anleitung (Seite 9 bis 23)

Auf den folgenden Seiten findest du eine allgemein verständliche Anleitung, mit der meiner Einschätzung nach jeder, der sich intensiv mit der Anleitung und den weiterführenden Quellen beschäftigt, eine eigene Webseite erstellen kann. Doch zunächst:

## 1. Blitz-Anleitung (für Insider, die webtechnisch bereits fit sind)

- 1) Lade die Webseitenvorlage, die in einer ZIP-Datei verpackt ist, mit dem Link [www.bandologie.de/webseitenvorlage](http://www.bandologie.de/webseitenvorlage) herunter, entzippe sie, lade die Dateien auf deinen lokalen Server oder online auf deinen Webspace und öffne die gleichen Dateien parallel dazu in einem entsprechenden Editor.
- 2) Lies die Hinweise, die in englischer Sprache innerhalb des Quelltextes vermerkt sind. Für einen Profi wie dich sollte das zur Orientierung genügen.
- 3) Passe die HTML-, PHP- und CSS-Dateien nach Belieben und nach deinen Vorstellungen an; wechsle Texte, Fotos, Farben, Formen, Positionen, baue nach Belieben Videos oder externe Code-Elemente deiner Wahl ein.
- 4) Damit deine Musiker-Webseite die beste aller Zeiten wird ☺, empfehle ich dir das Ebook: „Die wichtigsten Kriterien für eine erfolgreiche Musiker-Webseite“:  
[www.bandologie.de/ebook-musikerwebseite](http://www.bandologie.de/ebook-musikerwebseite)
- 5) Fertig. Glückwunsch zur neuen Webseite.

**So, und jetzt nochmal ganz langsam und mit vielen Erklärungen für Webmaster-Anfänger:**



# 1. Voraussetzungen um mit dieser Webseiten-Vorlage arbeiten zu können

Liebe Musiker,

aus dieser Vorlage kann man, je nach Band, eine verkaufstarke Webseite erstellen, die sehr zum Erfolg eurer Band beitragen kann. Soweit die guten Nachrichten. Nun kommt der Haken: Man muss sie selbst erstellen und das macht Mühe.

Von jemandem der mit einer solchen kostenlosen, gratis, geschenkten und wirklich absolut nichts-kostenden For-Free-Vorlage arbeitet, erwarte ich Eigeninitiative. Das bedeutet konkret: Du solltest in der Lage sein einige Hilfsmittel selbstständig per Internet-suchmaschine zu finden. Du solltest ein Verständnis für logische Zusammenhänge mitbringen, sowie die Fähigkeit und Geduld genau zu lesen. Ich denke es ist verständlich und einleuchtend, dass ich für dieses Gratis-Template<sup>1</sup> keinen Gratis-Support anbiete.

## Technische Voraussetzungen um mit dieser Webseiten-Vorlage arbeiten zu können

Falls du technisch bisher nicht so versiert bist, ist das kein Problem. Du wirst die folgenden Punkte im Laufe dieser Anleitung verstehen:

- 1) Du benötigst einen Webspace mit PHP-Funktion (Angebote ohne PHP-Funktion sind häufig preisgünstiger). Außerdem benötigst du eine Domain.
- 2) Um auf deinem Computer an der Gestaltung der Webseite arbeiten zu können, benötigst du ein kleines Programm, einen sogenannten lokalen Server. Damit hast du die Möglichkeit, die Webseite erstmal nur auf deinem Computer zu gestalten und Dinge auszuprobieren. Später kannst du die gleiche Webseite, die auf dem lokalen Server funktioniert ins Internet, also auf einen öffentlich zugänglichen Server, stellen. Wie du ein solches Programm findest, dazu gibt es gleich Tipps. Zudem benötigst du ein Textbearbeitungsprogramm oder einen Editor; auch dazu später genauere Infos.

## 2. Überblick: Was ist mit diesem Template möglich und was nicht?

**Das geht:** Mit Hilfe dieses anpassbaren Webseiten-Templates und dieser Anleitung kannst du mit etwas Eigeninitiative eine professionelle Webseite erstellen. Mit einer solchen Webseite kannst du beispielsweise eine Musikband, ein Kunstprojekt oder sonstige Leistungen als Kleinunternehmer präsentieren und erfolgreich verkaufen.

**Freigabe zur Nutzung (Rechtliches):** Ich gebe hiermit den von mir geschriebenen Quellcode zu dieser Webseite für die private und kommerzielle Verwendung frei. Bitte

---

<sup>1</sup> Der Vollständigkeit halber: Ein Template ist das Gleiche wie eine Vorlage.

beachte, dass Teile des Quellcodes (beispielsweise der JWPlayer, die Newsletter-, PayPal- und YouTube-Einbindungen) nicht von mir geschrieben, sondern per Copy/Paste eingefügt sind. Was ich dir hiermit schenke, ist der von mir geschriebene HTML-Quellcode. Mit etwas Übung wirst du sehr genau erkennen können, welche Teile eingefügt und welche von mir geschrieben sind. Der Verkauf oder die Vermietung dieses Templates, dieser Anleitung oder Teilen davon ist untersagt. Davon abgesehen kannst du mit diesem Template tun was du möchtest, solange es sich um eine legale Aktivität handelt. Alles, was du mit diesem Template tust, erfolgt auf deine Verantwortung.

**Das geht nicht:** Folgendes wirst du als webtechnischer Anfänger nicht auf Anhieb in deine Webseite einbauen können: Bewegte Elemente (Flash etc.), einen Login für Nutzer oder eine selbstprogrammierte Eingabemaske für einen redaktionellen Beitrag.

### 3. Warum sich dieses Webseiten-Template für deine Band lohnen kann

Der Umgang mit diesem Template ist vergleichsweise einfach. Für alle Anfänger in diesem Bereich möchte ich erklären, warum sich diese Vorlage zum Bau einer Webseite sehr lohnen kann; warum ich nicht einfach auf <http://wordpress.org> oder <http://de.jimdo.com> verweise und warum hier ganz bewusst nicht versprochen wird, dass du deine neue Webseite innerhalb von fünf Minuten online stellen kannst. Für dieses Template brauchst du keinen Login, musst dich nicht registrieren und du kannst auf Grundlage einer weltweit gängigen Sprache (HTML)<sup>2</sup> Anpassungen wirklich aller Details deiner Webseite vornehmen. Du bist hier nicht in einem Baukastensystem, sondern erhältst einen offenen Quellcode. Das bedeutet, dass du mit diesem Template so frei bist, wie es nach meinem Verständnis möglich ist.

Hier zwei wichtige Punkte für den Aufbau und Betrieb deiner Webseite:

#### 1) Kontrolle der eigenen Webseite

Viele Künstler (und auch andere Unternehmer) neigen dazu, sich von „dem ganzen Technik-Kram“ loszusprechen. Man möchte am liebsten einen guten Freund oder einen sehr günstigen Webmaster haben, der sich um alles kümmert, was im Internet stattfindet. Das funktioniert allerdings nur sehr selten und ist für einen selbstständig handelnden Musik-Act nur bedingt zu empfehlen. Selbst ein Management oder ein Musiklabel, das mit der Betreuung einer Webseite beauftragt wird, leistet nicht immer das, was für den Künstler am besten wäre. Kontrolle erlangt man, wenn man die Webseite selbst aufgesetzt oder selbst angepasst hat (wie es mit diesem Template möglich ist) oder dadurch, dass man einen ausführenden Webdesigner gut und dauerhaft für seine Arbeit bezahlt. Bei eurer Tätigkeit als Musiker seid ihr die Chefs und frei nach Hans Peter-Zimmermann (Schweizer Gesundheits- und Unternehmensberater): „Webseite ist Chefsache.“

---

<sup>2</sup> Übrigens: Im Zusammenhang mit HTML spricht man nicht von einer Programmiersprache, sondern von einer Auszeichnungssprache. Das ist auch der Grund, weshalb in dieser Anleitung nicht von „(HTML-)Befehlen“ sondern von „Angaben“ die Rede ist.

## 2) Hundertprozentige Verlässlichkeit bei Änderungen

Eine Webseite, die bei einer Umprogrammierung nicht genau das macht, was sie machen soll, nervt. Das Gleiche gilt für den Fall, dass man etwas in seine Webseite einbinden möchte, welches technisch sehr umständlich wird, weil man ein System verwendet hat, das mit gewissen Funktionen nicht kompatibel ist. Ich hoffe, dass dich diese Anleitung vor solchen Momenten bewahren kann. Die Sprache HTML ist leicht verständlich und logisch aufgebaut. Zudem ist HTML sehr weit verbreitet und man findet unter anderem Hilfe bei <http://de.selfhtml.org> – dieser Link könnte für dich bei der Erstellung deiner neuen Webseite sehr interessant werden. Für den Fall, dass du einen Teilbereich deiner neuen Webseite an einen professionellen Webdesigner übergeben willst, kennt der sofort die Sprache (genauer: die Auszeichnungssprache, HTML) auf der die Seite basiert. Zu guter Letzt kannst du mit diesem Template bei Bedarf in alle Bestandteile eingreifen. Bei anderen Systemen, insbesondere bei Schnellbaukästen, gibt es Bereiche, auf die man als Urheber der Webseite keinen Einfluss hat. Das muss nicht unbedingt schlecht sein; die hier vorliegende Variante ist mir persönlich wesentlich angenehmer. Ich hoffe, das trifft auch deine Vorstellungen. ☺

**Ein paar Worte zum Thema „Fünf-Minuten-Webseiten“:** Auch ich finde die Vorstellung verlockend innerhalb weniger Minuten eine brauchbare Webseite aufzubauen und online zu stellen. Jedoch benötigt man die meiste Zeit ohnehin nicht für das Erlernen der technischen Kenntnisse, sondern für den Aufbau der Inhalte wie Texte, Bilder, Videos und Musik.

## 4. So ist dieses Webseiten-Template aufgebaut

Dieses Template besteht aus 13 HTML-Dateien, 9 PHP-Dateien (eine Erklärung, was das ist, folgt gleich), den Ordnern „css“ (Datei: main.css), „images“ (inklusive Unterordner 26 Bilddateien), „mp3“ (inklusive einer Sounddatei) und dem Ordner „jwplayer“ (das ist ein Download von <http://www.longtailvideo.com/players/jw-flv-player/>) – in diesem Ordner befinden sich 7 Dateien. Zudem enthält die ZIP-Datei natürlich diese Anleitung.

Wenn du diese Dateien erhalten hast, ist die Vorlage vollständig. Die Dateien sind insgesamt circa 6 bis 7 Megabyte groß (variiert etwas, je nach Anzeige).

### **Dazu folgende Erklärung für Webmaster-Anfänger:**

Es gibt einen übergeordneten Trick, nach dem diese Webseite funktioniert: Die Einbindung von Inhalten in andere Inhalte. Das Ganze kann man sich folgendermaßen vorstellen: Wenn du beispielsweise bei Facebook ein YouTube-Video postest, dann liegt das Video weiterhin auf einem Server von YouTube, zusätzlich kann man es dann auch in deinem Profil und deinem Facebook-Status sehen. Ähnlich ist es mit deinem Profilbild: Das Bild wird einmalig hochgeladen, wird allerdings bei jedem deiner Beiträge angezeigt. Es ist eingebettet.

Nach diesem Prinzip funktioniert diese Webseite. Der Unterschied ist, dass hierbei nicht nur ein Foto, sondern ein Textabschnitt eingebettet wird; genauer gesagt ein sogenannter Quelltext (Quellcode). In diesem Text ist festgelegt, wie die Webseite angeordnet sein soll und wo, welche Text-, Foto-, Video-Inhalte und Formatierungen zu sehen sein sollen. Sobald der Quelltext von einem Internet-Browser gelesen (man sagt auch „ausgegeben“) wird, sieht man das Ergebnis: die Webseite.

**Konkret bedeutet das, man schreibt den Quelltext, der mehrfach auf der Webseite auftauchen soll, nur ein Mal.**

Ein Beispiel dafür ist die Datei „navigation.html“. Die Navigationsleiste einer Webseite soll in der Regel auf jeder Unterseite auftauchen. Für den Fall, dass ein Navigationspunkt umbenannt wird, soll das üblicherweise auch auf allen Unterseiten geschehen. Deshalb macht es Sinn, die Datei „navigation.html“ nur einmal zu schreiben und sie dann immer wieder einzubetten.

Um eine solche Einbettung vornehmen zu können, habe ich für dieses Template einen einzigen Befehl aus der Skriptsprache PHP genutzt. Dieser eine Befehl ist der Grund, weshalb es in diesem Template sowohl Dateien mit der Endung „.php“ als auch Dateien mit der Endung „.html“ gibt. Das ist auch der Grund, weshalb du einen Webservice mit PHP-Funktion benötigst, um diese Webseiten-Vorlage darauf laufen zu lassen. Der PHP-Befehl lautet „include“ (zu Deutsch: beinhalten, enthalten, umfassen, einschließen).

Der Befehl, „include“, befindet sich zum Beispiel in der Datei „index.php“ (siehe Dateien dieses Templates), in der Zeile 27. Diese Zeile lautet:

```
<? include ("./navigation.html"); ?>
```

In diesem Fall ist also innerhalb der Datei „index.php“ die Datei „navigation.html“ im Quelltext eingebettet. Sobald die Datei „index.php“ in einem Browser geöffnet wird, zeigt der Browser unter anderem den Inhalt aus der Datei „navigation.html“ an.

**Um die Sache logisch und nachvollziehbar aufzubauen habe ich ein kleines Schema entwickelt. Eine typische Unterseite dieser Webseitenvorlage besteht immer aus fünf Elementen und verläuft schematisch folgendermaßen:**

- 1) <? include ("./header.html"); ?>
- 2) <? include ("./navigation.html"); ?>
- 3) <? include ("./[[ **Der hauptsächlichste Inhalt dieser Unterseite** ]].html"); ?>
- 4) <? include ("./banner.html"); ?>
- 5) <? include ("./footer.html"); ?>

Diesen Aufbau kannst anhand der Abbildung auf der folgenden Seite nachvollziehen:

# Schematische Abbildung des Aufbaus der Webseite

Zum Verständnis des Befehls „include“ (siehe Seite 5) am Beispiel der Datei „index.php“:

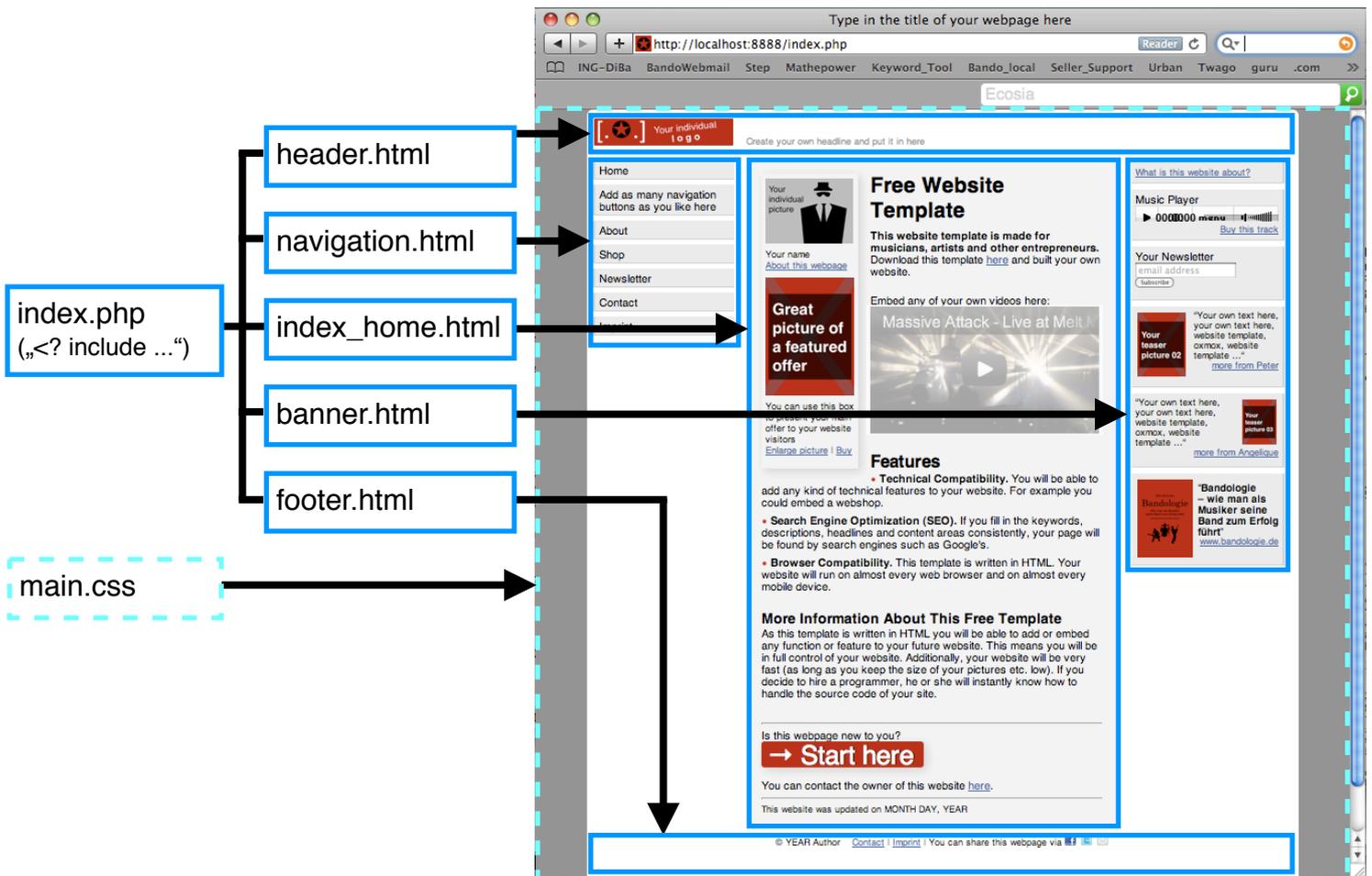


Abb. 1 Darstellung einer typischen Seite aus diesem Template.

Fast alle PHP-Dateien dieses Templates sind nach diesem Schema aufgebaut.

*Einzige Ausnahme: Auf der Unterseite „shop\_see\_all\_photos.php“ habe ich den Bereich „banner.html“ weggelassen, damit eure Produktfotos größer dargestellt werden können.*

Es gibt also vier Dateien, die auf jeder Unterseite, außer der gerade genannten, angezeigt, sprich vom Browser ausgegeben, werden: „header.html“, „navigation.html“, „banner.html“ und „footer.html“ (siehe Abbildung). Zudem gibt es jeweils eine fünfte, eingebundene Datei, in der sich der hauptsächliche Inhalt der jeweiligen Unterseite abspielt (in der Abbildung ist das die Datei „index\_home.html“).

## Beispiel

Ein Beispiel dafür ist die Datei „shop.html“, in der alles definiert ist, was den Shop ausmacht. Das Schema für die Datei „shop.php“ sieht dann folgendermaßen aus:

- 1) <? include („./header.html“); ?>
- 2) <? include („./navigation.html“); ?>
- 3) <? include („./shop.html“); ?>
- 4) <? include („./banner.html“); ?>
- 5) <? include („./footer.html“); ?>

Das bedeutet: Wenn du etwas am Inhalt deines Shops ändern möchtest, öffnest du die Datei „shop.html“. Falls du die HTML-Datei umbenennst, musst du sie also auch in der entsprechenden PHP-Datei, in der sie per „include“ eingebunden ist, umbenennen.

Im Browser wird bei diesem Template immer die „.php“ aufgerufen und angezeigt. Ein Link zu einer Unterseite deiner neuen Webseite wäre zum Beispiel:

<http://www.name-deiner-neuen-webseite.com/shop.php>

Um die Sache übersichtlich zu halten, habe ich die zusammengehörigen PHP- und HTML-Dateien entsprechend identisch benannt: „shop.php“ beinhaltet das gerade gezeigte Schema, also den Seitenaufbau. „shop.html“ beinhaltet alles, was den Shop ausmacht, beispielsweise beschreibende Texte zu den Produkten im Shop, Fotos und Links, mit denen man die Produkte kaufen kann.

*Bei dieser Benennung gibt es eine Ausnahme: Die Datei „index.php“ und die Datei „index\_home.html“ gehören ebenfalls zusammen. Der Grund, weshalb ich die HTML-Datei in diesem Fall anders benannt habe, ist, dass ein Web-Browser immer zuerst die Datei, die als „index“ benannt ist, ansteuert und anzeigt. Zudem würde, wenn es „index.php“ und „index.html“ geben würde, zuerst die HTML-Datei vom Browser angesteuert werden. Bei diesem Template soll natürlich die „index.php“ (inklusive Navigation etc.; siehe Schema, oben) angezeigt werden und nicht nur die Datei „index.html“ (Fließtext der Startseite).*

## 6. Blitz-Einblick für HTML-Einsteiger

Zum Verständnis einer HTML-Webseite: Die einfachste Form einer Seite wäre, einfach nur Text auf einer Seite zu zeigen. Das könnte folgendermaßen laufen:

```
<html>
<head> (Hier kommen einige Angaben hin, zu denen es im Folgenden Tipps gibt) </head>
<body> (Hier kommt der Text hin, der auf der Webseite sichtbar ist) </body>
</html>
```

Diese Webseite wäre fertig.

Interessanter wird es, wenn zumindest ein paar Links in die Webseite eingebaut sind oder es weitere Unterseiten gibt. Typischerweise wird bei einer Webseite nicht alles auf einer Seite präsentiert, sondern inhaltlich sortiert, auf mehreren Unterseiten.

### So kannst du HTML-Dateien bearbeiten

HTML-Dateien sind gewöhnlichen Textdateien sehr ähnlich. Das bedeutet, dass du sie bei Bedarf in einem ganz gewöhnlichen Textbearbeitungsprogramm (z.B. Microsoft Word, Open Office oder Apple Pages) bearbeiten kannst. Komfortabler für dich wird es, wenn du dir einen HTML-Editor besorgst. Es gibt diverse kostenlose Editoren (bitte dazu eine Suchmaschinenanfrage starten oder im Kapitel „Weiterführende Links“ in dieser Anleitung nachschauen.)

## HTML kann man in zwei Arten von Text unterteilen:

- 1) Angaben, die den Fließtext und andere Elemente formatieren und positionieren. Diese Angaben sieht man zwar im Quelltext, jedoch nicht auf der späteren Webseite. Die nicht auf der Webseite sichtbaren Elemente erkennt man daran, dass sie von den Zeichen „<“ und „>“ eingerahmt sind. Webmaster-Laien würden bei diesen Angaben von Befehlen sprechen.

Im gleich folgenden Beispiel sind diese Angaben rot markiert.

- 2) Den sichtbaren Fließtext und die Elemente der Webseite, wie beispielsweise Fotos und Videos.

Im gleich folgenden Beispiel sind diese Bestandteile schwarz.

Im Quelltext siehst du beides gemischt.

### Beispiel

Eine typische HTML-Zeile sieht so aus:

```
<p>You can contact the owner of this website <a href="contact">here</a>.</p>
```

## Wie wird aus den Dateien in diesem Template eine Webseite?

Die Dateien in diesem Template ergeben zusammen eine Webseite. Der Grund, weshalb die Dateien zusammengehören, ist handgemacht: Sie verlinken aufeinander, behandeln das gleiche Thema und sind stilistisch aufeinander abgestimmt. Mehr dazu im Kapitel „Der geheime Masterplan in der Hinterhand: CSS (Cascading Style Sheets)“.

## Teil zwei: Schritt-Für-Schritt-Anleitung

# 7. Jetzt geht es los: Blitz-Einstieg in den Webseiten-Aufbau

### So stellst du deine neue Webseite online – die Zusammenhänge zwischen Webhoster, PHP-Paket, FTP und lokalem Server

Um die Dateien aus diesem Template anpassen und das Ergebnis in einem Browser (z.B. Firefox) sehen zu können, musst du einen lokalen Server nutzen oder online in einem Webspaces arbeiten. Aus diesem Grund wird dieser Teil erklärt, noch bevor du gleich erfährst, wie du die HTML-Dateien nach deinen Wünschen anpassen kannst.

#### Webhoster

Ein Webhoster vermietet Webspaces und vermittelt in der Regel auch die Buchung von Domains. Häufig werden auch managed Server und diverse Zusatz-Services angeboten.

#### Webspaces

Ein Webspaces ist Speicherplatz im Internet. Für eine professionelle Webseite bucht man sich in der Regel ein Webpaket bei einem Webhoster. Ein solches besteht üblicherweise aus einer Domain und Speicherplatz im Internet, auch genannt Webspaces.

#### PHP-Paket

Ein PHP-Paket ist ein Webspaces-Angebot, das die Verwendung der Skriptsprache PHP erlaubt. Um mit diesem Template (ohne den Code grundlegend umzuschreiben) eine Webseite zu betreiben, benötigst du ein Web-Paket, bei dem die Verwendung von PHP möglich ist.

#### FTP

FTP (File Transfer Protocol) ist eine Übertragungstechnik, mit der man beispielsweise die nötigen Dateien, die zusammen eine Webseite ergeben, auf einen Webspaces laden kann. Es gibt kostenlose FTP-Programme, mit denen man beispielsweise die (bei Bedarf angepassten) Dateien aus diesem Template auf den eigenen Webspaces kopieren kann.

### Vorab-Wissen, um erstmals eine Webseite online zu stellen – Begriffe und Zusammenhänge, erklärt für Einsteiger

Hier ein paar konkrete Angaben, damit du in etwa einschätzen kannst, was du dir buchen könntest: Mein Webpaket kostet beispielsweise € 3,99 pro Monat also € 47,88 pro Jahr. Es enthält 6 GB Speicherplatz (wahrscheinlich erheblich mehr als du benötigen wirst) und drei Domains ([www.bandologie.com](http://www.bandologie.com), [www.bandologie.de](http://www.bandologie.de) und [www.nilskolonko.de](http://www.nilskolonko.de))<sup>3</sup>.

---

<sup>3</sup> Wenn dich interessiert, bei welchem Anbieter (Webhoster, Internet Service Provider) ich seit Jahren Kunde bin (möchte hier keine Werbung betreiben), findest du diese Info, wenn du eine meiner Domains bei [www.denic.de](http://www.denic.de) nachschlägst. Falls du gerade im Begriff bist erstmals Webmaster zu werden, ist die Webseite der Denic ohnehin eine Seite, die du kennen solltest und das Nachschlagen ist eine nette Übung. ;)

Von deinem Computer aus werden später zum Beispiel die Dateien aus diesem Template (plus alle von dir verwendeten Fotos, Musikdateien und Videos) auf den Webspaces kopiert. Ab diesem Zeitpunkt ist deine Webseite unter der (in der Regel vom Anbieter für dich verknüpften) Domain abrufbar.

Ein Webhoster betreibt speziell konfigurierte Festplatten, die jeden Tag 24 Stunden lang laufen. Diese Festplatten nennt man Server. Bei Webhosting-Angeboten, wie beispielsweise dem von mir gebuchten, hat nicht jede Webseite einen eigenen Server und unter Umständen auch keine eigene IP-Adresse, sondern ein Server wird vom Anbieter in mehrere Abschnitte unterteilt. Diese Abschnitte erhalten dann jeweils einen Domainnamen und lassen sich somit per Browser ansteuern. Mein Webspaces beispielsweise, ist also nur ein Teil eines Servers. Du könntest dir auch selbst einen Server kaufen oder mieten und betreiben, allerdings solltest du dann sehr genau wissen was du tust und dich wirklich exzellent mit entsprechenden Sicherheitsmaßnahmen auskennen. Das ist nur etwas für echte Webmaster-Cracks mit viel Erfahrung. Zusammenfassend: Eine Webseite ist auf einem Server (meist einem separat steuerbaren Teil davon, genannt Webspaces) gespeicherter Inhalt, der sich per Browser aufrufen lässt.

## So besorgst du dir einen lokalen Server

Um auch ohne Webspaces und Domain (wie gerade im Text beschrieben) eine Webseite aufbauen und testen zu können, kannst du das, was später ein Server erledigt, auf deinem Computer imitieren. Speziell für diesen Zweck gibt es kleine Programme, die genau das ermöglichen. Du kannst dir eines dieser Programme installieren, zum Beispiel „XAMPP“ (für Windows-Nutzer) oder „MAMP“ (für Apple-Nutzer). Starte einfach eine Suchanfrage in einer Internetsuchmaschine und du wirst schnell fündig werden.

**Tip:** Im Quellcode dieses Templates lautet der Name der Domain „http://localhost:8888“. Das bedeutet, du kannst entweder diesen Namen so lassen und mit deinem lokalen Server diese Adresse ansteuern oder den Namen entsprechend ersetzen. Wenn du beispielsweise die Domain „http://www.meine-eigene-webseite.de“ registriert hast und die Dateien auf deinen Webspaces lädst, müsstest du den Namen „http://localhost:8888“ durch „http://www.meine-eigene-webseite.de“ austauschen (beispielsweise mit der Funktion „Suchen & Ersetzen“, die in vielen Text-Editoren verfügbar ist).

**Tip:** Aus mir heute unerklärlichen Gründen mochte ich jahrelang die Idee nicht, eine Webseite auf meinem Computer zu erstellen. Ich wollte sofort online sein und alles online entwickeln. Das funktioniert zwar ebenfalls, allerdings ist man mit einem lokalen Server erheblich unabhängiger und kann zudem jede Änderung blitzschnell sehen. Meine Internetverbindung und der Server meines Webhosters sind zwar schnell, doch die Entwicklung mit dem lokalen Server (ich nutze MAMP) läuft schneller. Zudem kann ich bei Bedarf auch ohne Internetverbindung, beispielsweise im Zug oder in einem Café, basteln.

**Zusammenfassung:** Sobald du auf deinem lokalen Server (siehe oben) die Dateien für deine Webseite angepasst hast (dazu gibt es gleich Hilfe), kannst du sie per FTP auf deinen Webspaces bei deinem Webhoster laden. Sobald die Dateien dorthin kopiert sind, ist deine Webseite online. Falls du diese Schritte das erste Mal durchführst, kannst du jetzt

anfangen, das Template entweder auf deinem Webservice oder auf deinem lokalen Server zu bearbeiten.

## 8. Die wichtigsten Ausdrücke, um deine neue Webseite nach deinen Wünschen zu formen (HTML für Anfänger, im Bezug auf dieses Template)

**Legende:** Die HTML-Angaben, die man später auf der Webseite nicht sieht, sind rot markiert. Die Angaben, die man später auf der Webseite sieht sind schwarz.

Folgende Ausdrücke der Auszeichnungssprache HTML können dir beim Aufbau deiner neuen Webseite behilflich sein.

### Beispiele aus dem Webseitenbereich „<body>“:

`<a href="http://www.ein-eingefügter-Link.de"> [[ beschreibender Text zum Link ]] </a>`

→ Mit diesem Befehl kannst du an beliebiger Stelle einen Link auf der Webseite einfügen.

`<p> [[ beliebiger Text ]] </p>`

→ Fügt einen Absatz hinzu.

`<br />`

→ Fügt einen Zeilenumbruch hinzu.

`<strong> [[ beliebiger, wichtiger Text wird fett angezeigt ]] </strong>`

→ Die Auszeichnung `<strong>` markiert besonders wichtigen Text. Angeblich wird die Auszeichnung `<strong>` von Suchmaschinen entsprechend beachtet und die Phrasen werden als „wichtig“ erfasst<sup>4</sup>.

`<b> [[ beliebiger Text wird fett angezeigt ]] </b>`

→ Die Auszeichnung `<b>` hingegen ist rein optischer Natur und zeigt Text als fett an.

`<p style="font-size: 1.5em; font-color: #c20a0a;"> [[ beliebiger Text ]] </p>`

→ Fügt einen Absatz hinzu und formtiert Schriftgröße (font-size) und Textfarbe (font-color<sup>5</sup>). Der Text im Absatz hat aufgrund der „Style“-Angabe eine Größe von 1,5 im

---

<sup>4</sup> Es handelt sich um eine veraltete und bürgerliche Kategorisierung. Die einzig relevanten Kategorien der Postmoderne bleiben „witzig“ und „nicht witzig“. Anm. des Kängurus. (Das ist ein Insider-Witz für Leser und Hörer des Buches „Das Känguru-Manifest“.)

<sup>5</sup> Die Angabe „color“ wird im HTML-Text immer in der amerikanischen Schreibweise verwendet.

Verhältnis zu einem Geviert<sup>6</sup>. Die Größenangabe „em“ findest du in diesem Template sowohl in den CSS (über CSS erfährst du später mehr) als auch in den HTML-Dateien. Man könnte auch die Größenangabe „px“ (Pixel) verwenden, allerdings könnte man diese nicht zentral über die CSS verändern. Aus diesem Grund verwende ich lieber „em“.

```

```

→ Fügt ein Bild hinzu, das im Verzeichnis „/images“ liegt und den Dateinamen „Bild.png“ trägt. Das Bild wird aufgrund der „Style“-Angabe mit einer Breite von 170 Pixeln angezeigt (width). Es liegt am rechten Rand des Bereiches, in dem es sich befindet (float). Das Bild hat einen Rand von oben 0, rechts minus 5, unten 5 und links 15 Pixeln (margin).

Jetzt ein umfangreicheres Beispiel:

```
<table style="background: #ffffe0; padding: 15px 10px 10px 10px; border-left: 5px #c20a0a solid; border-right: 5px #c20a0a solid; box-shadow: 5px 5px 10px 0px #cccccc; width: 490px; ">
```

```
<tr>
```

```
<td style="vertical-align: top;">
```

```
<a href="shop.php"></a>
```

```
</td>
```

```
<td style="vertical-align: top; font-size: 1.8em; line-height: 1.3em; ">
```

```
<p>This is how you can get my individual product or offer:<br />
```

```
<a href="shop.php"><strong>Add your own link here</strong></a></p>
```

```
</td>
```

```
</tr>
```

```
</table>
```

→ Stammt aus dem Quellcode der Datei „about.html“ in diesem Template: Fügt eine Tabelle hinzu (table), in der sich links ein Bild und rechts Text mit einem Link befindet.

Die Tabelle hat einen beigefarbenen Hintergrund (table style="background: #ffffe0); innerhalb der Tabelle wird ein Rand gelassen – oben 15, rechts 10, unten 10 und links 10 Pixel (padding); es gibt einen Rahmen auf der linken Seite von 5 Pixeln Breite, im Bandologie-Rot und mit durchgezogener Linie (border-left: 5px #c20a0a solid); auf der rechten Seite ist die gleiche Rahmenlinie gezogen (border-right: 5px #c20a0a solid); es gibt einen grauen Schatten, dessen Position 5 Pixel horizontal und 5 Pixel vertikal von der Tabelle liegt, zudem ist ein Unschärferadius von 10 Pixeln definiert (box-shadow: 5px 5px 10px 0px #cccccc); die Tabelle hat eine feste Breite von 490 Pixeln (width: 490px).

<tr> fügt eine Tabellenzeile hinzu. (tr = table row = Tabellenzeile).

<td style="vertical-align: top;"> fügt eine Tabellenspalte hinzu, deren Text oben in der Tabelle beginnt (vertical-align: top). Die Standardeinstellung bei HTML wäre, dass der Text mittig in der Tabelle positioniert wäre.

<a href="shop.php"> verlinkt auf die Unterseite „shop.php“.

 fügt ein Bild hinzu, das im Verzeichnis „/images“ liegt und den Dateinamen „Ein\_Bild.png“ trägt. Das

---

<sup>6</sup> Ein Geviert ist eine fest definierte Größe innerhalb eines Schriftsatzes. Die Breite entspricht in etwa der Breite des Großbuchstaben „M“.

Bild wird mit einer Höhe von 120 Pixeln angezeigt (height) und hat einen Außenabstand von oben 0, rechts 10, unten 0 und links 0 Pixeln (margin).

`<td style="vertical-align: top; font-size: 1.8em; line-height: 1.3em; ">` fügt eine Tabellenspalte hinzu, deren Text oben beginnt, eine Textgröße von 1,8em (font-size) und einen Zeilenabstand von 1,3em (line-height) hat.

## Beispiele aus dem Webseitenbereich „<head>“:

Der <head> Bereich steht im Quellcode über dem <body> Bereich und wird demnach auch zuerst vom Browser ausgelesen. Ich habe hier den <body> Bereich zuerst mit Beispielen aufgeführt, weil das der Bereich ist, mit dem man als Webmaster üblicherweise die meiste Zeit verbringt. Der <head> Bereich dient dazu Angaben zu machen, die für die gesamte Unterseite gelten. Dieser Bereich ist später nicht für den Benutzer der Webseite sichtbar, allerdings elementar für die Funktionalität der Seite.

`<meta name="keywords" content="add, your, relevant, keywords, here" />`

→ Hier kann man beschreibende Schlagworte (Keywords; zu Deutsch: Schlüsselwörter) eingeben, die den Inhalt der Seite verdeutlichen.<sup>7</sup>

`<base href="http://www.Referenzwebseite.de" />`

→ Mit dieser Angabe wird eine Domain für alle relativen Links definiert. Eine Erklärung zu absoluten und relativen Links folgt gleich, im Beispiel 3 des nächsten Kapitels.

`<meta http-equiv="refresh" content="5; url=index.php">`

→ Ein schöner Trick: Mit dieser Angabe, die in der Datei „404.php“ in Zeile 9 vorkommt, leitet die Seite nach einer Pause von 5 Sekunden (content) auf die Unterseite „index.php“ um. Diese Funktion habe ich in das Template eingebaut, damit jemand, der eine ungültige Unterseite auf deiner Webseite aufruft, automatisch auf die Startseite geleitet wird. Die Standard-Einstellung bei HTML wäre, dass der Nutzer eine Fehlermeldung („Fehler 404“, nicht vorhandene Seite) lesen würde. Achtung: Eine solche Umleitung funktioniert nur, wenn du eine .htaccess-Datei auf einem Webspace nutzt und eine entsprechende Zeile darin vermerkt hast (eine Erklärung dazu folgt auf der nächsten Seite). Wie du gleich lernen wirst, habe ich bei dieser Umleitung einen relativen Link eingebaut. Dieser Link funktioniert nur, wenn die <base href=" korrekt ist (siehe oben).

## Schön für Nutzer und praktisch für Suchmaschinen

Ein anderes Praxisbeispiel für den Einsatz einer Umleitung ist die Seite [www.bandologie.de/gratis.html](http://www.bandologie.de/gratis.html). In diesem Fall nutze ich die Umleitung zu Marketing-Zwecken, denn „Bandologie.de/**gratis.html**“ kann man sich leicht merken. Im <head> dieser Datei habe ich die Angabe `<meta http-equiv="refresh" content="0; url=http://www.bandologie.de/index.php/gratis-infos-fuer-musiker">` vermerkt. Diese Zeile bewirkt, dass der Besucher ohne Verzögerung zu den Gratis-Dokumenten umgeleitet wird, die auf der komplizierter zu merkenden Unterseite [index.php/gratis-infos-fuer-musiker](http://www.bandologie.de/index.php/gratis-infos-fuer-musiker) liegen.

---

<sup>7</sup> Die Verwendung von Keywords und deren Wirkung ist umstritten. Ich persönlich habe gute Erfahrungen damit gemacht, Keywords zu verwenden, die in Überschriften und im Text auftauchen. Meine Webseite wird sehr häufig von der angesteuerten Zielgruppe per Suchmaschinentreffer gefunden.

Mit dieser Methode können Bandologie-Kenner alle Bandologie-Interviews nur mit Hilfe des Vornamen des jeweiligen Interviewpartners und der Endung „.html“ aufzurufen. Der Link [www.bandologie.de/kristian.html](http://www.bandologie.de/kristian.html) leitet zum Beispiel um auf [http://www.bandologie.de/doc\\_kolonko/20\\_bandologie\\_interview\\_mit\\_musikproduzent\\_kristian\\_nord.html](http://www.bandologie.de/doc_kolonko/20_bandologie_interview_mit_musikproduzent_kristian_nord.html). Wenn jemand in einer Suchmaschine „Interview + Musikproduzent + Bandologie“ oder den Namen „Kristian Nord“ eingibt, wird diese Seite von einer Suchmaschine (zumindest nach weit verbreiteter Einschätzung von Suchmaschinen-Experten) als besserer Treffer eingestuft, als wenn ich diese Datei einfach „Kristian“ oder „mein\_Interview\_2428“ genannt hätte. Mit der Umleitung ermögliche ich also sowohl Nutzern als auch Suchmaschinen bestmöglichen Zugriff.

Auf euer Webseite könntet ihr zum Beispiel mit euren jeweiligen Personennamen auf den entsprechenden Teil eurer Band-Biografie umleiten.

[www.eure-neue-webseite.de/jacob.html](http://www.eure-neue-webseite.de/jacob.html) könnte umleiten auf

[www.eure-neue-webseite.de/eure-bandbiografie.html#jacob-hoellenmeier-gitarrist](http://www.eure-neue-webseite.de/eure-bandbiografie.html#jacob-hoellenmeier-gitarrist)<sup>8</sup>

Falls du im Bereich der Webtechnik noch einen Schritt weiter gehen willst, kannst du Umleitungen nicht nur mit dem Befehl `meta http-equiv="refresh"` im `<head>` herbeiführen, sondern kannst dazu eine Zeile in einer sogenannten `.htaccess`-Datei schreiben.

## Exkurs: So funktioniert eine `.htaccess`-Datei

Mit einer sogenannten HTACCESS-Datei (auch `.htaccess`; gesprochen: *hate:l'æk.ses*) kannst du gewisse Grundeinstellungen deiner Webseite oder bei Bedarf einzelner Bereiche der Seite vornehmen. Ich, beispielsweise, erstellte das erste Mal eine `.htaccess`, als ich eine automatische Umleitung von einer nicht gefundenen Unterseite (sehr bekannt: ein „404 Fehler“) auf meine Startseite ermöglichen wollte. Wenn man jetzt eingibt [www.bandologie.de/ups-ich-habe-mich-vertippt](http://www.bandologie.de/ups-ich-habe-mich-vertippt) (oder was auch immer), wird man zum Inhaltsverzeichnis der Webseite geleitet. Wie gerade beschrieben, habe ich dieses Feature auch in diese Webseitenvorlage eingebaut.

Falls du an dieser Umleitung etwas ändern möchtest (zum Beispiel könnte man die Zeit bis zur Umleitung verkürzen) kannst du folgende Dateien und Zeilen verändern:

Datei: „`.htaccess`“, Zeile 3: „`ErrorDocument 404 /404page.php`“

Datei: „`404page.php`“, unter anderem Zeile 9

Datei: „`404page.html`“

Eine `.htaccess` kannst du entweder dadurch bearbeiten, dass du die Datei in einem HTML-Editor (beispielsweise Smultron) öffnest, während sie online auf deinem Webspaces liegt oder du könntest die Datei umbenennen in „`.htaccess`“ (ohne Punkt davor), um sie auf deiner Festplatte sehen und öffnen zu können. Alternativ dazu, für Fortgeschrittene,

---

<sup>8</sup> In diesem Link ist ein sogenannter Anker eingebaut. Das bedeutet, dass der Link nicht nur auf die Seite verweist, sondern den Nutzer direkt zum richtigen Abschnitt auf der Seite bringt. In diesem Fall heißt der Abschnitt „`jacob-hoellenmeier-gitarrist`“. Einen Anker erkennst du an dem Zeichen „`#`“. Mehr über Anker findest du bei SelfHTML. Auch in diesem Template habe ich Anker verwendet. Du findest sie im Quelltext, wenn du die Datei „`shop.html`“ nach dem Zeichen „`#`“ durchsuchst.

könntest du an deinem Computer einstellen, dass er dir sogenannte versteckte Dateien anzeigt. Dann würdest du auch die Datei „.htaccess“ auf deiner Festplatte sehen.

In diesem Template habe ich diese Datei extra mit „htaccess“ (ohne Punkt davor) benannt, damit jeder sie sehen und in einem Text-Editor öffnen kann. **Um die Datei für deine Webseite wirksam zu machen, benenne sie um in „.htaccess“ (mit Punkt davor).**

Die gerade erwähnte Möglichkeit, mit der .htaccess eine Umleitung herbeizuführen, funktioniert mit der Angabe „Redirect permanent“. Eine solche Zeile kann so aussehen:

```
Redirect permanent /jacob http://www.eure-neue-webseite.de/eure-bandbiografie.html#jacob-hoellenmeier-gitarrist
```

Wenn man diese Zeile in die .htaccess-Datei eurer Webseite schreibt würde man durch die Eingabe [www.eure-neue-webseite.de/jacob](http://www.eure-neue-webseite.de/jacob) (ohne .html) zu der besagten Unterseite kommen.

Mehr zum Thema htaccess: <http://de.selfhtml.org/servercgi/server/htaccess.htm>

## 9. So machst du aus dieser Vorlage Stück für Stück deine eigene Webseite

Falls du in diesem Kapitel der Anleitung einsteigst, dann solltest du entweder mindestens die vorherigen Seiten einmal genau gelesen und verstanden haben oder aus anderen Gründen in etwa wissen was du tust.

Ich habe HTML im Wesentlichen durch stundenlanges Herumprobieren erlernt und kann dir eine ähnliche Vorgehensweise empfehlen: Verschaffe dir einen Überblick (zum Beispiel mit Hilfe dieser Kurzanleitung und durch das Nutzen von <http://de.selfhtml.org>), dann lädst du dir einen sogenannten lokalen Server (soeben erklärt) auf deinen Rechner und fängst einfach an mit dem vorliegenden Template herumzuprobieren (Vorschläge dafür folgen gleich). Übrigens: Keine Sorge, du kannst in diesem Template nichts kaputt machen und im Zweifelsfall lädst du es einfach noch einmal herunter.

Es gibt jetzt drei praktische Tipps zur Handhabung von HTML-Dateien und dann geht es mit konkreten Beispielen aus diesem Template weiter.

**Tip:** Falls du eine Zeile im Quellcode nicht verstehst, kopiere die ganze Zeile oder Teile davon (per Copy/Paste) in eine Suchmaschine. Mit etwas Engagement findest du so zu nahezu jedem Thema im Bezug auf HTML Hilfe.

**Tip:** Um zeitgleich in allen geöffneten Dokumenten „http://localhost:8888“ durch deine eigene Domain zu ersetzen gibt es im HTML-Editor Smultron die „Erweiterte Suche“ (in der Symbolleiste, rechts neben dem Suchfenster „Livesuche“). Eine ähnliche Funktion wird in vielen HTML-Editoren verfügbar sein.

**Tip:** Mit dem FTP-Programm Cyberduck kannst du auf einem Apple-Computer mit einem Rechtsklick auf eine Datei zunächst „Bearbeiten mit“ und dann „Smultron“ anwählen. Somit kannst du Änderungen online vornehmen (die Alternative wäre, die Datei zuerst herunterzuladen, zu bearbeiten und wieder hochzuladen). Es brauchte einen Praktikanten, der mir diese Funktion zeigte, obwohl ich schon seit Jahren erfolgreicher Webmaster war. Vielleicht ist dir diese Möglichkeit gerade in diesem Moment neu. Eine ähnliche Funktion wird in vielen HTML-Editoren und auch unter Windows verfügbar sein.

Anhand von vier Beispielen möchte ich dir erklären, wie du Details dieses Templates verändern kannst.

## Beispiel 1 – Ändern einer Überschrift

Öffne die Datei „index\_home.html“ in einem Text-Bearbeitungsprogramm oder einem HTML-Editor und durchsuche die Datei (Shortcut: STRG + F) nach der Phrase „<!-- Main content start -->“. Falls du bereits einen HTML-Editor nutzt: Du findest diese Phrase in Zeile 33. Drei Zeilen darunter befindet sich die Zeile:

```
<h1>Free Website Template</h1>
```

Ändere die Worte „Free Website Template“ in die Worte „Test Überschrift eins“ und schaue dir das Ergebnis in einem Web-Browser an. Wenn du mit einem lokalen Server arbeitest und „http://localhost:8888“ als lokale Adresse nutzt, musst du dazu „http://localhost:8888/index.php“ (oder einfach nur http://localhost:8888) in deinem Browser öffnen. Die Dateien aus diesem Template müssen dafür natürlich im richtigen Ordner auf deiner Festplatte und der lokale Server gestartet sein (diese Anmerkung verstehst du, sobald du einen lokalen Server installiert und einmal erfolgreich verwendet hast). Wenn du bereits eine Domain und ein Webpaket bei einem Hoster gemietet hast, musst du die Dateien aus diesem Template per FTP auf deinen Webespace laden und rufst daraufhin deine Domain auf.

**Tip:** Um die Änderungen sichtbar zu machen, musst du möglicherweise deinen Browser aktualisieren oder den Cache-Speicher deines Browsers löschen. Wie Letzteres in deinem speziellen Browser geht, findest du mit einer Suchmaschinen-Anfrage heraus.

Wenn du diese Schritte durchführen konntest, bist du jetzt in der Lage, alle Text-Elemente dieser Webseiten-Vorlage nach deinen Wünschen zu verändern. Du musst die Elemente nur noch im Quelltext finden.

Um das zu tun, rufe dir die entsprechende Unterseite des Templates im Browser auf, beispielsweise „shop.php“ und durchsuche die zugehörige HTML-Seite (STRG + F) nach der Text-Phrase, die du ersetzen möchtest, beispielsweise „Describe your product here“. Anstelle dieses Textes schreibst du einen neuen. Auf diese Weise kannst du Stück für Stück die ganze Vorlage umbauen, bis schließlich dein gewünschter Text zu sehen ist.

**Tip:** Auch, wenn der HTML-Quelltext für dich zu Anfang unübersichtlich wirkt, wirst du ihn mit etwas Übung ähnlich schnell und sicher lesen können, wie einen normalen Fließtext.

## Beispiel 2 – Ändern eines Fotos

Öffne die Datei „index\_home.html“ in einem Text-Bearbeitungsprogramm oder einem HTML-Editor und durchsuche die Datei (Shortcut: STRG + F) nach der Phrase „your\_individual\_picture“. Du wirst die Zeile 15 finden, in der unter anderem vermerkt ist:

```

```

Öffne parallel dazu die Datei „your\_individual\_picture.png“, die sich im Ordner „images“ innerhalb dieses Templates befindet. Benenne die Datei um in „test\_bild.png“ und benenne ebenfalls die verlinkte Bild-Datei um, die in der oben zitierten Phrase aus Zeile 15 genannt ist – selbstverständlich in den exakt gleichen Namen, „test\_bild.png“.

Wenn du diese Datei öffnen und umbenennen konntest und sie nach der Aktualisierung deines Browsers noch immer (oder jetzt wieder) sichtbar ist, kannst du dir hoffentlich mit Leichtigkeit vorstellen, dass du diese Bild-Datei genauso gut durch ein Foto von dir austauschen oder mit einem Bildbearbeitungsprogramm umgestalten kannst. Auf die gleiche Weise kannst du alle eingebundenen Fotos und auf ähnliche Weise auch Videos austauschen. Bei eingebetteten Videos sind es meist mehrere Zeilen, die man von einer Video-Betreiberwebseite als „embed code“ oder unter dem Punkt „share“ (je nach Anbieter etwas anders benannt) in den eigenen Quellcode per Copy/Paste einsetzen kann.

## Beispiel 3 – Ändern eines Links

Nach zwei Beispielen, jetzt etwas versierter: Öffne die Datei „index\_home.html“ und gehe wieder zur Zeile 15 (dieses Mal ist sie hier vollständig wiedergegeben):

```
<a href="about.php"></a>
```

Die Angabe `<a href="` eröffnet innerhalb von HTML einen Link. In diesem Fall liegt der Link auf einem Bild (auf dem, das im vorherigen Beispiel Thema war). Anstelle des Bildes könnte hier auch Text verlinkt sein oder ein DIV-Bereich, wie du selbstständig beim Durchsehen der Dateien dieses Templates nachvollziehen kannst.

**Es gibt sogenannte absolute und relative Links.** Der Unterschied ist ganz einfach: Ein absoluter Link funktioniert immer (vorausgesetzt natürlich, die dahinterliegende Webseite lässt sich öffnen), denn er enthält sowohl die sogenannte Top-Level-Domain (Beispiel: `www.meine-eigene-musikerwebseite.org`) und zusätzlich gegebenenfalls ein Verzeichnis oder eine spezielle Datei auf dieser Domain, zum Beispiel `www.meine-eigene-musikerwebseite.org/shop.html`.

Ein relativer Link funktioniert nur, wenn in den `<head>` Daten die jeweils korrekte Domain angegeben ist. Man definiert eine Basis (Englisch: `base`) für alle folgenden Links. In jeder `.php` Datei in diesem Template findest du die Zeile `<base href="http://localhost:8888" />`. Damit ist es möglich, innerhalb dieser Webseite zum Beispiel einfach auf das Verzeichnis „shop.html“ zu verlinken, ohne jedes Mal die vollständige Domain anzugeben.

Das ist praktisch, wenn man zum Beispiel zuerst auf dem heimischen Rechner mit „http://localhost:8888“, arbeitet und erst später damit online geht, auf die Domain „www.meine-eigene-musikerwebseite.org“. Der Link auf „shop.html“ müsste im Quelltext nicht mehr verändert werden, sondern nur die besagte `<base href=` Angabe im `<head>` Bereich. Relative Links werden natürlich umso interessanter, desto mehr Links die Webseite enthält.<sup>9</sup>

Nun endlich zum Beispiel: Ändere den oben genannten Link `<a href="about.php">About this webpage</a>` in `<a href="contact.php">Hier kannst du mich kontaktieren</a>`. Wenn du das getan hast, hast du soeben den Link und den beschreibenden Text des Links verändert. Aktualisiere den Browser und schau dir das Ergebnis an.

## Beispiel 4 – Ändern einiger Formatierungen

Wieder die gleiche Zeile als Beispiel, dann haben wir alle Bestandteile dieser Zeile einmal behandelt:

```
<a href="about.php"></a>
```

Ändere die Angabe „style="width: 126px;“ in „style="width: 300px;“ und schau dir an, was geschieht. Das sollte selbsterklärend sein.

Beim sogenannten „padding“ (Innenabstand; bei Bedarf solche Vokabeln bitte bei SelfHTML oder in einem Online-Dictionary nachschlagen) habe ich hier einen kleinen Trick verwendet: Die erste Angabe („0“) ist immer oben, die zweite rechts, die dritte unten und die vierte ist immer die linke Seite. In diesem Fall habe ich den Abstand des Bildes unten auf 5 Pixel gesetzt. Ändere die Angabe beispielsweise in 25 Pixel oder verwende zusätzlich eine der anderen Angaben (die bisher mit „0“ angegeben sind) und schau dir an was geschieht.

Wenn du diese vier Beispiele bearbeitet hast, kannst du jetzt bereits viele Elemente des HTML-Textes in diesem Template selbstständig nach deinen Wünschen anpassen. Was dir jetzt noch fehlt sind ein paar Positionierungsangaben. Dazu findest du mehr Infos im kommenden Kapitel, „Der geheime Masterplan in der Hinterhand: CSS (Cascading Style Sheets)“.

---

<sup>9</sup> Nachdem ich einige Jahre lang absolute Links lieber mochte und mir „zur Sicherheit“ immer die Mühe gemacht habe, die vollständige Domain einzutippen, nutze ich mittlerweile „skrupellos“ auch relative Links, die mir zunächst etwas suspekt und unsicher erschienen. Keine Sorge, richtig angewendet funktionieren die genauso gut wie die absoluten Links ... und durch die Kürze gibt es weniger Fehlerquellen im HTML-Text.

# 10. Der geheime Masterplan in der Hinterhand: CSS (Cascading Style Sheets)

Formatierungen und Positionierungen, die man auf einer Webseite mehr als einmal verwenden möchte, kann man zentral lagern und dann abrufen. Das Prinzip dahinter ist das gleiche wie bei einer Einbettung (siehe Kapitel „So ist dieses Webseiten-Template aufgebaut“). Man erstellt sich sozusagen Stilvorlagen.

Im Fachjargon spricht man von stufenförmig geschichteten Gestaltungsvorlagen (Cascading Style Sheets). Besser beschreibend finde ich die Bezeichnung „zentral abgelegte Vorlagen für Schriftformatierungen und Webseitenbausteine“. Das Prinzip dahinter ist denkbar einfach: Man schreibt zunächst in einer Datei auf, wie die gewünschten Vorlagen aussehen sollen. Die Datei, in der diese Vorlagen beschrieben sind, nennt man CSS-Datei.

Die CSS-Datei dieses Templates befindet sich im Ordner „css“, die Datei heißt „main.css“.

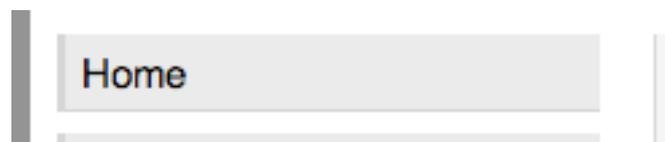
Innerhalb dieser Datei vergibt man für jede Stilvorlage einen Namen; man spricht in diesem Zusammenhang von einer ID. Mit diesem Namen, dieser ID, kann man die Vorlage daraufhin auf der ganzen Webseite immer wieder verwenden.

Im Folgenden zwei Beispiele mit Erklärungen zum Verständnis:

## Beispiel 1

In der Datei `main.css` steht geschrieben:

```
#navi_element_left {  
  width: 193px;  
  padding: 6px 0px 4px 6px;  
  border-bottom: solid 1px #cccccc;  
  border-left: solid 3px #cccccc;  
  color: #000000;  
  background: #e6e6e6;  
  margin: 0 0 8px 0;  
  text-decoration: none;  
  vertical-align: middle;  
}
```



**Abb. 2** Die Abbildung zeigt in der Mitte das Navigations-Element mit der ID „navi\_element\_left“ (den grauen Kasten mit der Aufschrift „Home“).

Zusammenfassend: Es handelt sich um die Formatierungen für ein sogenanntes „Block-Element, speziell um ein sogenanntes „div“. Ein „div“ ist nichts weiter als ein Bereich einer Webseite. In diesem Fall dient das div-Element als Navigations-Element, sozusagen als Knopf in der Navigation (siehe Abbildung).

Dem Element habe ich den Namen „`navi_element_left`“ gegeben. Es könnte auch einen beliebigen, anderen Namen, eine andere ID, tragen.

Aufgrund der oben geschriebenen Angaben (siehe rot markierte Zeilen), erhält das Navigations-Element folgende Eigenschaften: Eine feste Breite von 193 Pixeln (`width`); Innenabstand oben 6, rechts 0, unten 4 und links 6 Pixel (`padding`); einen Rahmen an der Unterseite (`border-bottom`), der durchgezogen sowie 1 Pixel breit (`solid 1px`) und grau ist (Farbcode: `#cccccc`); einen Rahmen auf der linken Seite (`border-left`), der durchgezogen sowie 3 Pixel breit (`solid 3px`) und grau ist (Farbcode: `#cccccc`); Textfarbe schwarz (`color: #000000`); Hintergrund hellgrau (`background: #e6e6e6`); Außenabstand oben 0, rechts 0, unten 8 Pixel und links 0 Pixel (`margin`); Text in normaler Formatierung, nicht fett, nicht kursiv etc. (`text-decoration: none`); Textausrichtung vertikal mittig (`vertical-align: middle`).

Somit ist schonmal definiert, wie ein Navigations-Element aussehen soll. Um ein solches Element auf der Webseite anzuzeigen, kann man es jetzt mit dem Stichwort „`navi_element_left`“ aufrufen. Im Fachjargon würde man sagen: „Das div-Element lässt sich jetzt mit der zugeordneten div-ID ansteuern“. Und das geht beispielsweise so (Auszug aus der Datei „`navigation.html`“):

```
<a href="/index.php">  
<div id="navi_element_left">Home</div>  
</a>
```

Diese Zeilen bewirken im Detail folgendes: Zunächst wird ein Link eröffnet, der auf die Datei „`index.php`“ verweist (`<a href="/index.php">`). Als Fläche für den Link wird in diesem Fall kein Text und kein Bild, sondern das besagte div-Element verwendet (`<div id="navi_element_left">`). Dieser Navigationsknopf ist beschriftet mit dem Wort „Home“. Das div-Element wird geschlossen (`</div>`). Der Link wird geschlossen (`</a>`).

Ein schöner Trick soweit. Das Ganze funktioniert allerdings nur, wenn im Text vorher erklärt wurde, wo sich die CSS-Datei befindet. Diese Angabe steht im `<head>` Bereich jeder PHP-Datei dieses Templates und lautet:

```
<link rel="stylesheet" href="http://localhost:8888/css/main.css" type="text/css" />
```

Somit weiß die Unterseite, wo die CSS-Datei liegt, die genutzt werden soll. Fertig.

Die Ganze Sache funktioniert in diesem Fall also folgendermaßen: Die CSS-Datei definiert die Eigenschaften des gewünschten div-Elements. Du kannst dir vorstellen, dort wäre eine Art kleiner Bauplan hinterlegt. Das Element trägt den von mir gewählten Namen, die ID „`navi_element_left`“. In der Datei „`navigation.html`“ wird dieser Name mehrfach aufgerufen und jeweils mit einem anderen Link versehen. Es entsteht eine Navigationsleiste (leicht nachzuvollziehen, mit einem kurzen Blick in die Datei „`navigation.html`“). Jede PHP-Datei in diesem Template greift unter anderem auf den Inhalt der Datei „`navigation.html`“ zu, weil diese Datei dort jeweils per „`include`“ eingebunden ist (leicht nachzuvollziehen beim Blick in eine beliebige Datei mit der Endung „`.php`“ aus diesem Template). Dadurch enthält jede Unterseite die Navigationsleiste. Die Folge: Sobald ein Detail, das sich auf die Navigation

bezieht, innerhalb der Datei „navigation.html“ oder in der Datei „main.css“ verändert wird, wirkt sich die Veränderung auf alle Unterseiten aus. Beispielsweise wäre es somit mit jeweils einem Handgriff möglich, den Navigationspunkt „Home“ in „Startseite“ umzubenennen oder alle Navigationspunkte orange zu färben.

**Auf die gleiche Weise wie die Datei navigation.html funktionieren die Dateien header.html, banner.html und footer.html.**

Mehr zum Thema div-Elemente gibt es hier:  
<http://de.selfhtml.org/html/text/bereiche.htm>

## Beispiel 2

In der Datei [main.css](#) steht geschrieben:

```
h1
{
  font-size: 2.0em;
  line-height: 1.2em;
  margin: 18px 0 10px 0;
}
```

Zusammenfassend: h1 (und folgende Nummern, h2, h3 etc.) sind „Headlines“, also Überschriften. Für die Überschrift mit dem Namen „h1“ sind in den roten Zeilen folgende Eigenschaften definiert: Schriftgröße von 2,0 im Verhältnis zu einem Geviert (font-size: 2.0em); Zeilenabstand im Verhältnis 1,2 zu einem Geviert (line-height: 1.2em); Außenabstand oben 18, rechts 0, unten 10 und links 0 Pixel (margin: 18px 0 10px 0). Es wäre problemlos möglich, weitere Eigenschaften, beispielsweise eine andere Textfarbe oder eine andere Schriftart, zu definieren.

Unter anderem in der Datei „index\_home.html“ wird auf diese Angaben aus der CSS-Datei Bezug genommen ... und zwar so:

```
<h1>Free Website Template</h1>
```

Wie auch im vorherigen Beispiel funktioniert das Ganze allerdings nur, wenn im Quelltext vorher erklärt wurde, wo sich die CSS-Datei befindet. In diesem Fall greift die Angabe aus der Datei „index.php“, in der vermerkt ist:

```
<link rel="stylesheet" href="http://localhost:8888/css/main.css" type="text/css" />
```

Der Zusammenhang läuft also folgendermaßen: In der CSS-Datei wird definiert, welche Eigenschaften die Überschrift „h1“ haben soll. In der Datei „index.php“ ist vermerkt, wo die CSS-Datei liegt (wie gerade beschrieben). Die Datei „index.php“ zeigt unter anderem den Inhalt der Datei „index\_home.html“, weil sie dort per „include“ eingebunden ist (siehe Abbildung auf Seite 6). Unter anderem in der Datei „index\_home.html“ wird die besagte Überschrift h1 aufgerufen. Fertig.

**Tipp:** Die Zusammenhänge erscheinen als Beschreibung wahrscheinlich komplexer als sie sind. Wenn du dich eine Weile mit den besagten Dateien (main.css, index.php und

index\_home.html) beschäftigst und einen Blick auf das Schaubild auf Seite 6 wirfst, wird dir schnell einleuchten, wie die Sache funktioniert.

Nach diesen zwei Beispielen solltest du einen Eindruck haben, was CSS (Cascading Style Sheets) sind. Die Alternative wäre übrigens, dass man alle Formatierungen und Positionierungen direkt im HTML-Text (dann in geringfügig anderer Form) niederschreibt. Das Praktische an CSS ist also, dass gewissermaßen „Baupläne“ für wiederkehrende Elemente zentral gelagert sind.

## 11. Deine Stimme und Ausdrucksmöglichkeit im Internet

Das war es mit dieser Gratis-Anleitung zum Gratis-Webseiten-Template. Da ich meine Zielgruppe, Musiker, immer sehr direkt anspreche, wollte ich es ja zunächst nicht zu laut sagen, doch mit diesem Template kannst du selbstverständlich alle möglichen Webseiten und nicht nur Musikerwebseiten erstellen. Wie bereits beschrieben kannst du damit von mir aus so viel Geld mit den Webseiten verdienen, wie du willst. Was ich hier erklärt habe, ist frei zugänglicher Code der Sprache HTML. Ich hoffe sehr, möglichst vielen Leuten mit dieser Anleitung die Möglichkeit zu geben, sich im Internet sinnvoll und unabhängig von Foren, Facebook und YouTube-Kommentarfeldern äußern zu können.

Zudem hoffe ich mit diesem Template eine Vorlage für funktionierende und verkaufsstarke Webseiten zu liefern und so manche Band davor zu bewahren, die typischen Künstler-Fehler (Shop vergessen, Bandologo viel zu groß aufgezogen etc.) zu bewahren. In diesem Sinne: **Viel Spaß mit deiner neuen, eigenen Webseite.**

Alles Gute und schöne Grüße, euer



*Nils Kolonko*  
Nils Kolonko

[www.bandologie.de](http://www.bandologie.de)

## 12. Weiterführende Links

<http://smultron.softonic.de/mac>

→ HTML-Editor für Apple-Computer (Mit einem HTML-Editor kann man auch beispielsweise PHP- und CSS-Dateien bearbeiten.)

<http://aktuell.de.selfhtml.org/links/html-editoren.htm#windows>

→ HTML-Editoren für Windows

<http://de.selfhtml.org/html/>

[http://de.selfhtml.org/css/eigenschaften/schrift.htm#text\\_shadow](http://de.selfhtml.org/css/eigenschaften/schrift.htm#text_shadow)

<http://de.selfhtml.org/html/tabellen/aufbau.htm>

[http://html-color-codes.info/webfarben\\_hexcodes](http://html-color-codes.info/webfarben_hexcodes)

→ Immer gut zu gebrauchen, wenn man an einer Webseite baut.

<http://www.longtailvideo.com/players/jw-flv-player/>

→ Dieser Player ist rechts oben im Template eingebaut, zu finden in der Datei „banner.html“, ab der Zeile <!-- Music Player START -->.

<http://www.alsacreations.fr/dewplayer-en.html>

→ Kostenloser MP3-Player zum Einbau in den Webseiten-Quellcode; Achtung: Flash ist nicht kompatibel mit einigen Apple-Produkten, zum Beispiel dem iPhone und dem iPad.

<http://sandbox.thewikies.com/vfe-generator/>

→ Videos als HTML 5 und mit Fallback-Möglichkeit auf Flash einbinden; wie das Begrüßungsvideo auf Bandologie.de – kompatibel mit iPhone und iPad. ☺

Mehr Informationen für Musiker auf [www.bandologie.de](http://www.bandologie.de)