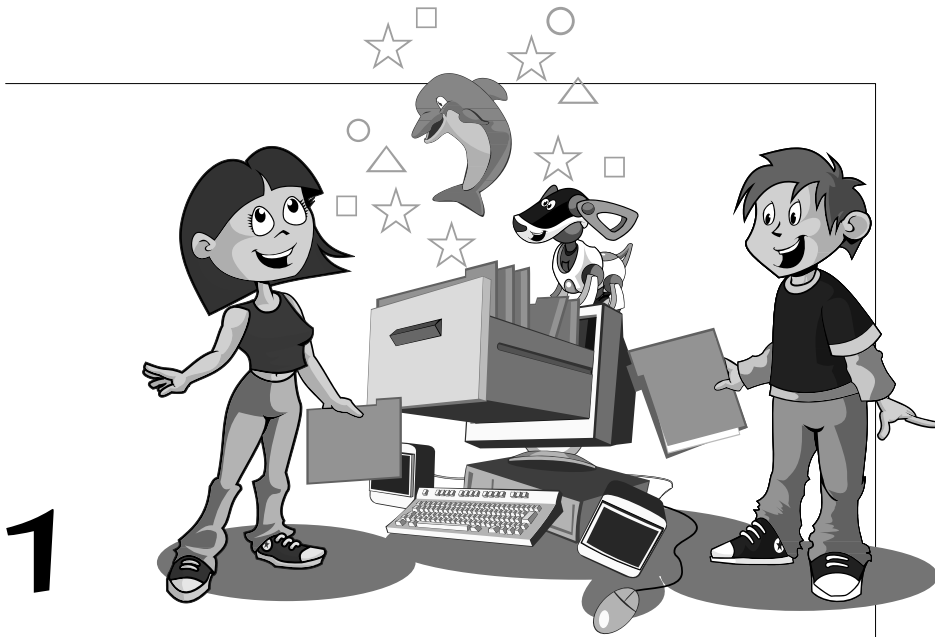




PHP und MySQL

FÜR **KIDS**





1

Deine coole Homepage mit HTML

Auf los geht's los! Wenn du mit PHP programmieren willst, musst du HTML können. HTML ist schließlich die Sprache, mit der eine Homepage geschrieben wird. Ohne HTML macht PHP keinen Sinn. Du kannst noch kein HTML? Nicht schlimm! In diesem Kapitel lernst du das Wichtigste zum Thema. Glaube mir, es ist kinderleicht. Na ja, fast ...

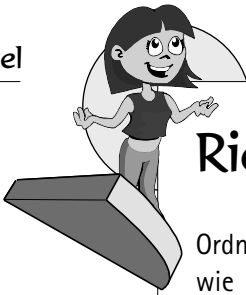
Du kannst schon HTML? Umso besser! Mache trotzdem mit, denn in diesem Kapitel zeige ich dir ein super Programm: einen sogenannten HTML- und PHP-Editor. Und ich verrate dir, wie du dein Projekt am besten organisierst.

In diesem Kapitel lernst du,

- ⊙ wie man eine HTML-Seite erstellt
- ⊙ wie man Überschriften und Absätze notiert
- ⊙ wie man Grafiken in die Seite einfügt und mit Farben arbeitet
- ⊙ wie man Querverweise setzt, die Hyperlinks
- ⊙ wie man Tabellen erstellt
- ⊙ wie man der Seite mit Style Sheets ein schickes Layout verpasst

Doch bevor es losgeht, besprechen wir gleich zu Beginn ein paar Dinge, die verdammt wichtig sind.

1



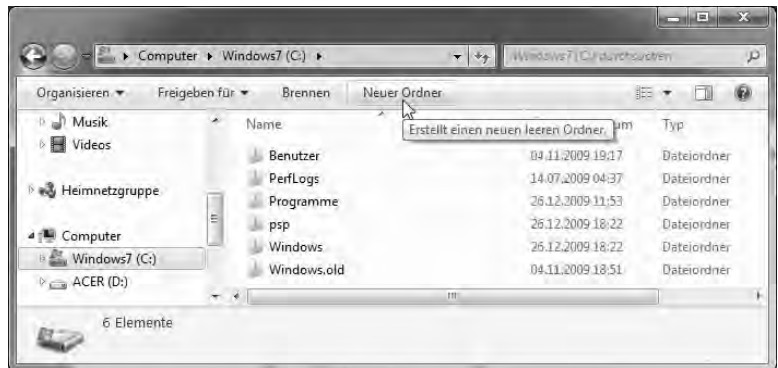
Richte dir einen Projektordner ein!

Ordnung ist das halbe Leben, geht dir dieser Spruch auch so auf die Nerven wie mir? Dabei kann ein wenig Ordnung im Nachhinein viel Zeit sparen! Und schon sind wir beim Thema Ordner. Richte dir zuerst einen Projektordner für deine Experimente ein. Diesen nennen wir `phpkid` und legen ihn direkt unter der FESTPLATTE C: ab!

Projektordner `phpkid` einrichten

Wie geht das? Ganz einfach! Hier die Schrittfolge für Windows XP bis 8:

- Starte zuerst den Windows Explorer. Die coolste Methode: Du hältst die Taste `Win` auf deiner Tastatur gedrückt. Das ist in der Regel die zweite oder dritte Taste von links in der unteren Reihe. Du erkennst sie am Windows-Logo. Jetzt tippst du mutig und unverzagt ein `E` wie »echt einfach« oder wie »Explorer«. Zack – schon startet das »Ordner-Verwaltungs-Programm« namens Windows Explorer.
- Was nun, sprach das Huhn? Achte darauf, dass dein Laufwerk C: (die Festplatte) markiert ist. Klicke also auf der linken Seite auf das Symbol für deine FESTPLATTE. Es befindet sich unterhalb des Symbols COMPUTER (Windows 7/8 bzw. Vista) bzw. ARBEITSPLATZ (Windows XP).
- Und nun richtest du deinen Ordner ein. In Windows 7/8 klickst du einfach auf die Schaltfläche NEUER ORDNER:



In Windows Vista gibt es diese praktische Schaltfläche leider noch nicht. Dort klickst du auf den Menüeintrag ORGANISIEREN und wählst den Menüpunkt NEUER ORDNER. In Windows XP ist der Weg noch etwas länger. Du klickst im Menü DATEI auf den Befehl NEU. Ein weiteres Menü klappt zur Seite. Wähle hier den Befehl ORDNER.



Richte dir einen Projektordner ein!



- Jetzt erscheint ein Platzhalter-Ordner mit der Bezeichnung *Neuer Ordner*. Dieser Platzhaltername ist markiert. Du kannst ihn also direkt überschreiben. Klicke noch nirgends, sondern tippe einfach los. Tippe den Namen des neuen Ordners, im Beispiel `phpkid`.
- Geschafft? Dann drücke einfach `Enter`! Fertig ist der neue Ordner.

Suche doch einmal deinen neuen Ordner. Du findest ihn im linken Bereich des Windows-Explorers. Er wird alphabetisch zwischen den anderen Ordnern einsortiert. Sollte dein Ordner nicht gleich zu sehen sein, ist das nicht schlimm. Der Windows Explorer ist manchmal ein »langsamer Geselle«, er vergisst das Aktualisieren der Ansicht. Hilf etwas nach, drücke auf die Funktionstaste `F5` auf deiner Tastatur.

Hoppla, du hast dich beim Ordnernamen verschrieben? Oder der Ordner wurde an der falschen Stelle eingerichtet? Kein Problem! Wenn du einen Ordner umbenennen willst, klickst du ihn kurz an. Drücke nun die Funktionstaste `F2`. Schon »öffnet« sich der Ordnername und du kannst ihn korrigieren. Bestätige die Umbenennungsaktion wieder mit `Enter`. Natürlich lässt sich so ein Ordner auch löschen. Markiere ihn und drücke die Taste `Entf` auf deiner Tastatur. Schon ist der Ordner weg!



Einen weiteren Unterordner einrichten

Weil das so gut geklappt hat, machen wir es gleich noch einmal. Erstelle diesmal einen Unterordner namens `html`. Dieser Ordner soll direkt unterhalb von `phpkid` entstehen. Diesen neuen Ordner nutzen wir für unsere ersten Gehversuche mit HTML.

- Markiere den Ordner `phpkid` im Windows Explorer. Klicke ihn also im linken Bereich an.
- Wähle nun wieder **NEUER ORDNER (ORGANISIEREN|NEUER ORDNER bzw. DATEI|NEU|ORDNER)** und richte den Unterordner `html` ein.



Gewöhne dir bei der Namensgebung für Ordner und Dateien generelle Kleinschreibung an. Die Ordner sollen also `phpkid` und `html` und nicht `Phpkid` und `Html` heißen. Das ist deshalb so wichtig, weil der Webserver (Homepage-Computer) später ganz pingelig zwischen Groß- und Kleinschreibung unterscheiden wird. Und wenn du hier etwas vermischst, wird die Homepage im Web nicht oder nicht richtig angezeigt. Bei genereller Kleinschreibung kann hier jedoch nichts »anbrennen«. Okay?





1

Und auf dem Mac?

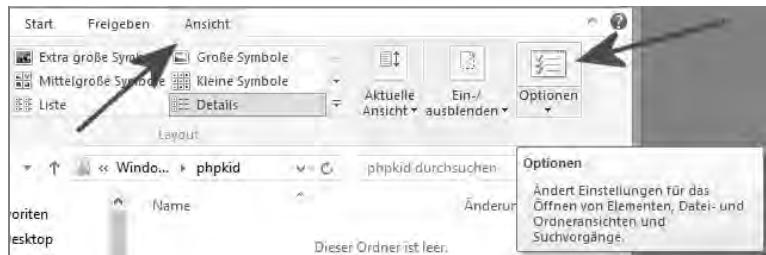
Hier richtest du Ordner über das Kontextmenü ein. Das bedeutet: Der Rechtsklick hilft dir in Zusammenarbeit mit dem Befehl NEUER ORDNER.

Bitte blende die Dateierendungen ein!

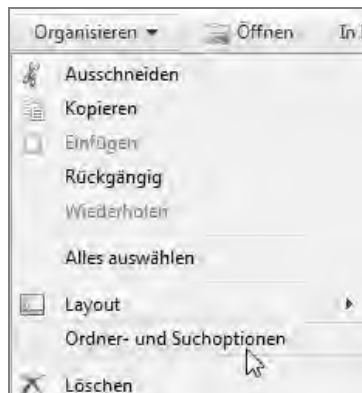
Ärgert dich auch, dass du die Dateierendungen unter Windows normalerweise nicht siehst? Jede Datei hat ja eine typische drei- bis vierstellige Endung, die nach einem Punkt an den eigentlichen Dateinamen angehängt wird. Mit `.doc` oder `.docx` kennzeichnet man Word-Dateien, `.txt` steht für Textdateien, `.html` für HTML-Dateien und `.php` für PHP-Dateien.

Von Hause aus siehst du diese Endungen nicht. Das ist sehr ärgerlich, denn wir brauchen sie! Für unseren Kurs musst du die Dateierendungen unbedingt eingeschaltet haben!

- Rufe den Windows Explorer auf. Wie ging das noch? Halte dafür beispielsweise die `[Win]`-Taste (die Taste mit dem Windows-Logo) auf deiner Tastatur gedrückt und tippe kurz ein `[E]`.
- In Windows 8 klickst du auf die Registerkarte ANSICHT und wählst ganz rechts das Symbol OPTIONEN.



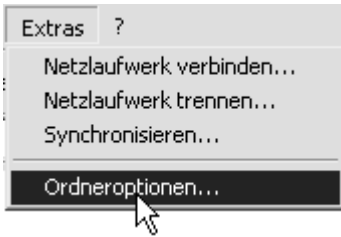
In Windows 7 und Vista wählst du im Menü ORGANISIEREN den Befehl ORDNER UND SUCHOPTIONEN.



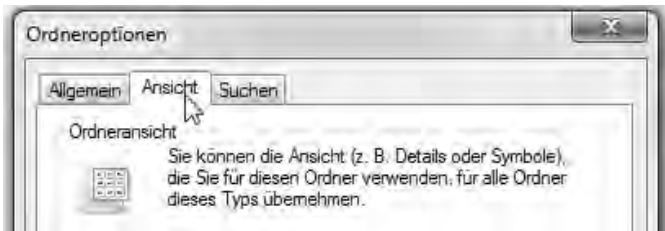
Richte dir einen Projektordner ein!



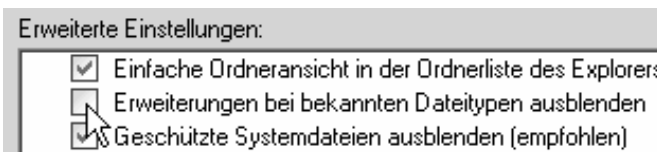
In Windows XP wählst du im Menü EXTRAS den Unterbefehl ORDNER-OPTIONEN.



- Geschafft? In allen Windows-Versionen erscheint nun das Dialogfenster *Ordneroptionen* – hier eine Bild aus Windows 7. Gehe ins Register ANSICHT, es ist das zweite Register.



- Suche nach einer Option, die je nach Windows-Version folgendermaßen heißt: *Erweiterungen* bzw. *Dateinamenerweiterungen bei bekannten Dateitypen ausblenden*. Sie ist abgehakt. Nimm das Häkchen weg!
- Bestätige deine Einstellungen durch Klick auf OK. Nun siehst du bei allen Dateinamen auch die typische Endung und weißt genau, um welchen Dateityp es sich handelt.



Nimm das entsprechende Häkchen weg, klicke es einfach an!

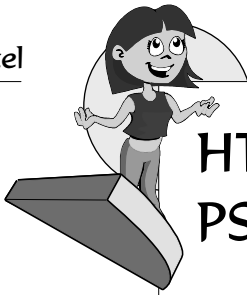
Ab sofort sind alle Dateieendungen sichtbar.

Und auf dem Mac?

Der Mac nennt die Endungen einfach Suffixe. Dateinamensuffixe, um ganz genau zu sein. Du steckst im Finder? Wähle im Menü FINDER den Befehl EINSTELLUNGEN. Hake die Option ab: *Alle Dateinamensuffixe einblenden*. Und schon erscheinen auch hier die Dateieendungen, pardon Dateinamensuffixe.



1



HTML auf Knopfdruck – mit PSPad

Vorhang auf, die Show beginnt. Ich präsentiere: PSPad, den freien Code-Editor für Windows! Es handelt sich praktisch um ein Programm zum Schreiben deiner Homepage. Neben HTML beherrscht PSPad aber auch PHP und andere Programmiersprachen. Geschrieben hat das tolle Programm Jan Fiala. Die Homepage von PSPad mit Downloadmöglichkeit findest du unter www.pspad.com. PSPad liegt aber auch auf der CD für dich bereit und diese Version werden wir jetzt gemeinsam installieren!

PSPad gibt es in zwei Varianten: In der ersten brauchst du es einfach nur zu entpacken – das System ist einsatzbereit. Die zweite funktioniert per Windows-Installationsprogramm. Welche Variante bevorzugst du?

Zip-Datei einfach nur entpacken

Ich rate dringend zur Variante 1. Mir gefällt es am besten, wenn man Dinge einfach nur zu entpacken braucht.

Warum rate ich von der zweiten Variante mit dem bequemen Windows-Installer ab? Weil sie nur Probleme bringt! Wenn du nicht aufpasst, wird sie dir eine Babylon Toolbar und weitere unnütze Tools unterjubeln:



Außerdem installiert sie PSPad im Programme-Ordner. Das ist eigentlich kein Problem, führt aber bei PSPad dazu, dass du bestimmte Einstellungen nicht speichern kannst!



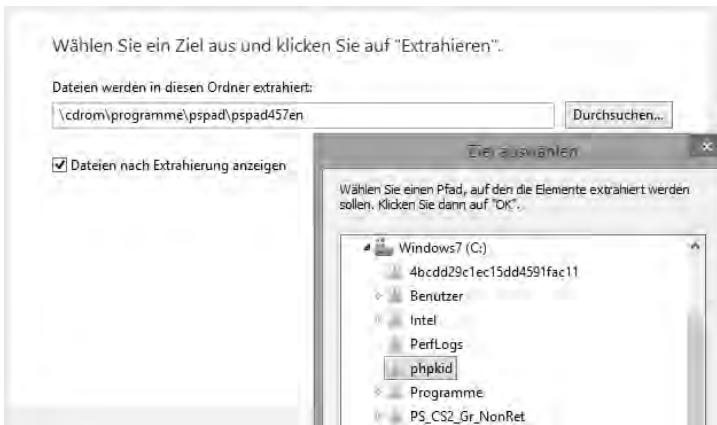


Also dann, auf zur von mir empfohlenen Installations-Variante 1:

- Starte den Windows Explorer, z.B. durch Gedrückthalten von **Win** und kurzem Tippen von **E**. Schau in den linken Bereich des Windows Explorers. Gehe zum Laufwerk für die CD und hangele dich durch bis zum Unterordner `programme/pspad`.
- Hier siehst du die Datei `pspad457en.zip`. Klicke mit der rechten Maustaste auf diese Datei. Wähle den Befehl **ALLE EXTRAHIEREN**. Der Assistent zum Extrahieren (Entpacken) erscheint.



- Lege fest, an welche Stelle du die Datei legen möchtest. Klicke auf die Schaltfläche **DURCHSUCHEN**. Wähle beispielsweise den Ordner `phpkid`, den du ja soeben unter `C:\` eingerichtet hast.



- Klicke je nach Windows-Version auf **EXTRAHIEREN** bzw. auf **WEITER**. Der Extrahiervorgang wird eingeleitet. Geduld. Dieser Vorgang kann einige Minuten dauern.

Du verwendest ein besseres und schnelleres Entpackprogramm, beispielsweise 7-Zip (www.7-zip.de)? Dann nimm es einfach! Der entsprechende Befehl im Kontextmenü lautet dann **7-ZIP/DATEIEN ENTPACKEN**.



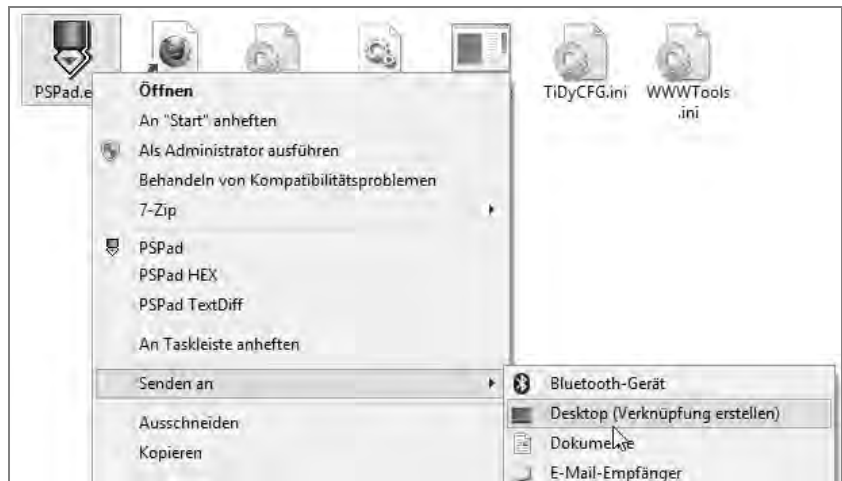
1



Nach erfolgreichem Entpacken findest du unter phpkid einen neuen Ordner namens PSPad editor. Hier befindet sich das Programm! Du startest es durch Doppelklick auf die Datei PSPad.exe.

Wenn du willst, kannst du diesen Ordner gerne an eine andere Stelle verschieben. Lege das Programm aber möglichst nicht in den Programm-Ordner von Windows (Program Files). Sonst funktioniert es ggf. nicht richtig!

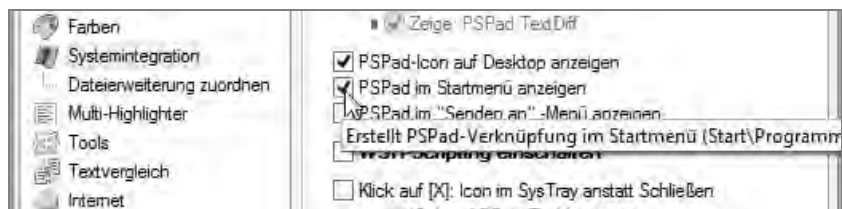
Mein Tipp: Richte dir zu dieser Datei eine Verknüpfung auf dem Desktop ein. So kannst du PSPad jederzeit bequem starten, ohne dich durch die Untiefen der Ordnerstruktur hangeln zu müssen.



So erstellst du eine Verknüpfung: Klicke mit rechts auf die Datei, wähle »Senden an/Desktop (Verknüpfung erstellen)«.



Es gibt noch einen zweiten Trick, eine derartige Verknüpfung einzurichten. Öffne PSPad und wähle EINSTELLUNGEN|PROGRAMM EINSTELLEN. Klicke auf der linken Seite auf den Eintrag *Systemintegration*. Hake rechts die Option *PSPad-Icon auf Desktop anzeigen* ab und klicke auf OK.



Wenn du möchtest, erstellt PSPad auch einen Eintrag im Startmenü für dich.



Eine HTML-Datei erstellen

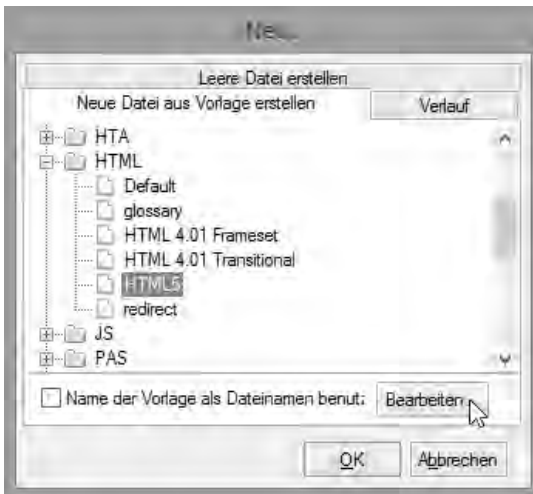
PSPad ist gestartet? Jetzt erstellst du im Handumdrehen deine erste HTML-Seite. Doch vorher sollten wir uns noch für die »richtige« HTML-Variante entscheiden. Wir nehmen natürlich HTML5, die derzeit aktuelle Version.

Hast du schon etwas Ahnung von HTML? Fragst du dich auch, welches die »richtige« HTML-Version sei? Das »klassische« und bewährte HTML 4 oder das neue HTML5? Die Antwort lautet: HTML5! Das ist die bisher einfachste und beste HTML-Variante. Wer HTML 4 kennt, wird HTML5 lieben. Die Unterschiede sind nicht groß, das »Klassische« ist geblieben, das »Komplizierte« gegangen. Fazit: Wer gerade erst einsteigt, findet mit HTML5 die bisher am leichtesten zu erlernende HTML-Version.



Und so erzeugst du in PSPad eine Musterseite in HTML5:

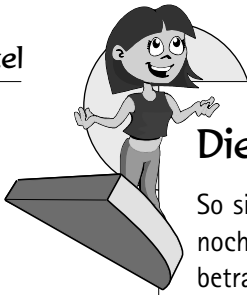
- Wähle im Menü DATEI den Befehl NEU. Das Dialogfenster *Neu* erscheint. Klicke auf die Registerzunge *Neue Datei aus Vorlage erstellen*.
- Scrolle zum Zweig *HTML* und klicke auf den Eintrag *HTML5*. Klicke danach auf *BEARBEITEN*, nicht auf *OK*. (*OK* würde eine neue HTML-Seite in den Editor einfügen, die auf diesem Grundgerüst beruht. Doch das Grundgerüst ist noch nicht ganz perfekt!)



Klicke auf die Schaltfläche »Bearbeiten«, noch nicht auf »OK«.

- Du hast auf *BEARBEITEN* geklickt? Super! Jetzt erscheint das eben erwähnte Grundgerüst. Und zwar direkt in der Vorlage! Du kannst und musst diese Grundgerüst-Vorlage jedoch wunschgemäß ändern.

1



Die HTML-Vorlage von PSPad anpassen

So sieht es aus – das HTML5-Grundgerüst. Für meinen Geschmack gibt es noch einige Schönheitsfehler. Und bevor wir die einzelnen Zeilen genauer betrachten, merzen wir diese Schönheitsfehler einfach aus.



Die von mir gezeigten Zeilennummern sind bei dir in PSPad nicht sichtbar? Wähle ANSICHT/ZEILENUMMERIERUNG.




```

1. <!DOCTYPE html>
2. <html lang='cs'>
3.   <head>
4.     <title></title>
5.     <meta charset='utf-8'>
6.     <meta name='description' content=''>
7.     <meta name='keywords' content=''>
8.     <meta name='author' content=''>
9.     <meta name='robots' content='all'>
10.    <!-- <meta http-equiv='X-UA-Compatible'
        content='IE=edge'> -->
11.    <link href='/favicon.png' rel='shortcut icon'
        type='image/png'>
12.  </head>
13.  <body>
14.
15.  </body>
16. </html>

```

Soviel vorweg: HTML besteht aus Text, der durch sogenannte Tags gesteuert wird. Jedes Tag steht in spitzen Klammern. Es gibt in der Regel ein Tag zum Einschalten und eins zum Abschalten. Das Abschalt-Tag bekommt zusätzlich einen Slash (/) vorangestellt.



Die Taste mit den spitzen Klammern findest du links unten auf der Tastatur. Drücke  für die öffnende spitze Klammer. Für die schließende Klammer  musst du zusätzlich die -Taste gedrückt halten.

Die Dinge, die mich stören, habe ich unterstrichen. Es sind ziemlich viele. In Zeile 2 entfernst du `lang='cs'`. (Es sei denn, du möchtest deine Webseiten auf Tschechisch erstellen. Unser Programmator Jan stammt aus Tschechien, das ist der Grund für `cs`.) Nach der Aktion steht in Zeile 2 nur noch `<html>`.

© des Titels »PHP und MySQL für Kids« (ISBN 978-3-8266-9465-3)

2013 by Verlagsgruppe Hüthig Jehle Rehm GmbH, Heidelberg.

Nähere Informationen unter: <http://www.mitp.de/9465>



Bleiben kann dann alles bis einschließlich Zeile 5. Zeile 5 kümmert sich übrigens um den sogenannten Zeichensatz (charset) für unsere HTML-Seite. UTF-8 ist der Universalzeichensatz, der die Zeichen der meisten Sprachen enthält. Damit decken wir praktisch alle wichtigen Sprachen der Erde ab, beispielsweise Englisch, Deutsch, Französisch, Spanisch, Russisch und sogar Chinesisch!

Ich tausche lediglich die Gänsefüßchen aus. Statt `charset='utf-8'` schreibe ich `charset="utf-8"`. Das ist Gewohnheit und gefällt mir persönlich besser. (Du könntest die Gänsefüßchen übrigens auch weglassen. Auch das ist in HTML5 neuerdings erlaubt.)

Zeile 6 bis 11 können entfallen. Nicht dass solche Dinge wie Beschreibung (description), Schlüsselwörter (keywords), Autor (author) oder Suchmaschinenroboter (robots) unwichtig wären. Doch für unsere Beispiele brauchen wir sie nicht.

Zeile 4 `<title></title>` dagegen ist sehr wichtig: Sie legt den Seitentitel fest. Diese Zeile möchte ich deshalb gerne etwas aufpeppen. Ich trage zwischen den Tags `<title>` und `</title>` die Zeichenfolge ein: *Hier Titel eintragen*. Damit du nie vergisst, den wichtigen Seitentitel einzutragen!

Das perfekte Basis-HTML5-Grundgerüst

Und so sieht das perfekte Basis-HTML5-Grundgerüst für unseren Kurs aus:

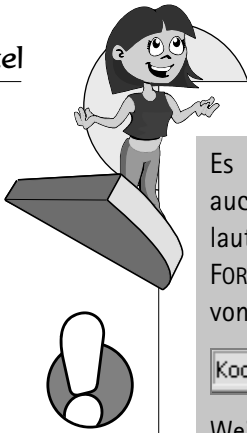
```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Hier Titel eintragen</title>
5     <meta charset="utf-8">
6   </head>
7   <body>
8
9   </body>
10 </html>
```

Die Einrückung führt dazu, dass es übersichtlicher aussieht. Das gelingt durch Leerzeichen.

Nicht vergessen: Die Zeilennummerierung schaltest du über ANSICHT/ZEILEN-NUMMERIERUNG ein. Merke es dir gut, wir werden diese Zeilennummerierung später beim Programmieren noch brauchen.

Du findest dieses Grundgerüst auch auf der Buch-CD. Schau in den Ordner `beispiele\kapitel101` und öffne die Datei `html-grundgeruest.txt`.

1



Es reicht nicht, in Zeile 5 `charset="utf-8"` zu schreiben, du musst auch in PSPad das Format UTF-8 einstellen. Nur dann werden deine Umlaute und Sonderzeichen richtig dargestellt! Wähle dafür im Menü **FORMAT** den Befehl **UTF-8** ist. Prüfe es nach: Schau in die Statuszeile von PSPad, also in die untere Zeile. Hier muss stehen: *Kodierung: UTF-8*.

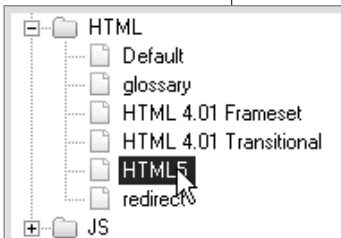
Kodierung: UTF-8

Weise PSPad zusätzlich an, neue Seiten immer in dieser eingestellten Kodierung zu öffnen. Wähle **EINSTELLUNGEN|PROGRAMM EINSTELLEN** und klicke links auf *Dateien und Verzeichnisse*. Schau zum Feld *Standard-CodePage für Dateioffnung*. Es muss so aussehen wie im Bild:

Format für neue Datei: Standard-CodePage für Dateioffnung
 DOS \Wie im Menü "Format" eingestellt

Und nun brauchst du diese aktualisierte Vorlage nur noch zu speichern. Wähle **DATEI|SPEICHERN**. Vorlagen legt PSPad automatisch im Ordner *Template* ab. Im Beispiel findest du die bearbeitete Datei unter dem Pfad `C:\phpkid\PSPad editor\Template\HTML5.html`.

Eine HTML-Datei erstellen



Voilà – jetzt endlich erzeugst du eine HTML-Seite. Wähle wieder **DATEI|NEU**, Register *Neue Datei aus Vorlage erstellen*. Navigiere erneut zum Zweig *HTML* und ... diesmal doppelklicke gleich auf *HTML5*. (Markieren und Klick auf **OK** geht auch.) Erneut entsteht das oben gezeigte, ideale Grundgerüst einer HTML5-Datei. Allerdings nicht als Vorlage, sondern jetzt als HTML-Datei. Und die schauen wir uns etwas genauer an!

Hinter Zeile 1 steckt die sogenannte *Dokumenttypdeklaration* (kurz DTD). Damit zeigen wir, dass es sich um HTML in der Version 5 handelt. Diese Dokumenttypdeklaration ist in HTML5 kurz und schmerzlos:

```
<!DOCTYPE html>
```

Der eigentliche Beginn steckt in Zeile 2, im Tag `<html>`. Hier geht es los. Wie die meisten Tags tritt auch dieses Tag paarweise auf: Auf das Einschalt-Tag `<html>` folgt ein Ausschalt-Tag. In Zeile 10 wird das Tag per `</html>` wieder abgeschaltet!

Als nächstes folgt der Kopf, der »Head«. Dieser steckt »zwischen« den Tags `<head></head>`, also zwischen Zeile 3 und Zeile 6. Wichtig ist der Titel (`<title></title>`). Was du zwischen diesen Tags notierst, erscheint



später in der Titelzeile bzw. im Tab des Browsers. (Wo genau, hängt vom verwendeten Browsertyp ab.)

- Trage hier doch einmal etwas Sinnvolles ein. Ersetze *Hier Titel eintragen* einfach durch den Text *Homepage von Computerhund Buffi!*

Was bedeutet Zeile 5? Mit diesem sogenannten Meta-Tag lege ich den Zeichensatz fest. Mit `charset="utf-8"` Sorge ich dafür, dass du mit Umlauten wie ä, ö, ü und dem ß arbeiten kannst. Ein Ausschalt-Tag ist in diesem Fall ausnahmsweise nicht nötig. Meta-Tags treten nicht paarweise auf.

Früher wurden Umlaute und Sonderzeichen wie das ß umschrieben. Statt ä schrieb man z. B. ä und aus ö wurde ö usw. usf. Das ß wurde durch ß ersetzt. Dank utf-8 ersparen wir uns diese Qual und ermöglichen den Einsatz praktisch beliebiger Zeichen. Deine Homepage ist damit gleich gut gerüstet für fast alle Sprachen der Welt!



In Zeile 7 wird der »Körper« des Dokuments eingeschaltet, der sogenannte »Body«. Zeile 9 schaltet den Body wieder ab.

Speichern nicht vergessen!

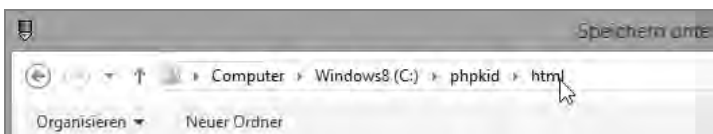
Vergiss nicht, dein Projekt zu speichern. HTML-Dokumente sind Textdokumente. Sie werden mit der Endung `htm` bzw. `html` versehen. Ich schlage den Namen `index.html` vor. Warum? Eine Hauptseite wird in aller Regel `index.html` genannt!

Und so speicherst du in PSPad:

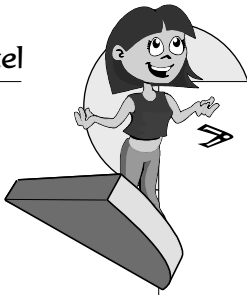
- Wähle im Menü DATEI den SPEICHERN-Befehl (oder tippe die Tastenkombination `[Strg] + [S]`, das geht viel fixer!). Das Dialogfenster zum Speichern erscheint, der Fokus steht auf dem Feld *Dateiname*.
- Tippe den Dateinamen ein, im Beispiel `index.html`.

Dateiname:	<input type="text" value="index.html"/>
Dateityp:	<input type="text" value="HTML Document (*.html;*.htm;*.shtml;*.hhc)"/>

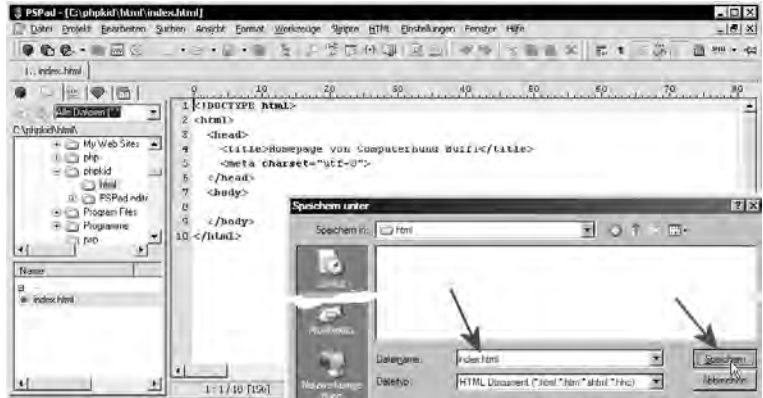
- Navigiere zu dem Ordner, in dem du speichern möchtest. Im Beispiel gehst du zu `C:\phpkid\html` – im Feld mit dem Speicherpfad muss am Ende `html` stehen.



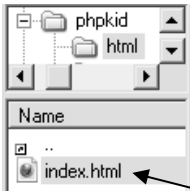
1



Klicke auf **SPEICHERN**, um die neue HTML-Datei zu sichern.



In Windows XP sehen die Dialoge noch etwas anders aus. Achte aber auch hier darauf, die Datei im richtigen Ordner abzulegen!



➤ Schau in den linken Bereich, zum Toolfenster mit dem Mini-Explorer. Diesen Mini-Explorer erreichst du über das zweite Register. Navigiere hier zu dem Ordner, in dem du gespeichert hast. Im Beispiel gehst du zu C:\phpkid\html und klickst den Ordner html an.

➤ Im unteren Bereich des Toolfensters findest du die Dateiliste. Hier sollte die index.html als Symbol zu sehen sein. Hat es geklappt?



Du kannst dieses pffiffige Toolfenster auch ausblenden. Wähle **ANSICHT|TOOLFENSTER** oder die Tastenkombination **Strg + F2**. Eine erneute Wahl dieses Befehls blendet das Toolfenster wieder ein.

So wird's perfekt: Überschriften, Absätze und mehr

Und, wie schaut's aus? Bewundere deine Seite doch einmal in der Vorschau! Dafür bietet dir PSPad folgende zwei Möglichkeiten.

Öffnen im internen Browser

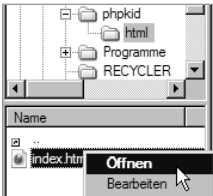
Das Öffnen im internen Browser ist ganz einfach. Drücke die Funktionstaste **F10**, klicke das »Weltkugel-Symbol« an oder wähle **HTML|VORSCHAU IM INTERNEN BROWSER**. Schon öffnet sich ein neues Fenster und zeigt deine Seite.



Noch besser jedoch ist die externe Browservorschau. Dabei öffnet sich der Browser in einem eigenen Fenster und du hast den vollen Überblick. Nur hier wird auch der ungeheuer wichtige Seitentitel dargestellt, also der Text zwischen den Tags `<title></title>`. Und das zeige ich dir jetzt:

Öffnen im externen Browser

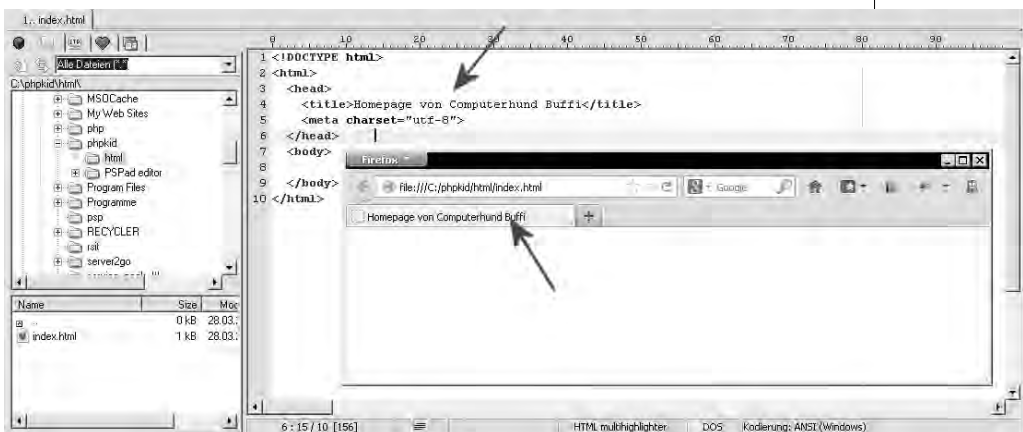
Schau in den linken unteren Bereich, zum Toolfenster. Wähle oben das zweite Register, das mit dem Ordnersymbol. Achte darauf, dass der Ordner `html` unter `phpkid` markiert ist. Schau dann etwas tiefer – und zwar dorthin, wo die einzelnen Dateien des markierten Ordners angezeigt werden.



Klicke mit der rechten Maustaste auf die zu öffnende Datei, rechtsklicke im Beispiel also auf `index.html`.

Das Kontextmenü erscheint. Wähle ganz oben den Befehl **ÖFFNEN**. Die HTML-Seite wird im Standardbrowser geöffnet, in aller Regel ist das der Internet Explorer, bei mir jedoch der Firefox.

Du möchtest einen anderen Browser wählen? Dann verwende den Kontextmenü-Befehl **ÖFFNEN MIT** und suche den anderen Browser heraus. Der Browser ist nicht aufgelistet? Suche ihn aus dem Dateisystem deiner Festplatte heraus. Das gelingt beispielsweise über die Schaltfläche **DURCHSUCHEN** bzw. **PROGRAMM AUSWÄHLEN**. Mozilla Firefox liegt unter dem Pfad `C:\Programme\Mozilla Firefox`. Die Startdatei heißt `firefox.exe`.



Die bisher noch leere HTML-Seite in der externen Vorschau. Als Browser dient der Firefox.

1



Zugegeben – die Seite ist leer, fast leer. Nur der Seitentitel von Zeile 4 steht schon da. Wo? Na oben im Tab des Browsers. Schau hin: Dort steht nun ebenfalls *Homepage von Computerhund Buffi*.



Du wunderst dich, warum bis jetzt noch kein weiterer Inhalt im Dokument zu sehen ist? Du hattest doch noch nichts hineingeschrieben! Der Bereich zwischen `<body></body>` ist schließlich noch leer! Nur der Seitentitel ist schon vorhanden und der wird ja auch angezeigt!

Leben einhauchen: Überschriften und Absatz

➤ Hauche deinem Dokument Leben ein: Klicke dazu in die bisher noch leere Zeile. Es ist die Zeile 8. Wir wagen uns nun an den Body. Schaffe ruhig durch mehrmaligen Druck auf `Enter` ein paar Zeilen Platz.

Auch der »Buch-Hund« Buffi wollte es sich nicht nehmen lassen, endlich mit einer eigenen Homepage aufzuwarten. Bitte sehr – in diesem Kapitel darf er endlich aktiv werden.

Orientiere dich an Buffis Beispiel. Schreibe es ab. Ich empfehle dir, die HTML-Befehle erst einmal exakt so aus dem Beispiel zu übernehmen. (Nur den Text kannst du so ändern, dass er auch auf dich zutrifft!)

```
<h1>Homepage von Buffi</h1>
<p>Guten Tag! Mein Name ist Buffi! Ich bin von Beruf
<b>Computerhund</b> mit <i>allumfassender
Allgemeinbildung</i>. Meine Freunde heißen Tim und
Nele.</p>
<h2>Ein Bild von meinen Vorfahren</h2>
```

```

0 10 20 30 40 50 60
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Homepage von Computerhund Buffi</title>
5     <meta charset="utf-8">
6   </head>
7   <body>
8     <h1>Homepage von Buffi</h1>
9     <p>Guten Tag! Mein Name ist Buffi! Ich bin von Beruf
10    <b>Computerhund</b> mit <i>allumfassender Allgemeinbildung</i>.
11    Meine Freunde heißen Tim und Nele.</p>
12    <h2>Ein Bild von meinen Vorfahren</h2>
13  </body>
14 </html>

```

So wird der Code in PSPad dargestellt. Du kannst ruhig immer wieder `Enter` drücken und zusätzliche Leerzeichen tippen, um den Code ordentlich auszurichten. Auf die Darstellung hat das keinen Einfluss!

© des Titels »PHP und MySQL für Kids« (ISBN 978-3-8266-9465-3)

2013 by Verlagsgruppe Hüthig Jehle Rehm GmbH, Heidelberg.

Nähere Informationen unter: <http://www.mitp.de/9465>



Was haben wir da gemacht? Wir fangen mit einer Überschrift an! Mit dem Tag-Paar `<h1></h1>` kennzeichnest du die *heading 1*, die Hauptüberschrift. Hier schmettert dir Buffi sein *Homepage von Buffi* entgegen.



Die Homepage beginnt mit einer Hauptüberschrift.

Im Text gibt es gleich noch eine weitere Überschrift, und zwar *Ein Bild von meinen Vorfahren*! Es ist eine *heading 2*, eine Unterüberschrift auf »zweiter Stufe«. Diese wird stets vom Tag-Paar `<h2></h2>` eingerahmt. Insgesamt gibt es übrigens sechs Überschriftsebenen, du könntest also auch eine `<h3></h3>`, `<h4></h4>` usw. einsetzen.

Doch ein HTML-Dokument besteht nicht nur aus Überschriften. Was ist mit dem Textabsatz mittendrin? Auf hierfür gibt es das passende Tag-Paar. Textpassagen »wickelst« du einfach mit dem Tag-Paar `<p></p>` ein. Das *p* steht dabei für *paragraph*, zu Deutsch Absatz.

Wenn du möchtest, kannst du den Text innerhalb von Tags durch das sogenannte Stil-Attribut `style` (Attribut = Eigenschaft) gestalten. Deine Überschriften und Absätze stehen normalerweise bündig am linken Rand, also linksbündig. Deine Hauptüberschrift soll außer der Reihe zentriert (mittig) ausgerichtet werden? Ergänze im Einschalt-Tag `<h1>` einfach das Attribut-Werte-Paar `style="text-align: center;"`. Und zwar nach einem Leerzeichen. Hinter diesem Stil-Attribut `style` verbirgt sich die Gestaltungssprache CSS. Die Angabe `text-align` bezieht sich auf die Textausrichtung und `center` steht für zentriert. Das sieht so aus:

```
<h1 style="text-align: center;">Homepage von Buffi</h1>
```

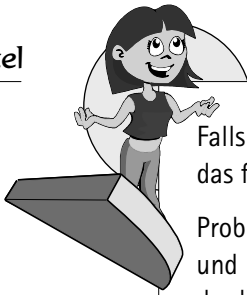
Homepage von Buffi

Guten Tag! Mein Name ist Buffi! Ich bin von Beruf Computerhund mit allumfassender

Auf diese Weise stellst du die Überschrift einfach in die Mitte!



1



Falls du die Überschrift dagegen nach rechts setzen möchtest, notierst du das folgendermaßen: `<h1 style="text-align: right;">`

Probiere es ruhig einmal aus! Es gelingt mit Überschriften und Absätzen und weiteren sogenannten Blockelementen. Genug probiert? Dann kannst du diesen Zusatz auch wieder löschen. (Ich verzichte im Beispiel auf diesen Zusatz!) Ich empfehle dir dringend, CSS zu lernen, damit du die wichtigsten Attribute und Werte zum Gestalten kennst!

Tags zur Hervorhebung von Zeichen

Sicher gibst du dich nicht mit blankem Text zufrieden. Das wäre doch langweilig! Schau einmal in den Absatz. Auch hier findest du schon weitere interessante Tags vor. Mit `` (b wie bold = fett) kannst du Wörter oder Wortgruppen hervorheben. Sie werden dann fett angezeigt. Das Tag-Paar `<i></i>` (wie *italic* = kursiv) dagegen sorgt für eine Kursivstellung der entsprechenden Passage.



Die Tag-Paare `` und `<i></i>` galten bis vor kurzem als veraltet. Als Alternative sollte man zu `` und `` greifen. Die gute Nachricht: `` und `<i></i>` leben in HTML5 weiter. Es gibt also keinen Grund, auf diese Tags zu verzichten!

Die wichtigsten HTML-Tags im Überblick

In dieser Tabelle zeige ich dir die wichtigsten HTML-Tags und ihre Bedeutung im Überblick:

Tag-Paar	Bedeutung	Wie sieht's aus?
<code><h1></h1></code>	Heading 1 – Überschrift 1	fett, sehr groß
<code><h2></h2></code>	Überschrift 2	fett, groß
<code><h3></h3></code>	Überschrift 3	fett, mittelgroß
<code><h4></h4></code>	Überschrift 4	fett, normal
<code><h5></h5></code>	Überschrift 5	fett, klein
<code><h6></h6></code>	Überschrift 6	fett, sehr klein
<code></code>	bold – fett	Text wird fett dargestellt
<code></code>	strong – stark	wird ebenfalls fett dargestellt
<code><i></i></code>	italics – kursiv	Text wird kursiv dargestellt
<code></code>	emphatic – hervorgehoben	Text wird ebenfalls kursiv dargestellt
<code><blockquote></blockquote></code>	langes Zitat	Textblock wird eingerückt
<code><small></small></code>	small – klein	Text wird verkleinert

Hier bin ich: Ein Bild einfügen



Wenn du eine Linie setzen möchtest, verwendest du das Tag `<hr>`. Dieses Tag braucht ausnahmsweise kein End-Tag.

Außerdem gibt es den sogenannten *break*, einen Zeilenumbruch. Wenn du ein Wort auf eine neue Zeile schieben möchtest, setzt du davor einfach das Tag `
`. Damit rutscht das Wort auf die neue Zeile, ohne dass gleich ein neuer Absatz beginnt. Auch `
` benötigt kein End-Tag.

An dieser Stelle gleich ein interessantes Detail: Die Schreibweise `<hr>` (für eine Linie) bzw. `
` (für den *break*, Zeilenumbruch) ist die klassische Schreibweise. Sie gilt weiterhin in HTML5. Du findest auf vielen Seiten auch die XHTML-Variante `<hr />` bzw. `
` vor. Dabei bekommt das Tag nach einem Leerzeichen einen internen Ausschalt-Slash, so einen Schrägstrich. Das macht man deshalb, weil in der strengeren XHTML-Schreibweise jedes Tag ausgeschaltet werden muss(te). Und da ein `<hr>`- oder `
`-Tag nun mal kein Ausschalt-Tag kennt, schaltet man es einfach intern ab. In der Praxis findest du beide Schreibweisen vor. Selbst PHP verwendet Elemente aus XHTML. (Wir aber bleiben bei der bewährten »klassischen« Schreibweise von HTML5.)



Genug gelabert, jetzt wollen wir unsere Homepage etwas aufpolieren! Wir binden ein Bild ein.

Hier bin ich: Ein Bild einfügen

Hast du irgendwo ein Foto auf der Festplatte herumliegen? Vielleicht vom Scanner oder der Digicam? Hauptsache es liegt im Format GIF, JPG oder PNG vor. (Andere Formate sind für das Internet meist nicht geeignet.)

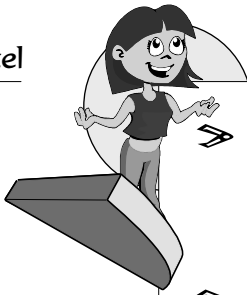
Du hast momentan keine derartige Grafik? Dann nimm zum Üben einfach ein Bild von der Buch-CD! Du findest die Datei `buffi.png` auf der CD im Ordner `beispiele/kapitel01`. Es zeigt die Vorfahren von Buffi.



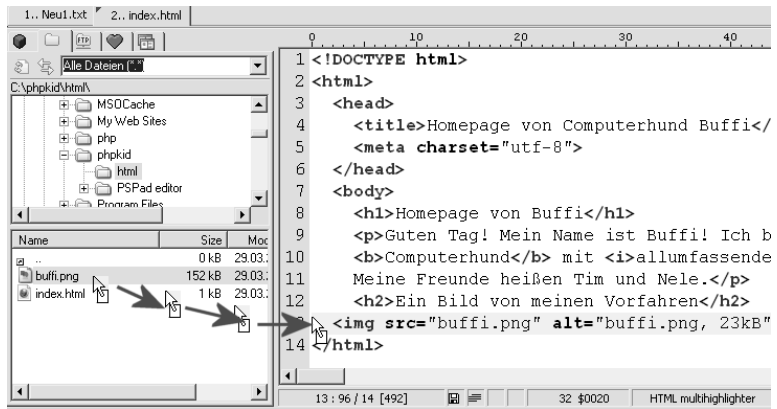
Wichtig: Dein Bild sollte nicht zu hoch und breit sein. Es sollte also, wenn du es im Grafikprogramm aufrufst, nicht die ganze Breite und Höhe des Bildschirms ausfüllen. Die Abbildung von »Buffis Vorfahren« ist 445 Pixel breit und 310 Pixel hoch. (Pixel sind Bildpunkte, eine Maßeinheit auf dem Bildschirm.)

Fügen wir das Bild nun ganz schnell in unsere Homepage ein! Und zwar unterhalb der schon vorbereiteten Überschrift 2.

1



- Achte zuerst darauf, dass die Grafik im gleichen Ordner liegt wie deine HTML-Datei. Das ist zwar nicht unbedingt nötig, macht das Einfügen aber einfacher. Kopiere das Bild also in den Ordner `html`. Zur Erinnerung: Die Beispielgrafik für Buffi heißt `buffi.png`.
- Sorge in PSPad dafür, dass der Mini-Editor mit der Dateiansicht sichtbar ist. Du weißt schon, per *Toolfenster*, zweites Register. Du schaltest das Toolfenster über das Menü ANSICHT ein bzw. aus.
- An welche Stelle möchtest du die Grafik einfügen? Schaffe hier eine Leerzeile, drücke also auf `Enter`. Und jetzt geht es los: Ziehe die Grafikdatei aus der Dateiliste direkt in das Dokument. Und zwar an die gewünschte Stelle:



Klicken und ziehen – so einfach fügst du eine Grafik ein.

- Hurra, die Grafik ist da! Das entsprechende Tag wurde fix und fertig eingefügt, mit allem Drum und Dran. Im Beispiel sieht die lange Zeile so aus:

```


```

- Das Attribut-Werte-Paar `border="0"` dient übrigens zum Unterdrücken einer eventuellen Umrandung.

Hintergrund dieser »border-Geschichte«: Wenn eine Grafik zum Hyperlink gemacht wird, bekommt sie dadurch automatisch eine kräftige blaue Umrandung. Das ist nun einmal so bei Hyperlinks und Grafiken. Um diese hässliche Umrandung zu unterdrücken, schreiben PSPad und andere Editoren zur Sicherheit das Attribut-Werte-Paar `border="0"`. Du könntest als Alternative auch CSS-Code schreiben `style="border: 0;"`. Das ist moderner, kommt im Endeffekt aber aufs Gleiche hinaus.





Grafik innerhalb eines Absatzes einfügen

Wir sind noch nicht fertig mit der Grafik. Wir packen die Grafik zusätzlich noch in ein sogenanntes Block-Element. Welches nehmen wir? Den Absatz! Setze also vor der Grafik das einleitende `<p>`-Tag. Schalte es hinter der Grafik mit `</p>` wieder ab.

```
<p>

</p>
```

Der Vorteil: Du kannst den Absatz nun zentrieren oder rechtsbündig ausrichten. Das Bild wird dadurch ebenfalls zentriert bzw. rechts ausgerichtet.

Was bedeuten die Attribute von ``?

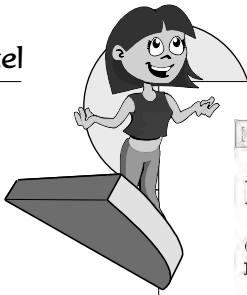
Schau dir das eben eingefügte Tag für die Grafik doch etwas genauer an. Es handelt sich um das Tag ``. Zuerst fällt dir vielleicht auf, dass auch `` kein Endtag benötigt. Wieder eine Ausnahme. Doch was bedeuten die Attribute?

- ◆ Das Attribut `src` steht für *source*, Quelle. Als Wert gibst du den Grafikenamen in Gänsefüßchen an. (Sollte sich die Grafik in einem Unterordner befinden, musst du den Pfad ebenfalls notieren.)
- ◆ Das Attribut `alt` sorgt nicht etwa dafür, dass die Grafik »alt« aussieht. Im Gegenteil! Dieses Attribut ist sehr modern und erzeugt einen sogenannten Alternativtext. Der ist wichtig für Suchmaschinen und für Leute, die nicht gut sehen können. (Sie lassen sich diesen alternativen Text vom Browser einfach vorlesen!) Nach dem Gleichheitszeichen tippt du ein beschreibendes Wort oder eine Wortgruppe.

Ersetze also den Platzhaltertext von PSPad beispielsweise durch eine Wortgruppe wie *Die Vorfahren von Hund Buffi*.

- ◆ Das Attribut `title` stelle ich dir in der nächsten Abbildung vor. Es erzeugt ... nun ... einen Titel für das Bild. Den Effekt siehst du, wenn du den Mauszeiger über die Abbildung führst. Auch hier könnte ruhig mehr Text stehen – im Beispiel schreibe ich *Buffis Vorfahren*.

1



- ❖ Die Attribute `width` und `height` »kümmern« sich um Breite und Höhe der Grafik. Die Grafik von Buffis Vorfahren ist im Beispiel 445 Pixel breit und 310 Pixel hoch. Die Reihenfolge, in der man die Attribute angibt, ist dabei egal. Du kannst sie auch vertauschen.



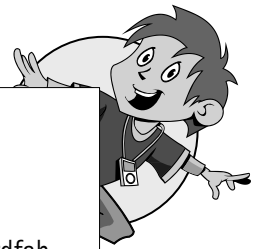
Es gibt übrigens eine modernere Schreibweise für die drei Attribute `border`, `height` und `width`. Und zwar mit der Stil- und Formatiersprache CSS. Ersetze `border="0" height="310" width="445"` einfach durch: `style="border: 0; height: 310px; width: 445px;"`. Du schreibst also nur noch das `style`-Attribut und setzt alle weiteren CSS-Anweisungen in Gänsefüßchen. Mehr zu CSS folgt weiter hinten!

Mit dem `style`-Attribut von CSS sieht das ``-Tag dann so aus:

```
<p>

</p>
```

Übrigens: Das hier verwendete Bild stammt aus dem alten Ägypten und ist mehr als 3000 Jahre alt. Es stellt altägyptische Pharaonenhunde dar. Ich habe es aus der Wikipedia entnommen, es stammt aus dem Medienarchiv *Wikimedia Commons*. Es ist urheberrechtsfrei, du kannst es also unbeschwert auf deiner Website einsetzen!



Meine Hobbys kurz aufgelistet

Noch nicht fertig! Schreibe auf, welchen Hobbys du frönst! Skateboardfahren? Inlineskaten? Lesen? Programmieren? Manga-Sammeln? Lego? Playmobil? Pferdebücher lesen? StarWars-Filme gucken? Was auch immer: Für solche Dinge bieten sich in idealer Weise die Listen an.

Mit Aufzählungspunkten: ungeordnete Liste

Im ersten Vorschlag verwenden wir eine sogenannte *unordered list*, eine »ungeordnete Liste«. Das entsprechende Tag-Paar heißt ``. Die einzelnen Listen-Einträge wiederum werden in `` eingekleidet. Dieses *li* steht dabei als Abkürzung für *list item* = Listenelement.

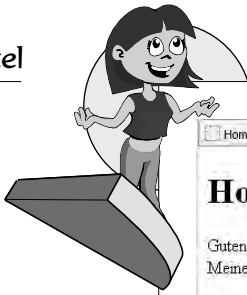
Schluss mit dem Englischunterricht, ran an das Beispiel. Ich zeige dir den gesamten Rest des Dokuments. Die Liste beginnt unter der neu eingefügten H3 *Ich habe folgende Hobbys*. Ergänze also noch die Überschrift und lege dann mit deinen Hobbys los!

```
...
<h2>Ein Bild von meinen Vorfahren</h2>
<p>

</p>
<h3>Ich habe folgende Hobbys:</h3>
<ul>
<li>Sterne betrachten</li>
<li>Inlineskaten</li>
<li>Harry Potter lesen</li>
<li>Homepages für Freunde erstellen</li>
</ul>
</body>
</html>
```

In einer solchen *unordered list* bekommt jeder Aufzählungspunkt einen kleinen Kreis vorangestellt. Sieht doch irgendwie knuffig aus, nicht wahr?

1



Homepage von Computerhund Buffi

Homepage von Buffi

Guten Tag! Mein Name ist Buffi! Ich bin von Beruf **Computerhund** mit *allumfassender Allgemeinbildung*. Meine Freunde heißen Tim und Nele.

Ein Bild von meinen Vorfahren

Ich habe folgende Hobbys:

- Sterne betrachten
- Inlineskaten
- Harry Potter lesen
- Homepages für Freunde erstellen

Skaten, Lesen, Homepage erstellen? Hier ist die Liste für deine Hobbys!

Eins, zwei, drei: Ordnung schaffen durch Nummerierung

Du möchtest deine Hobbys lieber durchnummerieren? Auch dieser Wunsch lässt sich leicht erfüllen! Bringe Ordnung ins Chaos: Verwende eine *ordered list*.

Ersetze einfach das Tag-Paar `` durch ``. Wie von Geisterhand werden deine Einträge nun durchnummeriert.

```
<h3>Ich habe folgende Hobbys:</h3>
<ol>
  <li>Sterne betrachten</li>
  <li>Inlineskaten</li>
  <li>Harry Potter lesen</li>
  <li>Homepages für Freunde erstellen</li>
</ol>
```

Das Schönste: Du kannst nun nachträglich so viele Einträge hinzufügen, wie du möchtest. Die Nummerierung passt sich immer wieder an!



»Hobby by Number«: Eine durchnummerierte Liste.

Cooler Links zu coolen Plätzen

Was wäre das Web ohne die Hyperlinks? Ein Nichts! Es würde gar nicht existieren! Erst durch diese Kreuz- und Querverweise werden deine Seiten richtig lebendig. Verlinke zu den schönsten und interessantesten Plätzen im Web. Wirf deine »Anker« aus.

Einfache Hyperlinks erstellen

Auf welche Seiten verlinkst du? Buffi wählt seinen »Heimatverlag« und netterweise auch die Seiten des Buchautors (Danke, Buffi!). Hier die entsprechenden Adressen, die sogenannten URLs.

◇ <http://www.it-fachportal.de>

◇ <http://www.phpkid.de>

Zum Erstellen von Querverweisen (Links) verwendest du das `<a>`-Tag. Das `a` steht für Anker. Du wirfst also einen »Anker« aus. Als Ziel verwendest du die entsprechende Webadresse. Und dafür brauchst du zusätzlich das Attribut `href`. Denn erst mit diesem Attribut kannst du die Zieladresse für `<a>` angeben. Geschafft? Danach notierst du den beschreibenden Text. Hier ein Beispiel:

```
<a href="http://www.it-fachportal.de">Mein Heimatverlag</a>
```

Nicht vergessen: Das Tag wird zum Schluss natürlich durch `` abgeschaltet.

Eine Web-Adresse muss bei Hyperlinks stets ausgeschrieben werden. Die einfache Abkürzung nach dem Motto `www.it-fachportal.de` oder gar `it-fachportal.de` ist hier nicht erlaubt. Im Zweifelsfall wird der Link nicht funktionieren.





1

So sieht der Quelltext für das gesamte Hyperlink-Beispiel von Hund Buffi aus:

```
<p>
<strong>Besuche doch einmal folgende Seiten</strong>:<br>
<a href="http://www.it-fachportal.de">Mein Heimatverlag</a> -
<a href="http://www.phpkid.de">Serviceseite zum Buch</a>
</p>
```

Auf eine Überschrift hat Buffi diesmal verzichtet. Der Abschnitt mit den Hyperlinks wurde in einen Absatz (`<p></p>`) eingehüllt. Doch was passiert innerhalb dieses Absatzes?

Zur Abwechslung findest du diesmal das Tag-Paar `` vor. Praktisch als »Überschrifts-Ersatz«. Dadurch wird die entsprechende Passage fett hervorgehoben und stärker betont.

Danach hat Hund Buffi einen Zeilenumbruch gesetzt, den schon besprochenen *break* `
`. Solch ein Umbruch erzeugt eine neue Zeile. Dabei entsteht jedoch kein neuer Absatz! (Du erkennst es in der Vorschau: Der Abstand zwischen den Zeilen ist geringer!)

Nun folgen die Hyperlinks nebeneinander, aus optischen Gründen nur durch einen Bindestrich voneinander getrennt.



Unbesuchte Hyperlinks sind per Voreinstellung blau, besuchte färben sich lila.

Parke doch einmal den Mauszeiger über einem Hyperlink. Schon erscheint das Verweis-Ziel in der Statuszeile, also in der unteren Zeile des Browsers.

Übrigens: Du musst nicht immer auf eine externe Webseite verweisen. Setze Links vor allem innerhalb deines »Web-Projekts«. Verweise von einer Seite auf die andere. Du willst auf der Startseite auf eine Unterseite namens `hobby.html` verweisen? Diese liegt im gleichen Ordner wie die Startseite? Als Linkbeschreibung soll der Text *Meine Interessen* verwendet werden? Dann sieht der interne Links so aus:

```
<a href="hobby.html">Meine Interessen</a>
```





Wie wäre es mit ein paar Tipps und Tricks zu Hyperlinks?

- Du möchtest, dass sich nach Klick auf den Link ein eigenes Fenster bzw. ein neuer Browsertab öffnet? Dann notiere ganz einfach zusätzlich das Attribut-Werte-Paar `target="_blank"` im `<a>`-Tag:

```
<a href="http://www.it-fachportal.de" target="_blank">☞  
Mein Heimatverlag</a>
```

- Du möchtest, dass beim Darüberfahren mit dem Mauszeiger ein informativer Beschreibungstext auf dem Hyperlink erscheint? Dann ergänze das schon bekannte Attribut `title` (Beschreibungstitel). Gib als Wert ein beschreibendes Wort oder eine Wortgruppe an:

```
<a href="http://www.bhv.de" title="Software für Kids">☞  
Lernsoftware und mehr</a>
```

- Du möchtest die Farbe der Hyperlinks ändern? Die Unterstreichung abschalten? Das alles gelingt mit den genialen Style Sheets (CSS). Blättere einfach ein Stück weiter nach hinten, dort verrate ich dir mehr zu diesem Thema.

Meine drei Lieblingsfächer in einer Tabelle

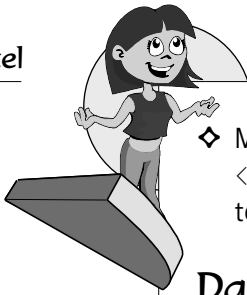
Schreiben wir zur Abwechslung doch mal eine Tabelle. Warum? Weil es Spaß macht! Und weil Tabellen auf HTML-Seiten eine ganz wichtige Rolle spielen. Und weil du Tabellen kennenlernen solltest. Sind das genug Gründe? Dann los!

Grundaufbau der Tabellen

Was musst du zu Tabellen wissen? Nicht viel!

- ◇ Jede Tabelle wird vom Tag-Paar `<table></table>` eingerahmt. (*table* ist das englische Wort für Tabelle!)
- ◇ Eine Zeile steht innerhalb von `<tr></tr>` (*table row* heißt Tabellenzeile).
- ◇ Eine einzelne Zelle (das Tabellenkästchen) wird zwischen `<td></td>` »gequetscht«. Mit *td* ist *table data* gemeint, frei übersetzt »Zellinhalt«.

1



- ❖ Möchtest du eine Zelle besonders hervorheben, wählst du statt `<td></td>` einfach `<th></th>` für *table head*. Eine solcherart gestaltete Zelle wird fett hervorgehoben. Der Inhalt wird zentriert.

Das Beispiel

Zuerst zeige ich dir das Beispiel. Eine Tabelle besteht aus Zeilen und Spalten. Diese werden durch die Tabellenkästchen (Zellen) gebildet.



Die Beispiel-Tabelle besteht aus vier Zeilen und zwei Spalten.

Das Beispiel sieht im Quelltext folgendermaßen aus:

```
<table border="1">
<tr>
<th>Lieblingsfach</th><th>Warum?</th>
</tr>
<tr>
<td>Informatik</td><td>Weil ich der Beste bin!</td>
</tr>
<tr>
<td>Physik</td><td>Weil wir lernen, wie ein Motor
funktioniert.</td>
</tr>
<tr>
<td>Zeichnen</td><td>Weil der Lehrer mit uns Manga-Comics
zeichnet.</td>
</tr>
</table>
```



Die Erklärung

Durchschaust du die Struktur? Zuerst setzen wir das einleitende Tag `<table>` und drücken schon ein paar Mal `Enter`. Damit schaffen wir etwas Luft. Nicht vergessen: Schalte das `<table>`-Einschalt-Tag gleich jetzt am Ende der Tabelle mit `</table>` wieder ab!

Damit die Tabelle einen Rahmen erhält, arbeite ich gleichzeitig mit dem Attribut-Werte-Paar `border="1"`. Das einleitende Tabellen-Tag sieht also so aus: `<table border="1">`. (Wenn du den Rahmen vergrößern willst, erhöhst du einfach den Wert hinter `border`. Mit `border="5"` bekommst du beispielsweise einen 5 Pixel dicken Rand.)



Danach definiere ich die erste von vier Zeilen. Ich schreibe `<tr>`, lasse danach ein paar Zeilen frei und setze das abschließende `</tr>`. Wenn du willst, kannst du alle vier Zeilen schon derart vorbereiten, hier zumindest zwei Zeilen:

```
<tr>
...
</tr>
<tr>
...
</tr>
```

Nun kümmern wir uns um die einzelnen »Kästchen« der Tabelle. Die erste Zeile bekommt zweimal `<th></th>` »spendiert«. Es soll schließlich der Tabellenkopf werden. In den übrigen Zeilen begnügen wir uns mit je zweimal `<td></td>` für den normalen »Zellinhalt«.

Wie gesagt, was dann in der Zelle selber erscheinen soll, wird einfach zwischen `<th></th>` bzw. `<td></td>` notiert. So weit, so gut. Aber ich wusste gar nicht, dass Buffi noch zur Schule geht ...

Schickes Layout mit Style Sheets

Cooler Seite? Von wegen! Bis jetzt sieht unser Projekt aus wie eine Kakerlake hinterm Küchenschrank – farblos und platt. Dagegen sollten wir etwas unternehmen, und zwar mit Style Sheets. Lege Schriftart, Schriftgröße und Farben fest! Verpasse den Hyperlinks einen »Mouseover-Effekt«. Du wirst staunen, mit wie wenig Aufwand man seine Seiten aufpeppen kann.

1



Style Sheets sind eine ganz fabelhafte Geschichte. Du legst einmal fest, wie deine Homepage auszusehen hat. Das machst du am besten in einer extra Datei. Dadurch erstellst du eine Art Musterschablone. Dieses Gestaltungs-Muster kannst du nun beliebig oft für weitere HTML-Seiten verwenden. Das spart viel Arbeit! Dabei gilt: Style-Sheet-Dateien sind einfache Textdateien mit der Endung `.css`.

Verweis auf CSS-Datei setzen

Also, frisch ans Werk. Worauf wartest du noch? Die extra »Stil-Datei« soll im Beispiel `phpkid.css` heißen. Sie wird im gleichen Ordner entstehen wie unsere HTML-Datei. Doch zuerst setzen wir einen Verweis auf diese Datei. Wir binden sie also in unser HTML-Dokument `index.html` ein. Das gelingt im Kopfbereich, im Head:

➤ Suche einmal diese Zeile:

```
<meta charset="utf-8">
```

➤ Klicke hinter diese Zeile und drücke `Enter`. Damit erzeugst du eine leere Zeile für den Verweis auf diese Datei. Schreibe nun den eigentlichen Verweis, notiere also folgende Zeile:

```
<link rel="stylesheet" href="phpkid.css">
```

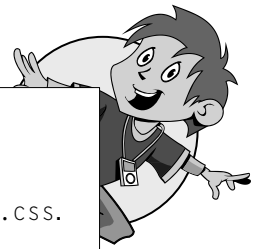
Insgesamt sieht der Anfang der HTML-Datei also so aus:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Homepage von Computerhund Buffi</title>
5 <meta charset="utf-8">
6 <link rel="stylesheet" href="phpkid.css">
7 </head>
```

Lass' dich von der Syntax dieser Zeile 6 nicht zu sehr entmutigen. Wichtig ist das Attribut `href`. Hier notierst du hinter dem Gleichheitszeichen den Namen der CSS-Datei, und zwar in Gänsefüßchen.

Du speicherst deine HTML-Datei, aktualisierst die Vorschau und – es passiert rein gar nichts? Na klar! Wir haben einen Link auf eine Datei erstellt, die es noch gar nicht gibt. Da wartet also ein Stück Arbeit auf uns.





CSS-Datei erstellen

Eine CSS-Datei ist nichts weiter als eine Textdatei mit der Endung `.css`. Also, erstelle eine neue Textdatei.

- Wähle in PSPad DATEI|NEU. Das Dialogfenster *Neu* erscheint, du solltest dich im Register *Neue Datei aus Vorlage erstellen* befinden. Wähle den Ordner *CSS* und doppelklicke auf *Default*.
- PSPad fügt eine fast leere CSS-Datei ein und schaltet auf die CSS-Farbhervorhebung um. Der Text `/* CSS Document */` beweist, dass es sich um eine CSS-Datei handelt. Lösche diese Platzhalter-Zeile!
- Ehe ich Style Sheets jetzt groß und breit erkläre, legen wir los! Tippe zuerst folgende drei Zeilen. (Die geschweiften Klammern erzeugst du mit gedrückter `[AltGr]`-Taste. Tippe dazu entweder die `[7]` bzw. `[0.]`)

```
body {  
    font-family: Verdana, Arial, Helvetica, sans-serif;  
}
```

- Speichere die Datei ab. Wähle als Dateinamen `phpkid.css`. Die Datei wird im Beispiel im gleichen Ordner abgelegt wie das HTML-Dokument, ich hatte den Ordner `html` vorgeschlagen. Achte darauf!

Die Schreibweise für CSS-Style-Sheets gleicht der von Programmiersprachen wie JavaScript oder PHP. So findest du auch in CSS die berühmten geschweiften Klammern. Diese geschweiften Klammern bilden einen Block. Sie halten praktisch zusammen, was zusammen gehört. Auch in CSS wird (fast) jede Zeile durch ein Semikolon (;) abgeschlossen. Es ist so, wie du es in PHP noch kennenlernen wirst!



Rufe das HTML-Dokument `index.html` doch einmal in der Browseransicht auf. Aktualisiere die Ansicht, z. B. durch Druck auf die Funktionstaste `[F5]`. Wahnsinn! Auf einen Schlag ändert sich die Schriftart im gesamten Dokument. Und das nur mit diesen simplen drei Zeilen. Wie geht das?

Nun, das gesamte Dokument befindet sich zwischen den Tags `<body>` `</body>`. Mit diesem Tag kannst du praktisch alle Textpassagen erreichen, `<body></body>` ist sozusagen die »große Klammer«, die alles zusammenhält. Also `body`, okay. Doch wo sind die spitzen Klammern? Die werden in CSS nicht mitgeschrieben! Du notierst also das »nackige Tag« ohne spitze Klammern. Wir schreiben einfach `body` und haben damit den sogenannten Selektor zu Papier gebracht. Wenn das kein Grund zum Feiern ist ...



1

Nun setzen wir nach einem Leerzeichen die öffnende geschweifte Klammer. Der Block beginnt. Mein Tipp: Drücke schon ruhig zweimal auf `Enter`. Setze gleich die schließende Klammer. (Denn hinterher vergisst man das oft, glaube mir!) Zwischen diesen Klammern notierst du nun die Formatierungsanweisungen. Im Beispiel handelt es sich um folgende Zeile:

```
font-family: Verdana, Arial, Helvetica, sans-serif;
```

Damit legen wir die *font-family* fest, die Schriftfamilie – hier also *Verdana*. Am Ende der Zeile setzt du ein Semikolon.



Wenn der Benutzer auf seinem Rechner kein *Verdana* hat? Dann wird die Schriftart verwendet, die als nächstes nach dem Komma folgt. Und wenn es auch hier heißt: Fehlanzeige? Dann probiert der Browser, ob er *Helvetica* findet. Sollte das nicht glücken, wählt der Browser irgendeine sachliche Schriftart. Dafür sorgt *sans-serif* am Schluss der Aufreihung.

Das ganze Gebilde mit Selektor und geschweiften Klammern nennt sich nun Regel. Und tatsächlich haben wir damit eine fabelhafte kleine Regel erstellt, mit der wir die Schriftart für das gesamte Dokument steuern.

Einige Attribute und Werte von CSS

Neugierig, welche Anweisungen es noch so gibt? Die folgende Tabelle zeigt dir einige wichtige Attribute und Werte von CSS. In der linken Spalte findest du jeweils das Beispiel. Rechts erkläre ich dir, was passiert.

Beispiel	verantwortlich für
<code>font-family: Arial, Helvetica;</code>	Schriftart Arial bzw. Helvetica
<code>font-size: 12pt;</code>	Schriftgröße 12 Punkt
<code>font-weight: bold;</code>	Zeichenformat fett
<code>font-variant: small-caps;</code>	KAPITÄLCHEN (DAS SIND KLEINE GROSSBUCHSTABEN)
<code>font-style: italic;</code>	Schriftstil (z. B. <i>kursiv</i>)
<code>line-height: 1.2;</code>	Zeilenabstand (z. B. 1,2-zeilig)
<code>margin: 10px;</code>	Rand rundherum 10 Pixel
<code>text-decoration: none;</code>	Texteffekt (Unterstreichung) abschalten
<code>border-style: solid;</code>	Rahmen rundherum ziehen

Mehrere Fliegen mit einer Klappe

Nun kennst du schon einige CSS-Eigenschaften. Wunderbar. Kümmere dich doch einmal um die *Schriftgröße*! Angenommen, du möchtest in Absätzen

Schön bunt hier: Farben zuweisen



(Selektor `p`), Tabellen (`table`) und Listeneinträgen (`li`) eine Schriftgröße von 10 Punkt einrichten.

Dann schlage doch einmal drei Fliegen mit einer Klappe! Reihe die Selektoren vor der öffnenden geschweiften Klammer auf. Dann gilt die Regel für alle drei Tags zugleich!

```
p, table, li {  
    font-size : 10pt;  
}
```

Warum haben wir in dieser Regel keine Schriftart definiert? Nun, diese hatten wir doch schon in der »body-Regel« festgelegt. Schließlich gilt *body* für das gesamte Dokument, also auch für `p`, `table` und `li`. In dieser Regel müssen wir nur noch die Schriftgröße bestimmen. Und zwar zusätzlich.



Schön bunt hier: Farben zuweisen

Natürlich kannst du in CSS nicht nur Texteingenschaften, Ränder und Rahmen festlegen. Arbeite doch einmal mit Farben! Für die Schriftfarbe nimmst du das Attribut *color*. Die Hintergrundfarbe wird dagegen durch *background-color* gesteuert.

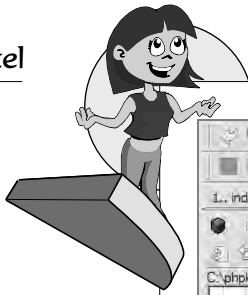
Farb-Beispiel	verantwortlich für
<code>color: blue;</code>	Farbe blau
<code>background-color: silver;</code>	Hintergrundfarbe hellgrau

Doch wie viele Farben kannst du verwenden? 16,7 Millionen! Und wie stellt man sie dar? Das geht zum einen mit dem sogenannten Hexadezimalwert. Das ist eine kryptische Zeichenfolge. Sie beginnt mit einer Raute, gefolgt von weiteren 3–6 Stellen. Diese beschreiben die Zusammensetzung aus den drei Grundfarben Rot, Grün und Blau. Ein Blau sieht so aus: `#0000FF`.

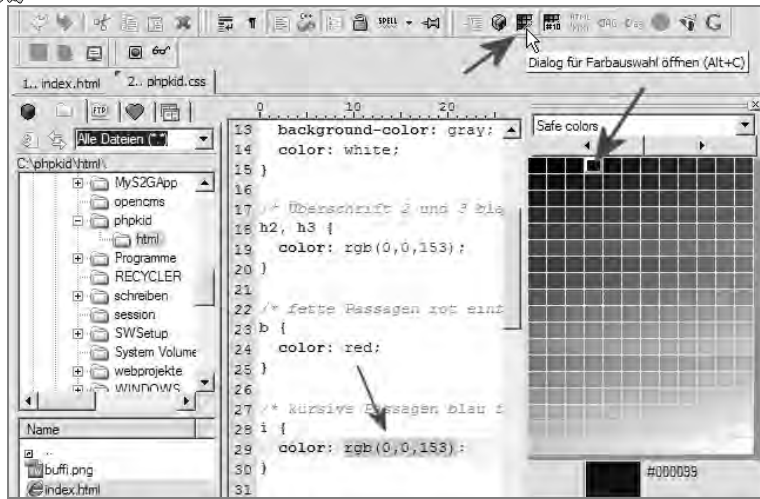
Zum anderen kannst du diese RGB-Anteile auch auf andere Weise beschreiben. Notiere `rgb()` und trage innerhalb der Klammern die Anteile der drei Grundfarben ein. Und zwar auf einer Skala von 0 bis 255 – getrennt mit Komma. Ein Blau sieht dann so aus: `rgb(0,0,255)`.

Du möchtest mit PSPad die gewünschte Farbe zuweisen? Wähle WERKZEUGE|FARBAUSWAHL bzw. die Farbauswahl-Schaltfläche. Setze den Cursor dort hin, wo der Farbwert stehen soll. Doppelklicke auf deine Wunschfarbe:

1



PSPad verwendet zum Einfügen der Farben die freundliche »RGB-Syntax«.



Neben den 16,7 Millionen unterschiedlichen Hexadezimalwerten gibt es aber auch »freundlichere Farbnamen«. Zumindest für die 16 Grundfarben. Wenn du dich also auf diese 16 Grundfarben beschränken möchtest, wählst du einen der 16 englischen Farbnamen aus.

Eine Gesamtübersicht über diese Grundfarben zeige ich dir in der Tabelle. Dabei ist es egal, ob du dich für den Farbnamen (linke Spalte), den Hexadezimalwert oder den RGB-Farbwert (rechte Spalte) entscheidest:

Farbname	dahinter steckt	Hexadezimalwert	RGB-Farbwert
black	schwarz	#000000	rgb(0,0,0)
silver	hellgrau	#c0c0c0	rgb(192,192,192)
gray	grau	#808080	rgb(128,128,128)
white	weiß	#ffffff	rgb(255,255,255)
maroon	kastanienbraun	#800000	rgb(128,0,0)
red	rot	#ff0000	rgb(255,0,0)
purple	lila	#800080	rgb(128,0,128)
fuchsia	helllila	#ff00ff	rgb(255,0,255)
green	dunkelgrün	#008000	rgb(0,128,0)
lime	hellgrün	#00ff00	rgb(0,255,0)
olive	olivgrün	#808000	rgb(128,128,0)
yellow	gelb	#ffff00	rgb(255,255,0)
navy	dunkelblau	#000080	rgb(0,0,128)
blue	blau	#0000ff	rgb(0,0,255)
teal	blaugrün	#008080	rgb(0,128,128)
aqua	himmelblau	#00ffff	rgb(0,255,255)

Buffi, gar nicht faul, hat seine Homepage inzwischen komplett mit Style Sheets aufgebürstet. So sieht seine neue »Layout-Page« aus.



Homepage von Computerhund Buffi

Homepage von Buffi

Guten Tag! Mein Name ist Buffi! Ich bin von Beruf Computerhund mit *allumfassender Allgemeinbildung*. Meine Freunde heißen Tim und Nele.

Ein Bild von meinen Vorfahren



Ich habe folgende Hobbys:

- Sterne betrachten
- Inlineskatzen
- Harry Potter lesen
- Homepages für Freunde erstellen

Besuche doch einmal folgende Seiten:
[Mein Heimatverlag](#) - [Serviceseite zum Buch](#)

Lieblingsfach	Warum?
Informatik	Weil ich der Beste bin!
Physik	Weil wir lernen, wie ein Motor funktioniert.
Zeichnen	Weil der Lehrer mit uns Manga-Comics zeichnet.

Welche eine Verwandlung!
CSS bringt Farbe ins Bild.

Die komplette CSS-Datei im Überblick

Möchtest du die gesamte CSS-Datei sehen, die Hund Buffi zur Steuerung seiner Seite geschrieben hat? Bitte sehr. Damit du durch den Code durchsteigst, habe ich mit beschreibenden Kommentaren gearbeitet.

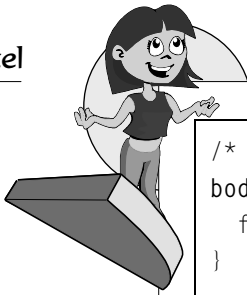
Kommentare sind Passagen, die nur für den Autor der Datei interessant sind. Sie werden nicht ausgeführt, bleiben praktisch unsichtbar. In CSS verwendest du die gleichen Kommentarzeichen, die du später auch in PHP setzen wirst. Ein Kommentar wird mit `/*` eingeleitet und endet mit `*/`. So kannst du der ersten Regel (also dem Gebilde in geschweiften Klammern) z. B. folgende Beschreibung vorstellen:

```
/* Schriftart im gesamten Dokument */
```

Außerdem empfehle ich dir, mit Leerzeilen für mehr Übersicht zu sorgen. Was hier wie die reinste Platzverschwendung aussieht, wird uns später bei PHP sehr helfen! Aber nun »Manege frei« für den CSS-Code:



1



```
/* Schriftart im gesamten Dokument */
body {
    font-family: Verdana, Arial, Helvetica, sans-serif;
}

/* Absätze, DIVs, Tabellen, Listen, Formulare in 10 Punkt */
p, div, table, li, form {
    font-size: 10pt;
}

/* Hintergrundfarbe grau mit weißer Schrift */
h1 {
    background-color: gray;
    color: white;
}

/* Überschrift 2 und 3 blau färben */
h2, h3 {
    color: rgb(0,0,153);
}

/* fette Passagen rot einfärben */
b {
    color: red;
}

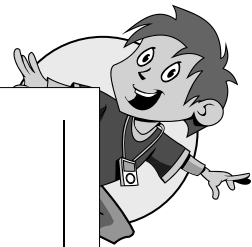
/* kursive Passagen blau färben */
i {
    color: rgb(0,0,153);
}

/* Link-Stile für Hover-Links */
a:link {
    color: rgb(0,0,153);
}

a:visited {
    color: gray;
}

a:hover {
    text-decoration: none;
}
```

So legst du die exakte Breite fest



```
color: red;
}

a:active {
  color: black;
}
```

Lies dir den Quelltext ganz in Ruhe durch. Die ersten beiden Regeln haben wir schon besprochen. Bei der zweiten Regel habe ich gleich die Tags `div` und `form` mit dazu geschrieben. So werden diese Elemente ebenfalls gestaltet. (Zu `div` und `form` gleich mehr!)

Schon mit der dritten Regel für `h1` erreichst du einen sehr interessanten »Rahmen-Effekt«. Die Schrift wird weiß eingefärbt. Doch der Hintergrund erhält die Farbe grau! Das sieht doch toll aus, oder?

Wenn du genau hinschaust, stellst du fest, dass Buffi auch die Tags `` und `<i></i>` aufpoliert hat. Fette Passagen werden zusätzlich rot, kursive dagegen blau eingefärbt.

Du wunderst dich über den letzten Abschnitt in der CSS-Datei? Was verbirgt sich hinter *Link-Stile für Hover-Links*? Das sind Sonderfälle, damit erzeugst du den berühmten Hovereffekt beim Darüberfahren mit der Maus! Mit den vier »Sonder-Selektoren« `a:link`, `a:visited`, `a:hover` und `a:active` kannst du die vier Linkzustände ansprechen. Einmal den normalen Link, dann den besuchten Link, danach den Link während des Darüberfahrens mit der Maus (Hover-Effekt!) und den aktiven Link (während des Klickens). Du solltest die Links übrigens stets in dieser Reihenfolge notieren, damit es keine Anzeigeprobleme gibt.



So legst du die exakte Breite fest

Zufrieden? Noch nicht ganz! Verändere doch einmal die Größe des Browserfensters. Ziehe das Fenster lang. Der ganze Text wandert mit.



Die Breite des Textes richtet sich nach der Breite des Browserfensters.

Das ist doch der reinste Wanderzirkus!



1

Zugegeben, auf manchen Seiten ist dieser Effekt beabsichtigt. Unser Hund wünscht jedoch eine ganz exakte Breite. Die Breite der Seite soll genau 600 Pixel betragen. Nicht mehr und nicht weniger.

Packe den Inhalt in einen »Div-Container«

Dafür eignet sich ganz ausgezeichnet ein sogenannter Div-Container. Es handelt sich um das Tag-Paar `<div></div>`. In dieses Tag-Pärchen darfst du ganze Textpassagen einwickeln. »Div« darf Überschriften, Absätze, Tabellen und Listen enthalten.

Füge direkt unter dem `<body>`-Einschalt-Tag zuerst eine leere Zeile ein. Setze dort hinein:

```
<div>
```

Schalte dieses »Div« kurz über `</body>` wieder ab. Nun sieht es so aus:

```
...
<body>
  <div>
    <h1>Homepage von Buffi</h1>
    ... viele, viele Zeilen dazwischen ...
  </table>
  </div>
</body>
</html>
```

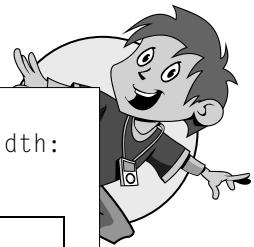
Breite festlegen über Inline-CSS

Fehlt noch die Breite (`width`). Richtig, das könnten wir in der CSS-Datei festlegen. Zu umständlich! Es handelt sich schließlich nur um eine einzige Angabe. Wir lösen das Ganze deshalb kurz und schmerzlos mit Inline-CSS.

Bei Inline-CSS werden die CSS-Anweisungen direkt im Tag notiert. Ich habe es dir in Zusammenhang mit `text-align` (Absatzausrichtung), `border` (Rahmen), `width` (Breite) und `height` (Höhe) schon gezeigt! Eine Inline-CSS-Passage beginnt mit dem Attribut `style="..."`. Die CSS-Regeln werden nun innerhalb der Gänsefüßchen notiert. Inline-CSS ist dann ideal, wenn man kurze Anweisungen festlegen möchte, die sich nicht wiederholen sollen.



Eine Umfrage! Wie findest du meine Page?



Gehe deshalb in das einschaltende `<div>`-Tag. Ergänze `style="width: 600px;"`. Insgesamt sieht das Tag also so aus:

```
<div style="width: 600px;">
```



Exakte Breite durch die CSS-Regel `width: 600px;`

Eine Umfrage! Wie findest du meine Page?

Du möchtest wissen, wie deine Seite bei den Besuchern ankommt? Erstelle eine Umfrage! Dazu nimmst du einfach eine neue Seite, die wir im Beispiel `umfrage.html` nennen. Speichere sie ebenfalls unter `C:\phpkid\html`.

➤ Baue die Seite auf dem schon bekannten Grundgerüst auf. Füge ebenfalls den schon verwendeten CSS-Link zur Datei `phpkid.css` ein!



Eine Umfrage mit Radioknöpfen – Der Besucher wählt eine Option aus.

Ein Formular mit Radioknöpfen

Und hier zeige ich dir den kompletten Quelltext dieser Umfrage. Denke zuallererst an den Titel. Passe den Bereich zwischen `<title></title>` an. Setze eine Überschrift in das Dokument.



1

Aus Platzgründen habe ich den HTML-Code im Gegensatz zur `index.html` nicht eingerückt. Es ist schließlich keine Pflicht. Und wenn du doch einrücken willst: Es gibt in PSPad eine clevere Automatik. Mehr dazu gleich ...

```
<!DOCTYPE html>
<html>
<head>
<title>Umfrage: Wie findest du meine Seite</title>
<meta charset="utf-8">
<link rel="stylesheet" href="phpkid.css">
</head>
<body>
<h2>Wie findest du meine Page?</h2>
<form action="umfrage.html">
<p>
<input type="radio" name="ergebnis" value="1"> echt super<br>
<input type="radio" name="ergebnis" value="2"> ziemlich gut<br>
<input type="radio" name="ergebnis" value="3"> geht so
</p>
<p><input type="submit" value="Absenden"></p>
</form>
</body>
</html>
```

Das eigentliche Formular beginnt mit dem `<form>`-Tag. Du musst hier außerdem das Attribut `action` eintragen. Wir tragen hier `umfrage.html` ein. (Vergiss nicht, das Formular mit `</form>` auch wieder abzuschalten.)

Im Formular selber arbeiten wir mit sogenannten Radiobuttons. Dieser Typ wurde tatsächlich von den Wellenschaltern alter Radioapparate abgeschaut. Drückt man einen Knopf rein, springt der andere heraus. Mittelwelle und UKW gleichzeitig gedrückt? Das geht bei Radiobuttons nicht!

Radiobuttons werden auch als *Kontrollkästchen* bezeichnet.

Radiobuttons werden mit `<input type="radio">` erzeugt. Gib allen zusammengehörigen Buttons den gleichen Namen. Dazu dient das Attribut `name`. Im Beispiel heißen alle Buttons `ergebnis`. Der `value` wiederum ist der Wert, den der Button beim Absenden »mitschickt«. Hier bekommt jeder Button einen eigenen Wert, z. B. 1, 2 und 3. Das dient zum späteren Wiedererkennen. Du als »Formularbesitzer« bekommst schließlich das Name-Werte-Paar zugeschickt, z. B. `ergebnis="1"`.





Submit! Doch wo bleibt die Action?

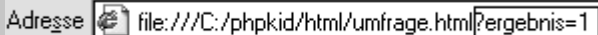
Jedes Formular benötigt zum Abschicken einen Submit-Button. Submit steht für Versenden, Übermitteln. Dieser Button sieht im Beispiel so aus:


```
<input type="submit" value="Absenden">
```

Der Wert hinter dem Attribut ist übrigens die Beschriftung. Du könntest hier auch »Los!« oder »Klicken auf eigene Gefahr« eintragen. Aber darauf will ich nicht hinaus.

Klappe die Erste. Und Action! Wenn du auf den Submit-Button klickst, geht die »Action« los. Es passiert ... nichts. Oder fast nichts. Kunststück. Schließlich passiert das, was wir im einleitenden `<form>`-Tag hinter `action` notiert haben. Doch da steht bis jetzt nur `umfrage.html`. Das bedeutet: Die Seite schickt den Formularinhalt an sich selber!

Schauen doch einmal in die Adress-Leiste des Browsers. Nach dem Klicken wird hinter die Adresse noch ein Fragezeichen angehängt. Nun steht hier plötzlich z.B. `umfrage.html?ergebnis=2` (je nachdem, welche Auswahl du getroffen hast). Das ist doch sehr interessant! Mehr können wir an dieser Stelle jedoch nicht machen. Zum Auswerten des Formularinhalts kommen wir allein mit HTML nicht weiter. Hier brauchen wir PHP.



Adresse  file:///C:/phpkid/html/umfrage.html?ergebnis=1

Ein Grund mehr, PHP zu lernen!



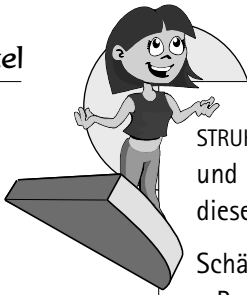
Tipps und Tricks zu PSPad

PSPad ist ein toller Editor mit vielen genialen Funktionen. Man muss sie nur kennen.

HTML-Quellcode automatisch ausrichten

Du möchtest den HTML-Quellcode schön einrücken und formatieren? Damit es ordentlich und übersichtlich aussieht? HTML-Kenner wissen: Zusätzliche Umbrüche und Leerzeichen im Editor werden nicht im Browser dargestellt. Fazit: Gliedere und formatiere den Quellcode so, wie du magst. Per Leerzeichen eingerückt, mit zusätzlichen Leerabsätzen und und und.

Keine Lust, das Gliedern von Hand zu erledigen? PSPad bietet dir gleich zwei Tools. Die »sanfte« Variante findest du unter HTML|HTML-CODE NEU



1

STRUKTURIEREN. Der Code wird ausgerichtet und automatisch eingerückt. Ab und an musst du noch ein überschüssiges Leerzeichen entfernen, denn in dieser Beziehung ist das Tool noch nicht perfekt.

Schärfere Geschütze führt »Tidy« auf, *tidy up* heißt saubermachen. Wähle z.B. HTML|TIDY|TIDY CLEAN DOCUMENT WRAP oder TIDY CLEAN DOCUMENT NOWRAP und staune! Tidy darfst du aber nur bei reinen HTML-Dateien anwenden. Sobald PHP mit dazukommt, ist dieser »Code-Verschönerer« tabu!



Tidy schreibt automatisch eine zusätzliche Meta-Zeile in das Dokument:
`<meta name="generator" content="HTML Tidy for Windows..."`
 Diese Zeile kannst du löschen.

Automatischen Zeilenumbruch einfügen

Du hast den Quellcode automatisch ausgerichtet, aber auf die Umbrüche verzichtet? Die Zeilen sind nun so lang, dass du ständig scrollen musst? Schalte die automatischen Zeilenumbrüche an. Wähle ANSICHT|ZEILENUMBRUCH. Der Umbruch erfolgt jetzt rein optisch für die Anzeige im Editor, wird jedoch nicht mitgespeichert.

Mehr Überblick in der Codeansicht

Der Quellcode wird zu klein angezeigt, du kannst ihn nicht lesen? Oder du möchtest alles auf einen Blick sehen? Na dann zoome doch einfach – so wie vom Internetbrowser oder aus der Textverarbeitung bekannt. Halte die Taste `[Strg]` gedrückt und drehe am Mousrad.

Lesezeichen für mehr Übersicht

Genial finde ich auch die Möglichkeit, mit Lesezeichen zu arbeiten. So kannst du dir besonders bei langen Codeabschnitten wichtige Zeilen besser »merken«. Mit `[Alt] + [←]` setzt du ein (gelbes) Lesezeichen. Mit `[Alt] + [→]` dagegen kannst du ein einmal gesetztes (gelbes) Lesezeichen wieder löschen. Mit den Tastentricks `[Alt] + [↓]` bzw. `[Alt] + [↑]` springst du zwischen mehreren Lesezeichen hin und her. Weitere (rote) Lesezeichen gibt es auch. Die setzt du am besten über das Kontextmenü.



Nach dem Schließen der Datei werden die Lesezeichen automatisch gelöscht.

Datei vom Windows-Explorer aus öffnen

Eine der genialsten Features von PSPad? Das ist für mich eindeutig die Möglichkeit, Dateien direkt vom Windows-Explorer aus laden zu können. Und zwar über das Kontextmenü. Klicke also mit der rechten Maustaste auf

© des Titels »PHP und MySQL für Kids« (ISBN 978-3-8266-9465-3)

2013 by Verlagsgruppe Hüthig Jehle Rehm GmbH, Heidelberg.

Nähere Informationen unter: <http://www.mitp.de/9465>



die zu öffnende Datei, beispielsweise die `index.html`. Wähle den Befehl PSPAD. Schon öffnet sich die entsprechende Datei. Klappt nicht? Dann probiere SENDEN AN|PSPAD. Klappt auch nicht?

Wähle EINSTELLUNGEN|PROGRAMM EINSTELLEN und gehe links in den Bereich Systemintegration. Achte darauf, dass die Option »PSPad« im Kontextmenü anzeigen aktiviert ist. Weg mit dem Häkchen vor Kontextmenü kaskadieren. Und setze zusätzlich ein Häkchen vor PSPad im »Senden an«-Menü zeigen. Manchmal klappt es jetzt – leider nicht immer!



Arbeit mit Verknüpfungen

Besonders nützlich ist auch die Arbeit mit Verknüpfungen. So kannst du dir zu den wichtigsten Ordnern Querverweise direkt in PSPad anlegen. Nach Klick springt PSPad sofort zum entsprechenden Ordner im Mini-Explorer. Verknüpfungen heißen in PSPad Links. Sie gehören ins Register mit dem Herz. Und so erstellst du einen Link zum Ordner `phpkid/html`:

- Achte darauf, dass das Toolfenster sichtbar ist (ANSICHT|TOOLFENSTER). Wechsle ins *Favoriten*-Register, du erkennst es am Herz.
- Klicke mit der rechten Maustaste auf das *Ordner*-Symbol und wähle den Befehl *Neuer Eintrag*.
- Das Dialogfenster *Neuer Eintrag* erscheint. Schau rechts neben das Feld *Pfad / URL* und klicke auf die Schaltfläche mit den drei Punkten. Suche den gewünschten Ordner heraus.



- Nach Klick auf OK ist der neue Link fertig – eine praktische Verknüpfung zum gewünschten Ordner!

Lege dir so viele Verknüpfungen an, wie du brauchst.

Werkzeuge und Suchfunktion

Schau dir doch auch die pfiffigen Werkzeuge von PSPad an. Du findest sie im gleichnamigen Menü. Mir hat es – neben der schon getesteten Farbauswahl – besonders der Code-Explorer angetan. Dieses Tool zeigt dir die Elemente der aktiven Datei in einer baumartigen Struktur. Ausprobieren! Mein zweitliebstes Tool heißt Text-Diff: Aktuelle Datei vergleichen. So



1

kannst du zwei verschiedene Versionen eines Dokuments laden. PSPad hebt die Unterschiede farblich hervor.

Auch die ausgefeilte Suchfunktion benötigte ich recht häufig, du findest sie im gleichnamigen Menü. Mein Favorit heißt *Suchen/Ersetzen in Dateien*. Damit kannst du beispielsweise ganze Ordner durchsuchen.

Schlussbemerkung

Uff! Das war eine ganze Menge HTML auf einmal. Gute HTML-Kenntnisse sind jedoch die Voraussetzung für die PHP-Programmierung. Immerhin bist du jetzt schon ein kleiner »HTML-Experte«.

Beim Thema HTML haben wir uns für die moderne Version HTML5 entschieden. Die bisher beste und leichteste HTML-Variante!

Du möchtest tiefer in HTML (und CSS) einsteigen? Dann empfehle ich dir die wunderbaren Bücher »HTML für Kids« und »CSS für Kids« von meinen Autorenkollegen Robert Aguilar bzw. David Sigos. Vor allem das letztgenannte Buch bietet dir eine hervorragende Einführung in schickes Seitenlayout.



Die Beispieldateien findest du übrigens auch auf der CD zum Buch. Schau in den Unterordner `beispiele/kapitel01`.

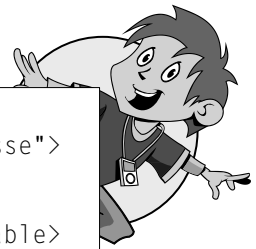
Zusammenfassung

- ◇ Du kannst mit dem PSPad blitzschnell das Grundgerüst eines HTML-Dokuments »zaubern«. Wähle im Menü DATEI den Befehl NEU und doppelklicke auf den entsprechenden Eintrag.
- ◇ Du fügst mit den entsprechenden Tags ganz lässig Überschriften (z. B. `<h1></h1>`) und Absätze ein. Setze den entsprechenden Text einfach innerhalb der Tags ein.
- ◇ Du weißt, wie du mit dem Tag `` eine Grafik in deine Homepage einfügst. Ziehe die Grafikdatei einfach aus der Dateiliste des Toolfensters in dein HTML-Dokument. Passe die Attribute und Werte des Tags dann wunschgemäß an.
- ◇ Du kannst mit `` Aufzählungen und mit `` Listen mit HTML erstellen. Du notierst beliebig viele Listen-Einträge zwischen ``.

© des Titels »PHP und MySQL für Kids« (ISBN 978-3-8266-9465-3)

2013 by Verlagsgruppe Hüthig Jehle Rehm GmbH, Heidelberg.

Nähere Informationen unter: <http://www.mitp.de/9465>



- ◇ Du erstellst Hyperlinks nach dem Muster `Text für Hyperlinks`.
- ◇ Du schreibst HTML-Tabellen mit den entsprechenden Tags `<table></table>`, `<tr></tr>`, `<th></th>` und `<td></td>`.
- ◇ Du erstellst externe CSS-Dateien und steuerst damit das Aussehen deiner Homepage. Das sind Textdateien mit der Endung `.css`. Hier notierst du mit Hilfe von Attributen, wie die einzelnen Tags formatiert werden sollen. Das Attribut für die Schriftart heißt z. B. `font-family`.
- ◇ Du arbeitest mit Farben, legst die Breite fest mit Hilfe eines `<div></div>`-Containers und dem Attribut `width`.
- ◇ Du weißt, wie du mit den `<form></form>`-Tags und mit Kontrollkästchen ein Umfrageformular vorbereiten kannst und hast den SUBMIT-Button ausprobiert.

Ein paar Fragen ...

Und jetzt bist du an der Reihe. Beantworte zuerst ein paar Fragen und löse dann ein paar Aufgaben. Viel Spaß und viel Erfolg!

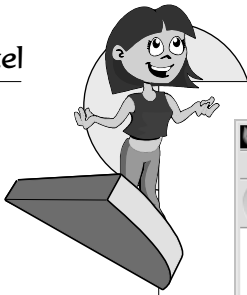
1. Zwischen welchen Tags wird der Kopfbereich des Dokuments notiert?
2. Wie heißt der Farbname für *kastanienbraun*? Wie lautet der dazugehörige hexadezimale Farbcode und wie die RGB-Syntax?
3. Wie sehen die Kommentarzeichen aus, die in CSS-Dateien verwendet werden?

Die Antworten zu den Fragen stecken allesamt auf der CD zum Buch. Schau im Ordner `fragen` nach!

... und ein paar Aufgaben

1. Schau in die Seite `index.html`. Füge unter der Aufzählung deiner Hobbys eine Linie ein. Welches Tag verwendest du dafür?
2. Setze ganz oben auf die Seite `index.html` einen neuen Absatz, und zwar unterhalb von `<div style="width: 600px;">`. Füge dort einen Verweis auf die zweite Seite `umfrage.html` ein. Schreibe als Linktext: *Zur Umfrage*. Setze ganz oben auf der Umfrage-Seite (direkt unter `<body>`) ebenfalls einen Absatz ein. Hier soll einen Link zurück verweisen. Dieser soll heißen: *Zur Startseite*.

1



Verlinke dein Projekt mit internen Querverweisen.



3. Prüfe deine Seiten auf Korrektheit. Surfe dazu zum »Prüfdienst« des W3C. (Das sind die Leute, die HTML erfunden haben.)

Surfe zu <http://validator.w3.org>. Gehe ins Register *Validate by File Upload*. Klicke auf den Button DURCHSUCHEN. Suche die zu prüfende Datei heraus und klicke auf CHECK.



Die Prüfung funktioniert nur richtig, wenn die Datei eine Dokumenttyp-Deklaration besitzt. Du darfst diese erste Zeile `<!DOCTYPE html>` also nicht weglassen.

4. Du legst mit PSPad eine neue Seite an und wählst dafür DATEI|NEU bzw. `Strg` + `N`? Dann gehst du ja immer ins Register *Neue Datei aus Vorlage erstellen* und wählst die angepasste Vorlage *HTML5*. Mache diese Vorlage noch perfekter. Ergänze den Link zur CSS-Datei. Füge also folgende Zeile nachträglich in die Vorlage ein:

```
<link rel="stylesheet" href="phpkid.css">
```

So sparen wir uns in Zukunft auch das Schreiben dieser Passage.

Die Lösungen für die Aufgaben findest du, falls möglich, stets auf der CD zum Buch, und zwar unter dem Ordner `loesungen`. Schau im Beispiel unter `loesungen/kapitel01` nach! Und wundere dich nicht – ich habe den HTML-Quellcode mit Tidy automatisch eingerückt.