

Design anpassen – eine kurze Einführung

Das gesamte Layout von Papoo basiert auf modernen CSS Layouts die vollständig ohne Layout Tabellen funktionieren.

Um schnell vorhandene Designs anpassen zu können sind Kenntnisse in CSS erforderlich. Die reine Anpassung wie z.B. der Austausch von Bilder oder ändern von Schriftfarben ist sehr simpel. Die Erstellung eigener Designs im Gegensatz dazu sehr komplex. An dieser Stelle soll es keine Einführung in CSS geben, dafür sind andere Quellen wesentlich besser geeignet, einiger dieser Quellen sind hier gelistet:

- YAML von Dirk Jesse (<http://www.yaml.de/>), dazu gibt es auch ein Buch. YAML verwenden wir in einer Reihe von Designs als Grundlage.
- <http://edition-w3c.de/TR/1998/REC-CSS2-19980512/> als komplette Übersicht
- <http://www.css4you.de/> zum Nachschlagen bestimmter Einstellungen

Papoo Designs

Mit der Installation von Papoo haben Sie mindestens 2 Designs zur Verfügung auf die Sie aufbauen können. Die Pro und Business Versionen besitzen mindestens 16 Designs aus denen Sie auswählen können.

Um diese Designs nun anpassen zu können müssen wir die Struktur der Ausgabe von Inhalten aus Papoo einmal kurz erläutern.

Papoo verwendet als Template Engine Smarty, so dass die Ausgabe und die Verarbeitung von Inhalten getrennt ist. Innerhalb des HTML findet man dann nur noch die Smarty tags die durch geschweifte Klammern gekennzeichnet sind. Beispiel: {\$ausgabe}. Weitere Details dazu finden Sie im Kapitel Frontend.

Für jedes Layout kann nun ein eigener Template Satz verwendet werden. Dieser ist im Verzeichnis /templates zu finden. Als Basis nehmen wir hier das Standard Design.

Als Basis der normalen Inhalte wird immer dir __index.html eingebunden. (2 Unterstriche). Aus dieser Datei heraus werden alle anderen Inhalte eingebunden.

Diese Datei ist hier nun einmal dargestellt:

Der Kopfbereich

```
{foreach item=error from=$page_error}<h1>{$error}</h1>{/foreach}
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="{ $lang_short}"
lang="{ $lang_short}" dir="{ $lang_dir}">
<head>
<!-- Seiten Kopf, wird komplett über Admin realisiert-->
{include file="_kopf.utf8.html"}
</head>
```

Der Kopfbereich der Datei braucht nur dann verändert zu werden wenn man den Doctype anpassen möchte. Die erste Zeile gibt hier eine Fehlermeldung aus wenn Probleme im System bestehen, man sollte diese also nicht ändern. Die Zeile: `{include file="_kopf.utf8.html"}` bindet alle Metatags, CSS Anweisungen, Javaskripte etc. ein.

Der body tag

```
{include file="_body_kopf.utf8.html"}
```

Der body tag wird ebenfalls über eine Include Anweisung eingebunden, da je nach Einstellung des Systems einige onLoad Anweisungne durchgeführt werden. Durch die Auslagerung kann man an dieser Stelle dann nichts falsch machen.

Der eigentliche Inhalt

Der eigentliche Inhalt von Papoo wird in 5 Blöcken realisiert. Diese 5 Blöcke findet man auch im Modulmanger wieder:

- Kopfbereich
- Links
- Rechts
- Mitte
- Fuß

Die einzelnen Module wie z.B. Menü, Hauptinhalt, Sprachauswahl, Kalender, Überschrift der Seite, Schrifvergrößerung etc. können frei zugeordnet werden.

Im Template findet man diese 5 Bereiche als Include Anweisungen wieder:

- **Kopfbereich** -> `{include file="_head.html"}`
- **Links** -> `{include file="_inhalt_links.html"}`
- **Rechts** -> `{include file="_inhalt_rechts.html"}`
- **Mitte** -> `{include file="_inhalt_mitte.html"}`
- **Fuß** -> `{include file="_fuss.html"}`

Diese Blöcke besitzen keine Layout Struktur, diese muß in der `__index.html` drumherum gelegt werden wie wir das hier sehen:

Inhalt – Kopfbereich

```
<div id="page_margins">
    <div id="page" class="hold_floats">

        <!-- Kopfbereich -->
        <div id="head">
            <div id="head_content" class="clearfix">
                {include file="_head.html"}
            </div> <!-- ENDE head_content -->
        </div> <!-- ENDE head -->
    <!-- Hauptbereich -->
    <div id="main">
```

Hier wird der Kopfbereich in die Seite mit div Blöcken eingebunden. Die Bezeichnung lehnt sich hier an die Bezeichnung der YAML Struktur an.

Inhalt Links

```
<!-- Inhalte in der linken Spalte -->
{if $spalte_links OR 1}
    <div id="coll">
        <div id="coll_content" class="clearfix">
            {include file="_inhalt_links.html"}
        </div> <!-- ENDE coll_content -->
    </div> <!-- ENDE coll -->
{/if}
```

Hier wird die linke Spalte in die div Blöcke mit den ids coll und coll_content eingebunden.

Inhalt rechts

```
<!-- Inhalte der rechten Spalte-->
{if $spalte_rechts OR 1}
    <div id="col2">
        <div id="col2_content" class="clearfix">
            {include file="_inhalt_rechts.html"}
        </div> <!-- ENDE col2_content -->
    </div> <!-- ENDE col2 -->
{/if}
```

Hier wird der Inhalt des rechten Blocks in die div Container col2 und col2_content eingebunden.

Inhalte Mitte

```
<!-- Hauptinhaltsbereich in der Mitte - Content !!-->
{if $spalte_mitte OR 1}
    <div id="col3">
        <div id="col3_content" class="clearfix">
            {include file="_inhalt_mitte.html"}
        </div> <!-- ENDE col3_content -->
        <!-- Clearing für IE -->
        <div id="ie_clearing">.</div>
    </div> <!-- ENDE col3 -->
{/if}
```

Hier wird die Mitte eingebunden in die div Container col3 und col3_content. Am Ende befindet sich noch ein sogenanntes «Clearing» div mit dem das Float Verhalten im Internet Explorer korrigiert wird. Die meisten fließenden Designs haben ein entsprechendes div integriert.

Inhalt Fuß

```
</div> <!-- ENDE main -->
    <div id="footer" class="floatbox">
        {include file="_fuss.html"}
    </div> <!-- ENDE footer -->
</div> <!-- ENDE page -->
</div> <!-- ENDE page_margins -->
```

Der Fußbereich wird ebenfalls in einen div Container eingebunden mit der id «footer».

Seite schließen

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

Die Seite wird hier mit den schließenden tags geschlossen.

Der Aufbau der div Blöcke

Die div Blöcke sind in dem oben beschriebenen HTML nach dem Muster des YAML CSS Frameworks von Dirk Jesse aufgebaut. Natürlich kann man für die Strukturierung auch andere Bezeichnungen nutzen.

Mit dieser Strukturierung kann man herkömmliche CSS Templates nutzen und in diese die Seitenstruktur von Papoo einbinden.

CSS Dateien speichern

Die CSS Dateien speichern Sie im Verzeichnis /css, wobei für jedes eigene Stylesheet ein eigenes Verzeichnis erstellt werden muß. Wenn Sie also ein eigenes CSS einbinden wollen erstellen Sie im Verzeichnis /css ein neues Verzeichnis z.B. mystyle.

Für die korrekte Einbindung sind einige Details zu beachten.

Copyright

Wenn Sie in den Fuß der Seite automatisch ein Copyright einbinden wollen, erstellen Sie eine Datei mit Namen cc.txt. Dort tragen Sie dann Ihren Copyright ein, dabei kann auch HTML genutzt werden.

Beispiel:

```
(c) 2007 Sitenam.com. Design by <a href="http://www.freecsstemplates.org/">  
Free CSS Templates</a>.
```

HTML Template

Wenn Sie ein eigenes HTML Template nutzen wollen, erstellen Sie eine Datei template.txt. Tragen Sie dort den Namen des Template Verzeichnisses ein. Dieses Verzeichnis sollte natürlich auch im Verzeichnis /templates vorhanden sein.

Module einbinden – css.xml

Bei der Installation eines neuen Stylesheets werden automatisch auch die Module zugeordnet. Um eine entsprechende Vorlage zu nutzen kopieren Sie einfach eine entsprechende css.xml Datei aus dem Verzeichnis eines anderen CSS Layouts in Ihr Verzeichnis.

Diese Datei ist unbedingt notwendig, da Sie ansonsten keine Module zuweisen können.

Sie können diese Datei später selber erstellen, indem Sie im Modulmanger nach Auswahl des Styls auf den Link css.cmls Datei erstellen klicken. Speichern Sie diese Datei dann manuell in Ihrem CSS Verzeichnis. So wird bei jeder

späteren Installation des Styles die Einstellung übernommen die Sie eingetragen haben.

Einbindung in das System

Um nun ein eigenes CSS Design in das System einzubinden gehen Sie in die Administration und klicken Sie dort auf System, dann auf CSS Layouts. Sie sehen dort

eine zweigeteilte Liste, zum einem die Styles die schon eingebunden sind und im Styleswitcher im Frontend zur Verfügung stehen, zum anderen die Styles die noch nicht eingebunden sind.

Um nun einen Style zu aktivieren und ihn einzubinden klicken Sie einfach auf den Namen des entsprechende Styles, hier (Abbildung 1) z.B. einfach auf sun oder auf blue. Durch den Klick wird das Style eingebunden. Um diesen Style wieder zu löschen klicken Sie auf den Namen in der oberen Liste, sie landen dann in der Bearbeitungsmaske. Dort können Sie den Style entweder zum Standard machen oder wieder löschen.



Beim löschen wird nur der Eintrag aus der Datenbank entfernt, die Dateien werden nicht gelöscht.

In der Bearbeitungsmaske können Sie die einzelnen Dateien auch aus der Administration heraus bearbeiten und verändern. Um die Daten allerdings speichern zu können brauchen die Dateien Schreibrechte, am besten 777.

Wenn Sie diese Rechte nicht einstellen, können Änderungen nicht gespeichert werden und Sie erhalten bei dem Versuch eine Fehlermeldung.

Außerdem können Sie hier auch den Templatesatz ändern/einstellen den Sie mit diesem CSS nutzen wollen.

Weitere Details zur Layout Gestaltung erfahren Sie in den folgenden Kapiteln.

Dateien auswählen von Style: LGBLue

Klicken Sie auf den Namen der Datei um sie zu bearbeiten.

- [css.xml](#) Dieser Style ist Standard-Style.
- [IEFixes_7_index.css](#) Dieser Style ist Standard-Style.
- [IEFixes_6_index.css](#) Dieser Style ist Standard-Style.
- [basemod.css](#) Dieser Style ist Standard-Style.
- [IEFixes_5_5_index.css](#) Dieser Style ist Standard-Style.
- [index.css](#) Dieser Style ist Standard-Style.
- [schriften_und_farben.css](#) Dieser Style ist Standard-Style.
- [base.css](#) Dieser Style ist Standard-Style.
- [print.css](#) Dieser Style ist Standard-Style.

Name des Styles

Bezeichnung

Wählen Sie den Template Satz

Templatesatz

Bezeichnung ändern

[Diesen Style zum Standard machen.](#)

Diesen Style löschen

Klicken Sie hier um den Style sofort zu löschen!!
Achtung Wenn Sie auf den Link klicken wird der Style gelöscht.
[Style löschen](#)

Abbildung 2: CSS Layout - Einstellungen eines Styles