



FinishWeb 3

Layout Erstellung



Informationen zum Dokument

Versionierung

| Version | Datum | Status | Änderungen und Bemerkungen | Autor |
|---------|----------|---------|---|-------------------|
| 0.1.0 | 01.09.06 | Draft | Erste Version | Bernd Zumoberhaus |
| 0.2.0 | 06.10.06 | Draft | Zweite Version | Tamara Zeiter |
| 0.2.1 | 12.10.06 | Release | Korrekturen und Release | Tamara Zeiter |
| 1.0 | 01.11.06 | Update | Änderungen und Release | Ralf Zenklusen |
| 1.1 | 06.11.06 | Update | Erklärungen zu den Layout-Dock's | Bernd Zumoberhaus |
| 1.2 | 14.11.06 | Update | Korrekturen | Bernd Zumoberhaus |
| 1.3 | 24.01.07 | Update | FAQ | Ralf Zenklusen |
| 1.4 | 30.01.07 | Update | Docks | Claudia Müller |
| 1.5 | 10.04.07 | Update | Docks | Ralf Zenklusen |
| 1.6 | 08.01.08 | Update | Language | Tamara Zeiter |
| 1.7 | 21.01.09 | Update | „Umfang der Layout-Designstudie“ und „Designangaben und Grafiken“ erweitert | Bernd Zumoberhaus |
| 1.8 | 22.01.09 | Update | „HTML-Layout-Datei anpassen“ erweitert | Bernd Zumoberhaus |
| 1.9 | 25.11.09 | Update | Ergänzungen | Bernd Zumoberhaus |
| 2.0 | 20.08.13 | Update | Überarbeitung | Bernd Zumoberhaus |
| 2.5 | 20.07.16 | Update | Komplette Überarbeitung | Bernd Zumoberhaus |

Freigabe

| Name, Kurzzeichen | Rolle, Aufgabe | Datum, Visum |
|-----------------------|--------------------|----------------|
| Bernd Zumoberhaus, BZ | Internet & Verkauf | 20.07.2016, BZ |

Verteiler

| Name, Kurzzeichen | Kontakt | Funktion |
|-------------------|---------|----------|
| Öffentlich | | |

Zusatzinformationen

| | |
|----------------|--|
| Dateiname: | K:\Bar Informatik\Internet Doku\FinishWEB\FinishWeb 3\Dokumentation\Layout\Layout Erstellung.doc |
| Speicherdatum: | 20 Juli, 2016 15:15:00 |



Inhalt

| | | |
|----------|---|---|
| 1 | Einführung | 4 |
| 1.1 | Begriffe..... | 4 |
| | <i>Layout-Beispiel</i> | 5 |
| 1.2 | Layout-Assistent | Fehler! Textmarke nicht definiert. |
| 1.3 | Individuelles Layout | 5 |
| 2 | Erstellung einer individuellen Layout-Designstudie | 6 |
| 2.1 | Art der Layout-Designstudie | 6 |
| 2.2 | Umfang der Layout-Designstudie | 7 |
| 2.3 | Designangaben und Grafiken | 13 |
| 3 | Individuelles Layout für FinishWeb aufbereiten | 14 |
| 3.1 | HTML- und CSS-Dateien | 14 |
| 3.2 | Zusätzliche Arbeiten | 15 |
| 4 | Individuelles Layout in FinishWeb einfügen | 16 |
| 4.1 | HTML-Layout-Datei layout.html Grundgerüst..... | 16 |
| 4.2 | HTML-Layout-Datei anpassen..... | 17 |
| | <i>Allgemeines zu den Docks</i> | 17 |
| | <i>Dynamischer Inhalt (vom Menü zugewiesen)</i> | 18 |
| | <i>Fixer Inhalt (vom Layout zugewiesen)</i> | 18 |
| | <i>Dynamischer Inhalt (Menü)</i> | 18 |
| | <i>Javascript und Bilder im Layout</i> | 22 |
| 4.3 | Konfigurations-Datei layout-config.html | 22 |
| 4.4 | CSS-Datei style.css Grundgerüst und Anpassung..... | 23 |
| 4.5 | Abschluss..... | 26 |
| 4.6 | Zusatz-Layout mit Abweichungen..... | 26 |
| 5 | FAQ | 27 |
| 5.1 | Geht – geht nicht..... | 27 |



1 Einführung

Unter Layout versteht man alle Elemente, die das Aussehen einer Webseite bestimmen aber nicht deren Inhalt.

Zum Layout gehört also der Grundaufbau - was erscheint auf einer Webseite wo. Zum Layout gehört auch, wie etwas erscheint - in welcher Farbe, welche Schriftart oder Linienbreiten usw. Zusätzlich können zu einem Layout auch Text und Bilder gehören. Dieser Text und die Bilder erscheinen dann aber auf jeder Seite mit diesem Layout. Beispiele dafür sind ein Firmenlogo oder der Firmenname, die auf jeder Seite angezeigt werden sollen und daher direkt im Layout verankert werden.

FinishWeb kann mehrere Layouts verwalten und man kann Layouts je Menüpunkt zuweisen. Häufig wird nur ein Standard-Layout für die ganze Website benutzt. Wichtig ist aber, an ein Drucklayout zu denken.

1.1 Begriffe

| Begriff | Erklärung |
|--------------|---|
| Website | Alle Webseiten und Dateien eines Webauftrittes ergeben die Website. |
| Layout | Das Layout teilt die Webseite in Docks auf und bestimmt in welchem Dock welche Inhalte auf einer Webseite erscheinen. Das Layout wird in der Datei layout.html gespeichert. |
| Dock | Ein Platz im Layout an dem ein Inhalt (eine Inhaltsseite oder einzelnes Foto oder einzelner Text) angezeigt werden kann. Ein Dock entspricht einer Zelle einer Tabelle. |
| CSS | Mit der CSS-Datei wird das Aussehen (Farbe, Schriftart, Abstände usw.) aller Inhalte eines Layouts beschrieben. Die CSS-Daten werden in der Datei standard.css gespeichert. |
| Menü | Das Menü ist die Navigation einer Website. Über die einzelnen Menüeinträge sind die Webseiten aufrufbar. |
| Webseite | Eine einzelne Seite die beim Aufrufen eines Menüeintrages in Browser erscheint. Die Webseite besteht, je nach Layout, aus einer oder mehreren Inhaltsseiten und statischen Dingen wie Bilder und Text. |
| Inhaltsseite | Eine Inhaltsseite erscheint gemäss Layout an einem bestimmten Platz (Dock) einer Webseite. Die Inhaltsseite besteht aus Elementen. |
| Element | Elemente sind die kleinsten Inhaltsbestandteile mit denen Inhaltsseiten aufgebaut werden können. Beispiele dafür ist das Element „Bild“, „Text“ oder auch „Titel“. |
| Webserver | Ein Webserver führt Programme aus, die üblicherweise Ein- und Ausgaben über Webseiten machen. Der FinishWeb-Server ist ein solcher Server. Ebenso wie der Webserver auf dem die öffentlich zugänglichen Webseiten, die mit FinishWeb erstellt wurden, gespeichert sind. |

Layout-Beispiel

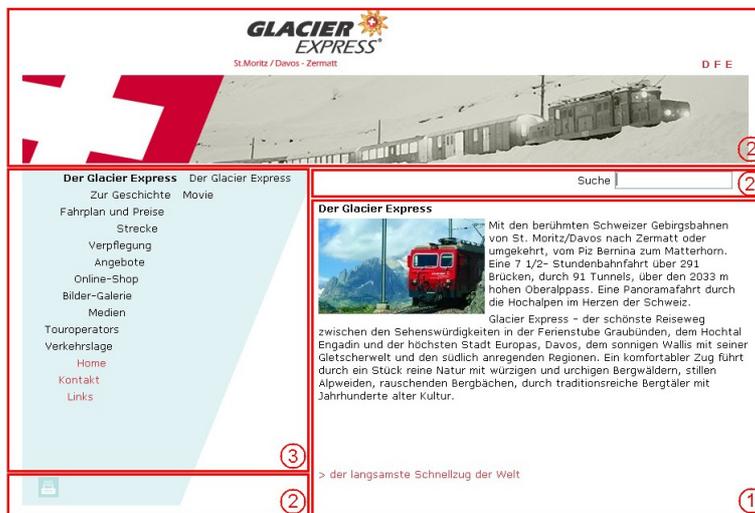


Abbildung 1-1 – Layout mit 5 Docks

Beschreibung

Layout mit 5 Docks

- 1 Dock mit Inhaltsseite, gefüllt mit verschiedenen Elementen. Dieses Dock erscheint bei jedem Menüpunkt und kann dort verändert werden.
- 2 Docks mit statischen Teilen, einzelnes Foto oder statische Inhaltsseite. Diese Teile erscheinen im Layout-Ordner und können dort verändert werden.
- 3 Dock mit Menü. Dies ändert automatisch, wenn der Menübaum verändert wird.

1.2 Individuelles Layout

Das Layout kann komplett individuell erstellt werden.

Die Erstellung eines Layouts läuft in drei Phasen ab:

- Erstellung einer Layout-Designstudie
- Individuelles Layout für FinishWeb aufbereiten
- Individuelles Layout in FinishWeb einfügen

Grundsätzlich kann man fast alle Design-Ideen umsetzen. Der Aufwand steigt aber mit der Anzahl der individuellen Details oder bei ausgefallenen Arrangements.



2 Erstellung einer individuellen Layout-Designstudie

Die Designstudie ist eine Veranschaulichung, für Kunde und Designer, wie die Webseite nach der Verwirklichung aussieht.

Möglicherweise müssen mehrere verschiedene Layouts erstellt werden - z.B. eines mit einer Inhalts-Spalte und eines mit einer zusätzlichen "Werbespalte" rechts. Häufig braucht man ein Drucklayout, das dann komplett anders aussehen kann als das Original-Layout.

Grundsätzlich müssen für die spätere Umsetzung alle notwendigen bzw. gewünschten Details sichtbar sein.

2.1 Art der Layout-Designstudie

Die Layout-Designstudie kann auf verschiedene Arten gemacht werden:

- 1 Mehrere hochqualitative PNG-Bilder der Website
- 2 Direkt entsprechende HTML Seiten
- 3 Website in PDFs inkl. Vermassung
und ein 1:1 Screenshot als hochqualitatives PNG-Bild
- 4 Photoshop-Vorlagen (bitte ohne Ebenen-Ordner)
- 5 Darstellungsvarianten für die jeweiligen Breakpoints (responsive Design). Es kann ein 12er- oder 24er-Grid-System verwendet werden. Das Bootstrap-Framework wird unterstützt.

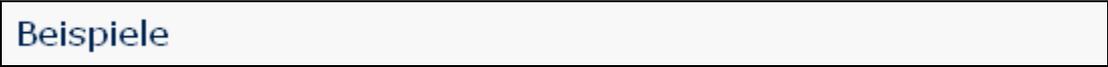
2.2 Umfang der Layout-Designstudie

Neben einer Frontseite sollten einige zusätzliche Folgeseiten erstellt werden, auf denen zumindest folgende Grundelemente, weitere Elemente und das Menü mit den entsprechenden Formatinformationen und Designangaben (siehe Punkt 2.3) ersichtlich sind.

Grundelemente

Maximal 6 verschiedene Aussehen bzw. Varianten pro Element definierbar.

- **Titel**

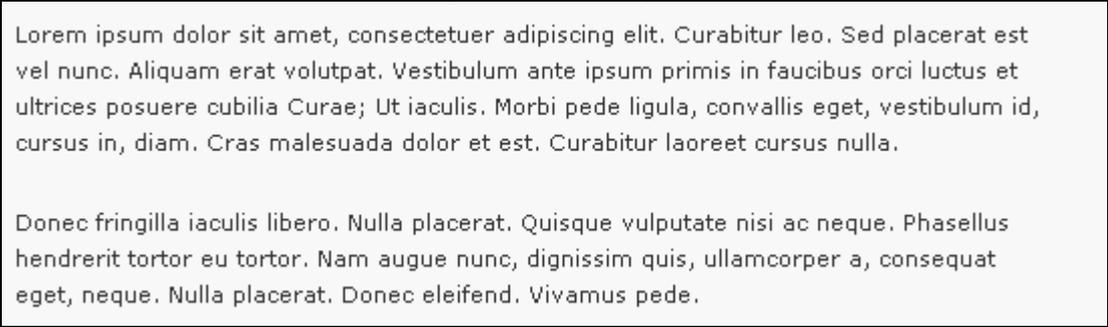


Beispiele

Abbildung – Titel

- **Text**

Normaler Text, Link im Text, Auflistung und Aufzählung



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur leo. Sed placerat est vel nunc. Aliquam erat volutpat. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Ut iaculis. Morbi pede ligula, convallis eget, vestibulum id, cursus in, diam. Cras malesuada dolor et est. Curabitur laoreet cursus nulla.

Donec fringilla iaculis libero. Nulla placerat. Quisque vulputate nisi ac neque. Phasellus hendrerit tortor eu tortor. Nam augue nunc, dignissim quis, ullamcorper a, consequat eget, neque. Nulla placerat. Donec eleifend. Vivamus pede.

Abbildung – Normaler Text mit zwei Abschnitten



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur leo. Sed placerat est vel nunc. Aliquam erat volutpat.

Abbildung – Formatvariante mit kursivem Text und einem Rahmen

- Auflistung
 - Punkt 2
 - Punkt 3
 -
1. Aufzählung
 2. Punkt 2
 3. Punkt 3
 4.

Abbildung – Auflistung und Aufzählung

• **Abschnitt**

Titel mit Text neben Bild, Bildausrichtung beidseitig oder einseitig auswählbar oder fix festlegbar, fixe Bildgrösse oder verschiedene wählbare Bildgrößen, Bildmaskierung

Abschnitt ohne Bild

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur leo. Sed placerat est vel nunc. Aliquam erat volutpat. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Ut iaculis. Morbi pede ligula, convallis eget, vestibulum id, cursus in, diam. Cras malesuada dolor et est. Curabitur laoreet cursus nulla.

Donec fringilla iaculis libero. Nulla placerat. Quisque vulputate nisi ac neque. Phasellus hendrerit tortor eu tortor. Nam augue nunc, dignissim quis, ullamcorper a, consequat eget, neque. Nulla placerat. Donec eleifend. Vivamus pede.

Abbildung – Abschnitt mit Titel ohne Bild

Abschnitt mit Bild links, Text umfliessend



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur leo. Sed placerat est vel nunc. Aliquam erat volutpat. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Ut iaculis. Morbi pede ligula, convallis eget, vestibulum id, cursus in, diam. Cras malesuada dolor et est. Curabitur laoreet cursus nulla.

Donec fringilla iaculis libero. Nulla placerat. Quisque vulputate nisi ac neque. Phasellus hendrerit tortor eu tortor. Nam augue nunc, dignissim quis, ullamcorper a, consequat eget, neque. Nulla placerat. Donec eleifend. Vivamus pede.

Abbildung – Abschnitt mit Titel, Bild links und umfliessenden Text



Abschnitt mit Bild rechts, Text nicht umfließend

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur leo. Sed placerat est vel nunc. Aliquam erat volutpat. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Ut iaculis. Morbi pede ligula, convallis eget, vestibulum id, cursus in, diam. Cras malesuada dolor et est. Curabitur laoreet cursus nulla.

Donec fringilla iaculis libero. Nulla placerat. Quisque vulputate nisi ac neque. Phasellus hendrerit tortor eu tortor. Nam augue nunc, dignissim quis, ullamcorper placerat. Donec eleifend. Vivamus pede.



Abbildung – Abschnitt mit Titel, Bild rechts und nicht umfließenden Text

- Tabelle**

Tabelle mit Titelzeile, einer beliebige Anzahl Spalten und Zeilen sowie einer Leerzeile, zwei alternierende Farben bei den Zeilen möglich, variable oder fixe Spaltenbreiten

| h1 | h2 | h3 |
|----|----|----|
| a | a | a |
| b | b | b |
| c | c | c |
| | | |
| d | d | d |

Abbildung – Tabelle mit einer Titelzeile (h), mehreren Spalten und Zeilen sowie einer Leerzeile (unter Zeile c) ohne alternierende Farben

- Linkliste**

Titel mit Auflistung von Links, zwei alternierende Farben bei den Links möglich

Linklisteüberschrift

- [BAR Informatik AG](#)
- [RhoneWeb](#)
- [FinishWeb](#)

Abbildung – Linkliste mit Titel und Links ohne alternierende Farben

- Link**

Eigenständiger Link mit Text und Icon sowie einer optionalen Bezeichnung

Bezeichnung:  [Alleinstehender Link](#)



- **Bild**

Bild mit optionaler Bildbeschreibung, fixe Bildgrösse oder verschiedene wählbare Bildgrössen, Bilder einzeln oder mehrere nebeneinander, Bildmaskierung



Abbildung – Bild mit Beschreibung

Weitere Elemente

- **Formular-zu-E-Mail**

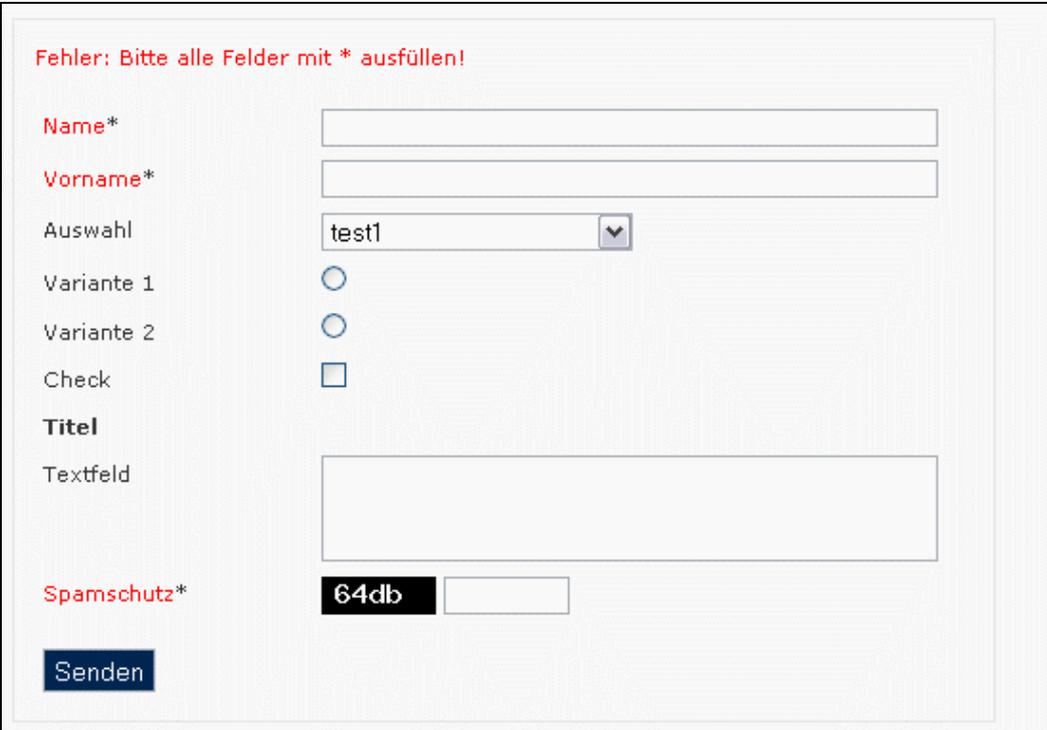
Frei konfigurierbares Formular-zu-E-Mail mit individuellen Feldern inkl. Spamschutz, definierbare Spaltenbreite



The screenshot shows a form with the following elements:

- Name***: Text input field
- Vorname***: Text input field
- Auswahl**: Dropdown menu with 'test1' selected
- Variante 1**: Radio button
- Variante 2**: Radio button
- Check**: Check box
- Titel**: Section header
- Textfeld**: Text input field
- Spamschutz***: A black box containing 'cb82' and an empty text input field
- Senden**: Blue button

Abbildung – Formular-zu-E-Mail Erstansicht



The screenshot shows the same form as above, but with a red error message at the top: **Fehler: Bitte alle Felder mit * ausfüllen!**. The labels for **Name***, **Vorname***, and **Spamschutz*** are highlighted in red. The 'Senden' button is still visible.

Abbildung – Formular-zu-E-Mail Ansicht bei unvollständig ausgefülltem Formular



Besten Dank für Ihre Mitteilung!

Abbildung – Formular-zu-E-Mail Ansicht bei erfolgreich verschicktem Formular

- **Dateiliste**

Zwei alternierende Farben bei den Dateieinträgen und frei definierbare Anzahl Dateien pro Seite möglich



Abbildung – Dateiliste mit einem Dateieintrag pro Seite und Navigation

- **Galerie**

Übersichtsseite mit Vorschaubildern, Anzahl Bilder (Spalten und Zeilen pro Seite), Navigationsdesign mit Pfeilen als Bild oder als Text, Detailbild auf separater Detailseite oder in Lightbox, Lightbox mit oder ohne Abdunklung des Hintergrunds, Lightbox mit oder ohne runden Ecken, Wasserzeichen

- **Teaser**

Grösse, Platzierung der Navigation / Pfeile, Inhalte der Teaserpanels

Menü

- **Menü und Untermenü**

Menüebenen über maximal 5 Bereiche auf der Website verteilbar (Bsp: Menüebene 1 oben horizontal, Menüebene 2-3 links und Menüebene 4 rechts). Jeweils separates Aussehen für ausgewählten und nicht ausgewählten Menüpunkt.

- **Untermenü im Inhalt**

Darstellung eines wählbaren Untermenüs im Inhalt, z.B. im Kacheldesign, mit oder ohne Bild.

Nicht verwenden

Diese Elemente sollte man im Layout grundsätzlich nicht verwenden:

- iFrame und Frames
- Bilder als Menüpunkt



2.3 Designangaben und Grafiken

Folgende Angaben muss man für den nächsten Schritt aus der Designstudie übernehmen können:

- Farben im HTML Hexformat
- Schriftart
 - Bei einer Speziialschrift zusätzlich die entsprechende Schrift als TTF oder PS (zur Bildgenerierung der entsprechenden Titel oder Menüpunkte) oder alternativ eine Web-Schrift zur Einbindung ins CSS.
 - Die korrekte Lizenzierung der Schrift ist Sache des Kunden.
- Schriftgrösse in Pixel
- Abstände und Abstandsgrößen in Pixel bei den Elementen und der kompletten Website
- Ränder und Randdicken, Randabstand in Pixel
- Links im Text inkl. Hover-Effekt
- Auflistung- und Aufzählung im Text
- Menüs inkl. Hover-Effekt und Aussehen, wenn ausgewählt
- Bildgrößen in Pixel:
 - Elemente Bild & Abschnitt
Maximal zehn verschiedene Bildgrößen und dazugehörige Bildmasken (z.B. runde Ecken über Bild) definiert werden.
Die Bildgrößen können fix oder variabel für die Breite und Höhe festgelegt und jeweils individuell benannt (z.B. Bild schmal, Bild breit, etc.) werden.
 - Element Galerie
Grösse für Vorschau- und Detailbild: Wenigstens ein Wert (Breite oder Höhe) muss fix sein, der andere Wert (Breite oder Höhe) kann variabel sein.
- Bildmasken als transparente GIFs
 - Sofern die Bilder mit Maskierungen versehen werden sollen (z.B. runde Ecken), müssen diese als transparente GIFs verfügbar sein.
- Benennung von allfälligen zusätzlichen Inhalten, z. B. Rechte Spalte, Werbespalte etc.
- Darstellungsvarianten für alle Breakpoints
- Darstellung der Off Canvas-Navigation (Mobile-Ansicht)

Alle Grafiken müssen zumindest in Webqualität (72dpi) vorliegen. Fotos sollten wenigstens in 1920x1080 Pixel (Full HD) aufgelöst sein.

Es ist von Vorteil wenn die Grafiken so vorliegen, dass sie direkt und ohne Änderungen in die Website eingebunden werden können. D.h. Qualität und Form (GIF, JPEG oder PNG) müssen stimmen.

Dies gilt auch für Schaltflächen und Knöpfe, sowie Grafik-Überschriften und Flash Animationen.



3 Individuelles Layout für FinishWeb aufbereiten

Die vorhandene Layout-Designstudie muss nun aufbereitet werden. Es müssen zwei, für jedes Layout zwingend vorhandene Dateien, erstellt werden. Falls nötig, müssen zudem die im Layout vorkommenden Grafiken beschnitten und aufbereitet werden.

3.1 HTML- und CSS- und/oder SCSS-Dateien

Für jedes Layout braucht es zwingend die folgenden Dateien:

1. HTML-Datei layout.html
Ein leeres Grundgerüst mit dem der grundsätzliche Seitenaufbau festgelegt wird. Also wohin z.B. ein Menü kommt oder wo der Inhalt angezeigt werden soll etc.
2. Konfigurations-Datei layout-config.html
Hier werden Standardwerte wie Seitentitel, CSS-Dateiname etc. festgelegt.
3. Basis CSS-Datei base.css im Ordner css
Hier ist die CSS-Basiskonfiguration der FinishWeb-Elemente abgelegt. Die Datei wird von uns zur Verfügung gestellt und darf nicht geändert werden.
4. CSS-Dateien style.css und print.css im Ordner css
 - a) Allgemeines
Die CSS-Datei style.css legt das genaue Aussehen jedes Details auf der Webseite fest. Je mehr Details wie Farben, Linienstärken, Schriften etc im Designstudio bei Punkt 1 definiert wurden, je aufwändiger wird die CSS-Datei.
 - b) Bearbeitung mit SCSS und Bootstrap-Framework (empfohlen)
FinishWeb enthält einen SCSS-Parser und erstellt daraus die Dateien style.css und print.css selbstständig. Die SCSS-Dateien sind im Ordner scss bereitgestellt. Das Bootstrap-Framework ist dort ebenso abgelegt und kann verwendet werden. Ein JS für die Off Canvas-Navigation in der Mobile-Ansicht steht auch zur Verfügung.
 - c) Bearbeitung mit CSS (nicht empfohlen)
Für sehr einfache und rudimentäre Websites können durchaus auch style.css und print.css direkt auf die konventionelle Art bearbeitet werden. Eine responsive Funktionalität und die Verwendung vom Bootstrap-Framework entfällt dann in der Regel.



3.2 Zusätzliche Arbeiten

Je nach Art der Designstudie gibt es verschiedene Arbeiten auszuführen:

- Sie benutzen die in der Designstudie erstellten Übersichts-Bilder der Website. Dort können nun die einzelnen Grafiken ausgeschnitten und die Farben, Abstände usw. ausgelesen werden. Dies ist jedoch nicht immer einfach und wegen den Einschränkungen von HTML Webseiten nicht immer zu 100% möglich.
- Wenn Sie in der Designstudie HTML Webseiten erstellt haben, so kann man diese nun direkt als Gerüst für das Layout verwenden. Die Seiten müssen dazu aber möglichst frei von automatisch generiertem HTML-Müll sein. HTML-Editoren wie Dreamweaver oder GoLive erzeugen recht gute Seiten. HTML-Seiten aus Exporten, wie z.B. von Word oder Photoshop, enthalten zu viele unerwünschte und unnötige Inhalte und müssen aufwändig gesäubert werden um ein optimales Resultat zu erhalten.

Am Ende enthalten die HTML- und die CSS-Dateien alle notwendigen Angaben. Zusätzlich stehen alle Layout-Grafiken zur Verfügung.



4 Individuelles Layout in FinishWeb einfügen

Im letzten Schritt gilt es nun, alles korrekt in FinishWeb einzufügen.

4.1 HTML-Layout-Datei layout.html Grundgerüst

Das nachfolgende HTML-Grundgerüst ist Pflicht für ein FinishWeb-Layout-Template. Im `<body>` kann dann das individuelle Seitengerüst integriert werden.

```
<!-- include layout-config.html -->
[ LAYOUT name="Standard" name_de="Standard" name_en="Standard" ]
[ format=php ]
<? header('Content-Type: text/html; charset=utf-8'); ?>
<!DOCTYPE html>
<html lang="[lang]">
  <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <meta name="robots" content="all" />
    <meta name="author" content="[head dc.creator]" />
    <meta name="publisher" content="FinishWeb Content Management System, rhone.ch GmbH und BAR Informatik AG" />
    <meta name="keywords" content="[head htmlkeywords]" />
    <meta name="description" content="[head description]" />
    <title>[doctitleprefix][doctitle][doctitlesuffix]</title>
    <link href="[layoutroot]favicon.ico" type="image/x-icon" rel="shortcut icon" />
    <link href="[layoutroot]<?=(( '[standardstyle]' != '' ) ? '[standardstyle]' : 'style.css' )?>" type="text/css" rel="stylesheet" media="all" />
    <?=(( '[standardprint]' != '' ) ? '<link href="[layoutroot][standardprint]" type="text/css" rel="stylesheet" media="print" />' : '' )?>
    <?=(( '[layoutspecificstyle]' != '' ) ? '<link href="[layoutroot][layoutspecificstyle]" type="text/css" rel="stylesheet" media="all" />' : '' )?>
    <?=(( '[layoutspecificprint]' != '' ) ? '<link href="[layoutroot][layoutspecificprint]" type="text/css" rel="stylesheet" media="print" />' : '' )?>
    <?=(( '[fwdomainspecificstyle]' != '' ) ? '<link href="[layoutroot][fwdomainspecificstyle]" type="text/css" rel="stylesheet" media="all" />' : '' )?>
```



```

    <?=(('[fwdomainsspecificprint]' != '')) ? '<link
href="[layoutroot][fwdomainsspecificprint]" type="text/css"
rel="stylesheet" media="print" />'. "\n" : '' )?>

    <?=(('[fwdomainlayoutspecificstyle]' != '')) ? '<link
href="[layoutroot][fwdomainlayoutspecificstyle]" type="text/css"
rel="stylesheet" media="all" />'. "\n" : '' )?>

    <?=(('[fwdomainlayoutspecificprint]' != '')) ? '<link
href="[layoutroot][fwdomainlayoutspecificprint]" type="text/css"
rel="stylesheet" media="print" />'. "\n" : '' )?>

    [STOP HEAD]

</head>
<body>
    <div class="wrapper">

        ...

    </div>

    <script src="[layoutroot]/js/doubletapgoto.min.js"></script>
    <script>
        $( function() {
            $( '#navigation li:has(ul)' ).doubleTapToGo();
        });
    </script>
</body>
</html>
<!-- include hr.html -->

```

4.2 HTML-Layout-Datei anpassen

In der neuen HTML-Layout-Datei layout.html wird nun das Seiten-Gerüst im <body> eingefügt. Im Gerüst selber werden die Platzhalter von FinishWeb eingefügt, die sogenannten Docks. Nachfolgend die wichtigsten Infos zu den Docks:

Allgemeines zu den Docks

Um an einem Ort des HTML-Gerüst einen Inhalt anzuzeigen, kann folgende Befehlsstruktur eingefügt werden. Einen solchen Inhalt - eine Art Fach - nennen wir Dock.

```
[STOP name="Inhaltsname, Standardsprache (Deutsch)"
name_de="Inhaltsname, Deutsch" name_en="Contentname, English"
docktype="xxxxxx"]
```

Mit dem Namen kennzeichnet man den Inhalt. Es gibt den Standard- den deutschen und den englischen Namen. Der Name ist frei wählbar, sollte aber der Einfachheit halber möglichst logisch gewählt werden.



Im Docktype wird der Typ des Docks festgelegt. Je nach dem Typ, kommt noch ein weiteres Feld „file“ dazu. Die zwei üblichsten „docktypes“ werden unten erklärt:

Dynamischer Inhalt (vom Menü zugewiesen)

```
[STOP name="Inhaltsname" name_de="Inhaltsname" name_en="Contentname"
docktype="content1"]
```

Die Namen sollten möglichst passend zum Inhalt gewählt werden. Beispielsweise bei einem zweiseitigen Layout „Inhalt links“ und „Inhalt rechts“.

Den Docktype legt man auf content1 fest. Damit weiss FinishWeb, dass es sich um einen dynamischen Inhalt handelt, der übers Menü angegeben werden kann.

Bei mehreren dynamischen Inhalten verwendet man fortlaufend „content1“, „content2“ usw. In der Menüverwaltung werden die Links in der entsprechenden Reihenfolge angezeigt. Die maximale Anzahl vom Menü zugewiesener Inhalte ist 5.

Fixer Inhalt (vom Layout zugewiesen)

```
[STOP name="Inhaltsname" name_de="Inhaltsname" name_en="Contentname"
docktype="fix" file="/Inhalte Layout/Fusszeile"]
```

Um einen fixen Inhalt dem Layout zuzuweisen, verwendet man den Docktype „fix“ zusammen mit „file“. Im „file“ gibt man den Pfad zur Inhaltsseite an. Bei weiteren Sprachen verwendet man zusätzliche file-Attribute mit dem Suffix xx, wobei das xx für das jeweilige Sprachkürzel steht. Beispiel: file_en (verweist auf eine englische Inhaltsseite).

Dynamischer Inhalt (Menü)

Das Menü wird ähnlich wie ein fixer Inhalt eingebunden. Mit dem Unterschied, dass man zusätzliche so genannte Levels für die einzelnen Menüebenen definiert. Ansonsten wird genauso ein Name definiert, der Docktype auf „fix“ festgelegt und im „file“ ein Verweis auf die Seite mit dem Menü erstellt.

Menülink

Der Menülink wird durch folgenden HTML-Code generiert:

```
[object currentitem][object currentitem atag]
```

Alternativ kann folgender HTML-Code verwendet werden, dieser bietet den Vorteil auch etwaige CSS-Klassen und ein title-Attribut direkt in den <a>-Tag einzubinden:

```
[object currentitem abeginopentag]" title="[object currentitem title]"
class="cssclass-name[object currentitem aendopentag][object
currentitem title][object currentitem aclosetag]
```



Menü mit einer Ebene

```

<nav class="nav" id="navigation" role="navigation">
<!-- MENU1 name="Menüname" name_de="Menüname" name_en="Menüname"
docktype="fix" file="/Inhalte/Layout/Standard/Menü 1"-->

    <!-- LEVEL1_ITEM -->
        <li class="menu1">[object currentitem][object currentitem atag]
    <!-- /LEVEL1_ITEM -->

    <!-- LEVEL1_ITEM_SEL -->
        <li class="menu1 menu1sel">[object currentitem][object currentitem
atag]
    <!-- /LEVEL1_ITEM_SEL -->

    <!-- LEVEL1_SEPARATOR_FIRST -->
        <ul class="menu1items">
    <!-- /LEVEL1_SEPARATOR_FIRST -->

    <!-- LEVEL1_ITEM_END -->
        </li>
    <!-- /LEVEL1_ITEM_END -->

    <!-- LEVEL1_SEPARATOR -->
        <li class="menuseparator1"></li>
    <!-- /LEVEL1_SEPARATOR -->

    <!-- LEVEL1_SEPARATOR_LAST -->
        <li class="clear"></li>
    </ul>
    <!-- /LEVEL1_SEPARATOR_LAST -->

<!-- /MENU1 -->
</nav>

```



Weitere Ebenen

Weitere Ebenen können mit Menü mittels entsprechenden Level-Angaben eingefügt werden:

Zum Beispiel:

```
<!-- LEVEL2_ITEM -->
```

Jede weitere Ebene sollte immer mit allen entsprechenden ITEM-Angaben integriert werden. D.h. ITEM, ITEM_SEL, ITEM_END, SEPARATOR_FIRST, SEPARATOR, SEPARATOR_LAST.

Menü-Separator

Will man zwischen den einzelnen Menüpunkten einen Separator platzieren, kann man diesen HTML-Kommentar (hier in diesem Beispiel mit der Ebene 1) verwenden:

```
<!-- LEVEL1_SEPARATOR -->
    <li class="menuseparator1"></li>
<!-- /LEVEL1_SEPARATOR -->
```

Alternativ lässt sich dies aber auch via CSS realisieren, was letztlich die technisch schönere Variante ist.

Sprachwahl

Um eine Sprachauswahl einzubinden, fügt man das Tag ‚LANGUAGE‘ ein.

Falls eine Sprachwahl mit Bildern realisiert wird, so müssen diese Bilder unter den Layout-Ressourcen mit den entsprechenden Sprachkürzeln abgespeichert werden. Im Layout-Template wird bei der Sprachwahl danach auf diese Bilder verwiesen.

Ein Beispielcode einer Sprachwahl mit Bildern sieht wie folgt aus:

```
<!-- LANGUAGE name="Sprachwahl" name_de="Sprachwahl"
name_en="Language" docktype="fix"
file="/Inhalte/Layout/Standard/Sprachwahl" -->
<ul class="langitems">

    <!-- ITEM -->
        <li class="lang">[object currentitem abeginopentag][object
currentitem aendopentag][object currentitem aclosetag]</li>
    <!-- /ITEM -->

    <!-- SEPARATOR -->
        <li class="separator"></li>
    <!-- /SEPARATOR -->

    <!-- ITEM_SELF -->
        <li class="lang langsel">>[object currentitem
abeginopentag][object currentitem aendopentag][object currentitem
aclosetag]</li>
  <!-- /ITEM_SELF -->
  <li class="clear"></li>
</ul>
<!-- /LANGUAGE -->
```

Ein Beispielcode einer Sprachwahl mit Text sieht wie folgt aus:

```
<!-- LANGUAGE name="Sprachwahl" name_de="Sprachwahl"
name_en="Language" docktype="fix"
file="/Inhalte/Layout/Standard/Sprachwahl" -->
<ul class="langitems">

  <!-- ITEM -->
  <li class="lang">[object currentitem abeginopentag][object
currentitem aendopentag][object currentlang][object currentitem
aclosetag]</li>
  <!-- /ITEM -->

  <!-- SEPARATOR -->
  <li class="separator"></li>
  <!-- /SEPARATOR -->

  <!-- ITEM_SELF -->
  <li class="lang langsel">[object currentitem abeginopentag][object
currentitem aendopentag][object currentlang][object currentitem
aclosetag]</li>
  <!-- /ITEM_SELF -->

  <li class="clear"></li>
</ul>
<!-- /LANGUAGE -->
```



Javascript und Bilder im Layout

Javascript einbinden

Wird für das Layout ein Javascript benötigt, z.B. für Menü-Animationen, sollte das komplette Javascript in eine separate Datei layout.js im Ordner js ausgelagert werden.

Die Einbindung in die HTML-Layout-Datei layout.html erfolgt mit <head> mit folgender Anweisung:

```
<script type="text/javascript"
src="[layoutroot]js/layout.js"></script>
```

Bilder einbinden

Werden Bilder direkt in die HTML-Layout-Datei layout.html eingebunden, so verwendet man im src-Attribut des vor dem Dateinamen des Bildes den Platzhalter [layoutroot]. Die Bilder selbst sind im Ordner images abzulegen. Beispiel:

```

```

Der Platzhalter [layoutroot] wird von FinishWeb automatisch durch den korrekten Layout-Pfad ersetzt.

4.3 Konfigurations-Datei layout-config.html

Hier werden Standardwerte wie Seitentitel, Kurztext- oder Wortplatzhalter fürs Layout, CSS-Dateiname etc. festgelegt.



4.4 CSS-Datei style.css oder style.scss Grundgerüst und Anpassung

Das nachfolgende CSS-Grundgerüst ist Pflicht für ein FinishWeb-Layout-Template.

Im ersten Abschnitt sollten die jeweilige Seiten-URL, die Firma sowie die Autoren eingetragen werden. Unter Changes ist sinnvollerweise ein Logfile über die Änderungen zu führen, jeweils mit Datum, dem Autorenkürzel sowie der ausgeführten Arbeit.

```
/**
 * Custom Stylesheet for FinishWeb Site www.yourdomain.com
 *
 * Copyright:
 * BAR Informatik AG
 * Brig-Glis, Switzerland
 *
 * Authors:
 * vn Vorname Name, Firma
 *
 * Changes:
 * DD.MM.YYYY      vn      initial version
 */
```

Nur bei direkter Bearbeitung der CSS-Datei (ohne SCSS)

Nach dem Abschnitt „Layout-styling“ sollten die CSS-Angaben zur Grundstruktur hinterlegt werden. Die CSS-Angaben zu den Elementen von FinishWeb bzw. den Inhalten der Seite sollten unter dem Abschnitt „Layout-defined styling of Finishweb elements“ folgen.

```
/******
 * Import base stylesheet for FinishWeb sites: base.css
*****/
@import url("base.css");
```



```

/*****
* Default styling of HTML-tags
*****/

* {
  margin: 0;
  padding: 0;
}

html {
  height: 101%;
}

body {
  font-family: Arial Regular, sans-serif;
  font-size: 70%;
  line-height: 1.6em;
  margin-top:0;
  color:#6F7171;
}

/*****
* Custom classes
*****/

/* Floating */
.float-left { float: left; }
.float-right { float: right; }
.clear { clear: both; font-size:0px; height:0px; line-height:0px;
display:block; }
.clear-left { clear: left; }
.clear-right { clear: right; }

/*****
* Default styling of FinishWeb elements
*****/

.fwelement { }

```



```
/* fwtitle, h1, h2, h3... */

/* color, line-height */
h1,
h2,
h3,
.fwelement h1,
.fwelement h2,
.fwelement h3 { }

/* font-size, font-weight */
h1,
.fwelement h1 { }
h2,
.fwelement h2 { }
h3,
.fwelement h3 { }

/* fw...

...

*/

/*****
* Layout-styling
*****/

/* Seiten-Layout */

/*****
* Layout-defined styling of Finishweb elements
*****/

/* Elemente */
```



4.5 Abschluss

Alle erstellten Dateien können nun im gewünschten Layout im Ordner "Layout" hochgeladen werden. Jetzt kann man dieses Layout in jedem Menüpunkt auswählen. Es ist auch direkt benutzbar und z.B. in der Schnellvorschau sichtbar.

4.6 Zusatz-Layout mit Abweichungen

Wird ein zusätzliches Layout mit minimalen Abweichungen erstellt, z.B. einspaltig (sofern das Haupt-Layout beispielsweise mehrspaltig ist), kann das erstellte Layout kopiert werden, wobei lediglich die Dateien layout.html, hr.html und der Ordner scss mit den SCSS-Dateien style.scss und print.scss kopiert werden sollten.

Die Links zu den jeweiligen Ressourcen (CSS und JS) sollten im neuen layout.html manuell auf das andere Layout gesetzt werden, damit die CSS (bzw. SCSS) und JS-Dateien sich nur in einem Ordner befinden. Selbiges gilt für die verlinkten weiteren SCSS-Dateien im style.scss und print.scss.

Die Abweichung fürs zweite Layout wird in einem weiteren CSS/SCSS definiert. Dessen Dateiname heisst z.B. style-einspaltig.css oder .scss.

Auf dieses zusätzliche CSS muss in der Konfigurationsdatei layout-config.html verwiesen werden (layoutspecificstyle).



5 FAQ

Hier einige Fragen und Antworten zur Layouterstellung.

5.1 Geht – geht nicht

Was kann nicht realisiert werden:

- Direktes öffnen eines Popup oder einer anderen externen URL (Webseiten-Adresse) vom Menü aus geht nicht. Ein direktes Öffnen eines Popups oder einer anderen externen URL von der Seite aus, ist mit dem Element „Weiterleitung“ möglich.
- Frames werden nicht unterstützt.

Kann der Seitenaufbau beliebig sein?

Grundsätzlich ja. Das Layout definiert dabei wo feste Inhalte (Bilder, Texte, fixe Inhaltsseiten) vorkommen, wo welche Menüebenen erscheinen bzw. wo Inhaltsseiten erscheinen.

Können Menüebenen an verschiedenen Orten sein und verschieden aussehen?

Ja, jede Menüebene kann individuell formatiert und platziert werden.

Können Menüeinträge aus einem Bild bestehen?

Ja, dies muss dann aber pro Menüeintrag manuell erstellt und eingefügt werden. Es können jeweils zwei Bilder angegeben werden: Menüeintrag inaktiv bzw. aktiv. Wir empfehlen dies nicht, da es für den FinishWeb-Benutzer schwierig ist weitere Menüeinträge zu erstellen.

Können Menüeinträge zusätzlich Bilder enthalten?

Ja, man kann jede Menüebene individuell mit CSS formatieren und zusätzlich Bilder anzeigen lassen. Z.B. ein Pfeil vor dem Menüeintragstext.

Können in Menüs und Titeln spezielle Schriften benutzt werden?

Ja, Menüeinträge und Titel können aus beliebigen TTF- oder PS-Schriften erstellt werden. Die Schriftdateien müssen dazu ins FinishWeb geladen werden und das Menü sowie der Titel entsprechend konfiguriert werden. Liegt die Schrift weder als TTF noch als PS vor, so muss die Schrift vom Kunden gekauft werden.

Ebenso kann eine Webschrift zur Einbindung ins CSS genutzt werden. Etwaige Lizenzkosten seitens der Webschrift-Anbieter sind von Kundenseite zu tragen.

Sind dynamische Menüs möglich, die sich z.B. bei einem "Mouse-Over" öffnen?

Ja, dies kann mittels CSS einfach realisiert werden. Fadeln-Animationen von Menüebenen können CSS oder Javascript ins Layout integriert werden.

Kann pro Menüebene oder Menüeintrag das Aussehen der Seite ändern?

Ja, pro Menüeintrag kann ein anderes Layout angegeben werden. Die Layoutwahl wird an Untermenüs vererbt.

Die erlaubte Anzahl Layouts hängt von der abonnierten FinishWeb-Version ab.

Kann der Benutzer gewisse Dinge in einem Layout später selber ändern. Z.B. ein Bild?

Ja, man kann dem Benutzer dies erlauben. Diese Inhalte sollten als fixe Inhaltsseiten im Layout erscheinen. Diese Inhaltsseiten kann der Benutzer danach sehr einfach ändern.



Müssen alle Untermenüs immer angezeigt werden.

Nein, man kann auch nur das aktuelle Untermenü eines Hauptmenüeintrags anzeigen lassen. Es können auch alle Untermenüs immer offen sein.

Können so genannte Quicklinks als eigenständiges Menü (zusätzlich zum eigentlichen Hauptmenü) auf der Website realisiert werden?

Ja, das ist möglich.