

7 Layout mit <DIV>-Containern: Das 960 Grid System

Worum es geht

Fast alle Webseiten verfügen über ein Layout, das über die simple Gliederung durch den Textfluss (von links nach rechts und von oben nach unten) hinaus geht: Die Inhalte werden gezielt im Browserfenster platziert. Technisch wird die Implementierung des Layouts mit CSS realisiert, wobei eine ungemeine Vielzahl von Techniken und Tricks gibt. Im Rahmen des Kurses können wir aus Zeitgründen nur einige von vielen möglichen Wegen aufzeigen, wie man zu einem eigenen Layout kommt. Wir benutzen dazu als erstes das 960-Grid System, welches bei vielen WebdesignerInnen durchaus beliebt ist und dir nach einer kurzen Einarbeitungszeit ermöglicht, mittels <DIV>-Containern komplexe Layouts zu erstellen. Auf die Gestaltung «responsiver» Layouts (mittels `display: flex;`) gehen wir später ein. Du wirst nach dem Durcharbeiten dieses Kapitels folgende Techniken beherrschen:

- Funktionsweise des 960 Grid System
- Gestalten eigener Seiten anhand des 960 Grid Systems
- Umsetzen eigener Seiten anhand des 960 Grid-Systems

Material und Referenzen

<http://webdesign.kanti-baden.ch/vorlagen/960> (an unsere Bedürfnisse angepasste Version des 960.gs)

<http://www.960.gs> (Zuhause des originalen 960 Grid Systems)

Kapitel 4.8 und Kapitel 4.9 im vorliegenden Kurshandbuch

sung zur Verfügung, die auf den Namen «960_12_ohne_stege.css» hört.

Innerhalb dieser Datei findest du mehrere Klassen definiert, die du benötigst um ein auf 12 Spalten basierendes Layout zu entwickeln. Neben der `.container_12`-Klasse (welche du der DIV-Box zuweisen musst, welche die eigentlichen Inhalt-DIVs enthält) sind das die Klassen für die einzelnen Spalten («`.grid_1`» - «`.grid_12`» für 1 bis 12 Spalten Breite).

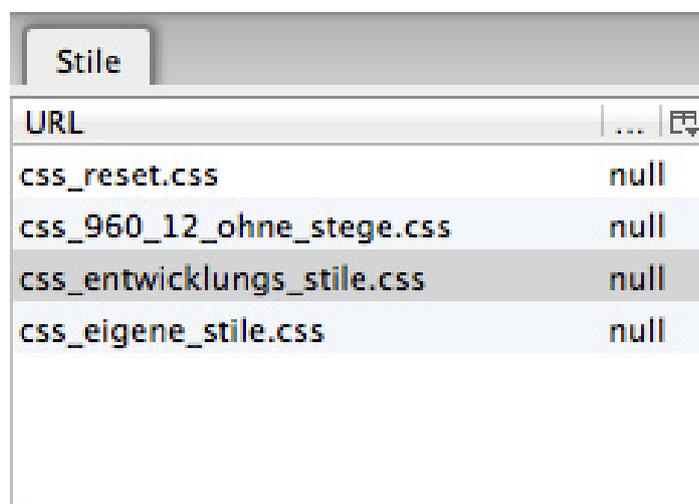
Mittels der Klassen «`.prefix_X`» kannst du eine Box um X Spalten nach rechts verschieben. Mit den Klassen «`.suffix_X`» kannst du rechts von Boxen Platz frei halten.

Ausserdem beinhaltet die Datei «960_12_ohne_stege.css» noch den schon im Kapitel 4.9.5 erwähnten Clearfix Hack («`.clearfix`»).

Neben der «960_12_ohne_stege.css»-Datei beinhaltet unsere Vorlage auch die zur CSS-Standardausstattung gehörende Datei «`reset.css`», die von Eric Meyer zur Vereinheitlichung des Browserverhaltens entwickelt wurde (siehe: <http://meyerweb.com/eric/tools/css/reset>).

Es empfiehlt sich weder «`reset.css`» noch «960_12_ohne_stege.css» zu berühren, bevor du nicht genau weisst, was du tust. Hingegen kannst du die übrigen CSS-Dateien durchaus verändern.

Die Datei «`entwicklungs_stile.css`» ist nur so lange vonnöten, wie du dein eigenes Layout noch nicht fertig gestellt hast. Sie sorgt dafür, dass deine DIV-Boxen einen sichtbaren Rand haben dass alle Textabsätze grau hinterlegt sind. Du kannst diese Datei löschen, sobald du deine eigenen Angaben in der Datei «`eigene_stile.css`» definiert hast.



URL	
<code>css_reset.css</code>	null
<code>css_960_12_ohne_stege.css</code>	null
<code>css_entwicklungs_stile.css</code>	null
<code>css_eigene_stile.css</code>	null

Abbildung 7.2: Die verschiedenen CSS-Dateien im verschlankten 960 Grid System.

6. Weise jeder weiteren DIV-Box die nötigen Klassen zu, die ihre Breite und Position bestimmen sollen.

Also z.B.

`class="grid_4"` für eine 4 Spalten breite Box.

`class="grid_5 prefix_3"` für eine Box, die 5 Spalten breit ist und links von sich Platz von drei Spalten ausspart.

`class="grid_1 suffix_2"` für eine box, die 1 Spalte breit ist und rechts von sich Platz von zwei Spalten ausspart.

7. Weise jeder DIV-Box eine eindeutige ID zu, z.B. `id="nav"` für den Menübereich. Diese IDs wirst Du benötigen, um die Formatierung dieser Box in der CSS-Datei «eigene_stile.css» vornehmen zu können.

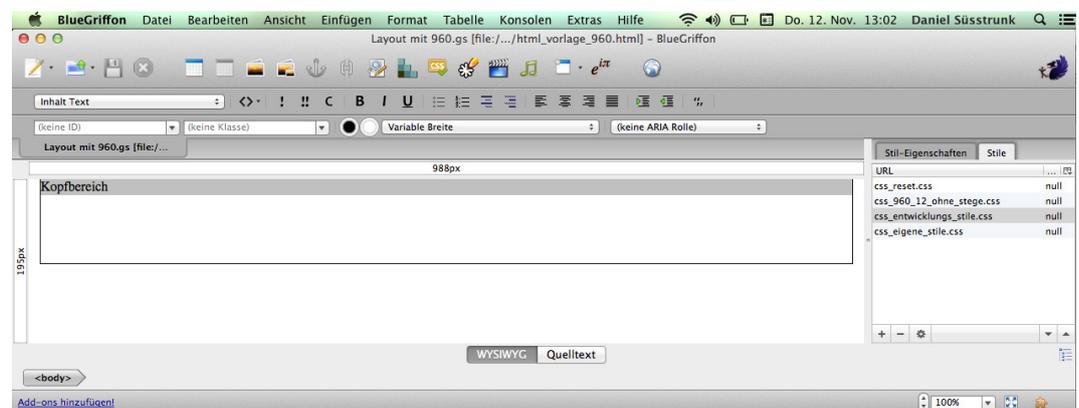


Abbildung 7.4: WYSIWYG-Ansicht des Quellcodes unter Abbildung 7.3

