



# HTML und CSS im Griff

Vorstellung nützlicher Testtools und Einführung in CSS-Frameworks  
Beispiele: 960.gs + YAML



# Testtools

Beispiele:

- Firebug
- Web Developer Toolbar
- YSlow

= Add-ons für Firefox

# Firebug

bietet u.a.

- Überblick HTML-Struktur (DOM)
- zeigt die eingebundenen CSS-Dateien an
- gibt beim Fokussieren eines HTML-Tags die verbundenen CSS-Deklarationen an, inkl. Zeilennummer in der CSS-Datei
- die Einflüsse von CSS-Angaben untereinander (Kaskade/Vererbung) werden sichtbar

# Web Developer Toolbar

-> sehr mächtige Werkzeugleiste

bietet u.a.

- Validierung (HTML + CSS) über die W3C-Seiten
- CSS-Dateien anzeigen, editieren, ausblenden, alternatives CSS anzeigen
- div. Programmelemente deaktivieren
- Formular- und Grafikkontrolle

# YSlow

Ergänzung zu Firebug (Yahoo-Tool)

bietet u.a.

- Performance-Tests
- Statistiken zu den Ladenzeiten der diversen Seitenkomponenten (HTML, CSS, Bilder, Skripte)
- macht Vorschläge zur Verbesserung der Performance
- enthält weitere Tools wie z.B. Smush.it(TM)

# CSS-Frameworks

## bekannte CSS-Probleme

- IE-Browser
- Routinearbeiten
- Codeschnipsel
- Wiederverwendbarkeit



**Sind Frameworks  
die Lösung?**



## Was ist ein CSS-Framework?

- Gestaltungsrahmen oder Raster (Grid)
- Sammlung von Basisvorlagen in (X)HTML- und CSS
- für bekannte Browser-Probleme gibt es Alternativen (Workarounds)

## Unterschiedliche Ansätze

- Fixe Seitenbreite
- feste Rastervorgaben (Grid)
- Elastische Seitenbreite (fluid)
- Spaltenraster (Columns)

## Pro

- viele Browser-Probleme gelöst
- weniger Entwicklungszeit
- Grundlagen bereits vorhanden

## Kontra

- Starres Layout
- Aufgeblähter Quellcode

# Beispiel: 960.gs

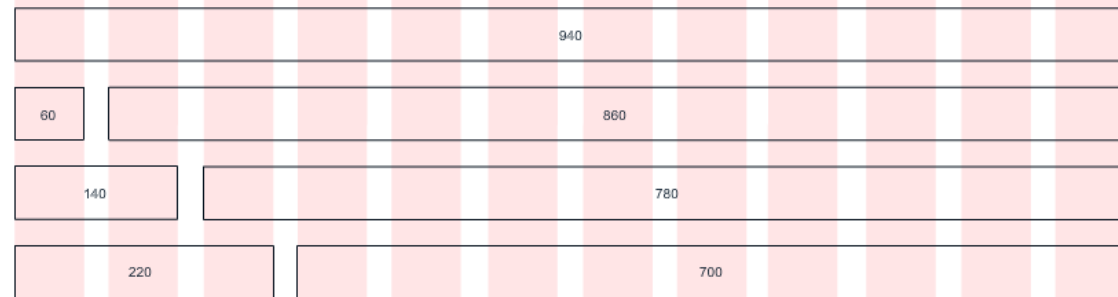
<http://960.gs>

# Basis

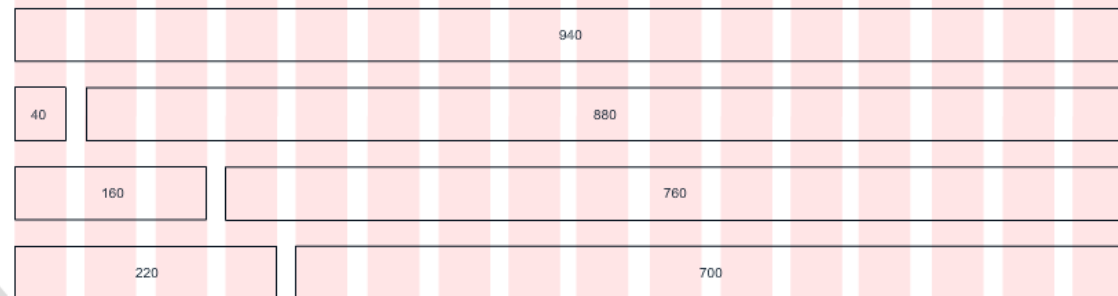
- Raster mit 960 px Breite
- Spalten mit fixen Rasterweiten (12 Spalten = 60 px, 16 Spalten = 40 px)
- Spaltenabstände von 10 px nach rechts und links erzeugen einen äusseren Rand von 10 px und innere Abstände von je 20 px
- Standard-CSS für Browser-Reset, Textformate und Spalten-Raster
- Clearfix für A-Grade-Browser (IE6/7, Firefox 2.+ , Safari 3.+ , Opera 9.+)

# Funktionsweise

## 12 Column Grid



## 16 Column Grid



Quelle: <http://960.gs/demo.html>

# CSS-Beispiel

## Beispiel 12er Raster

- linke Spalte (220 px) + rechte Spalte (700 px)

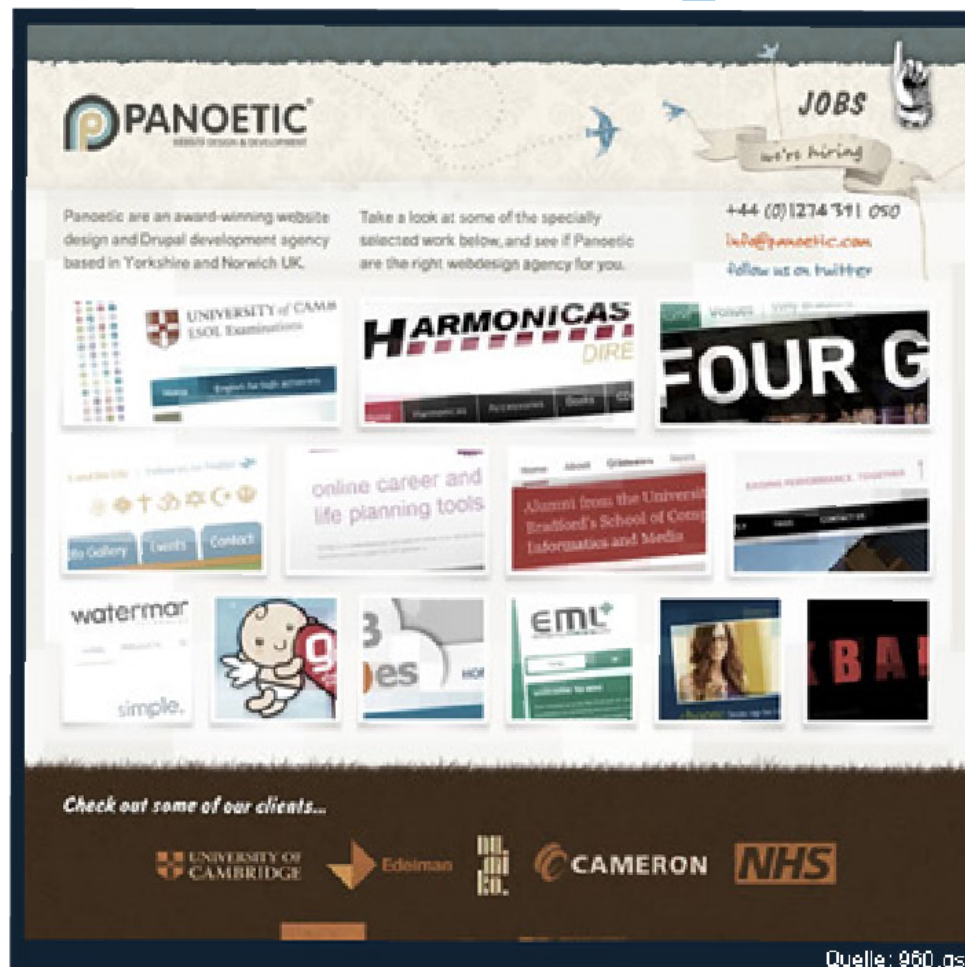
```
/* Hauptcontainer */
.container_12 {
margin-left: auto;
margin-right: auto;
width: 960px;
}

/* Floating und Abstände für alle Spalten (Auszug) */
.grid_3, .grid_9 {
display: inline;
float: left;
margin-left: 10px;
margin-right: 10px;
}

/* Container im 12er Grid (Auszug)*/
.container_12 .grid_3 {
width: 220px;
}
.container_12 .grid_9 {
width: 700px;
}

/* Variation für Hauptcontainer ohne Aussenabstand */
.alpha {
margin-left: 0;
}
.omega {
margin-right: 0;
}
```

# Seitenbeispiel





<http://960.gs/>

SSB.in

# Beispiel YAML

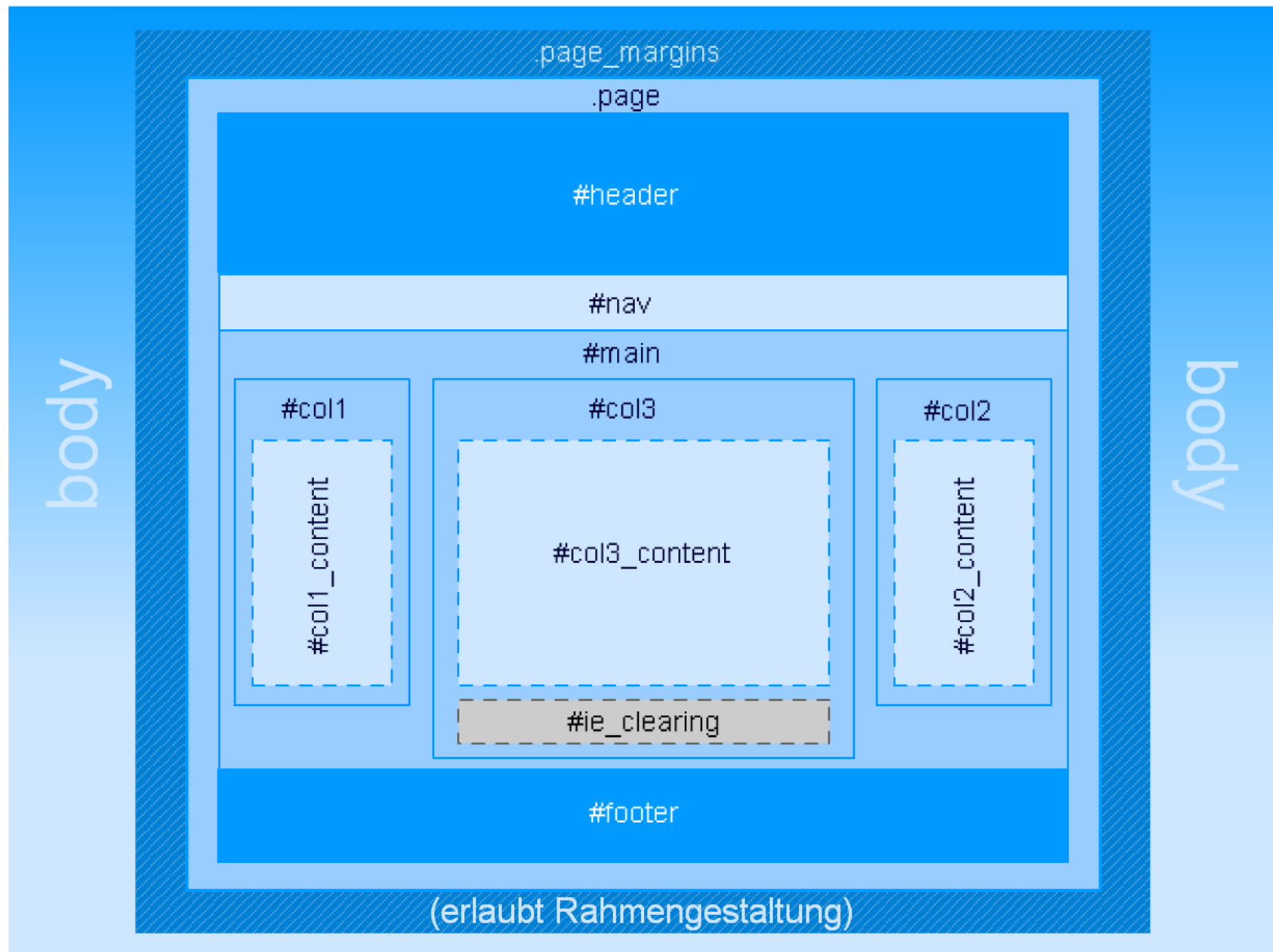
<http://www.yaml.de/filesadmin/examples/index.html>

# Basis

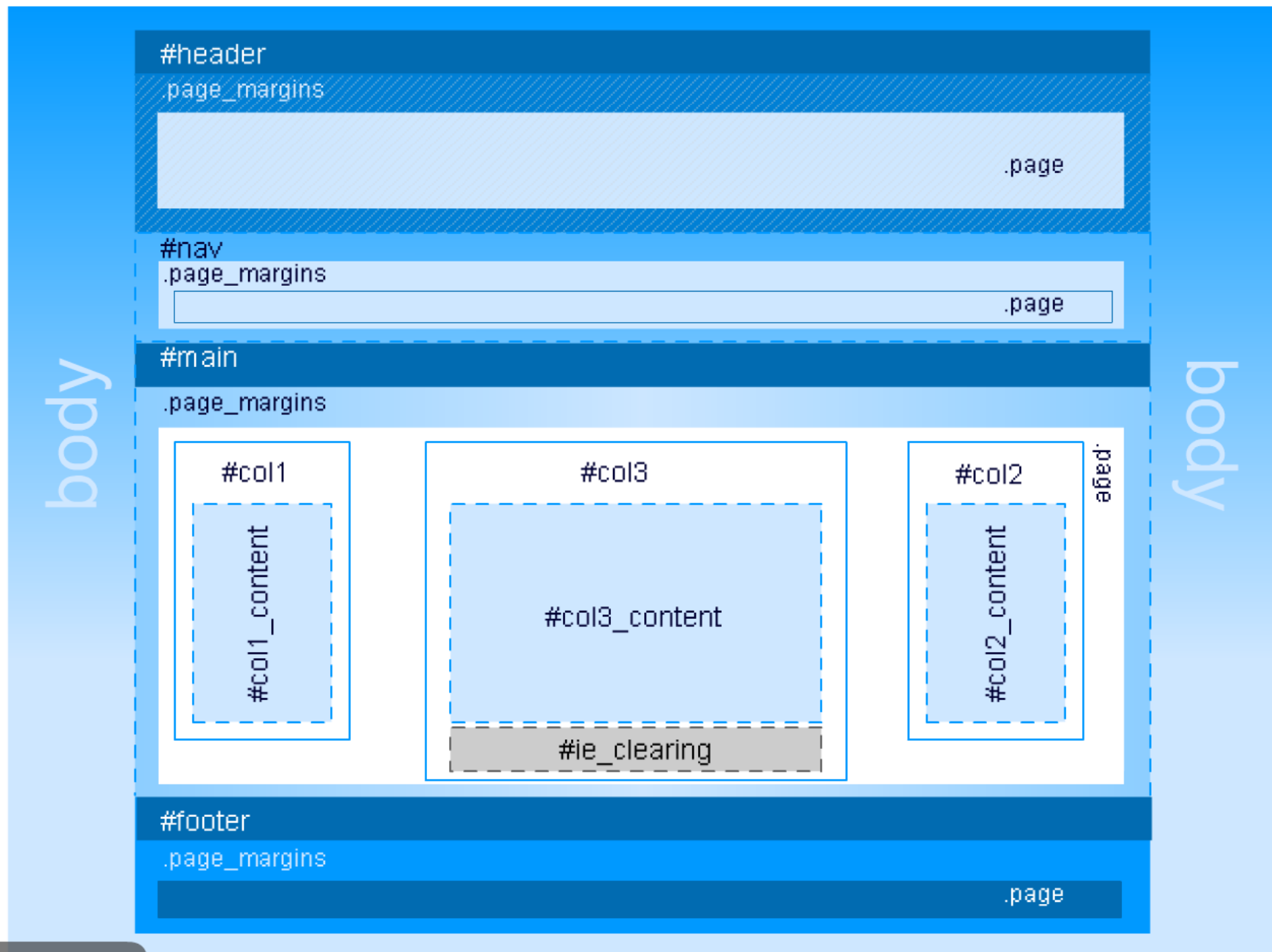
- Ganzheitliches Konzept
- TOP-DOWN Prinzip = Basiselemente sind vorhanden, können aber weggelassen oder überschrieben werden
- Grundlage: Floats = floatende Boxen oder Spalten
- macht Clearing notwendig (clear: both|left|right)

# Funktionsweise

# XHTML-Struktur



# Variation



# Erweiterung mit Subcolumns

.subcolumns = umgebende Klasse  
für die Spaltenaufteilung  
z.B. c50l + c50r

Mögliche Anordnung innerhalb der .subcolumns-Klasse

- 50% / 50% Teilung (Klassen c50l und c50r)
- 33% / 33% / 33% Teilung (Klassen c33l und c33r)
- 25% / 25% / 25% / 25% Teilung (Klassen c25l und c25r)
- 20% / 20% / 20% / 20% / 20% Teilung (Klassen c20l und c20r)

## Mögliche Aufteilung innerhalb der .subcolumns-Klasse

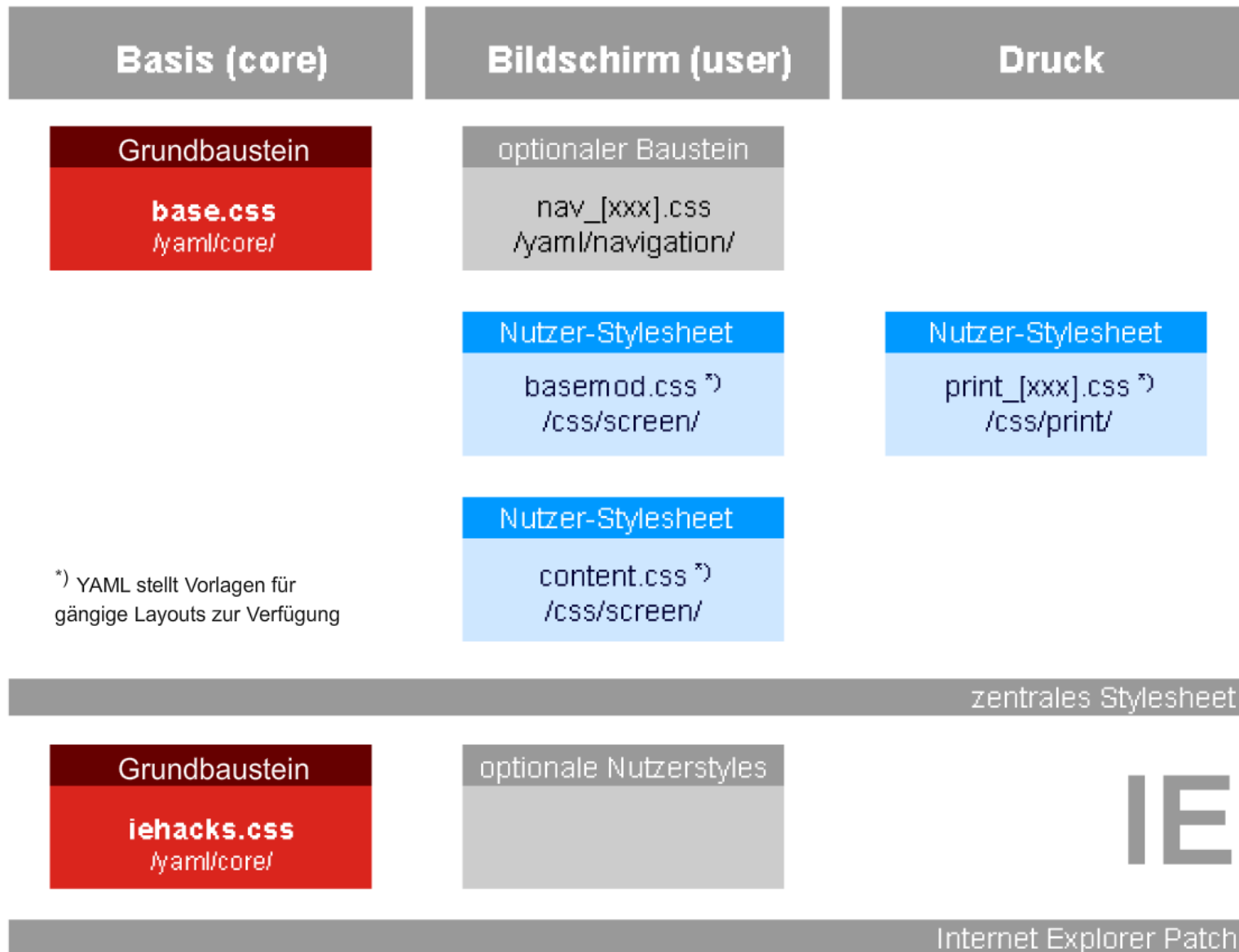
- 50% / 50% Teilung (Klassen c50l und c50r)
- 33% / 66% Teilung (Klassen c33l und c66r sowie c66l und c33r)
- 25% / 75% Teilung (Klassen c25l und c75r sowie c75l und c25r)
- Goldener Schnitt (Klassen c38l und c62r sowie c62l und c38r)



# Ergänzungen

u.a. fluide Breiten (auch für IE6/5)  
durch externes Javascript »minmax.js«

# CSS-Basis



# Clearing-Methode

`<div>` oder `<br />` mit `clear: both/left/right` wird ersetzt durch markupfreies Clearing

- Standard = `.clearfix`
- Alternativ = `.floatbox`

# Auszug aus /core/base.css (ohne IE)

```
/* (de) Clearfix-Methode zum Clearen der Float-Umgebungen */  
.clearfix:after {  
clear:both;  
content:".";  
display:block;  
font-size:0;  
height:0;  
visibility:hidden;  
}
```

```
/* (de) Diese Angabe benötigt der Safari-Browser zwingend !! */  
.clearfix { display:block; }
```

```
/* (de) Alternative Methode zum Einschließen von Float-Umgebungen */  
.floatbox { display:table; width:100%; }
```

```
/* (de) IE-Clearing:Benötigt nur der Internet Explorer, ist über iehacks.css  
zugeschaltet */  
#ie_clearing { display:none; }
```

# So sieht die CSS-Alternative für den IE aus: Auszug aus /core/ie\_hacks.css

```
* @affected IE 5.x/Win, IE6, IE7
* @css-for IE 5.x/Win, IE6, IE7
* @valid yes
*/

.clearfix { height:1%; } /* hasLayout aktivieren */

* @workaround
* @affected IE 5.x/Win, IE6, IE7
* @css-for IE 5.x/Win, IE6, IE7
* @valid no
*/

html #ie_clearing {
/* (de) Nur eine kleine Hilfe zur Fehlersuche */
position:static;

/* (de) Container sichtbar machen im IE */
display:block;

/* (de) Kein Fix im IE5.x möglich, daher normales Clearing */
\clear:both;

/* (de) IE-Clearing mit 100%-DIV für IE6 bzw. übergroßem Container im IE7 */
width:100%;
font-size:0px;
margin:-2px 0 -1em 1px;
}

* html #ie_clearing { margin:-2px 0 -1em 0; }
```

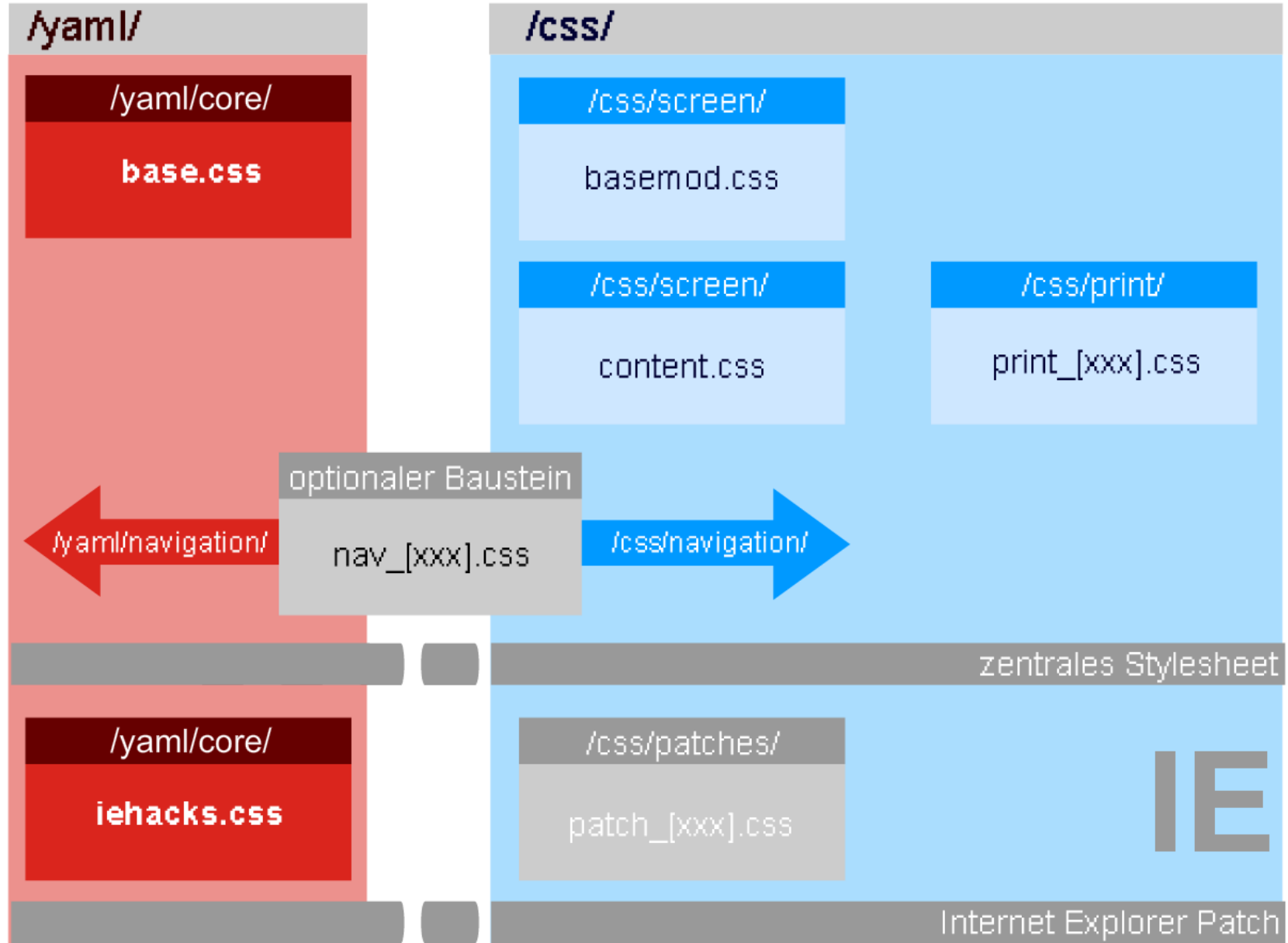
# Anwendung

User Struktur  

- Trennung von YAML-Core und User-CSS
- Standarddateien nur kopieren, Core unberührt lassen für evtl. YAML-Update

# Projektordner Nutzer

index.htm + sonstige Nutzerordner und -dateien



# YAML-Regeln

Regel 1: YAML ist kein Fertiglayout

Regel 2: YAML basiert auf dem TOP-DOWN-Prinzip

Regel 3: CSS-Grundbausteine

Regel 4: Trennung von YAML- und Nutzer-CSS

Regel 5: Viel Spaß mit YAML!



<http://www.yaml.de/fileadmin/examples/index.html>

**TINVA**

# Linktipps

## Frameworks:

- <http://www.yaml.de/> -> YAML-Homepage
- <http://builder.yaml.de> -> Online-Tool zum erstellen eigener YAML-Layouts (engl.)
- <http://www.t3net.de/dokumentation.htm> -> YAML gibt es für diverse CMS, u.a. Typo3, Joomla, Drupal
- <http://de.xtreme-theme.com/2010/11/xtreme-one-released/> -> xtreme One-theme für Wordpress, basiert auf YAML Hinweis: (kostenpflichtig)
- <http://developer.yahoo.com/yui/grids/> -> Homepage des Yahoo-Developer-Networks mit Basisdateien, YUI-Builder, Dokumentation etc.
- <http://960.gs/> -> System, HTML- + CSS-Genrator, Design-Hilfe
- <http://www.blueprintcss.org/> -> System, Tools, Tutorials
- <http://www.primarycss.com/> -> System und Online-Tool
- <http://grideasy.blogpimp.de> -> inkl. Design-Hilfe für Photoshop (deutsch)
- <http://www.zurb.com/playground/css-grid-builder> -> Online-Tool (schön einfach)
- <http://lwis.net/free-css-drop-down-menu/> -> Framework nur für CSS-basierte Menüs

## Testtools

- <http://getfirebug.com/> -> Firebug-Homepage
- <http://developer.yahoo.com/yslow/> -> Tool für die Webseiten-Analyse, benötigt Firebug
- <http://chrispederick.com/work/web-developer/help/> -> Entwicklerseite der Web-Developer-Browser-Extensions

## Tutorials:

- <http://net.tutsplus.com/articles/news/a-detailed-look-at-the-960-css-framework/>
- <http://net.tutsplus.com/tutorials/html-css-techniques/css-diy-organization/>  
-> Tutorial u.a. mit Hinweisen zu Namenskonventionen und strukturiertem CSS-Code

## Literatur

- <http://devsnippets.com/article/complete-guide-to-css-frameworks.html>  
Umfangreiche Übersicht zu CSS-Frameworks
- <http://www.smashingmagazine.com/2007/04/14/designing-with-grid-based-approach/> Beispiele und Know-How zu Grid-Design
- <http://www.webdesignerdepot.com/2009/05/the-960-grid-system-toolbox-and-resources/> Übersicht über diverse Frameworks
- [http://css-discuss.incutio.com/wiki/Three\\_Column\\_Layout](http://css-discuss.incutio.com/wiki/Three_Column_Layout) umfangreiche Übersicht zu 3-Spalten-Layouts
- <http://www.elmastudio.de/tipps-und-tools/css-grid-systeme-und-grid-generatoren/> Übersicht Grid-Generatoren und CSS-Frameworks
- <http://www.drweb.de/magazin/css-raster-generatoren-grid-generators/> Übersicht Online-Tools und Grid-Generatoren

# Vielen Dank für Ihre/Eure Aufmerksamkeit!

Kontakt: [r.hermanns@css-manufaktur.de](mailto:r.hermanns@css-manufaktur.de)

Twitter: [@renaade](https://twitter.com/renaade)