

# **RESPONSIVE DESIGN**

**1**

**MMWP2024 - LV03**

## INHALTSVERZEICHNIS

- Organisation
- Adaptive/Responsive Web Design
- Layout-Muster
- Navigations-Elemente
- Aktuelle Trends

## ÄNDERUNG DER BELEGSTRUKTUR

- 4 Belege als Prüfungsvorleistung für die schriftliche Prüfung (120min)
- 4. Beleg kann aber auch als Prüfungsprojekt gewählt werden (wenn vorher alle 3 anderen Belege erfolgreich absolviert wurden)
- Note für das Prüfungsprojekt vor Ende des Semesters
- Falls Sie mit der Note nicht einstanden sind, können Sie auch an der schriftlichen Prüfung teilnehmen  
**(diese Note zählt dann!)**

## INHALTSSCHWERPUNKTE

- Gerätearten und Darstellung
- Adaptive Web Design
- Responsive Web Design
- Layout-Muster
- Seitennavigation

## VORAUSSETZUNG

Der Ausgangspunkt dieser Vorlesungsreihe ist das Wissen über HTML-CSS-Layouting

- Fixed-Box layout
- Flex-Box layout
- Grid-Layout
- Responsive CSS-Regeln

## ZIELE

- Dynamische Bildbreiten bei Webdesign
- Vorstellung von Webdesigns auf Basis von responsive Layouting
- Definition von Layoutmuster
- Erklärung von Navigationsmustern und Navigationselemente

# AKTUELLE BILDSCHIRMGRÖSSEN (2023-2024)

## Aktuelle Trends



Quelle

## PROBLEM VON BILDGRÖSSEN

- Pixeldichte auf Geräten unterschiedlich (relative Größe ungleich absoluter Größe)
- Hohe Auflösungen auf kleinen Mobilgeräten
- Absolute Einheiten wie cm, mm, px, in, pt = (1/72in)
- Px ist relativ zum jeweiligen Gerät und Abhängig vom Gerät
- Absolute Angaben wie Pixel können somit zu Fehlerhaften darstellungen führen

# PX = PIXEL ?

Aktuelle Trends

## Device Pixel Ratio

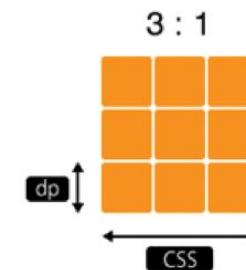
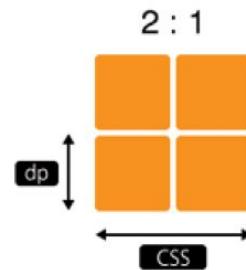
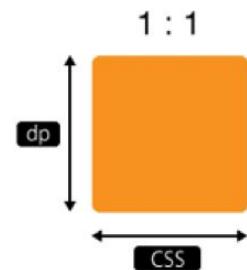
dpr=1.0



dpr=2.0 (e.g Moto G)



dpr=3.0 (e.g iPhone 6 Plus)



Quelle

## VERSCHIEDENE DARSTELLUNG

- Die Device-Pixel-Ratio gibt an, wie viele Hardware-Pixel das jeweilige Display verwendet
- Pixelanzahl in Höhe und Breite hängt von Hardware ab (Hardware-Pixel)
- Ausmaße des physikalischen Bildschirms in einer Längeneinheit → daraus resultiert die Entfernung zum Auge und die notwendige Schriftgröße
- Software-Pixel: je physikalischer Längen- und Höheneinheit
- Software-Pixel ungleich Hardware-Pixel

## NUTZUNG VON VIEWPORT FÜR RELATIVE GRÖSSEN - 1

- Viewport ist der für den Benutzer sichtbare Bereich einer Webseite
- Der Viewport hängt vom jeweiligen Gerät ab und ist auf einem Mobiltelefon kleiner als auf einem Computerbildschirm
- Möglichkeit der Verwendung von CSS-Media-Queries zur Anwendung unterschiedlicher Stile für kleine und große Bildschirme

## NUTZUNG VON VIEWPORT FÜR RELATIVE GRÖSSEN - 2

```
<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no">
```

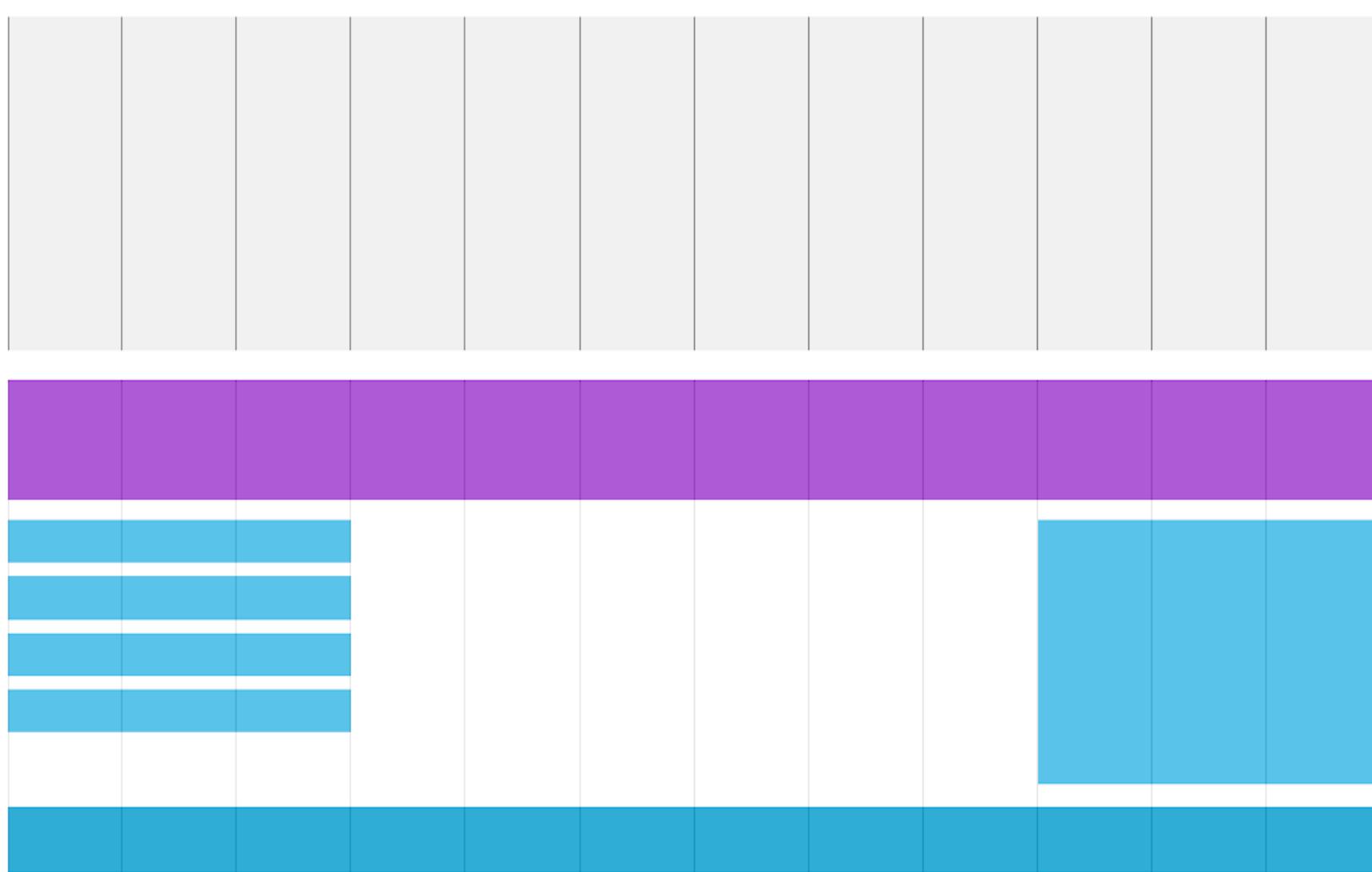
- width=device-width legt die Breite der Seite so fest, dass sie der Bildschirmbreite des Geräts entspricht
- Initial-scale=1.0 legt die anfängliche Zoomstufe fest, wenn die Seite zum ersten Mal vom Browser geladen wird

## VIEWPORTS ERMÖGLICHEN GRID-WEB-DESIGN

- Der Ansatz des Grid Designs ist es, die Webseite senkrecht in eine fixe Anzahl von „Streifen“ mit „Lücken“ zu teilen
- Je nach horizontaler Raumaufteilung werden Zonen der Webseite zu verschiedenen div-Boxen nebeneinander zusammenzufassen
- Viele Webseiten basieren auf dem Grid-Design
- Bootstrap benutzt bis zu 12 Spalten für die Aufteilung von Elementen (1/12 von Container breite)
- **Grid-View in der Browserconsole**

Aktuelle Trends

# GRID-VIEW



Quelle



## BEISPIEL GRID-DESIGNS

- [960.gs](#)
- [CSS-Grid-examples](#)
- [Grid-Website-templates](#)

## FLUID GRID DESIGN

- Grids sollten sich der Bildbreite anpassen
- Notwendigkeit, diese Grids in relativen Größen (Prozenten) planen
- Dabei wird die max -width festgelegt, da es immer höher auflösende, breitere Bildschirme gibt
- Auch Textverläufe, Schriftgrößen, Bilder, Videos mit ihren (div-)Boxen müssen sich mit der Größe der Webseite proportional ändern, zumindest in gewissen Grenzen

## FLUID GRID FRAMEWORKS

- Bootstrap 5 Breakpoints
- W3.CSS
- Vorgefertigte CSS-Klassen
- Foundation Framework
- Less-Framework

## ADAPTIVE WEB DESIGN

- Der Begriff wurde von Aaron Gustafson in seinem Buch 2011 geprägt
- Grundidee ist die Auswahl eines festen Sets von Bildschirmgrößen, die dann zu Schranken für Designwechsel der Website werden
- Fluid Grid Frameworks basieren auf diese Idee
- Verschiedene Breakpoints zur festlegung von Gridgrößen
- Most Used Responsive Breakpoints

## RESPONSIVE WEB DESIGN

- Der Begriff geht auf Ethan Marcotte und seinen [Artikel](#) vom 25.05.2010 zurück.
- Um das Aussehen der Webseite dem Anzeigegerät anzupassen, werden CSS3, Media Queries, die @media-Regel und Fluid Grids unter Benutzung von Prozentmaßen benutzt
- Ergänzend wird mit Maßen in em bzw. rem, flexiblen Bildern, flexiblen Videos und flexibler Schriftgröße gearbeitet

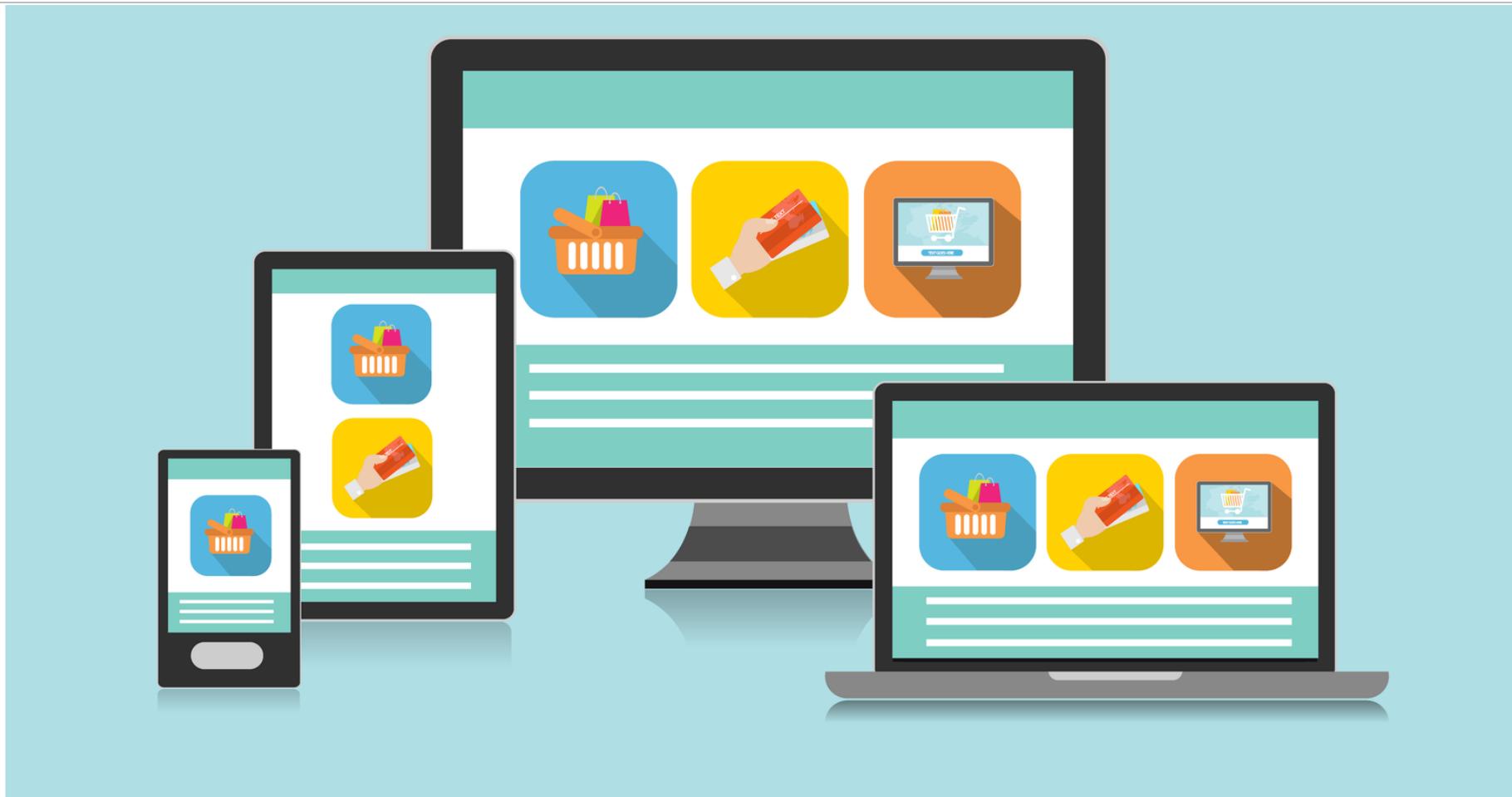
## MEDIA QUERIES

Media Queries ist eine Technik, die seit CSS3 eingeführt wurde, um mithilfe von @media – Regeln Eigenschaften nur für den Fall, dass die Bedingung erfüllt ist, festzulegen

```
@media only screen and (max-width: 600px) {  
    body {  
        background-color: lightblue;  
    }  
}
```

# MEDIA QUERIES

Aktuelle Trends



Quelle

## PRINZIPIEN VON RESPONSIVE WEB-DESIGN - 1

- Content First, Navigation Second
- Eine Art des Herangehens an das Design von Websites, die die Flächenaufteilung der Webseiten und die inhaltlich-semantisch gesehenen Rollen (Landmark Roles) der Teile der Webseiten in den Vordergrund stellt
- Eine Art des Designs, die alle adressierten Größen und Auflösungen von Bildschirmen gleich mitdenkt (und nicht erst ein Design fertigstellt, bevor der Adoptionsversuch gemacht wird)

## PRINZIPIEN VON RESPONSIVE WEB-DESIGN - 2

- Mobile First und Progressive Enhancement
- Die Entwicklung wird mit dem Design für kleine Bildschirme begonnen
- Die Funktionalität einer Webseite wird schrittweise an die Fähigkeiten größerer Bildschirme und der Browser angepasst
- Es wird in der Reihenfolge der Schichten HTML (mit strukturiertem Inhalt), CSS (Darstellungen) und JavaScript (Verhalten bei Interaktion, in der Zeit) vorgegangen

## PRINZIPIEN VON RESPONSIVE WEB-DESIGN - 3

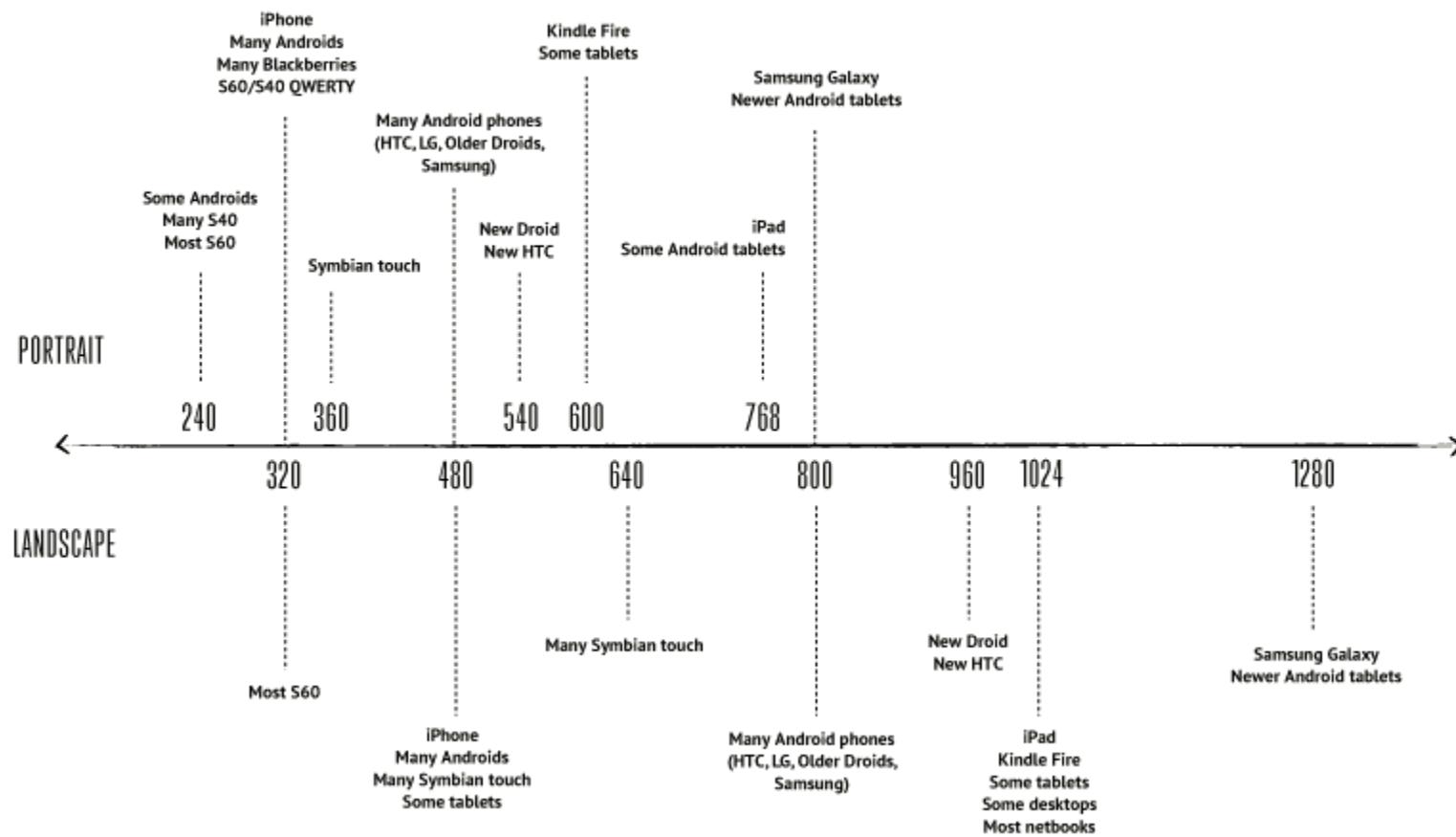
- Desktop First und Graceful Degradation
- Zuerst wird das Design für den größten gedachten Bildschirm konzipiert, danach wird stufenweise das Design für kleinere Bildschirmgrößen und -auflösungen durch „Abspecken“ der Inhalte und Umsortieren des Designs geschaffen
- Alte Webseiten können so renoviert werden, viele Webdesigner werden bei ihren traditionell erarbeiteten Techniken abgeholt
- Tendenz Potenziale auszuschöpfen macht Reduktion schwer

## PRINZIPIEN VON RESPONSIVE WEB-DESIGN - 4

- Für welche Bildschirm- und Browserauflösungen wird gestaltet?
- Alle Bildschirme ohne Format- und Auflösungsbeschränkungen.-max-width wird auf 960px (oder 920px / 1024px oder 1440px) gesetzt, die minimale Größe sind ältere Smartphones hochkant
- min –width wird gesetzt, um nicht jedes alte Gerät noch bedienen zu müssen
- Breakpoints nach Notwendigkeiten des Designs

Aktuelle Trends

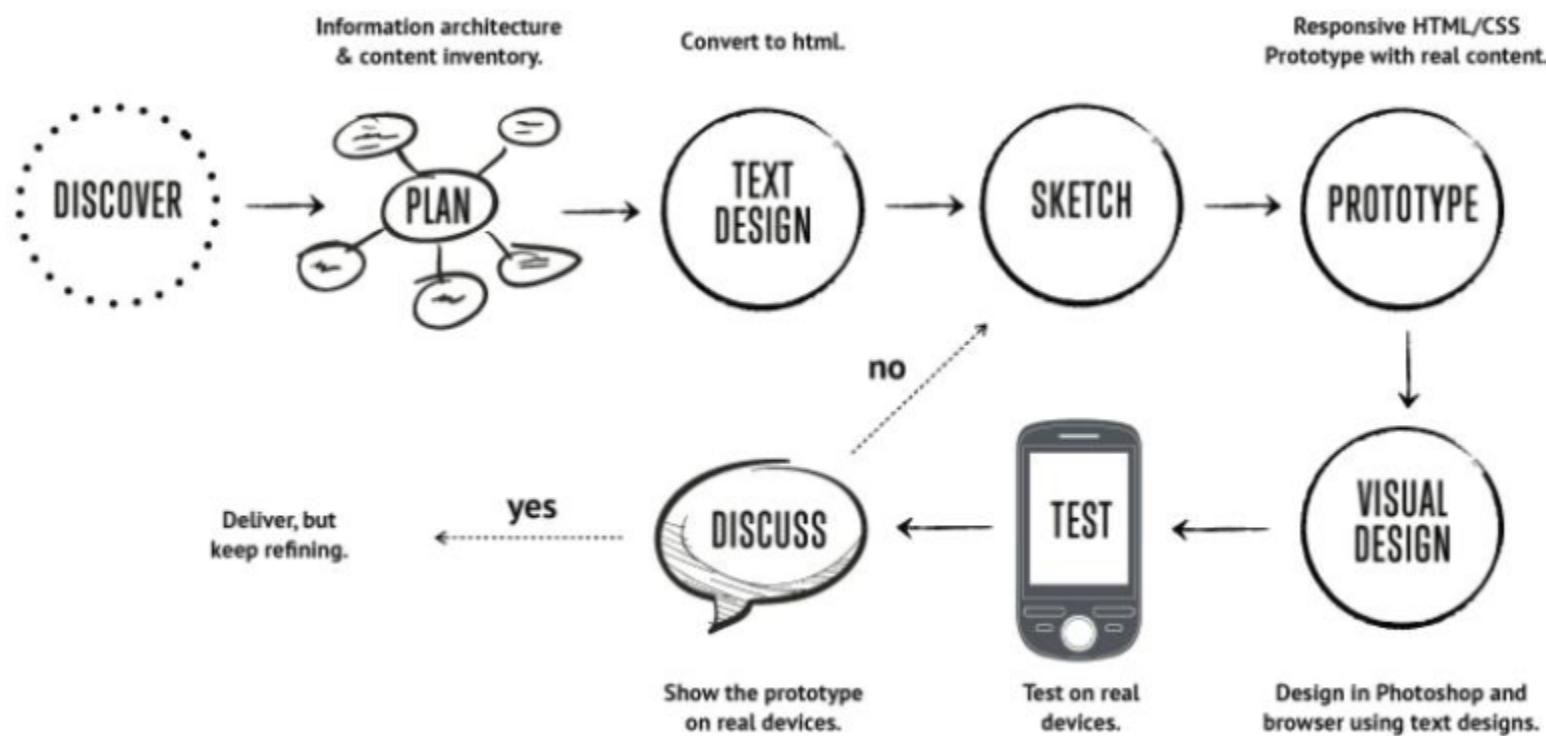
# VIELE MÖGLICHE BREAKPOINTS



Quelle

Aktuelle Trends

# ENTWICKLUNGSWORKFLOW



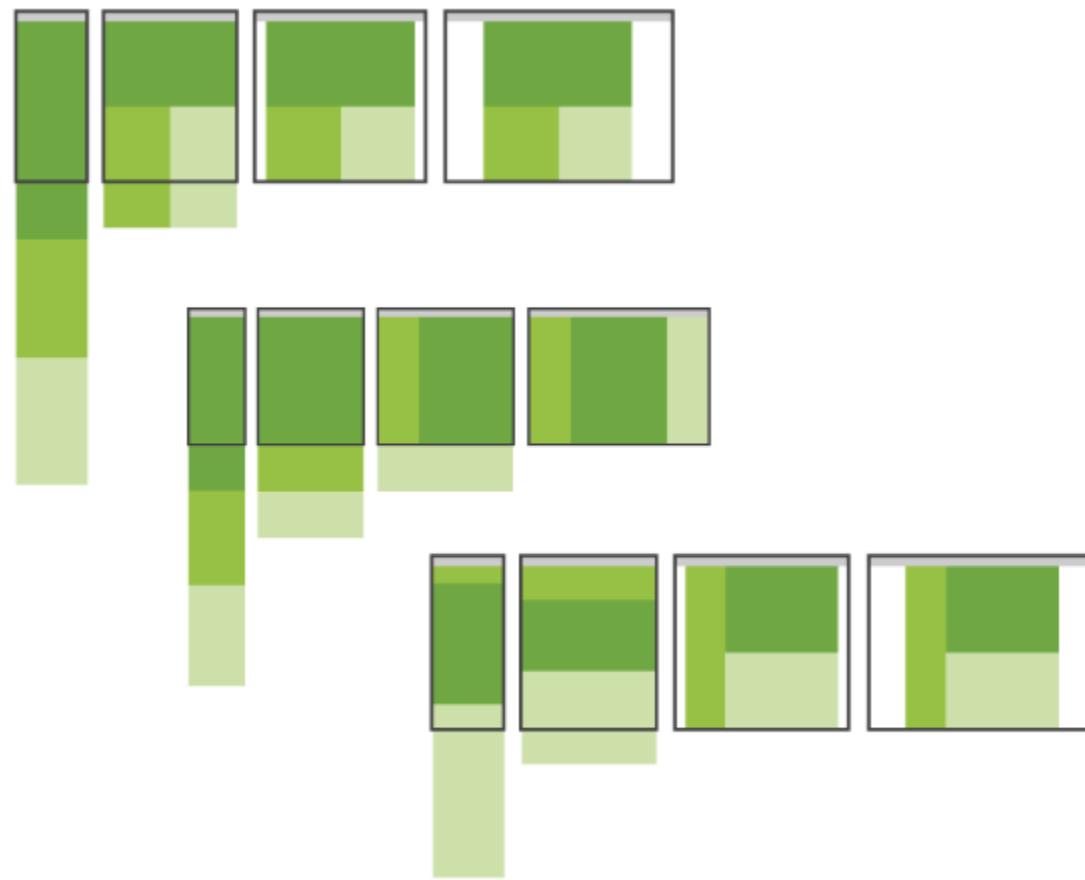
Quelle

## VERSCHIEDENE MUSTER FÜR INHALTE - 1

- Das klassische Boxendesign wird mit dem semantischen Ansatz fortgeführt
- Großen Layout-Flächen wie header, section, aside, footer etc. erhalten über CSS die Eigenschaft display:block bzw. flex
- [Luke Wroblewski](#) machte 2012 fünf Layout-Muster aus, die als Designansätze grundsätzlich denkbar sind

Aktuelle Trends

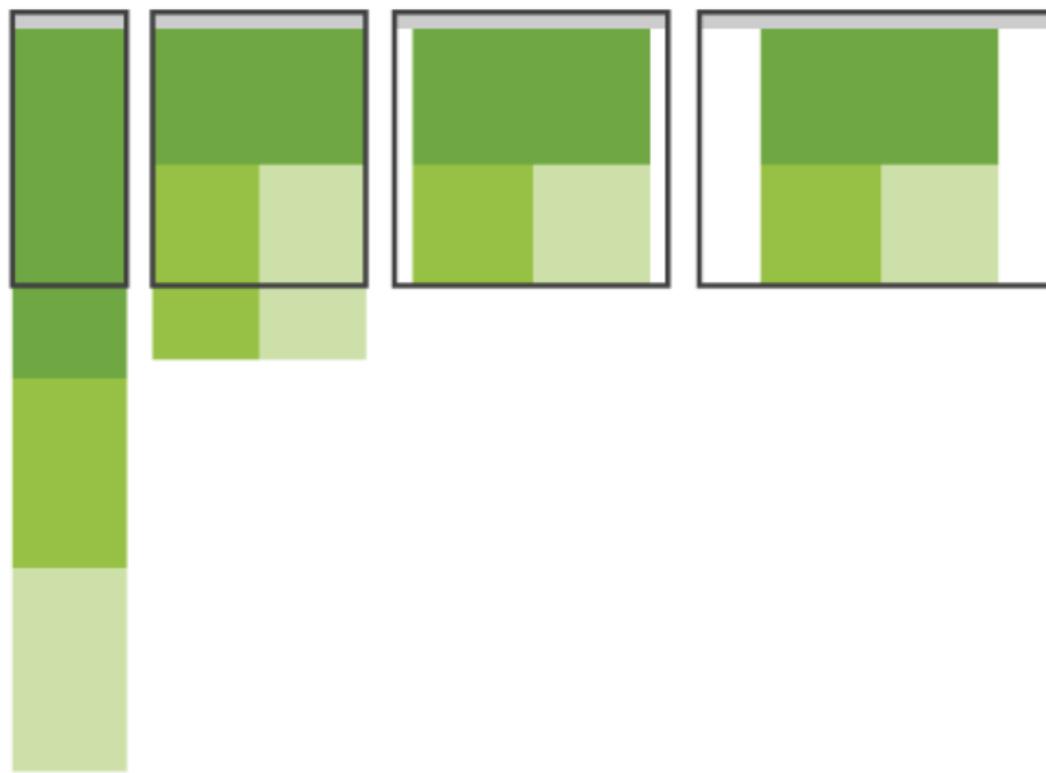
## VERSCHIEDENE MUSTER FÜR INHALTE - 2



Quelle

Aktuelle Trends

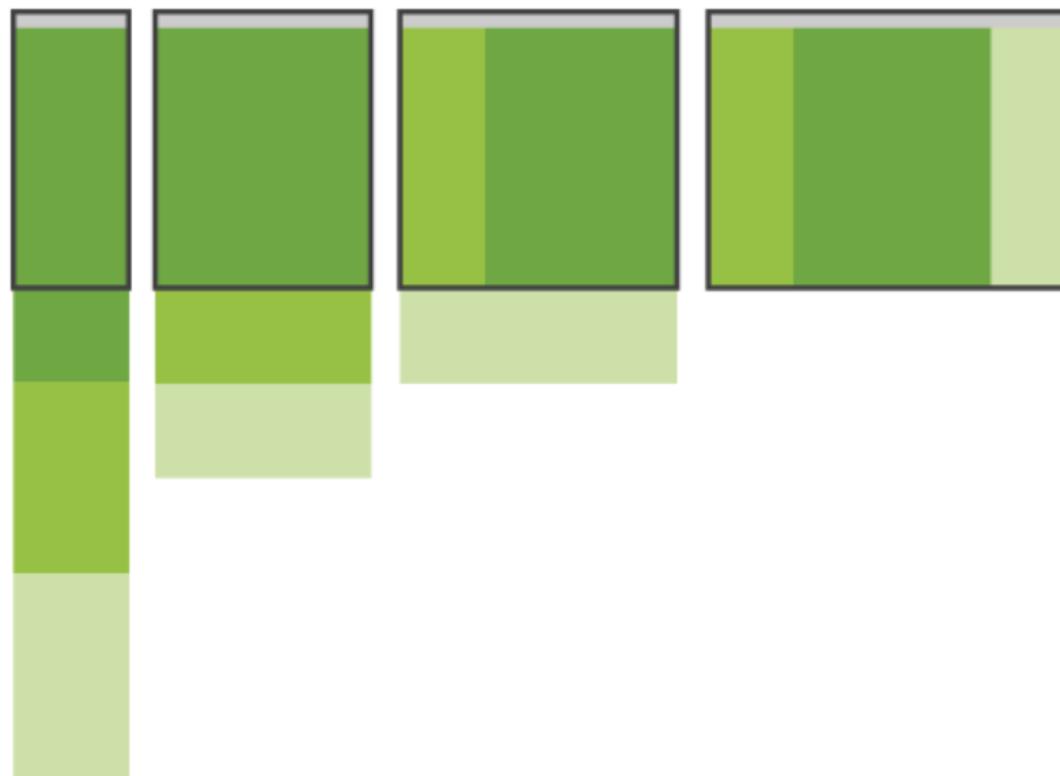
## LAYOUT-MUSTER - MOSTLY FLUID



Der Ansatz ist, die verschiedenen Breiten nach Breakpoints zu betrachten und simultan die daraus resultierenden Layouts für die eigene Inhalts- und Designidee zu gestalten.

Aktuelle Trends

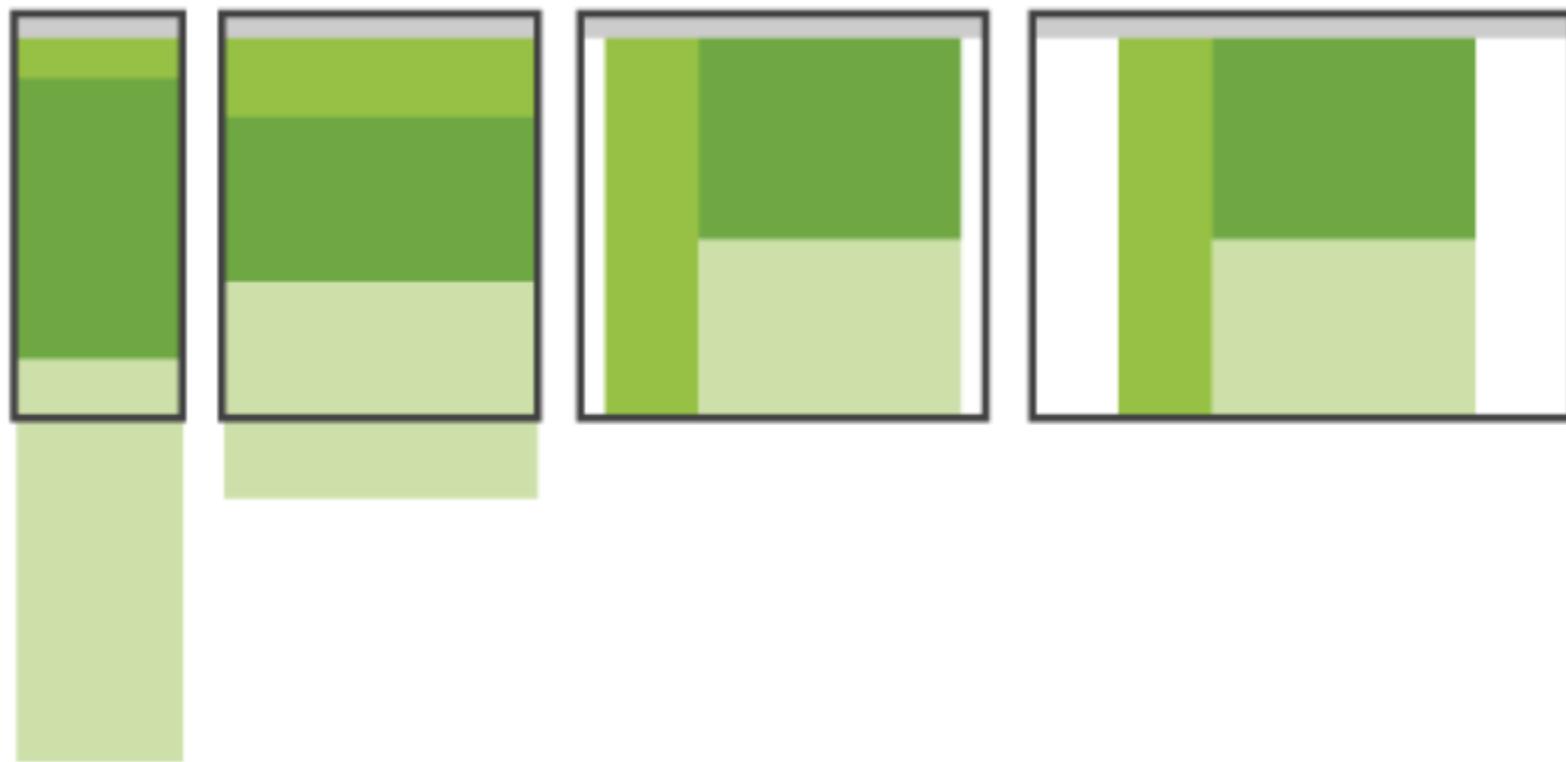
## LAYOUT-MUSTER - COLUMN DROP



Der Ansatz geht von Desktop First aus. Er löst mehrspaltiges Design durch „Fallenlassen“ einzelner Elemente unter den (teils noch mehrspaltigen) Hauptinhalt schrittweise auf.

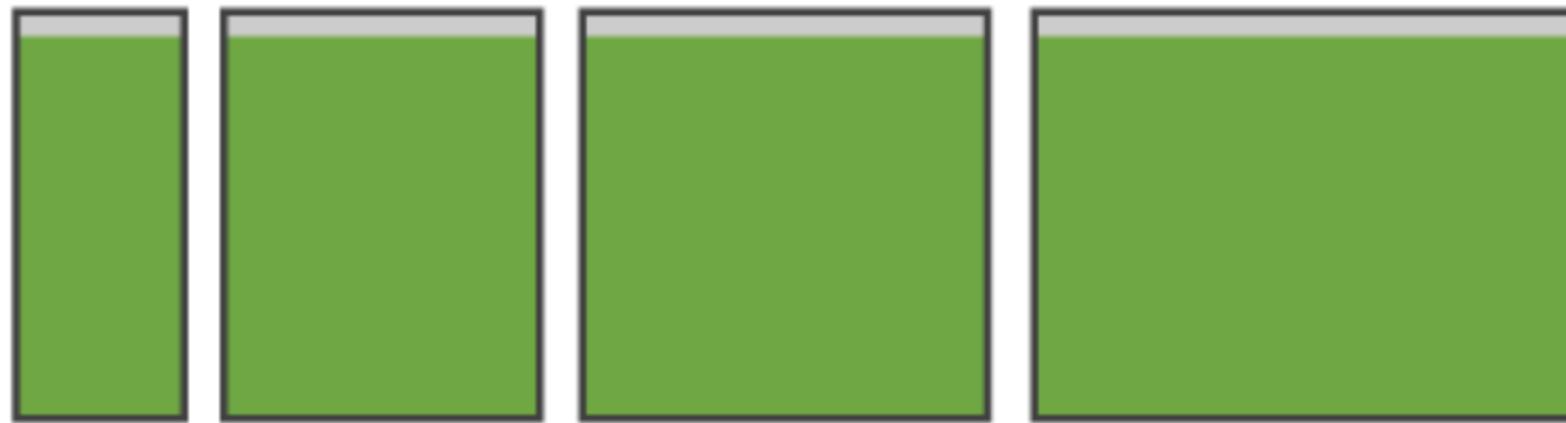
Aktuelle Trends

## LAYOUT-MUSTER - LAYOUT SHIFTER



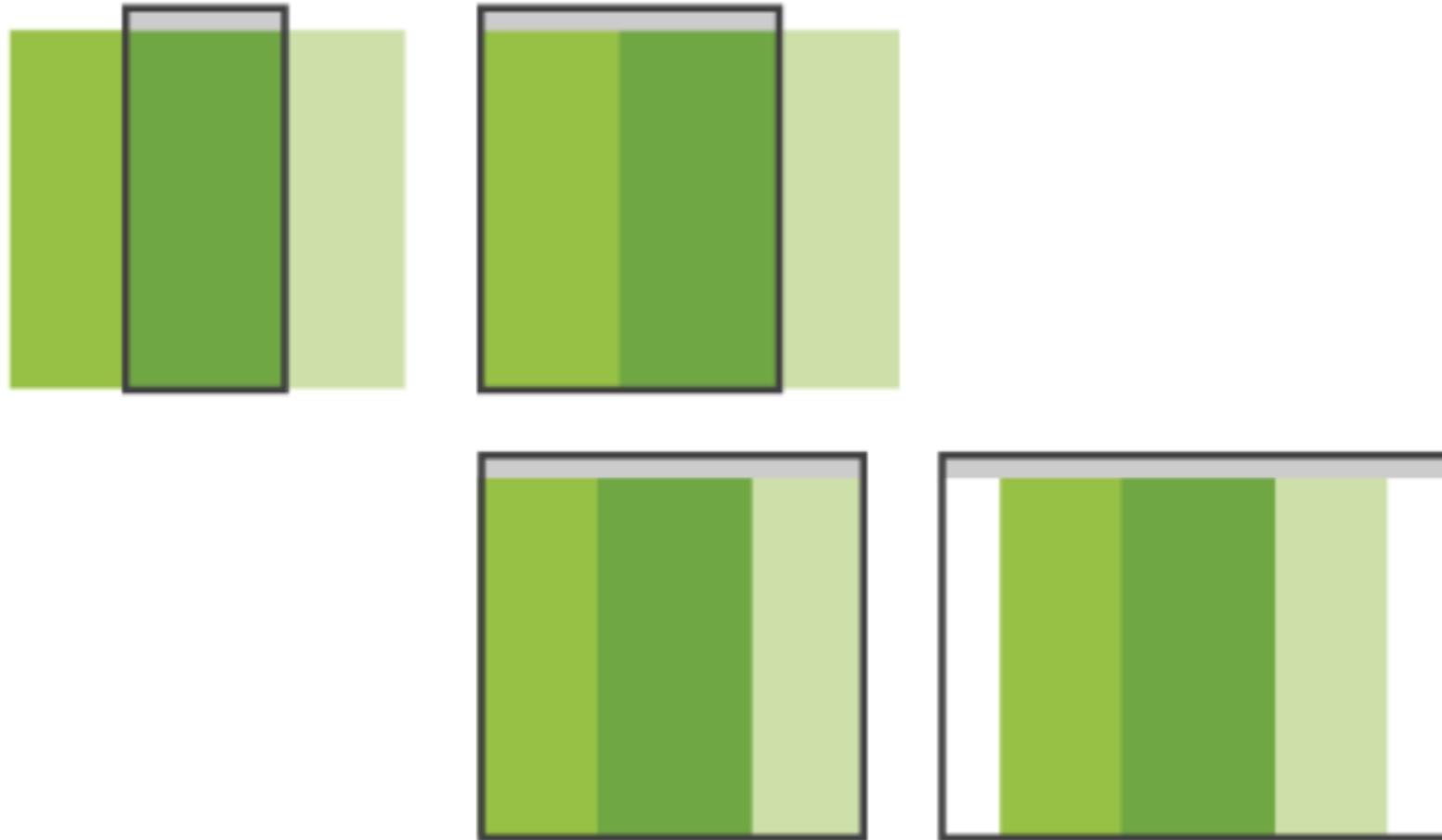
Dieser Ansatz erweitert die Ideen zuvor um Neudesign bestimmter Elemente bei Formatänderungen (horizontale versus vertikale Navigation, Bildformate und Dateiformate etc.).

## LAYOUT-MUSTER - TINY WEEKS



Der Schwerpunkt liegt hier auf einem einfachsten Layout (meist einer Spalte) kombiniert mit zahlreichen Maßnahmen der fluiden Anpassung von Schriftgrößen und -layout, Bild- und Videodateien, farblicher Gestaltung.

# LAYOUT-MUSTER - OFF CANVAS



Es wird versucht, die Fähigkeit von Smartphones und Tablets zu nutzen, Spalten links und rechts erst auf Anforderung in den Bildschirm über- deckend oder verdrängend einzufahren.

## LAYOUT-MUSTER - VERWEISE

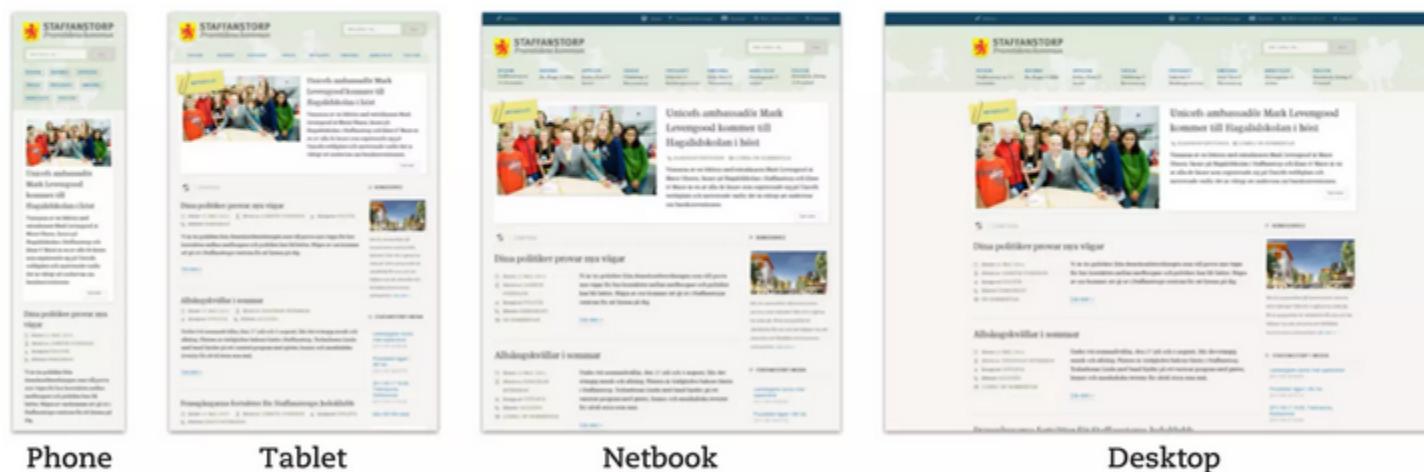
- Ein anderer Beitrag diskutiert diese Ansätze im Detail und verweist auf aktuellere Ideen dazu:  
<https://bradfrost.github.io/this-is-responsive/patterns.html>
- Siehe auch

# LAYOUT-MUSTER - BEISPIELE - 1

Aktuelle Trends

## WHAT IS RESPONSIVE WEB DESIGN?

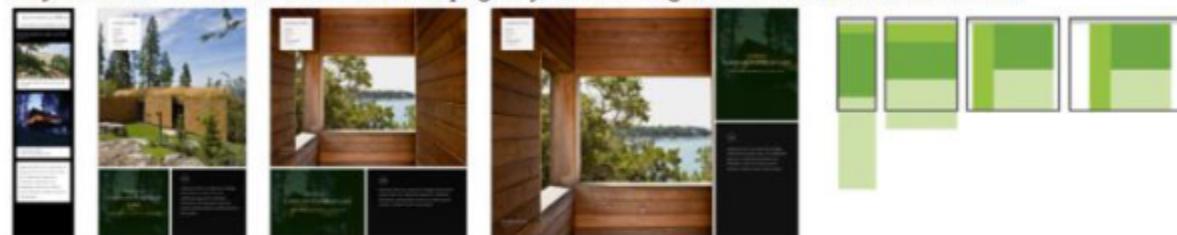
**A design where the layout and content adapt to the user's environment, which includes their screen size, platform, and even orientation.** Basically, it means using good code to create a site that works on a majority of devices, including those that haven't been invented yet.



# LAYOUT-MUSTER - BEISPIELE - 2

## RESPONSIVE WEB DESIGN LAYOUT PATTERNS

Layout Shifter: The site uses different page layouts for large, medium, and small screens.



Off Canvas: Content is hidden until the user asks for it. This technique is showing up in mobile apps.



Note: These patterns and examples are documented by Luke Wroblewski in his article "[Multi-Device Layout Patterns](#)." Please review his article for more information about each pattern and the pros/cons for them.

Copyright © 2013 by IQ Agency



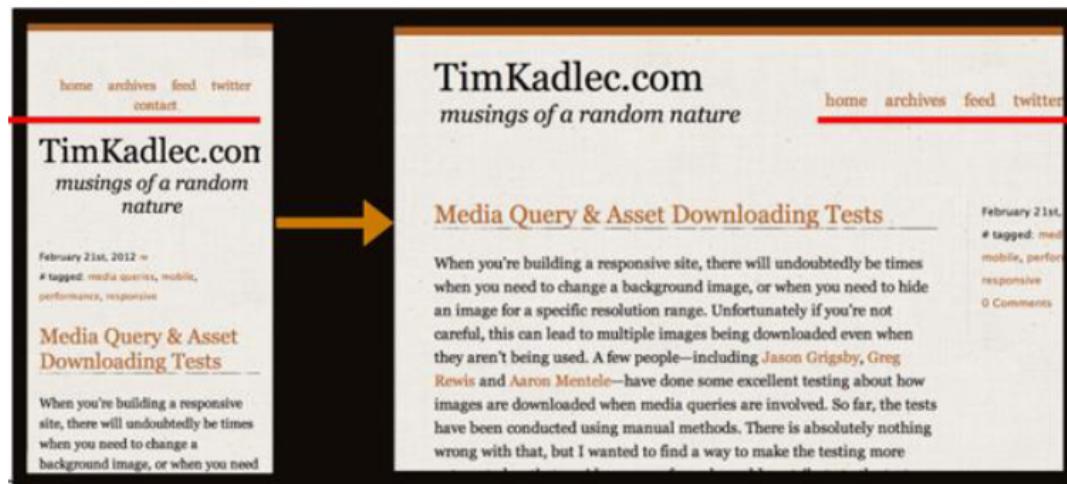
## Source

## NAVIGATIONS MUSTER

- Navigation zwischen Informationsangeboten ist sehr wichtig
- Dafür werden einfache und verschachtelte Menüs eingesetzt
- Je kleiner der Bildschirm, um so mehr konkurrieren vorhandener Platz, Navigationsübersicht und Schriftgröße
- Für Touchscreens ist sowieso auf größere Schaltflächen auszuweichen

# NAVIGATIONS MUSTER - TOP NAVIGATION

## Top Navigation



Dieser Ansatz ist eigentlich „keiner“ – bei Navigationen ohne Verschachtelungen und mit wenigen Einträgen ordnet man diese oben horizontal an.

Bei kleinen Bildschirmen lässt man die Navigation einfach mehrzeilig werden.

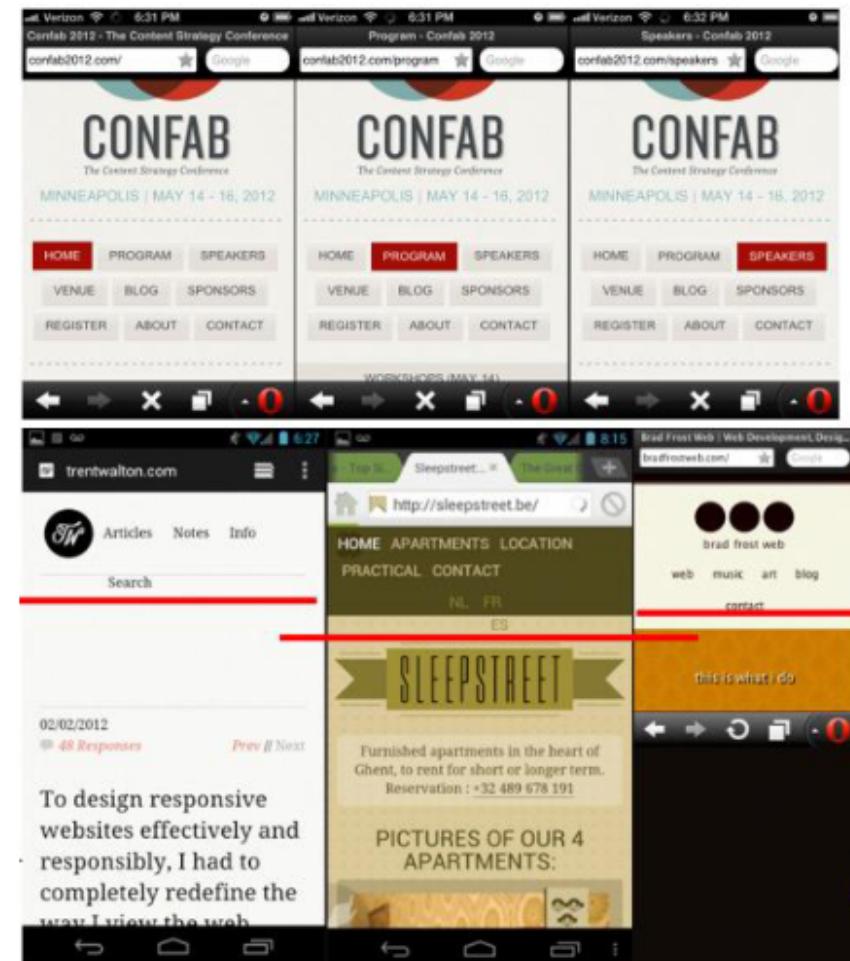
Aktuelle Trends

# NAVIGATIONS MUSTER - TOP NAVIGATION - PROBLEM

## Top Navigation

Problemaspekte:

- Höhe der Navigation zu Bildschirmhöhe
- Aussehen auf anderen Geräten, Design-Bugs



Aktuelle Trends

# NAVIGATIONSMUSTER - ICON NAVIGATION

## Navigation mit Icons



## NAVIGATIONS MUSTER - FOOTER (ANKER) NAVIGATION

- Da größere Navigation im Kopf auf kleinen Bildschirmen stört, bringt man sie im Footer an
- Per Verweis-Button im Kopf kann evtl. dorthin gesprungen (durchgescrollt) werden
- Über die Navigation oder mit einem Rücksprung ist eine Navigation möglich
- Diese Navigation ist gewöhnungsbedürftig, aber leicht erweiterbar, wenn die Website wächst
- Z.B.

# NAVIGATIONSMUSTER - SELECT MENU NAVIGATION

## Select Menu Navigation

Eine klassische Navigation für die Desktopvariante einer Website wird bei einer Grenze zu kleineren Bildschirmen in ein Select Menu (z.B. Drop-Down-Menü) verwandelt.



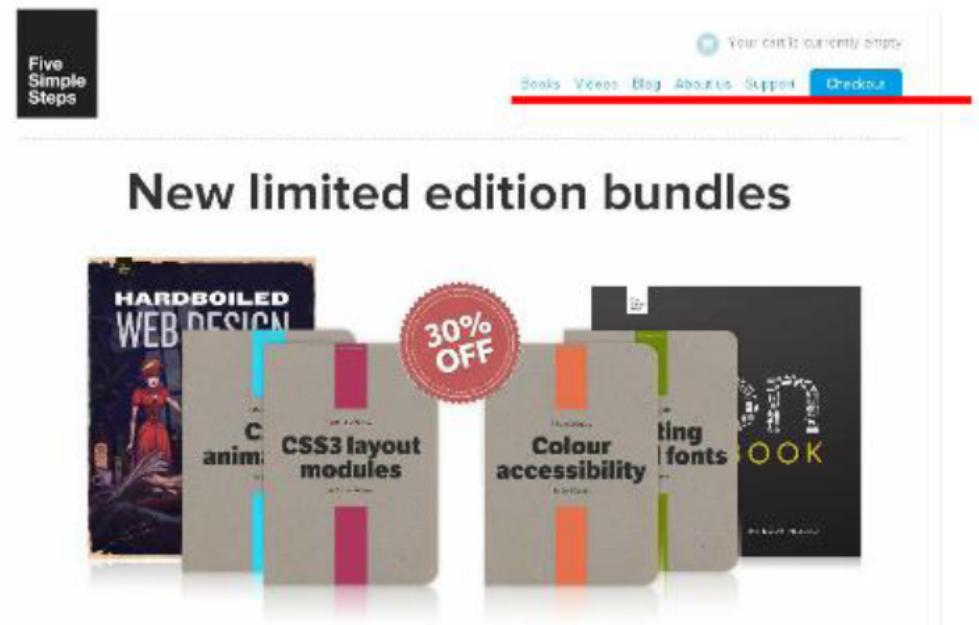
## **NAVIGATIONS MUSTER - SELECT MENU NAVIGATION - PROBLEM**

- Navigation wird zu Drop-Down-Box (festes Design)
- Nur Workarounds für Untermenüs

Aktuelle Trends

# NAVIGATIONS-MUSTER - SELECT MENU NAVIGATION

## Select Menu Navigation

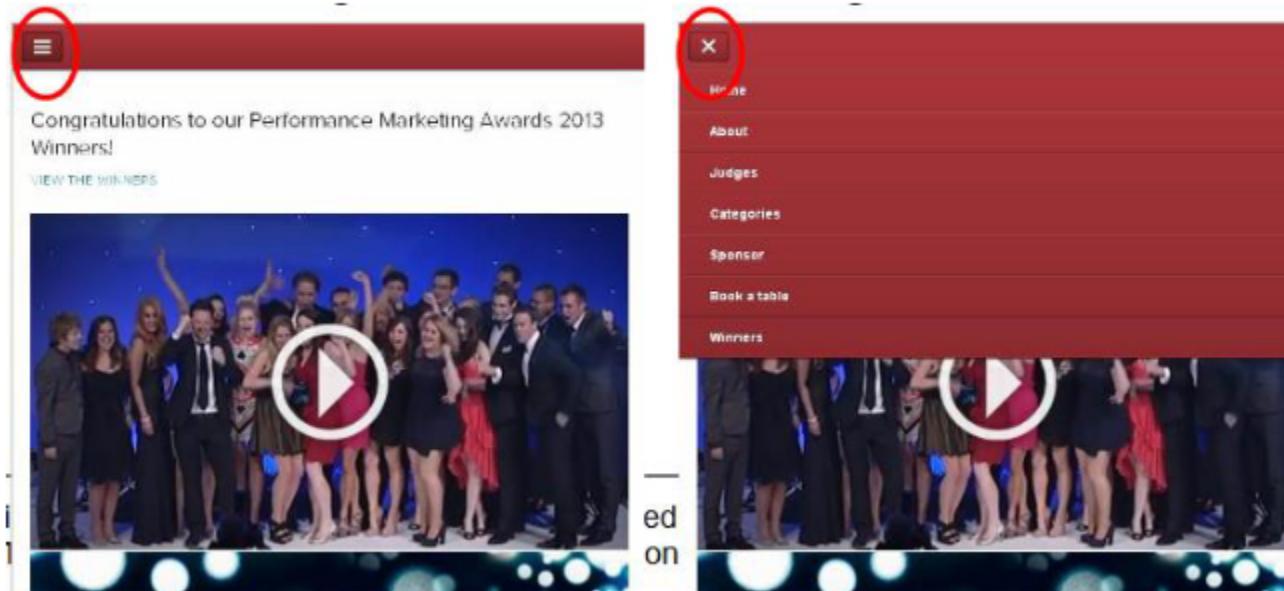


## TOGGLE-MENU („BURGER MENU“)

- Das Menu wird bei kleineren Bildschirmen nur noch als Ikone im Kopf dargestellt und nur bei Nutzerinteraktion dynamisch ein- bzw. wieder ausgeblendet
- Die Richtung der Überblendung ist wählbar
- Bootstrap biete hierfür z.B. eine größere **Auswahl an Möglichkeiten**

# NAVIGATIONS MUSTER - SELECT MENU NAVIGATION

## Toggle-Menu („Burger Menu“)



Aktuelle Trends

---

## OFF-CANVAS MENU

- Auch hierfür bietet Bootstrap eine **Auswahl an Elementen**

# NAVIGATIONS MUSTER - OFF-CANVAS MENU

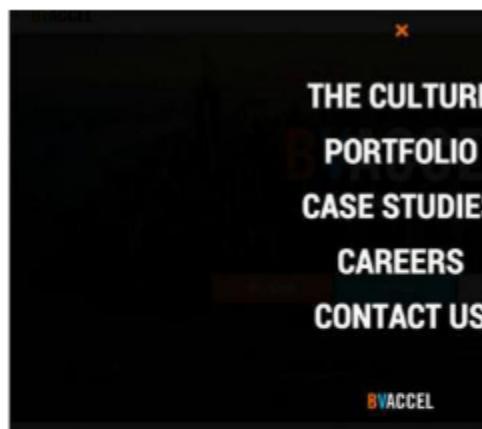
## Off-Canvas Menu

Die Grundidee dieses Menüs ist eigentlich ein Designtyp - das Verstecken der Navigation links oder rechts hinter dem Bildschirmrand kleiner Geräte, hervorzuholen durch Antippen und Ziehen.

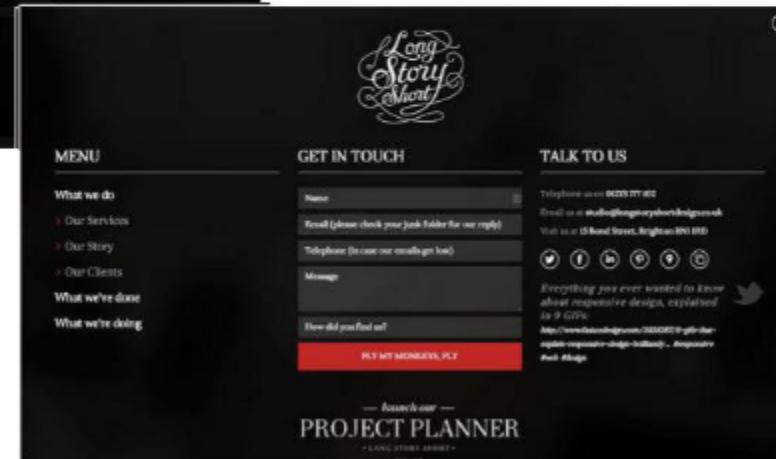


Aktuelle Trends

# NAVIGATIONSMASTER - FULL SCREEN MENUS



## Full Screen Menus



## NAVIGATIONS MUSTER - WEITERE MUSTER

- Eine andere Variante des Off-Canvas Menüs ist so konstruiert, dass das Menü unter der betrachteten Webseite liegt
- Die obere Webseite schnappt dann zurück, wenn nicht, navigiert wird
- [Weitere Beispiele](#)

## NAVIGATIONS MUSTER - WEITERE ANSÄTZE

- Neuester Trend ist es, nur noch mit der Touch-Geste Swipe oder mit der Maus-Geste Klicken-und-Ziehen zu arbeiten
- Die Angebote müssen dafür einfach strukturiert (Tiny Tweeks, Off-Canvas) und nicht zu umfangreich sein
- Die Navigation erfolgt dann über Verlinkung zum nächsten Angebot / Seitenzustand

## AKTUELLE LAYOUT TRENDS

- Der Trend geht dahin, Boxendesign zu vermeiden und die semantischen Rahmenelemente mehr in einer (!) Spalte zu verwalten
- Unterschiedliche Designs sollten dann durch eine Art „Choreografie“ (Tanz der Rahmenelemente) zu flexiblen neuen Arrangements entstehen

## AKTUELLE LAYOUT TRENDS

- Alles untereinander zu schieben, wäre dann die schwächste Lösung – Einrücken zwischen andere Elemente, seitliches Nebeneinanderstellen, verschieben aus einem Rahmenelement in ein anderes deutlich stärker
- Single Page applications Frameworks wie Angular, Next.js oder React basieren auf Responsiven Elementen und Designs

## AKTUELLE LAYOUT TRENDS

- Beispiele für neue Ansätze
- Content Folding
- Google Dev Guide für Responsive Design
- What is Responsive Design

## LITERATUR

- Andreas Hecht, Responsive Navigation: Design-Trends 2015, Febr. 2015, [Quelle](#) - bebilderte  
schöne Lösungen mit Links zu den technologischen Lösungen
- Patrick McNeil, 3 Essential Navigation Trends for 2015, [Navigationstrends](#)
- [Off-Canvas Vorstellung](#)
- [Off-Canvas Navigation](#)
- [Bootstrap 5](#)

## QUELLEN - 1

- Responsive und Adaptive Web Design:[Quelle](#)
- Ryan Boudreaux, Unterschied zwischen Adaptive und Responsive Web Design:[Quelle](#)
- Nathan Smith, [Quelle](#)
- [Mobile Responsive](#)
- [Responisve Design](#)
- [Fluid Design](#)
- [Fluid and responsive Design](#)

## QUELLEN - 2

- Ethan Marcotte, Fluid Grids, [Quelle](#)
- Fluid Grid Layout Tutorial For Responsive Web Design - [Quelle](#)
- F55+ Great and Useful Tools for Responsive Webdesign, [Quelle](#)
- 2012, Vorschläge für Grenzen bei {3,4,6}-Media-Query-Grenzen, [Quelle](#)
- Aug. 2020, 20 CSS Frameworks and Boilerplates
- [Quelle](#)
- Buch: Andrea Ertel, Kai Laborenz, „Responsive Webdesign: Anpassungsfähige Websites programmieren und gestalten“, Galileo Computing, 12.12.2013 / 12.12.2014 – future books

## **ABSPANN**

Drittes Level geschafft weitere Folgen!

Fragen und Feedback?