HTML & CSS – fortgeschrittene Techniken

Webseiten müssen heute flexibel sein und gleichermaßen auf Smartphones, Tablets als auch Desktop-Geräten funktionieren – responsives Webdesign macht das möglich. Sie sehen, wie Sie attraktive responsive Webseiten erstellen und lernen die neuen CSS-Layoutmodule Flexbox und Gridlayout kennen. Thema sind außerdem fortgeschrittene HTML-Elemente.

Inhalte:

- Fortgeschrittene HTML-Elemente (meta-Elemente etc.)

- Fortgeschrittene Formulartechniken

- Grundlagen des responsiven Webdesigns

- Funktionsweise von Flexbox

- Funktionsweise von Gridlayout

- Navigationsleisten gestalten

- Vollständige Layouts entwickeln

[Fortgeschrittene HTML-Elemente 2](#_Toc25156483)

[Tabellen erstellen 2](#_Toc25156484)

[1. Übung Tabellen 2](#_Toc25156485)

[Fortgeschrittene Formulartechniken 3](#_Toc25156486)

[2. Übung Formular 3](#_Toc25156487)

[meta-Elemente 4](#_Toc25156488)

[CSS-Wiederholung und Vertiefung 4](#_Toc25156489)

[Funktionsweise von Flexbox 4](#_Toc25156490)

[3. Übung Navigation in Varianten 4](#_Toc25156491)

[4. Übung Sticky Footer 5](#_Toc25156492)

[5. Übung (fakultativ) Anordnen mit Flexbox 5](#_Toc25156493)

[6. Übung Produktpräsentation 5](#_Toc25156494)

[Funktionsweise von Gridlayout 6](#_Toc25156495)

[7. Übung Gridlayout 6](#_Toc25156496)

[8. Übung Elemente ausrichten mit Leerraum 6](#_Toc25156497)

[9. Übung (fakultativ) Gridlayout ausrichten 6](#_Toc25156498)

[10. Übung (fakultativ) „Breaking out of the box“ 7](#_Toc25156499)

[Grundlagen des responsiven Webdesigns 7](#_Toc25156500)

[11. Übung Webseite untersuchen 7](#_Toc25156501)

[12. Übung Media Queries 8](#_Toc25156502)

[13. Übung Gridlayout responsive machen 8](#_Toc25156503)

[14. Übung (fakultativ) responsive Darstellung mit Gridlayout 8](#_Toc25156504)

[Layout 1 - Grundlegende Anpassungen 9](#_Toc25156505)

[Layout 2 - Navigation erweitern 10](#_Toc25156506)

[Layout 3 - Tabelle ergänzen 10](#_Toc25156507)

[Layout 4 - Formular ergänzen 10](#_Toc25156508)

[Layout 5 - Meta-Angaben und weitere Angaben im head-Bereich 10](#_Toc25156509)

[Layout 6 - Anpassungen an der Formatierung 10](#_Toc25156510)

[Layout 7 - Tabelle formatieren 10](#_Toc25156511)

[Layout 8 - Formularelemente anordnen 11](#_Toc25156512)

[Layout 9 – Formularelemente formatieren 12](#_Toc25156513)

[Layout 10 (fakultativ) – Checkbox im Formular integrieren 12](#_Toc25156514)

[Layout 11 - Fußzeile immer unten 13](#_Toc25156515)

[Layout 12 – Produkte ergänzen 13](#_Toc25156516)

[Layout 13 - Layout responsiv machen 13](#_Toc25156517)

[Layout 14 (fakultativ) - Tabelle responsive machen 13](#_Toc25156518)

[Layout 15 - Formular responsive machen 14](#_Toc25156519)

[Layout 16 - Navigation responsive machen – einfache Variante 15](#_Toc25156520)

[Layout 17 (fakultativ) – Klappnavigation integrieren 15](#_Toc25156521)

[Layout 18 (fakultativ)- Diashow ergänzen 15](#_Toc25156522)

Hinweis: Die vollständigen Listings des Buchs finden Sie im Ordner listings\_html\_css\_dummies. Die Dateien, die Sie für die Übungen brauchen, befinden sich direkt im Ordner dateien.

## 

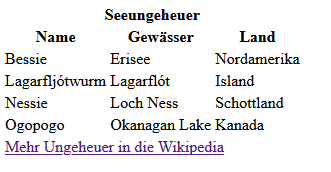
## Fortgeschrittene HTML-Elemente

### Tabellen erstellen

Buch S. 186 ff.

#### 1. Übung Tabellen

Ergänzen Sie die Tabelle mit einer weiteren Zeile und einem Fußbereich, der sich über drei Spalten erstreckt (*tabelle\_thead\_tbody.html*)

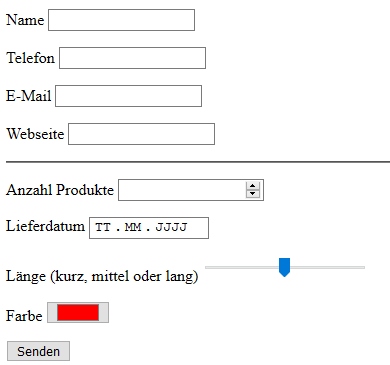


### Fortgeschrittene Formulartechniken

Buch S. 165 ff

#### 2. Übung Formular

Erstellen Sie das folgende Formular in einem HTML-Grundgerüst und speichern Sie die Datei unter dem Namen *formular.html* ab.



* Überlegen Sie sich, welche Formularelemente immer am besten geeignet sind.
* Für den Button nehmen Sie bitte ein button-Element (dieser lässt sich besser formatieren als ein input-Element).
* Verwenden Sie label-Elemente für die Beschriftungen.
* Beschriftung und Formularelement sollen immer durch Absätze (p) zusammengefasst werden, also beispielsweise so:

<p>

<label for="name">Name</label>

<input type="text" name="name" id="name" required>

</p>

* Die folgenden Felder sollen obligatorisch sein:
* Name
* E-Mail
* Anzahl Produkte

### meta-Elemente

Buch S. 98

Weitere Infos <https://wiki.selfhtml.org/wiki/HTML/Kopfdaten/meta>

## CSS-Wiederholung und Vertiefung

Buch S. 196 ff.

Google-Fonts: Buch S. 252 ff.

Überblick über Selektoren, Buch S. 202 ff.

### Funktionsweise von Flexbox

Buch S. 337

#### 3. Übung Navigation in Varianten

Nehmen Sie die Datei *flexbox-navi.html* als Basis.

Wie ließen sich folgende Fälle realisieren:

a) Service und Kontakt am Ende



b) Start alleine am Ende



c) Start am Ende, Platz zwischen den Elementen

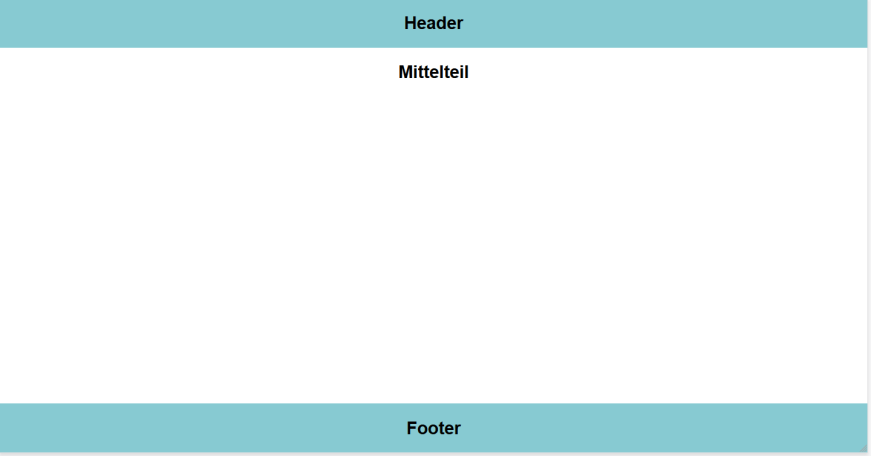


d) Start am Ende, Elemente nehmen allen verfügbaren Platz



#### 4. Übung Sticky Footer

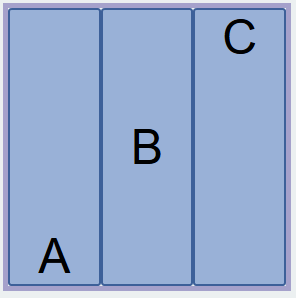
Häufig möchte man, dass eine Fußzeile auch bei weniger Inhalt sich immer ganz unten am Viewportrand befindet. Das kann man gut über Flexbox realisieren.



Sie finden das Beispiel mit Hinweisen als *flexbox\_sticky\_footer.html*.

#### 5. Übung (fakultativ) Anordnen mit Flexbox

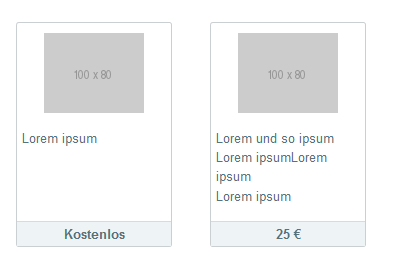
Wie ließe sich die folgende Anordnung realisieren?



Nehmen Sie als Basis die Datei *flexbox\_ausrichten\_komplex.html*

#### 6. Übung Produktpräsentation

Folgende Darstellung soll mit Flexbox erreicht werden. Nehmen Sie als Basis *flexbox\_produktpraesentation.html*. Hinweise stehen im Dokument selbst.

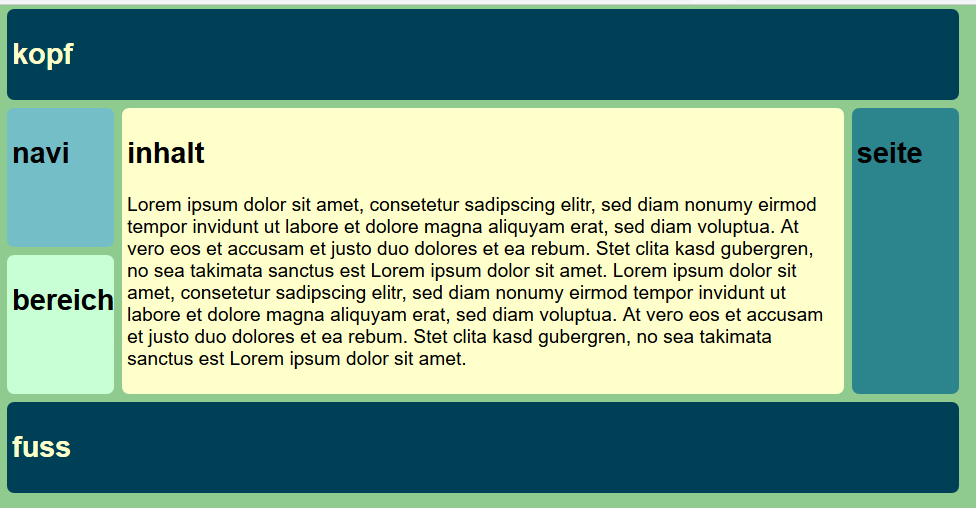


### Funktionsweise von Gridlayout

Buch S 352

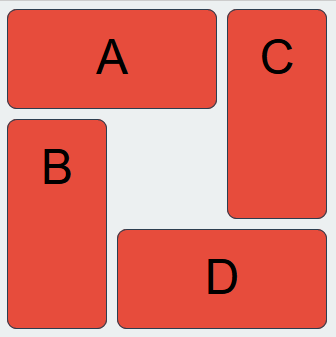
#### 7. Übung Gridlayout

Erstellen Sie folgendes Layout mit Gridlayout (die responsiven Anpassungen brauchen Sie noch nicht durchzuführen, das kommt später). Als Basis dient die Datei *grid\_rwd.html*.



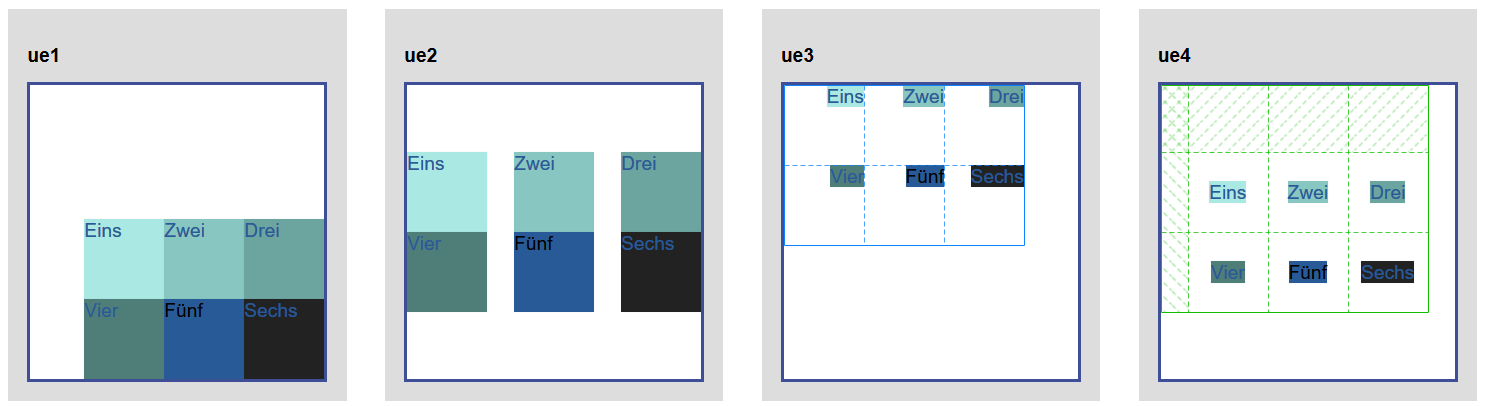
#### 8. Übung (fakultativ) Elemente anordnen mit Leerraum

Wie lässt sich die folgende Anordnung mit Gridlayout umsetzen? (Datei *grid\_leerraum.html*)



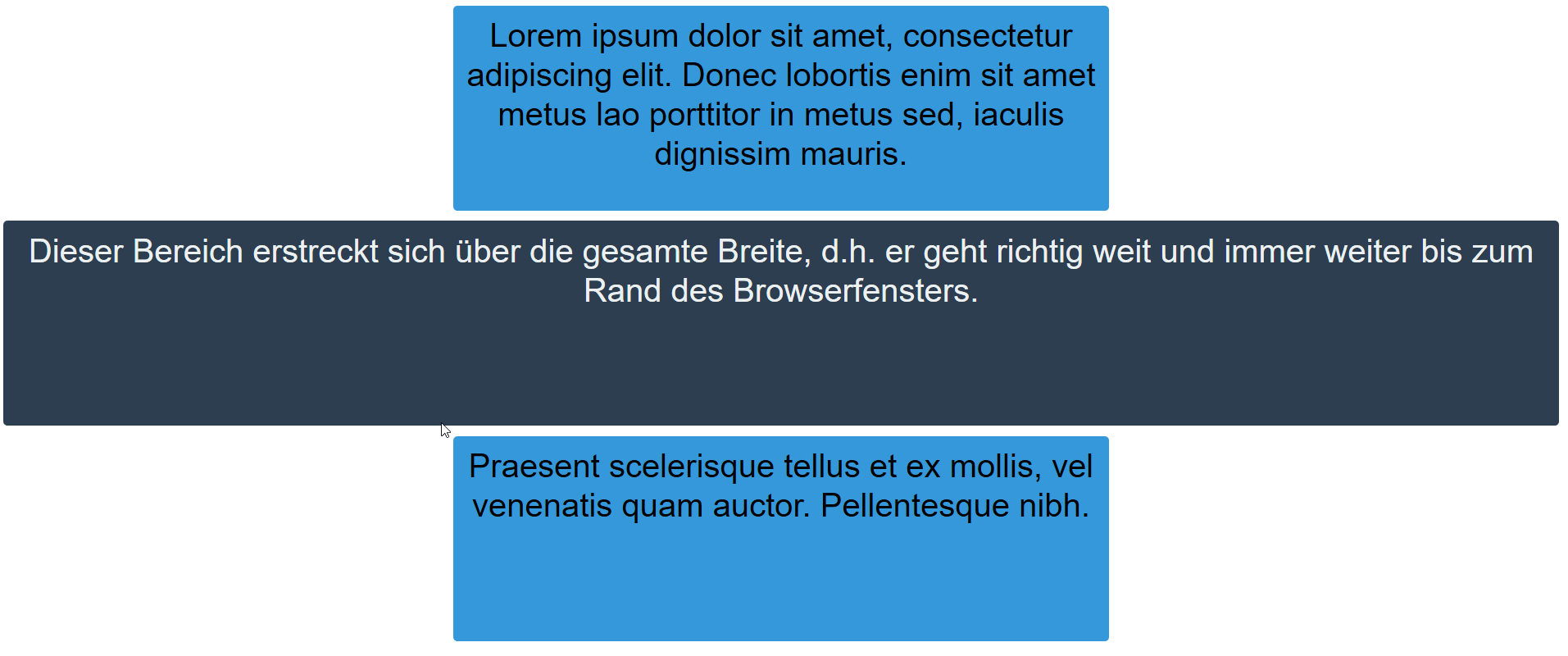
#### 9. Übung (fakultativ) Gridlayout ausrichten

Wie kann man die folgenden Ausrichtungen mit Gridlayout (*grid\_ausrichtung.html*) umsetzen?



#### 10. Übung (fakultativ) „Breaking out of the box“

Wie lässt sich die folgenden Darstellung mit Gridlayout (*grid\_breaking\_out\_of\_the\_box.html*) erstellen?



### Grundlagen des responsiven Webdesigns

Buch S. 365

#### 11. Übung Webseite untersuchen

Untersuchen Sie eine der folgenden Seiten:

* <https://www.tagesschau.de/>
* <https://www.spiegel.de/>
* <https://www.zeit.de/>
* <http://www.gutenbergschule.eu/>
* <https://www.theguardian.com/uk>

1. Welche Strategie wird gewählt (responsiv, adaptiv oder separate mobile Webseite)?
2. Was beobachten Sie sonst noch?
3. Was finden Sie gelungen, was finden Sie weniger gut?

Media Queries

Buch S. 372

#### 12. Übung Media Queries

Erweitern Sie da Beispiel *mediaqueries\_einstieg.html:*

* Ergänzen Sie noch eine Überschrift (h1).
* Die Überschrift soll bei kleinen Screens 1rem groß sein, bei mittleren 1.5 rem und bei größeren 2rem.
* Ab einer Größe von 855px soll der Hintergrund rot sein und die Schrift kursiv.

#### 13. Übung Gridlayout responsiv machen

Machten Sie das Layout *grid\_rwd.html* responsiv. Es soll bei weniger Platz zweispaltig sein und bei kleinen Viewports einspaltig.



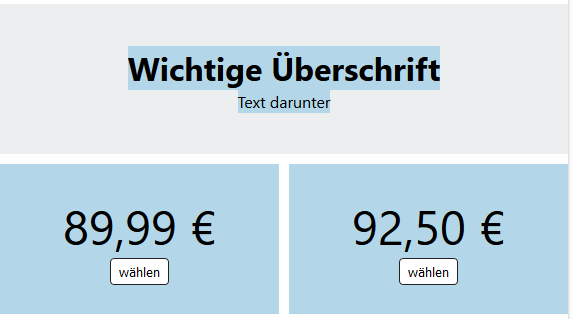
#### 14. Übung (fakultativ) responsive Darstellung mit Gridlayout

Setzen Sie die folgende Darstellung mit Gridlayout um (*gridlayout\_anordnen.html*).

Bei viel verfügbarem Platz ist die linke Spalte doppelt so breit wie die rechte.



Bei weniger Platz sind die Spalten gleich breit:



Und bei kleinem Viewport ist die Anordnung einspaltig:



Beispielwebsite

An einem Layoutbeispiel wenden Sie die HTML-&CSS-Techniken praktisch an. Die Basisdateien finden Sie im Ordner *layout*.

## Layout 1 - Grundlegende Anpassungen

* Ändern Sie die Texte.
* Ersetzen Sie die ursprünglichen Bilder.

## Layout 2 - Navigation erweitern

Ergänzen Sie weitere Navigationspunkte auf folgende Dateien mit passenden Linktexten (wenn Sie andere Dateinamen verwenden, müssen Sie später die Dateien ebenfalls unter den neuen Namen abspeichern).

* produkte.html
* tabelle.html
* kontakt.html

## Layout 3 - Tabelle ergänzen

Erstellen Sie durch Kopieren eine neue Unterseite mit dem Namen *tabelle.html* und fügen Sie die Tabelle aus *tabelle\_thead\_tbody.html* als Inhalt des main-Elements ein.

Sorgen Sie dafür, dass der aktuelle Navigationspunkt hervorgehoben ist.

## Layout 4 - Formular ergänzen

Erstellen Sie durch Kopieren eine neue Unterseite mit dem Namen *kontakt.html* und fügen Sie das Formular aus *formular.html* als Inhalt der ersten article-Elements ein!

Sorgen Sie dafür, dass der aktuelle Navigationspunkt hervorgehoben ist.

## Layout 5 - Meta-Angaben und weitere Angaben im head-Bereich

Passen Sie bei allen bisher gestellten Seiten die meta-Angaben und andere Angaben im head an.

* Wählen Sie jeweils eine passende description.
* Verwenden Sie gute Seitentitel.
* Welche weiteren meta-Angaben sind sinnvoll?

## Layout 6 - Anpassungen an der Formatierung

* Ändern Sie die Farben Ihres Layouts.
* Wählen Sie einen Google Font für die Überschriften (und eventuell andere Texte).

## Layout 7 - Tabelle formatieren

Vergeben Sie bei main die Klasse einspaltig und setzen Sie .einspaltig auf display: block.

Ergänzen Sie bei Ihrer HTML-Tabelle die Klasse datentabelle.

Erstellen Sie in Ihrer CSS-Datei einen weiteren Bereich für die Tabellenformatierung, den Sie durch einen CSS-Kommentar einleiten:

/\* Tabellenformatierung \*/

Für .datentabelle geben Sie an:

border-collapse: collapse;

width: 100%;

Die Tabellenüberschriften sollen nicht mittig sein, sondern so angeordnet werden wie die anderen Tabellen-Elemente. Dafür wählen Sie diese mit .datentabelle th aus und geben dann an

text-align: inherit;

Die Tabellenzellen sollen alle unten angeordnet werden, dafür wählen Sie diese aus und schreiben:

vertical-align: bottom;

Vergeben Sie außerdem etwas Innenabstand (padding) und einen Rahmen nach unten (border-bottom).

## Layout 8 (fakultativ)- Formularelemente anordnen

Ergänzen Sie bei Ihrem HTML-Formular die Klasse produktformular.

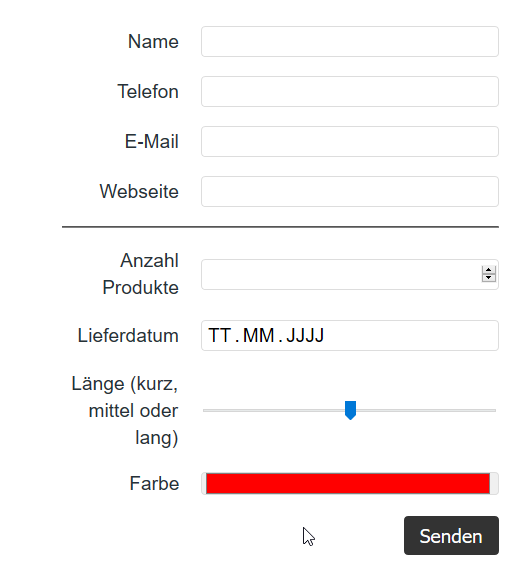
Erstellen Sie in Ihrer CSS-Datei einen weiteren Bereich für die Formularformatierung, den Sie durch einen CSS-Kommentar einleiten:

/\* Formularformatierung \*/

Definieren Sie folgende Formatierungen

* Der Absatz innerhalb des Formulars soll zum Flexboxcontainer werden mit display: flex; Definieren Sie außerdem für die Ausrichtung align-items: center.
* Die label im Formular erhalten eine Breite von 40%, einen Innenabstand nach rechts (padding-right) und werden rechtsbündig angeordnet.
* Der Button wird mit margin-left: auto nach rechts verschoben.

## Layout 9 – Formularelemente formatieren



Für den Button definieren Sie:

* ein padding
* keinen Rahmen (border: none)
* ein bisschen border-radius
* Hintergrund- und Textfarbe
* Damit die Schriftgröße wie gehabt ist, geben Sie font-size: inherit an.
* Beim :focus- und :hover-Zustand nehmen Sie andere Farben.
* Damit der Übergang nicht so abrupt abläuft, verwenden Sie eine transition für den Ursprungszustand.

Für die input-Elemente definieren Sie:

* Innenabstand nach links
* font:inherit, damit dieselbe Schrift genommen wird wie sonst
* dünnen Rahmen
* 100% Breite
* transition
* Für den :focus-Zustand definieren Sie outline: 0 und einen box-shadow ohne Verschiebung.

## Layout 10 (fakultativ) – Checkbox im Formular integrieren

Ergänzen Sie eine Checkbox mit Beschriftung im Formular. Was muss man machen, damit die Checkbox korrekt dargestellt wird?

## Layout 11 - Fußzeile immer unten

Bei weniger Inhalt befindet sich die Fußzeile in der Mitte des Bildschirms. Schöner wäre, sie wäre immer unten. Setzen Sie das um unter Verwendung der Technik von *flexbox\_sticky\_footer.html*.

## Layout 12 – Produkte ergänzen

* Speichern Sie die Seite *index.html* unter dem Namen *produkte.html* ab.
* Ergänzen Sie bei main die Klasse produktuebersicht.
* Innerhalb von main kopieren Sie den HTML-Code für die Produktkärtchen aus *flexbox\_produktpraesentation.html*. Insgesamt sollen es mindestens 7 Produkte sein.
* Ergänzen Sie dann auch den benötigten CSS-Code innerhalb von *stil.css*.
* Sorgen Sie dafür, dass die Produkte automatisch in der nächsten Zeile dargestellt werden, wenn nicht genügend Platz vorhanden ist.

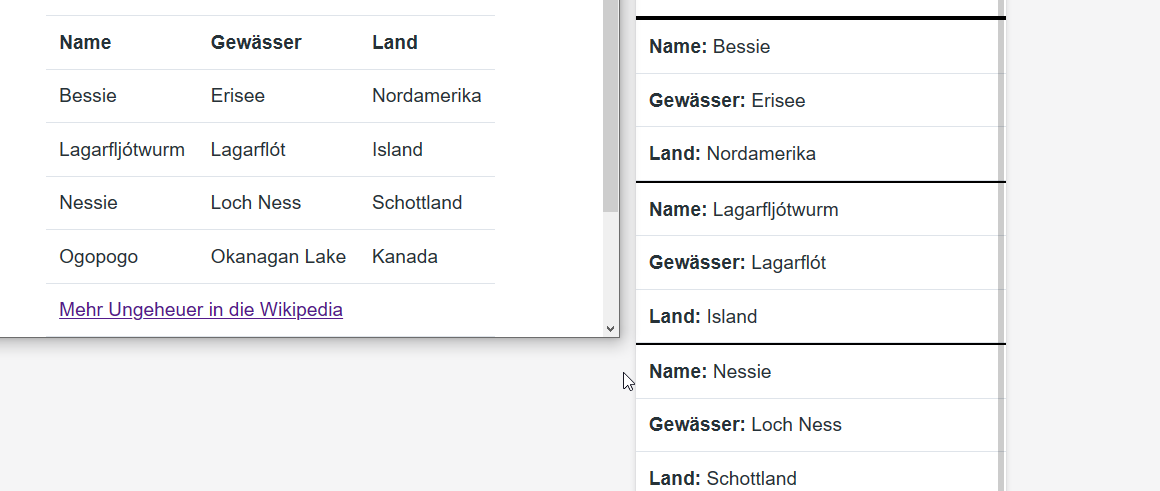
## Layout 13 - Layout responsiv machen

Machen Sie das Layout responsiv.

* Die Bilder sollen sich flexibel anpassen (flüssige Bilder).
* Unterhalb von 530px (Media Query mit max-width) soll main nicht mehr Flexcontainer sein, sondern normales Blockelement. Die Breite von article soll außerdem auf auto gesetzt werden.

## Layout 14 (fakultativ) - Tabelle responsiv machen

Es gibt verschiedene Möglichkeiten, Tabellen responsiv zu machen. Beispielsweise kann man die Inhalte untereinander darstellen lassen – muss aber dabei die Beschriftung zusätzlich ergänzen. Dies kann dann aussehen wie rechts unten gezeigt.



Überlegen Sie zuerst, für welche Größen die angepasste Darstellung gelten soll und formulieren Sie eine entsprechende Media Query mit max-width. Eine sinnvolle Breitenangabe wäre 450px. Innerhalb der Media Query nehmen Sie die folgenden Formatierungen vor:

* Machen Sie tr, td und th zu Blockelementen (display: block).
* Ergänzen Sie unterhalb der tr-Elemente einen Rahmen (border-bottom).
* Blenden Sie die Überschriftszellen innerhalb von thead aus (display: none).
* Vor der ersten Zelle ergänzen Sie den Text „Name“. Dazu schreiben Sie:

td:nth-child(1)::before {

content: "Name: ";

font-weight: bold; /\* fakulatativ \*/

}

* Genauso geben Sie auch die anderen Beschriftungen (Gewässer und Land an).
* Überlegen Sie sich, welche weitere Anpassungen sinnvoll wären!

## Layout 15 - Formular responsiv machen

Für Geräte mit weniger Breite als 780px definieren Sie:

* dass die Absätze und label-Elemente zu Blockelementen werden sollen.
* Setzen Sie text-align auf left.
* Vergeben Sie eine Breite von 90%.

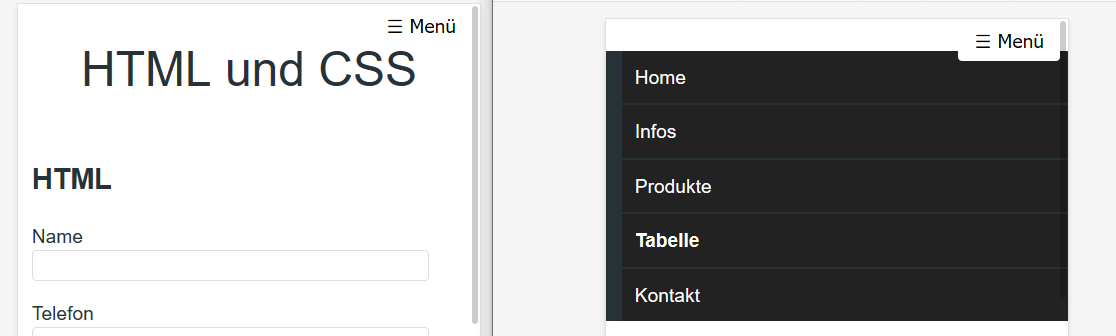
## 

## Layout 16 - Navigation responsiv machen – einfache Variante

Die einfachste Möglichkeit, um die Navigation responsiv zu machen, ist flex-wrap: wrap beim Flexboxcontainer der Navigation (nav ul) zu ergänzen.

## Layout 17 (fakultativ) – Klappnavigation integrieren

Binden Sie die im Buch auf S. 390ff beschriebene Klappnavigation in Ihr Beispiel ein!



## Layout 18 (fakultativ)- Diashow ergänzen

Für viele fortgeschrittene Effekte können Sie die jQuery-Plugins einsetzen.

jQuery-Plug-ins sind auf jQuery aufbauende Skripte, die spezielle Aufgaben erledigen. Am Beispiel einer Diashow sehen Sie, wie das prinzipiell funktioniert.

Die Bilder, die Sie in einer Diashow anzeigen lassen wollen, müssen Sie innerhalb eines div-Containers in Ihrer Webseite einbinden.

<div>

<img src="bilder/steine1.jpg" alt="Steine" >

<img src="bilder/steine2.jpg" alt="Steine" >

<img src="bilder/steine3.jpg" alt="Steine" >

</div>

Als Nächstes benötigen wir das jQuery-Plug-in für die Diashow, was wiederum eine JavaScript-Datei ist. Sie finden es unter <http://jquery.malsup.com/cycle2/download/>.

Innerhalb des head-Bereichs Ihres Dokuments schreiben Sie dann zuerst den Verweis auf jQuery und dann auf das jQuery-Plug-in.

<head>

<script

src="https://code.jquery.com/jquery-3.3.1.min.js"

integrity="sha256-FgpCb/KJQlLNfOu91ta32o/NMZxltwRo8QtmkMRdAu8="

crossorigin="anonymous"></script>

<script src="cycle.js"></script>  
</head>

Zum Starten der Diashow müssen Sie dem Container mit den Bildern eine vorgegebene Klasse zuweisen, nämlich cycle-slideshow:

<div id="bilder" class="cycle-slideshow">

<img src="bilder/steine1.jpg" alt="Steine">

<img src="bilder/steine2.jpg" alt="Steine">

<img src="bilder/steine3.jpg" alt="Steine">

</div>

Bei diesem Plug-in müssen Sie sogar keinen weiteren JavaScript-Code schreiben. Falls Sie die Anzeige noch modifizieren wollen, also beispielsweise eine andere Art des Übergangs wählen möchten, geht das ebenfalls rein über Angaben im HTML-Code:

<div id="bilder" class="cycle-slideshow"  
 data-cycle-fx="scrollHorz"  
 data-cycle-pause-on-hover="true"  
 data-cycle-speed="200">  
 <img src="bilder/steine1.jpg" alt="Steine">  
 <img src="bilder/steine2.jpg" alt="Steine">  
 <img src="bilder/steine3.jpg" alt="Steine">  
</div>

Dadurch ist definiert, dass als Funktion für den Übergang scrollHorz verwendet wird, beim Hovern soll außerdem die Diashow anhalten und die Geschwindigkeit beträgt 200 Millisekunden.



Statt die Dateien herunterzuladen, können Sie sie auch direkt von der Pluginseite verwenden. Sehen Sie sich dafür das Beispiel bei <http://jquery.malsup.com/cycle2/demo/basic.php> an.

Erstellen Sie eine weitere Unterseite und integrieren Sie die Diashow innerhalb des main-Bereichs!

Ergänzen Sie dann einen Link auf diese Unterseite in den anderen Seiten Ihres Layouts.