

HTML und CSS - Grundkurs für Einsteiger

Herzlich willkommen 😊

Kursinhalt

- HTML und CSS bilden ein mächtiges Werkzeuggespann, mit dem sich schon nach kurzer Lernphase private Webseiten wie auch einfache Firmenauftritte hervorragend gestalten lassen.
- Kursinhalt: Grundstruktur und Syntax von HTML; Meta-Informationen; Einfügen von Texten, Bildern und Links; Tabellen; Navigationsstruktur erstellen; Grundlagen CSS; Einbinden von CSS in HTML-Dokumente; Formatierung von Schrift, Absätzen, Tabellen, Rahmen etc.; Boxmodell und weitere CSS-Besonderheiten; Barrierefreiheit und Benutzerfreundlichkeit.

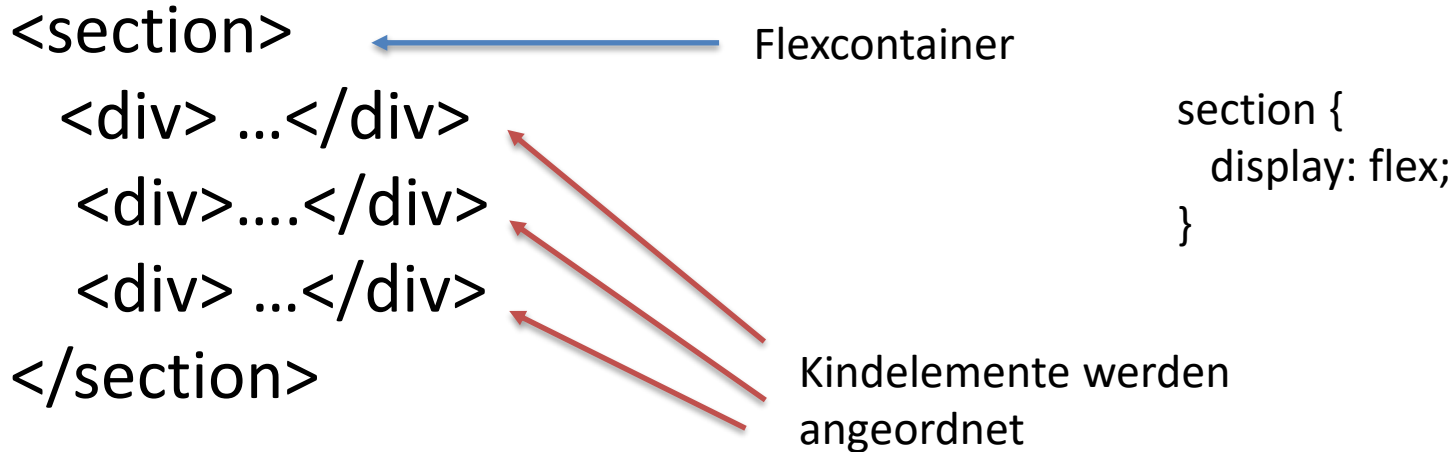
LAYOUTS ERSTELLEN

Techniken für Layouts

- float: Heute nur noch zur Anordnung einzelner Elemente, für Layouts veraltet
- position: Heute nur noch zur Anordnung einzelner Elemente, für Layouts veraltet
- Flexbox 😊
- Gridlayout 😊

Flexbox

- Mit Flexbox ordnen Sie die Kindelemente eines Containers an



Flexbox

- Mit `display: flex` werden Elemente nebeneinander angeordnet
- Zur Anordnung untereinander ergänzen Sie `flex-direction: column`

Layoutaufgabe 10

- Ergänzen Sie für nav ul die Angabe display: flex
- Ergänzen Sie auch für main die Angabe display: flex
- Machen Sie bitte die Links in der Navigation (nav a) zu Blockelementen über display: block

... UND RESPONSIV

Responsiv „anpassend“

- Sie können über so genannte CSS Media Queries die Eigenschaften des Ausgabegeräts abfragen
- ... und dann besondere Formatierungen definieren

Medien Abfragen

```
@media screen and (max-width: 700px) {  
    /* das gilt nur bis 700px Viewportbreite */  
    p {  
        background-color: red;  
    }  
}
```

Aufgabe 1

Testen Sie einmal die folgende Media Query aus und ändern Sie dann die Viewportgröße, um zu sehen, ob alles klappt.

```
@media screen and (max-width: 700px) {  
    /* das gilt nur bis 700px Viewportbreite */  
    body {  
        background-color: red;  
    }  
}
```

Ergänzen Sie weitere Formatierungen. Welche Anpassungen scheinen Ihnen sinnvoll?

Layoutaufgabe 11

Ergänzen Sie bitte am Ende Ihrer CSS-Datei Folgendes

```
@media screen and (max-width: 530px) {  
    main {  
        display: block;  
    }  
    article {  
        width: auto;  
    }  
}
```

Layoutaufgabe 12

Damit Bilder nie größer werden als die sie umgebenden Container, schreiben Sie außerdem in Ihre CSS-Datei

```
img {  
    max-width: 100%;  
    height: auto;  
}
```

Layoutaufgabe 13

- Erstellen Sie eine neue Unterseite durch Kopieren und speichern Sie sie unter dem Namen infos.html ab.
- Ändern Sie einzelne Texte/Inhalte ab.

USABILITY UND BARRIEREFREIHEIT

Usability und Barrierefreiheit

- Sind sehr wichtig.
- Wenn man korrektes HTML und CSS so schreibt, wie sie gedacht sind, macht man schon vieles richtig.

Barrierefreiheit

- Prüfen, ob sich Website mit Tastatur bedienen lässt
- Kontrast?
- Alternativtexte für Bilder?
- Keine Bilder mit Text
- Keine Informationen nur über Farben vermitteln
- Die passendsten HTML-Elemente verwenden

Usability

- Feedback: Was geschieht gerade, wo bin ich?
- Sprache/Bilder für Benutzer, nicht Firmenslang
- So wenig Einschränkungen des Benutzers wie möglich
- Konsistente Darstellung
- Besser Fehler vermeiden, als Nutzer auf Fehler hinweisen
- Mehr unter <https://www.nngroup.com/articles/ten-usability-heuristics/>

FERTIGE LAYOUTS UND MEHR LAYOUTTECHNIKEN

Fertige Layouts

- Layoutvorlagen nutzen, z.B.
https://www.w3schools.com/w3css/w3css_templates.asp
- Content Management-Systeme wie WordPress bieten viele kostenlose und kostenpflichtige Layoutvorlagen „Themes“:
<https://de.wordpress.org/themes/>

Framework nutzen

- Kein fertiges Layout, aber fertige Komponenten
- Bootstrap ist das bekannteste Frontend-Framework: <https://getbootstrap.com/>

Bootstrap - Grundprinzip

- CSS-Datei im Kopf verlinken
- JavaScript-Datei vor `</body>` verlinken
- Komponenten suchen, einfügen, anpassen

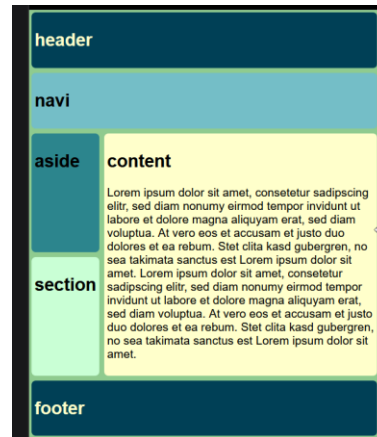
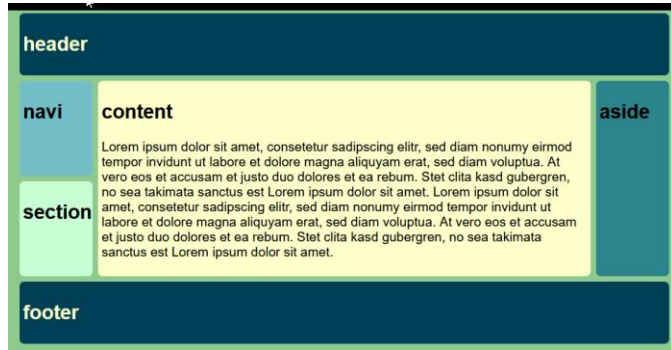
- <https://codesandbox.io/s/admiring-stonebraker-pqovl?file=/index.html>

Bootstrap ausprobieren

- <https://codesandbox.io/s/admiring-stonebraker-pqovl?file=/index.html>
- Ersetzen Sie die Links in der Card-Komponente durch Buttons.
- Die Buttons sollen unterschiedliche Formatierungen haben.
- Ergänzen Sie ein Accordion. Ändern Sie die Texte

Rasterlayout erstellen mit Grid

- Rasterlayout erstellen
- Basis: <https://codepen.io/FlorenceM/pen/QWgxdoj>
- Ergebnis: <https://codepen.io/FlorenceM/full/KKqeabV>



Feedback

- Was hat der Kurs Ihnen in Hinsicht auf Ihre Ziele gebracht?
- Was möchten Sie sonst noch loswerden?
- Haben Sie Anregungen, was die Form des Online-Unterrichts anbelangt?