

Kapitel 2 - Frontend

Flexbox

Flexbox, offiziell CSS Flexible Box Layout Module, ist eine neue Art und ein neues Konzept um eindimensionale Layouts auf Webseiten umzusetzen. Die herkömmliche Art Objekte auf einer Webseite zu positionieren ist, fixe Positionen und Maße zu vergeben.

Doch bei Flexbox werden bestimmte Regeln festgelegt, diese machen das Verhalten der Webseite vorhersagbar bei einer Veränderung der Bildschirmgröße. Anschließend ist es dem Browser überlassen, die Breite, Höhe, Position und Anordnung zu wählen.

Das Konzept

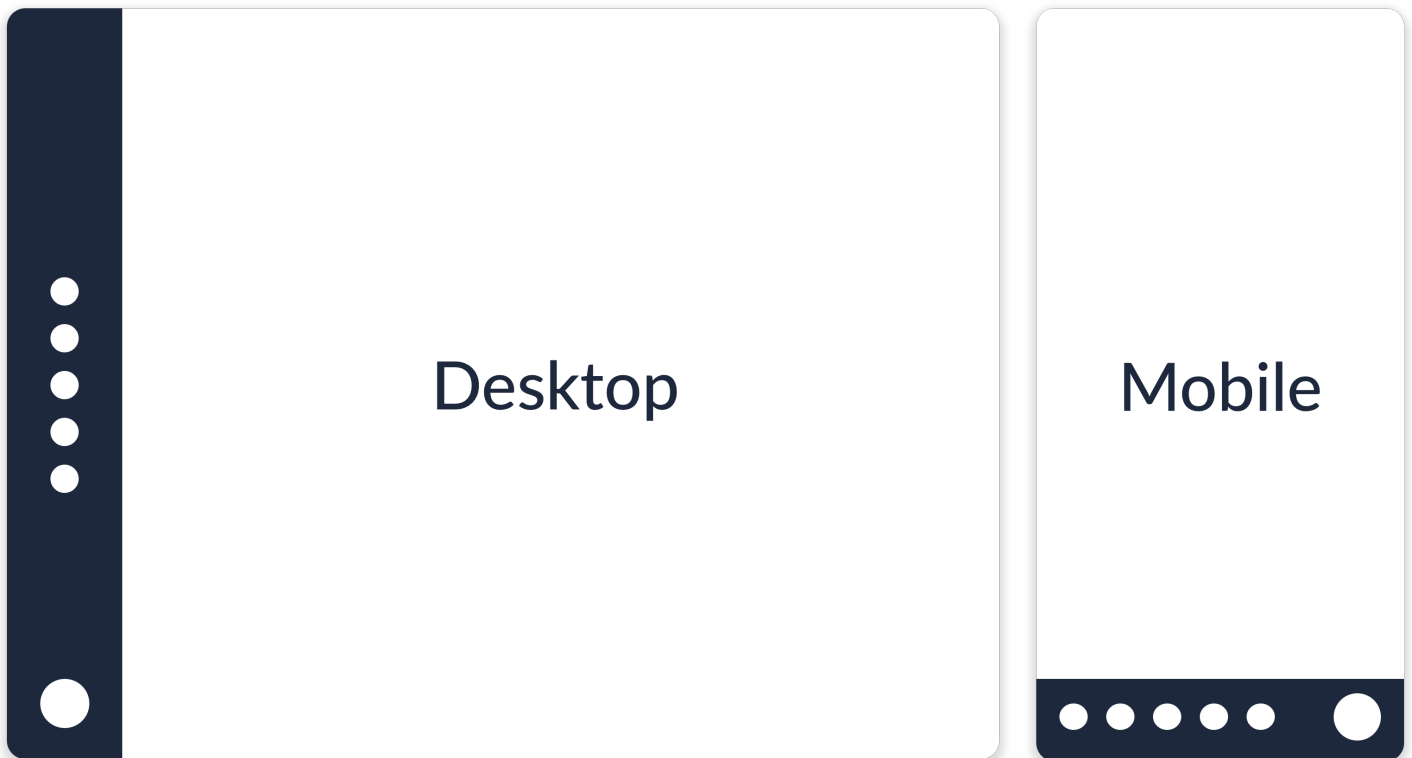
Die Grundidee ist es, dem Flex-Container die Möglichkeit zu geben, die Maße der Elemente so zu verändern, dass der Platz auf unterschiedlichen Bildschirmauflösungen bestmöglich ausgenutzt ist. Um das zu erzielen lässt das Elternelement die Tochterelemente je nach Bedarf wachsen oder schrumpfen.

technische Spezifikation

Innerhalb eines Divs können die einzelnen Elemente ihre Größe "flexibel" verändern. Sie wachsen, um freien Platz zu verwenden oder schrumpfen, um innerhalb des Elternobjekts zu bleiben und einen Overflow zu vermeiden. Der große Vorteil des Flexbox Layouts ist die Richtungsunabhängigkeit. Dadurch ist es sehr flexibel, was Orientierungsänderungen bei mobilen Geräten oder Auflösungsänderungen auf Desktop Geräten betrifft.

Erklärung anhand eines realen Beispiels

Auf dem Dashboard soll eine seitliche Navigation angezeigt werden, die auf mobilen Geräten an den unteren Rand des Bildschirms wandert, siehe Abbildung 1.



Mithilfe von Flexbox ist dieses Verhalten einfach zu erzielen.

Ich erstelle ein Elternelement mit folgenden Eigenschaften:

```
1 | .parent{  
2 |   display: flex;  
3 |   overflow: hidden;  
4 | }
```

Die Tochterelemente dieser Flexbox werden auf der horizontalen Hauptachse ausgerichtet. Der Overflow auf der X- und Y-Achse wird ausgeblendet. Die Navigation auf der Seite ist in folgendem Code-Block beschrieben.

Dieses Element ist durch `order:1` das erste Element in der Flexbox. Der Overflow auf der Y-Achse ist versteckt, um die Leiste zu fixieren. Weiters werden die Elemente innerhalb vertikal und horizontal zentriert und sind entlang der Y-Achse positioniert.

```
1 | .side-nav{  
2 |   display: flex;  
3 |   order: 1;  
4 |   justify-content: center;  
5 |   align-items: center;  
6 |   flex-direction: column;  
7 | }
```

Das Inhaltselement hat `order:2` damit es neben dem ersten auf der X-Achse positioniert wird. Ebenso ist

der Overflow auf der Y-Achse versteckt.

```
1 .content{
2   overflow-y: hidden;
3   display: flex;
4   justify-content: center;
5   flex-direction: column;
6   order: 2;
7 }
```

Damit die Navigation auf mobilen Geräten am unteren Rand positioniert ist, benötigen wir eine Media Query. Mithilfe dieser können CSS-Stile anhand von verschiedenen Eigenschaften wie z.B. Bildschirmauflösung oder Seitenverhältnis manipuliert werden. Im untenstehenden Code-Block wird dies veranschaulicht. Indem wir die Hauptachse des Flexbox Elternelements auf die Y-Achse ändern, werden die beiden Tochterelemente nun vertikal verteilt. Damit nun auch die Navigation unter dem Inhalt positioniert ist ändern wir die order auf 2. Weiters müssen die Höhe und Breite angepasst werden.

```
1 @media (max-width: 576px){
2   .parent{
3     flex-direction: column; //changed
4   }
5
6   .side-nav{
7     order: 2; //changed
8     width: 100vw; //changed
9     height: 66px; //changed
10  }
11 }
```