

# LU04a - Flexbox

## CSS Display

Die CSS-Anweisung „display“ bestimmt, wie ein HTML-Element auf der Webseite dargestellt wird. Sie kontrolliert das Layout und das Verhalten des Elements im Browserfenster.

### Die verschiedenen Display-Werte

Die Auswahl des richtigen „display“-Werts hängt von den Anforderungen des Layouts und dem gewünschten Verhalten der Elemente ab. Durch die geschickte Verwendung von „display“ können Entwickler das Erscheinungsbild und die Funktionalität von Webseiten effektiv steuern. Hier ist eine Übersicht über die wichtigsten Werte, die für „display“ verwendet werden können:

Wert	Beschreibung
<b>block</b> (Block-Element)	Block-Elemente nehmen die gesamte verfügbare Breite ihres übergeordneten Elements ein und beginnen auf einer neuen Zeile. Beispiele für Block-Elemente sind <code>&lt;div&gt;</code> , <code>&lt;p&gt;</code> , <code>&lt;h1&gt;</code> - <code>&lt;h6&gt;</code> , <code>&lt;header&gt;</code> , <code>&lt;footer&gt;</code> .
<b>inline</b> (Inline-Element)	Inline-Elemente nehmen nur so viel Breite wie nötig ein und bleiben im selben Absatz oder auf derselben Zeile wie benachbarte Inline-Elemente. Beispiele für Inline-Elemente sind <code>&lt;span&gt;</code> , <code>&lt;a&gt;</code> , <code>&lt;b&gt;</code> .
<b>inline-block</b> (Inline-Block-Element)	Kombiniert Eigenschaften von Block- und Inline-Elementen. Es nimmt nur so viel Breite wie nötig ein und kann dennoch mit anderen Inline- oder Block-Elementen in derselben Zeile stehen. Häufig verwendet für Elemente, die als Block formatiert werden müssen, aber in einer Zeile bleiben sollen, z. B. <code>&lt;button&gt;</code> .
<b>none</b> (Ausgeblendetes Element)	Das Element wird nicht gerendert und nimmt keinen Platz in der Layout-Struktur ein. Es ist praktisch, um Elemente auszublenden oder dynamisch zu verbergen. Verwendet für Elemente, die vorübergehend nicht angezeigt werden sollen, z. B. auf verschiedenen Browsergrößen.
<b>flex</b> (Flexbox-Element)	Aktiviert das Flexbox-Layoutmodell für das Element, mit dem Sie flexible Layouts erstellen können, indem Sie den Inhalt auf verschiedene Weise anordnen und ausrichten. Flexbox ist besonders nützlich, um komplexere Layouts zu erstellen und das Responsiveness von Websites zu verbessern.
<b>grid</b> (Grid-Element)	Aktiviert das Grid-Layoutmodell für das Element, mit dem Sie Inhalte in einer zweidimensionalen Rasteranordnung organisieren können. Grid ist leistungsstark für die Erstellung von Layouts mit mehreren Spalten und Zeilen und bietet präzise Kontrolle über die Positionierung von Elementen.

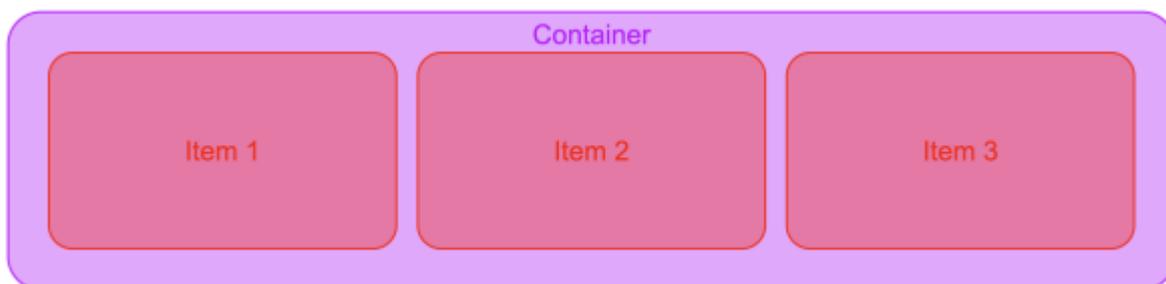
Eine komplette Liste aller verfügbaren Werte finden Sie [hier](#).

# CSS Flexbox

CSS Flexbox ist ein leistungsstarkes Layoutmodell in CSS, das entwickelt wurde, um das Design von Webseiten und Webanwendungen zu vereinfachen und flexible, dynamische Layouts zu ermöglichen. Es bietet eine effiziente Möglichkeit, Elemente innerhalb eines Container-Elements zu organisieren, auszurichten und zu verteilen, unabhängig von ihrer Grösse oder Reihenfolge.

## Grundkonzept

Die Grundkonzepte von Flexbox umfassen das Container-Element und die darin enthaltenen Items. Der Flex-Container fungiert als Rahmen für das Flexbox-Layout und organisiert die Flex-Items innerhalb. Diese Items können horizontal oder vertikal angeordnet werden, je nach Ausrichtung der Hauptachse. Die Hauptachse definiert die Richtung, in der die Flex-Items im Container angeordnet sind, während die Querachse senkrecht dazu steht und die Ausrichtung der Items in Bezug auf die Hauptachse beeinflusst. Das Verständnis dieser Grundkonzepte ist entscheidend für die effektive Nutzung von Flexbox zur Erstellung dynamischer und flexibler Layouts in CSS.



## Eigenschaften des Containers

Eigenschaft	Beschreibung
flex-direction	Bestimmt die Richtung der Hauptachse. Mögliche Werte sind: row, row-reverse, column, column-reverse.
flex-wrap	Kontrolliert, ob die Items in einer einzigen Zeile bleiben oder bei Bedarf in mehrere Zeilen umgebrochen werden. Mögliche Werte sind: nowrap, wrap, wrap-reverse.
justify-content	Definiert die Ausrichtung der Items entlang der Hauptachse. Dies beeinflusst, wie überschüssiger Platz auf der Hauptachse verteilt wird.
align-items	Bestimmt die Ausrichtung der Items entlang der Querachse, wenn sie auf der Hauptachse nicht den gesamten verfügbaren Platz einnehmen.
align-content	Steuert die Ausrichtung und Verteilung mehrerer Zeilen von Items entlang der Querachse.

## Eigenschaften des Items

Eigenschaft	Beschreibung
order	Ändert die Reihenfolge der Items innerhalb des Flex-Containers. Items mit einer niedrigeren Reihenfolge werden zuerst angezeigt.
flex-grow	Legt fest, wie viel zusätzlichen Platz ein Flex-Item entlang der Hauptachse einnehmen kann, im Verhältnis zu anderen Flex-Items.

Eigenschaft	Beschreibung
flex-shrink	Definiert, wie stark ein Flex-Item kleiner werden kann, um sich an den verfügbaren Platz anzupassen, im Verhältnis zu anderen Flex-Items.
flex-basis	Legt die bevorzugte Anfangsgrösse eines Flex-Items entlang der Hauptachse fest, bevor zusätzlicher Platz verteilt wird.
flex	Eine verkürzte Schreibweise für flex-grow, flex-shrink und flex-basis in einem Wert.
align-self	Überschreibt die Ausrichtungsvorgaben des Flex-Containers für ein einzelnes Flex-Item.

## Ressourcen

- [CSS-Tricks: A Complete Guide to Flexbox](#)
- [W3S: CSS Flexbox](#)

From:

<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:

<https://wiki.bzz.ch/modul/m287/learningunits/lu04/flexbox?rev=1742537962>

Last update: **2025/03/21 07:19**

