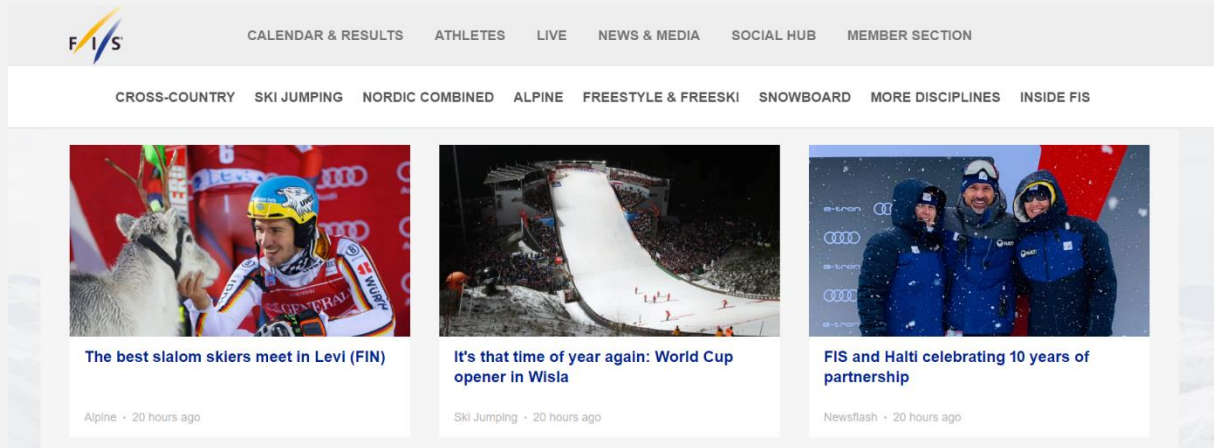


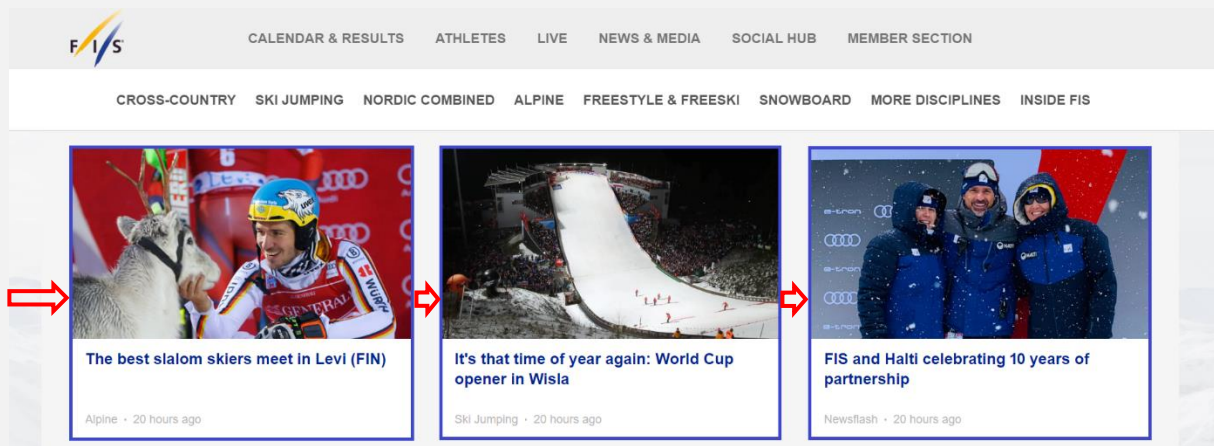
## Umfließen

Für die gestalterische Freiheit ist es notwendig, **Objekte nebeneinander zu positionieren**.

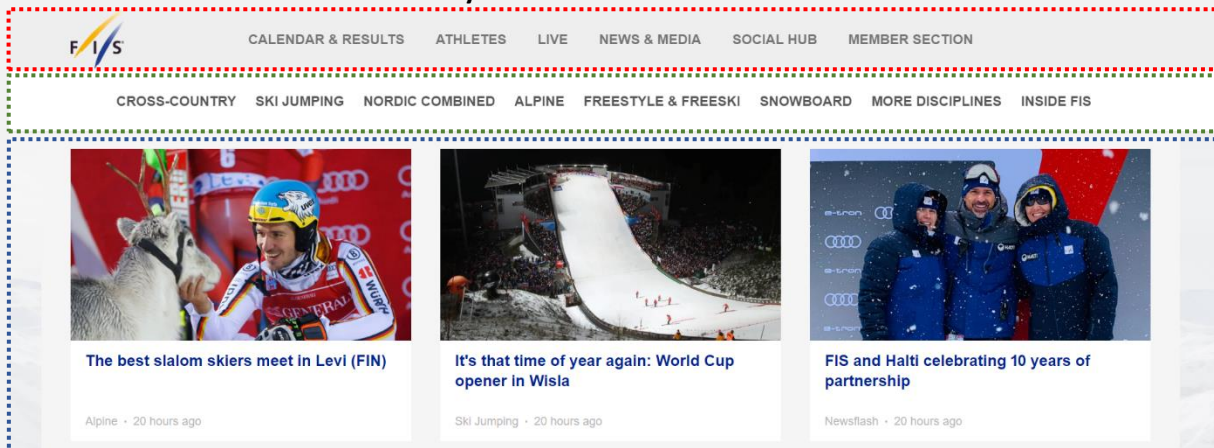
## Beispiel FIS Homepage




## Analyse: 3 Spalten Layout




## Analyse im Detail: mehrere Zeilen




## Analyse im Detail: mehrere Spalten pro Zeile

	CALENDAR & RESULTS	ATHLETES	LIVE	NEWS & MEDIA	SOCIAL HUB	MEMBER SECTION	
CROSS-COUNTRY	SKI JUMPING	NORDIC COMBINED	ALPINE	FREESTYLE & FREESKI	SNOWBOARD	MORE DISCIPLINES	INSIDE FIS




**The best slalom skiers meet in Levi (FIN)**

Alpine • 20 hours ago



**It's that time of year again: World Cup opener in Wisla**

Ski Jumping • 20 hours ago



**FIS and Hälti celebrating 10 years of partnership**

Newsflash • 20 hours ago

## Analyse im Detail: mehrere Spalten pro Zeile

	CALENDAR & RESULTS	ATHLETES	LIVE	NEWS & MEDIA	SOCIAL HUB	MEMBER SECTION	
CROSS-COUNTRY	SKI JUMPING	NORDIC COMBINED	ALPINE	FREESTYLE & FREESKI	SNOWBOARD	MORE DISCIPLINES	INSIDE FIS



**The best slalom skiers meet in Levi (FIN)**

Alpine • 20 hours ago



**It's that time of year again: World Cup opener in Wisla**

Ski Jumping • 20 hours ago



**FIS and Hälti celebrating 10 years of partnership**

Newsflash • 20 hours ago

## Analyse im Detail: mehrere Tags pro Spalte

	CALENDAR & RESULTS	ATHLETES	LIVE	NEWS & MEDIA	SOCIAL HUB	MEMBER SECTION	
CROSS-COUNTRY	SKI JUMPING	NORDIC COMBINED	ALPINE	FREESTYLE & FREESKI	SNOWBOARD	MORE DISCIPLINES	INSIDE FIS



**The best slalom skiers meet in Levi (FIN)**

Alpine • 20 hours ago



**It's that time of year again: World Cup opener in Wisla**

Ski Jumping • 20 hours ago



**FIS and Hälti celebrating 10 years of partnership**

Newsflash • 20 hours ago

## Flexboxen | Positionierung mit Eigenschaft *flex*

- Nebeneinander platzierte Elemente müssen eine **definierte Breite (*width*)** besitzen.
- Die Elemente werden in einem **Container** platziert, der als Flexbox fungiert. Diesem wird die CSS-Anweisung ***display: flex*** zugewiesen.

### Syntax

```
display: flex;
```

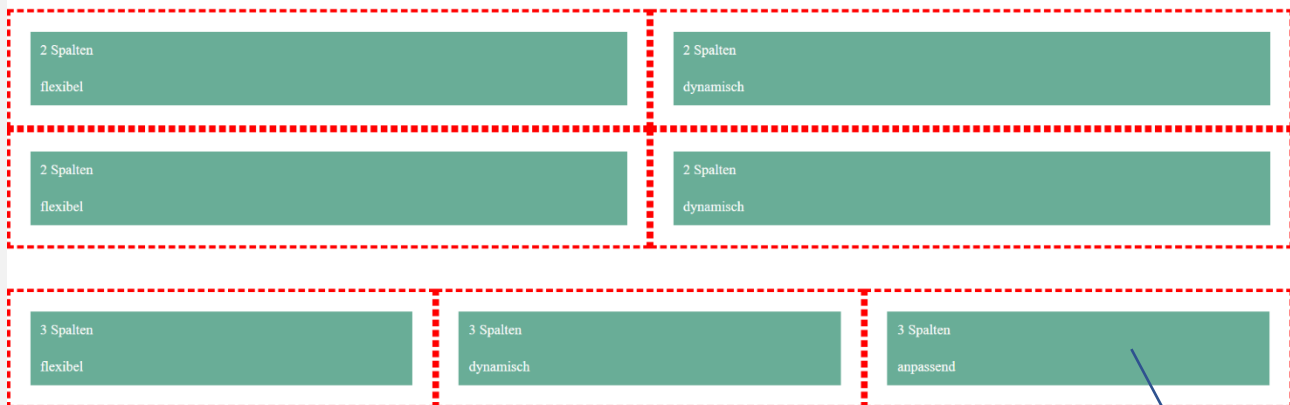
### Beispielcode:

Unterordner 8.1\_Beispiel\_CssFlex

## Übung 29 Fließende Boxen

Erstelle eine Webseite mit folgendem **bildschirmfüllenden Layout**.

### Spaltenlayout 1



```
<div class="threeColumns">...
```

```
<p>3 Spalten</p>  
<p>anpassend</p>
```

### Hinweis:

- Verwende für die **Boxen DIV-Container!**
- Verwende **Flexboxen!** Benutze das Beispiel 8.1\_Beispiel\_CssFlex als Vorlage und passe es an!
- Du musst keinen roten Rahmen setzen, dieser zeigt nur die einzelnen DIV-Container

## Umfließen mit Float und Display [VERALTETE TECHNIK]

Positionierung mit Eigenschaft *float**(engl. gleiten, schwimmen, fließen)*

- Umfließende Elemente müssen eine **definierte Breite (*width*)** besitzen.
- Dem gleitenden Element wird die CSS-Anweisung ***float: left*** zugewiesen.
- Das Gleiten der Tags wird mit ***clear: both*** beendet.

## Syntax

```
float: left | right | none;  
clear: left | right | both;
```

## Beispielcode:

Unterordner 8.1\_Beispiel\_CssFloat

Positionierung mit Eigenschaft *display*

Es gibt drei unterschiedliche Anzeigevarianten:

- **Inline Element:**
  - Benötigt exakt so viel Platz wie der Inhalt (Text)
  - Kann per CSS keine Breite und keine Höhe haben
  - Elemente werden nebeneinander platziert (in gleicher Zeile)
- **Block Element:**
  - Benötigt per Default gesamte Breite
  - Kann eine eigens gewählte Breite und Höhe haben
  - Elemente beginnen in einer neuen Zeile
- **Inline-Block Element:**
  - Beansprucht wie ein Blockelement einen bestimmten Platz
  - Verhält sich aber wie ein Inline Element: Platzierung in der gleichen Zeile

## Syntax

```
display: inline | block | inline-block;
```

## Beispielcode:

Unterordner 8.1\_Beispiel\_CssDisplay



## HTML Codehilfe für Floating und Inline-Blöcke [VERALTETE TECHNIK]

### ➤ Möglichkeit 1: Inline-Block

```
<div class="row">
  <div class="twoColumns">
    <!-- Inhalt von Spalte 1 hier -->
  </div>
  <div class="twoColumns">
    <!-- Inhalt von Spalte 2 hier -->
  </div>
</div>
```

```
.row {
  font-size: 0;
}
```

```
.twoColumns {
  font-size: initial;
}
```

CSS-Code

Achtung: bei Inlineblöcken erscheint im Browser leider ein kleiner unerwünschter Abstand zwischen den Blöcken. Ein Trick ist, die Schriftgröße des übergeordneten Elements auf 0 zu setzen.

### ➤ Möglichkeit 2: float

```
<div class="clearfix">
  <div class="box">
    <!-- Inhalt von Zelle 1 hier -->
  </div>
  <div class="box">
    <!-- Inhalt von Zelle 2 hier -->
  </div>
  <div class="box">
    <!-- Inhalt von Zelle 3 hier -->
  </div>
</div>
```

```
.clearfix::after {
  content: "";
  clear: both;
  display: table;
}
```

CSS-Code

Achtung: Floating Elements nehmen keinen Platz im Browser ein, solange nicht mit *clear:both* das Umfließen beendet wird. Für ein Spaltenlayout benötigt man deshalb noch einen DIV-Container (clearfix), der das Umfließen sauber beendet.

## CSS Codehilfe für Inline-Blöcke [Übung 29]

### ➤ *twoColumns* Klasse

```
display: inline-block;
box-sizing: border-box;
width: 50%;
padding: 20px;
```



*box-sizing: border-box* bewirkt, dass border und padding die Gesamtbreite nicht mehr verändern – sehr praktisch bei prozentuellen Werten!

### ➤ *threeColumns* Klasse

- Analog wie *twoColumns*
- *width: 33.3333%*



Prozentuelle Werte beziehen sich immer auf das übergeordnete Element. Damit lassen sich flexible Layouts umsetzen.

### ➤ *<p>* Tag

- Schriftfarbe #fff
- Hintergrundfarbe #69ad97
- Innenabstände 10 Pixel