

## Positionieren

### Positionsarten

Die **CSS Eigenschaft *position*** definiert die Positionierungsmethode:

- |                             |                    |   |
|-----------------------------|--------------------|---|
| ➤ <b>position: static</b>   | <b>statisch</b>    | normaler Platzanspruch im Boxmodell<br>keine besondere Positionierung                         |
| ➤ <b>position: relative</b> | <b>relativ</b>     | normaler Platzanspruch im Boxmodell<br>Verschiebung ausgehend von der ursprünglichen Position |
| ➤ <b>position: absolute</b> | <b>absolut</b>     | kein Platzanspruch im Boxmodell<br>Verschiebung ausgehend vom Elternelement                   |
| ➤ <b>position: fixed</b>    | <b>fixiert</b>     | kein Platzanspruch im Boxmodell<br>feste Positionierung am Viewport (Bildschirm)              |
| ➤ <b>position: sticky</b>   | <b>unbeweglich</b> | normaler Platzanspruch im Boxmodell<br>ab gewissem Offset feste Positionierung am Viewport    |

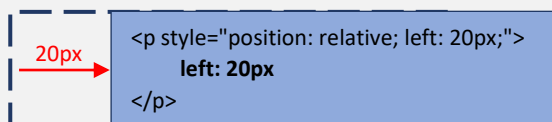
### Positionsverschiebungen

Für die Positionsarten relative, absolute, fixed und sticky kann man nun Verschiebungen angeben:

- **top**
- **right**
- **bottom**
- **left**

**Achtung:** die Verschiebungen geben den Einzug der jeweiligen Seite an. Zum Beispiel:

**left: 20px;** rückt ein Element um **20 Pixel nach rechts!**



### Beispiele

Unterordner 9.1\_Beispiel\_Positionierung

**W3Schools:** [https://www.w3schools.com/css/css\\_positioning.asp](https://www.w3schools.com/css/css_positioning.asp)

## Übung 32 Fixed Angular

Kopiere die Daten deiner fertigen Angular Webseite (Übung 31).

Erstelle eine im Browserfenster **fixierte Navigationsleiste**, d.h. sie soll immer sichtbar sein und auch nicht beim Scrollen verschwinden. Die Position bleibt unverändert (wie bisher ganz oben).

### How-To:

- DIV-Container des Headers *position: fixed;*  
*top: 0px;*  
*width: 100%;*
- Vermutlich ist jetzt der erste Teil der Webseite am Anfang abgeschnitten.  
In diesem Fall dem Inhalt (#main ID) oben mehr Innenabstand/Padding geben,  
damit der Inhalt weiter nach unten rutscht und ganz sichtbar wird.
- Kontrolliere, ob die Navigationsleiste auf beiden Seiten (Home und Features) funktioniert und sauber positioniert ist.
- Wahrscheinlich wirst du auch den Effekt sehen, dass die Nav-Leiste sich teils hinter Elementen versteckt. Diese Verdeckung musst du mit dem z-index beheben:



Bei Verschiebungen entstehen zwangsläufig Überlappungen – Elemente verdecken sich gegenseitig. Die Frage ist: welches Element ist voll sichtbar und welches Element wird verdeckt? Dazu kann man jedem Tag einen **z-index** zuweisen (jede beliebige ganze Zahl  $z \in \mathbb{Z}$ ). Jenes Tag mit dem höheren Index gewinnt und ist oben (und somit voll sichtbar).

```
<p style="z-index: 2;">  
  z-index: 2  
</p>
```

```
<p style="z-index: 1;">  
  z-index: 1  
</p>
```

Achtung: *position* darf nicht static sein, sonst wird der z-index ignoriert!

## Übung 33 Sticky Angular

Kopiere die Daten deiner fertigen Angular Webseite (Übung 31 oder 32).

Erstelle im Browserfenster eine **Sticky Navigationsleiste**, d.h.

- über der Navigationsleiste ist anfangs ein Banner (die erste Zeile des ursprünglichen Contents)
- beim Scrollen verschwindet der Banner und die Navigationsleiste wird oben fixiert

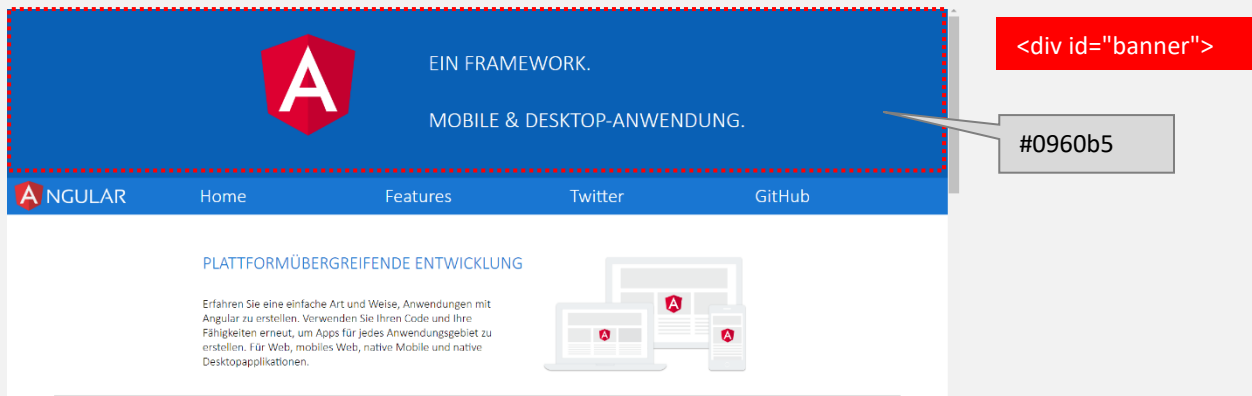
### How-To:

- Schritt 1: schichte deinen HTML-Code um, damit der Banner ganz oben ist.
- Schritt 2: ändere die Hintergrundfarbe des Banners
- Schritt 3: weise dem Header *position: sticky* und *top: 0px* zu.

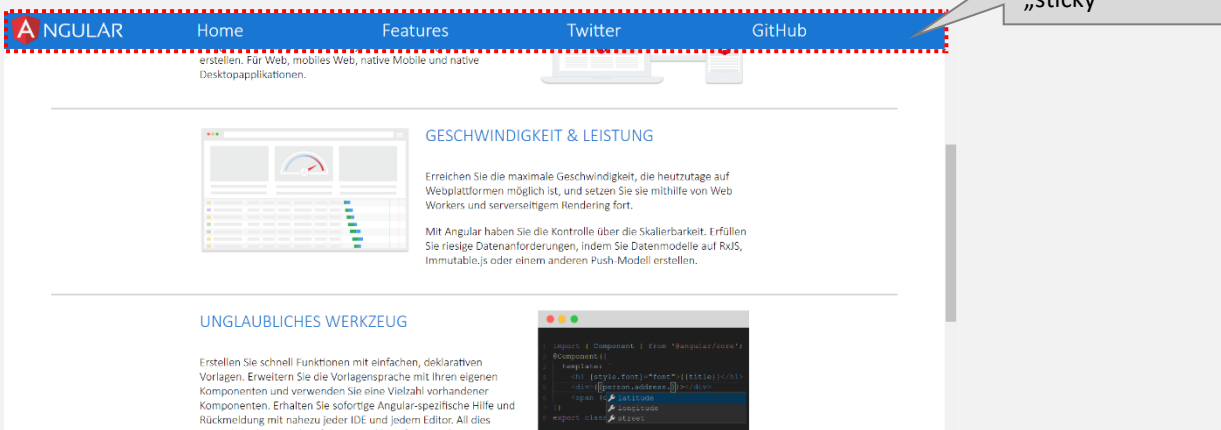


Bei *position:fixed* und *position:sticky* bestimmt der Wert *top* die Verschiebung im fixierten Browserfenster. Bei *top:0px* daher keine Verschiebung, das fixierte Element ist ganz oben.

### Screenshot (ohne Scrollen)



### Screenshot (mit Scrollen)



## Übung 34 Makro Header

Kopiere die Daten deiner fertigen Makrofotografie Webseite (Übung 30).

Ändere deinen HTML und CSS Code, sodass die Hauptüberschrift mittig über dem Bild platziert ist!

**How-To:**

- Das Bild und die Überschrift zusammen in einen Container verschachteln. Dieses Elternelement bestimmt den Bezugspunkt der Verschiebung.

```
<div id="header">
  
  <h1>Makrofotografie</h1>
</div>
```

- In CSS: Header relativ positionieren (als Bezugspunkt)

```
#header {
  position: relative;
}
```

- Das <h1> Tag umpositionieren:

```
position: absolute;
top: 50%;
left: 50%;
transform: translate(-50%, -50%);
```

- Oberen und unteren Rahmen des <h1> Tags definieren

## Übung 35 Surprise!

Schreibe eine Homepage mit **zwei Überraschungsboxen!**

Wenn der Mauszeiger über einer Box hovers, soll das Logo durch ein Auto ausgetauscht werden.

**Löse die Übung auf 2 Arten** (d.h. insgesamt vier Überraschungsboxen):

➤ **Variante 1**

- Verwende **<img> Tags**, die einander verdecken (das Logo liegt über dem Autobild)
- Verwende für die Positionierung des Vorhangs **position: absolute** (die beiden Bilder Vorhang und Inhalt liegen dadurch direkt übereinander)
- Verwende für die Überdeckung den **z-index**.
- Mit **hover** muss dann nur der **z-index** der Bilder richtig ausgetauscht werden.  
Du wirst dabei zwei Hover-Events brauchen
  1. Für das Verstecken des Logos
  2. Für das nach vorne Schieben des Autos

➤ **Variante 2**

- Verwende **<div> Boxen**, die als **Hintergrundbild per CSS** das Logo anzeigen.
- Mit **hover** wird dann einfach das Hintergrundbild ausgetauscht (beim Hover-Event nur die Eigenschaft `background-image: url(...)` verändern)

**Ausgangsdateien für die Übung:** siehe Ordner 9.1\_Surprise!

**Beispiel:**



Denke an die **Entwicklerumgebung** (F12 oder **Rechtsklick -> untersuchen**).

So kannst du nachsehen, wo und warum ein Element gerade so platziert wurde!