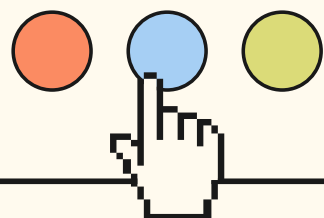




07/11/2024



FOUNDATIONS OF WEB DESIGN

CLASS 6

**Box Model
and CSS
Layout**

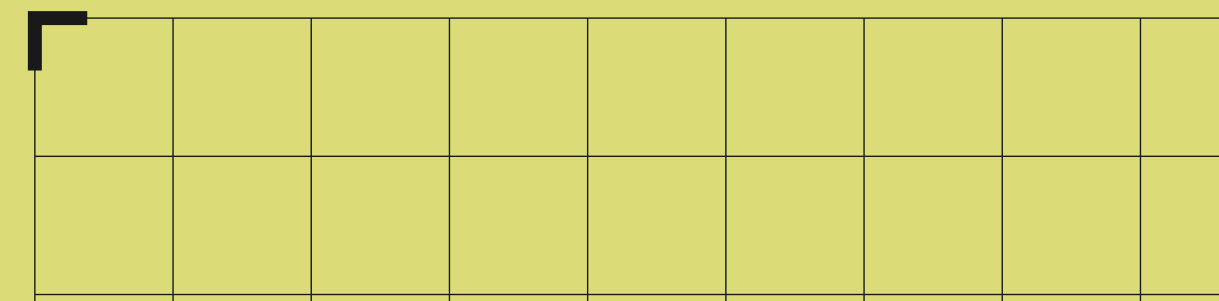
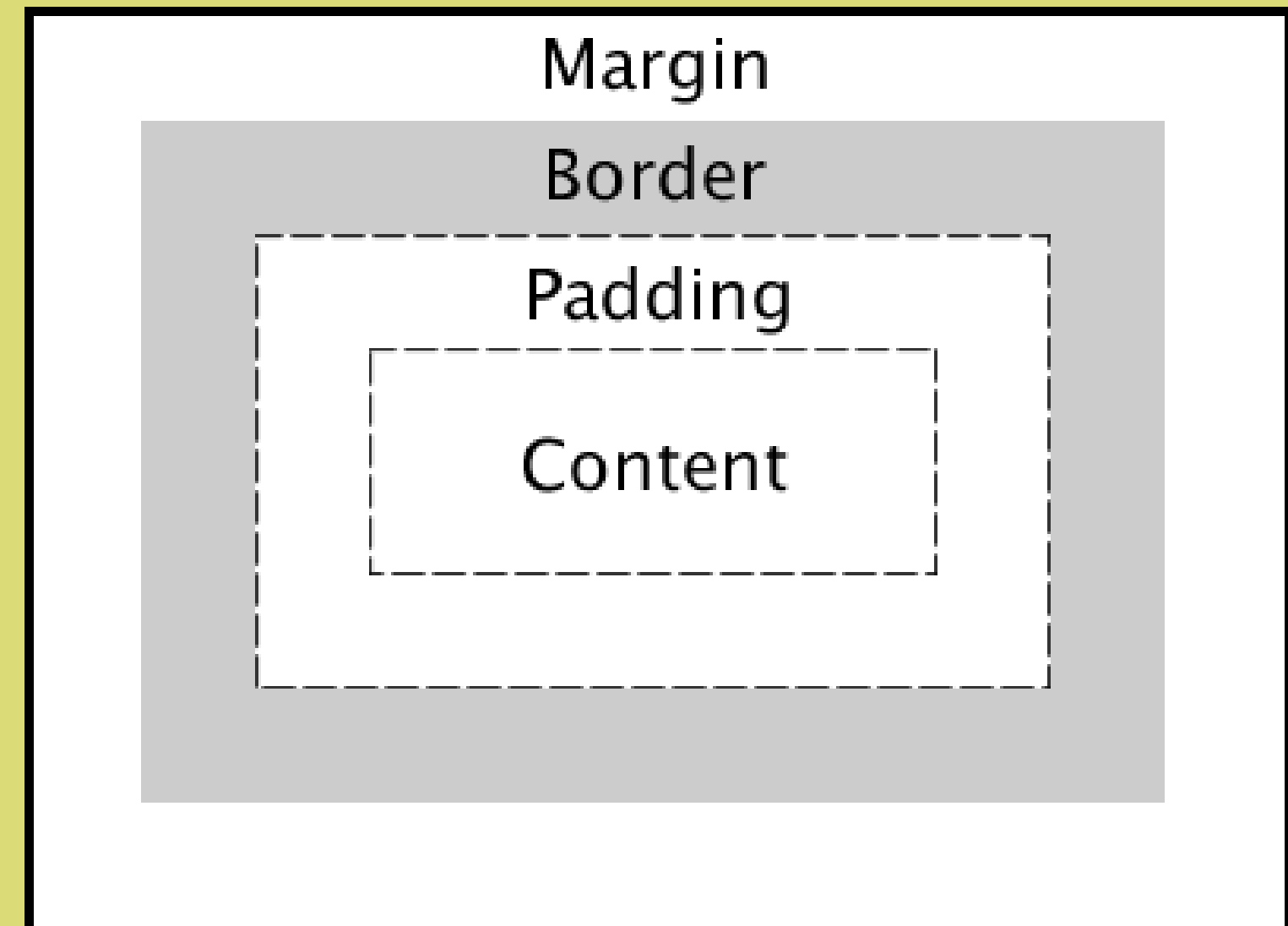
<https://dem-1323.github.io/foundations-of-web-design/>

CSS BOX MODEL



Structure

- **Content**
- **Padding**
- **Border**
- **Margin**



CSS POSITIONING

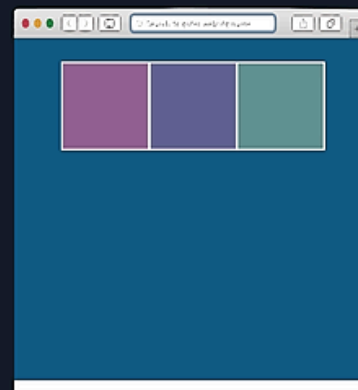


Lets you place HTML elements where you want them

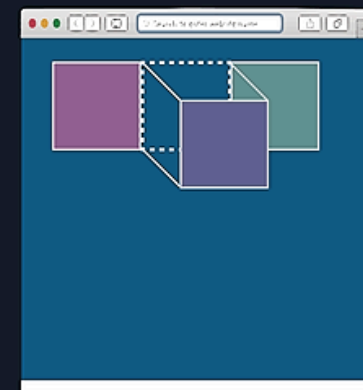
Types of Positioning

- **Static (default)**
- **Relative**
- **Absolute**
- **Fixed**
- **Sticky**

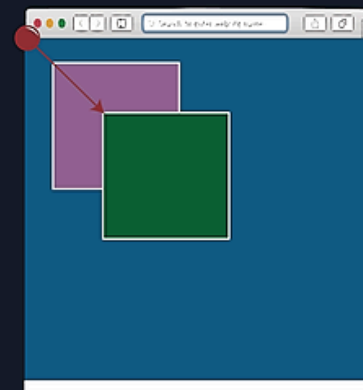
CSS Position Property



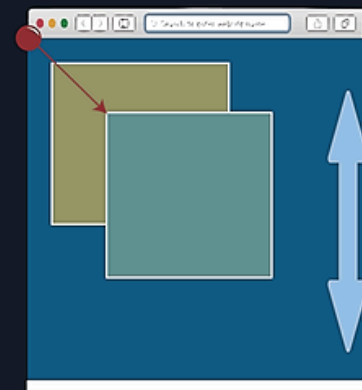
Static



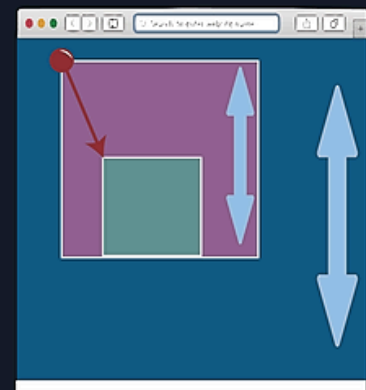
Relative



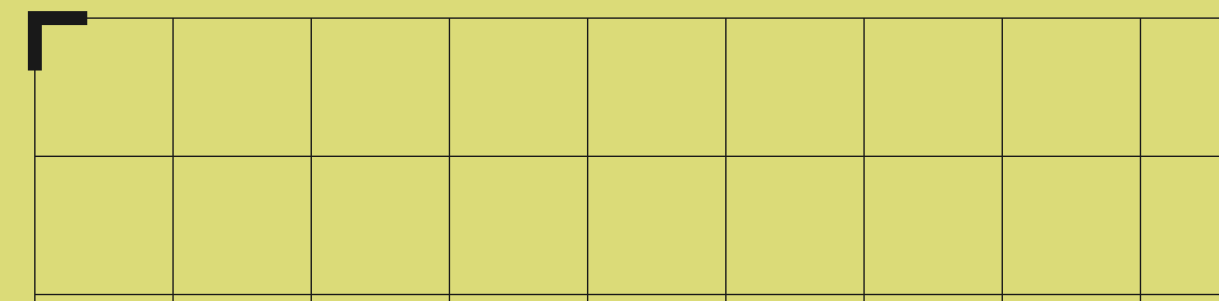
Absolute



Fixed



Sticky



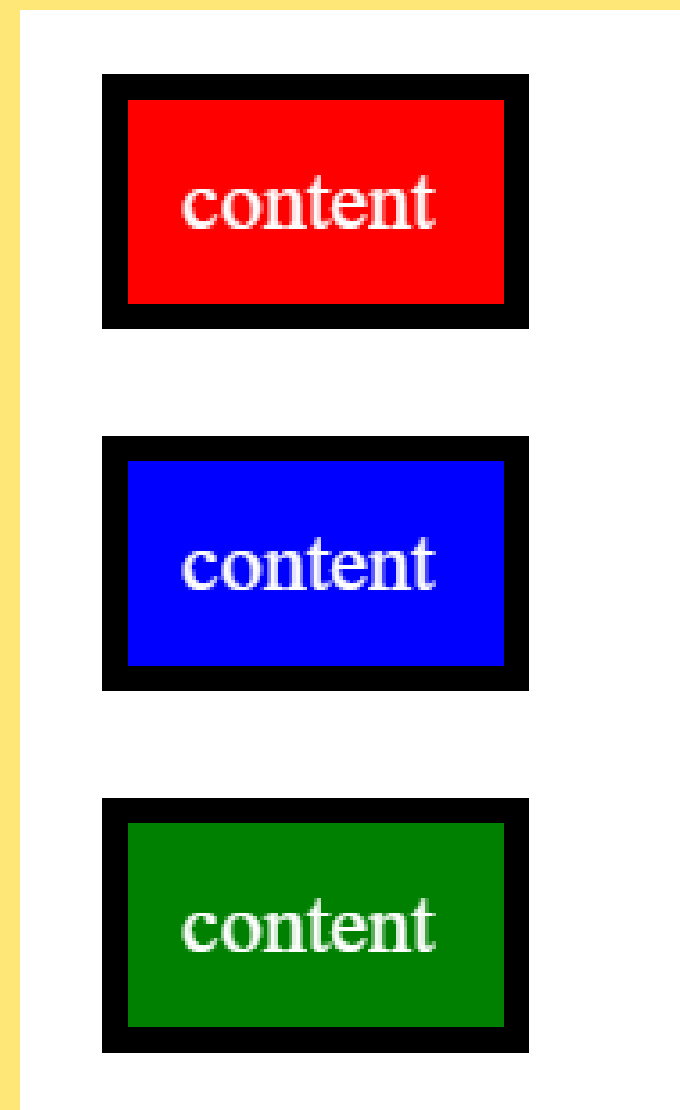
RELATIVE

An element with 'position: relative;' is positioned relative to its normal position.

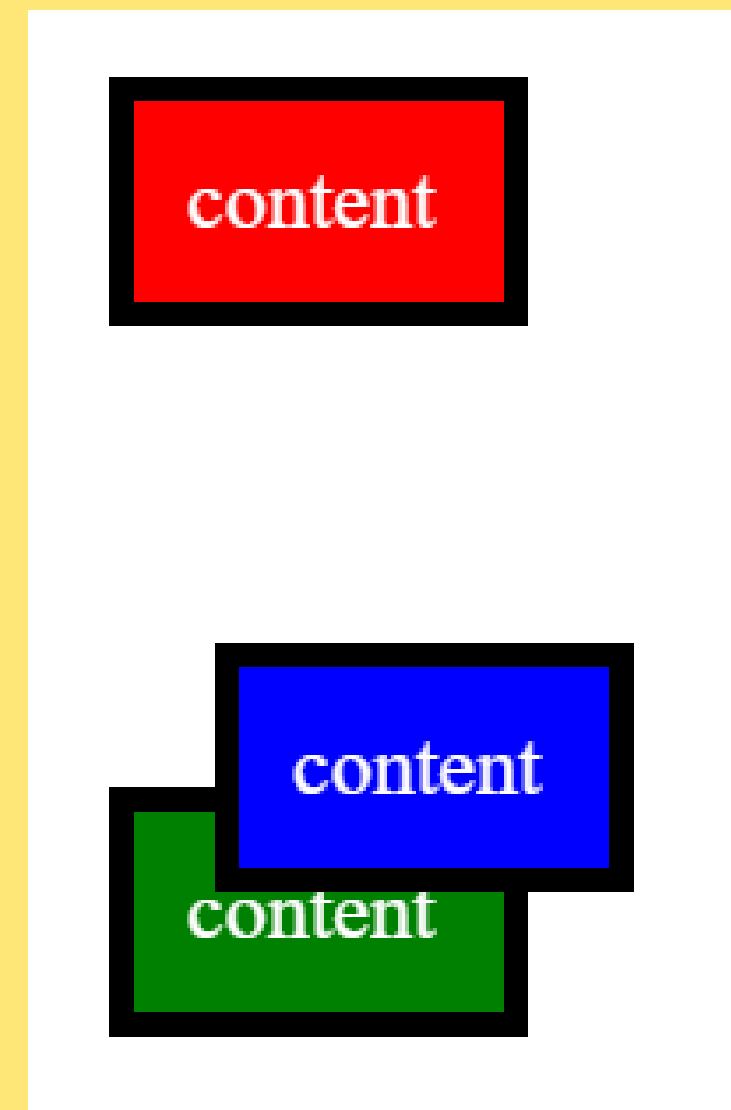
Example:

```
#blue {  
  background-color: blue;  
  position: relative;  
  left: 20px;  
  top: 40px;  
}
```

Static



Relative



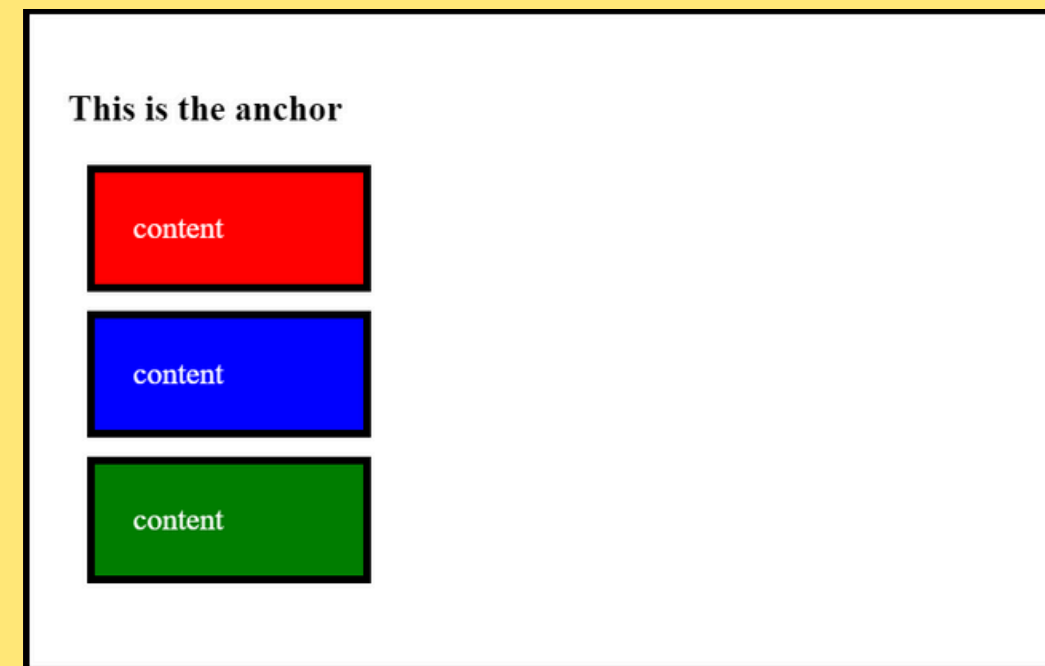
ABSOLUTE

Static

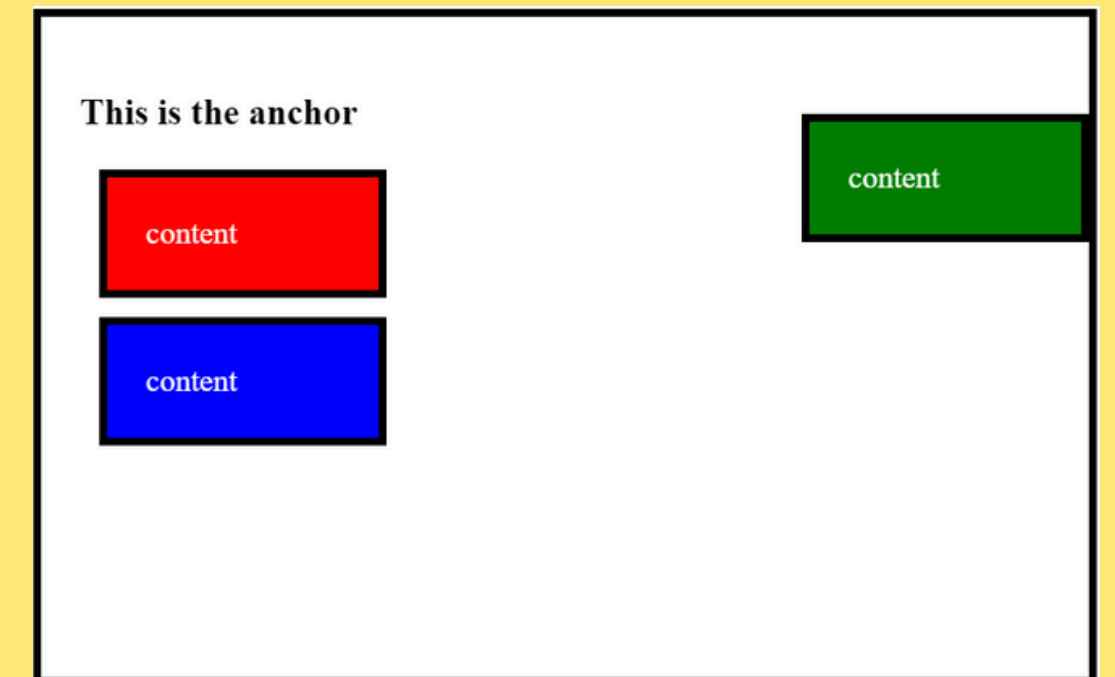
Absolute positioning places an element exactly where you want it, based on its nearest positioned ancestor.

Example:

```
#red {  
  background-color: red;  
  position: fixed;  
  left: 18px;  
  top: 354px;  
  margin: 0;  
}
```



Absolute



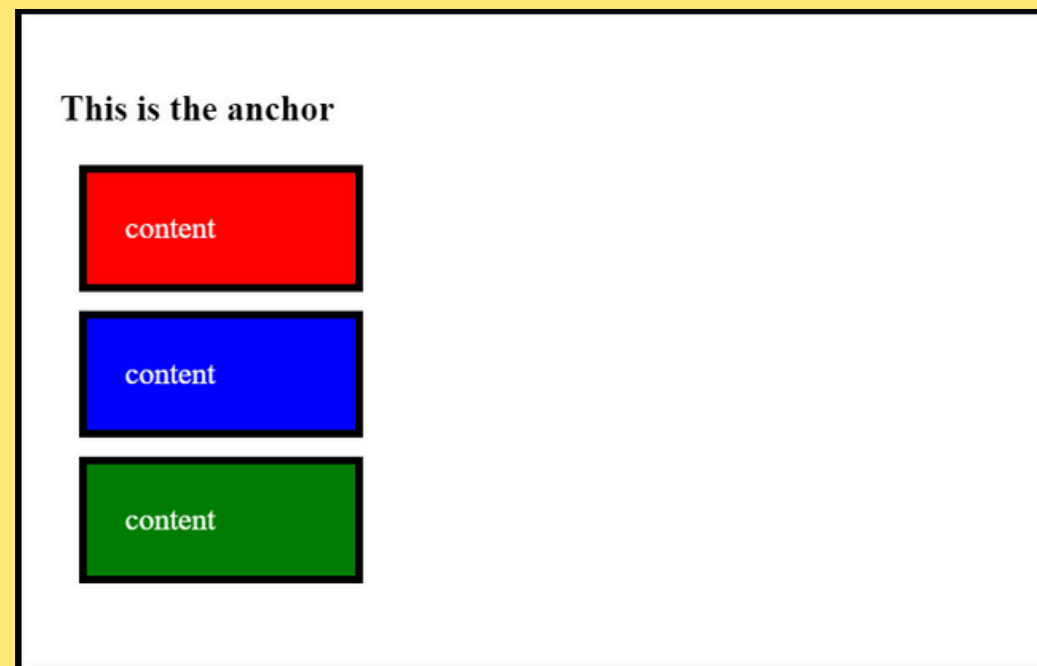
FIXED

Static

Fixed positioning keeps an element in the same place in the browser window, even when you scroll.

Example:

```
#red {  
  background-color: red;  
  position: fixed;  
  left: 18px;  
  top: 354px;  
  margin: 0;  
}
```



Fixed



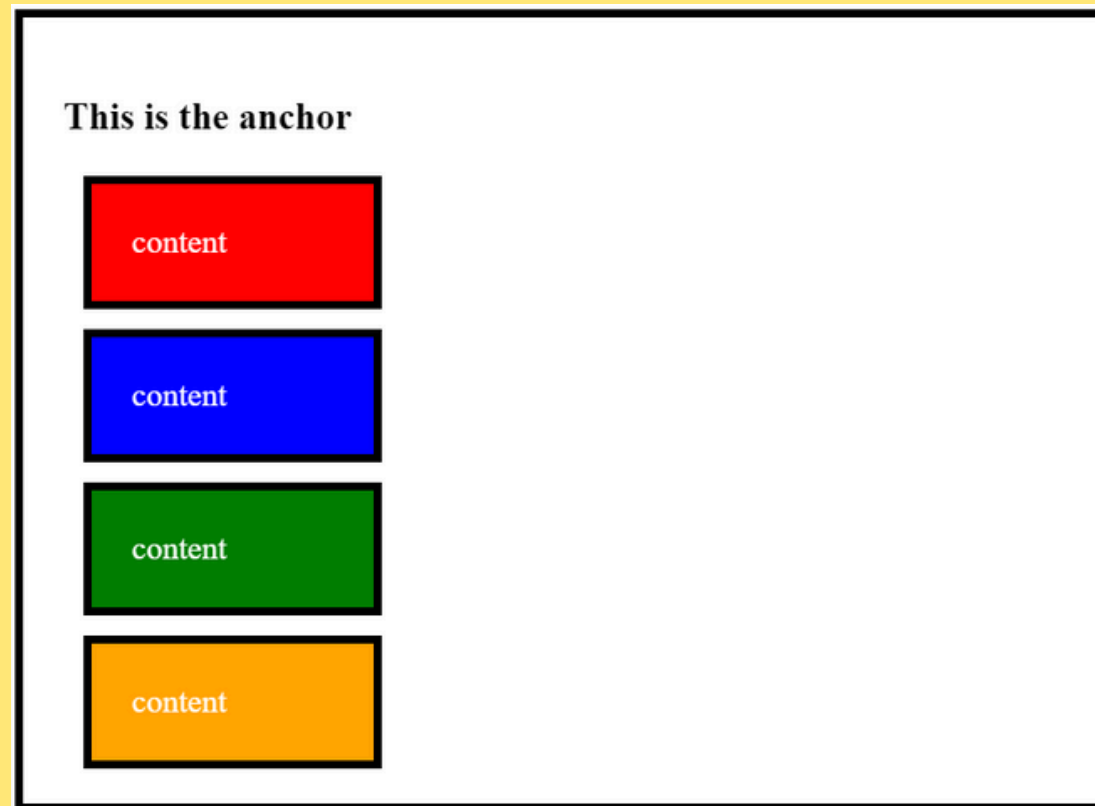
STICKY

Sticky positioning is a mix between relative and fixed. It's like a sticky note that moves with the page until it reaches a certain point, then it stays there.

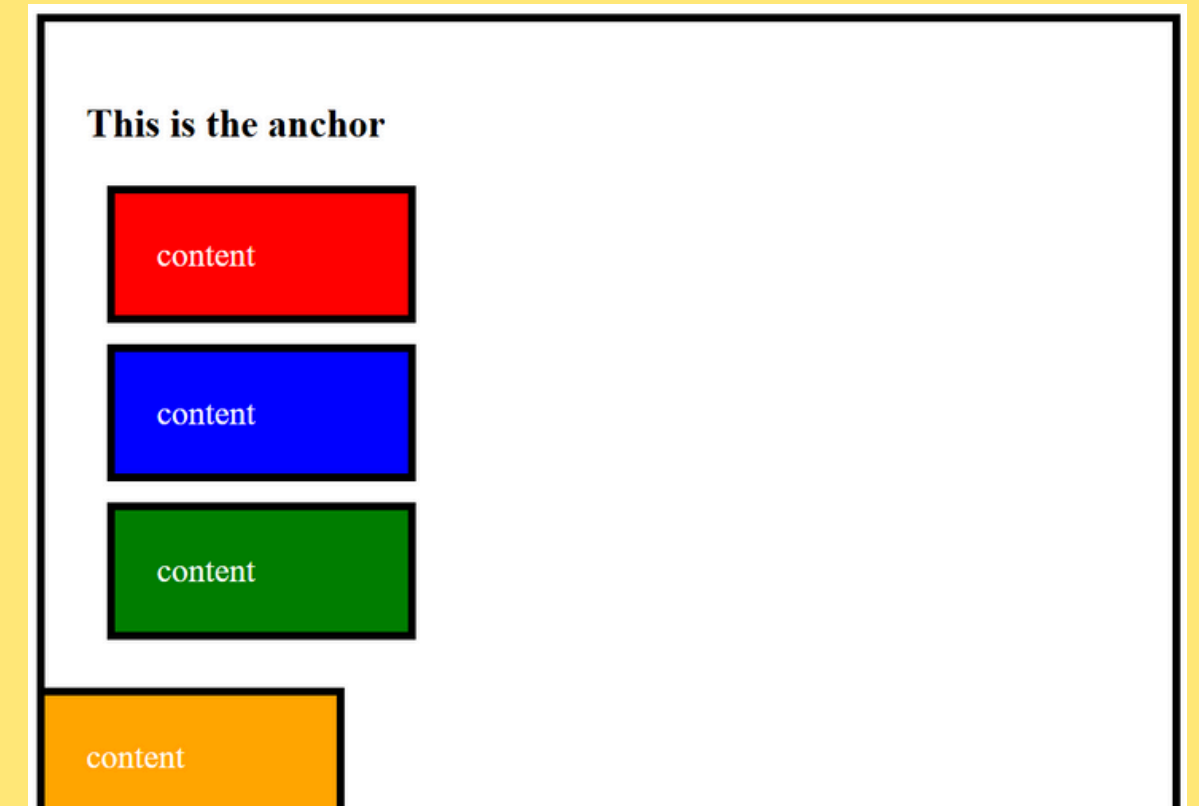
Example:

```
#orange {  
  background-color: orange;  
  position: sticky;  
  right: 0px;  
  bottom: 0px;  
}
```

Static



Sticky



FLEXBOX LAYOUT



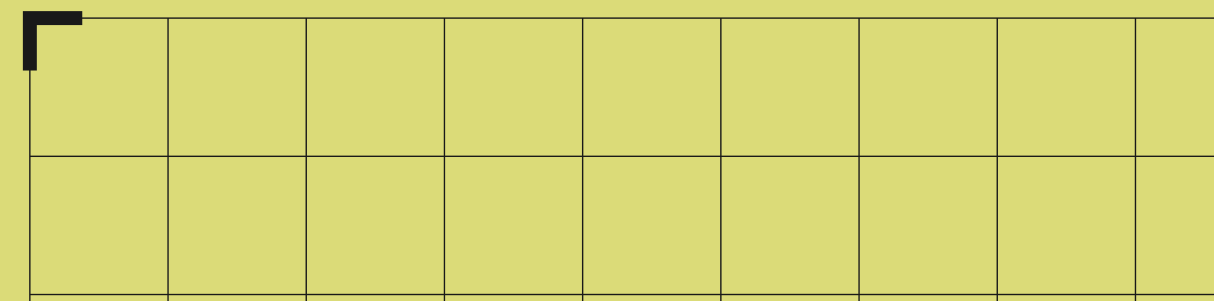
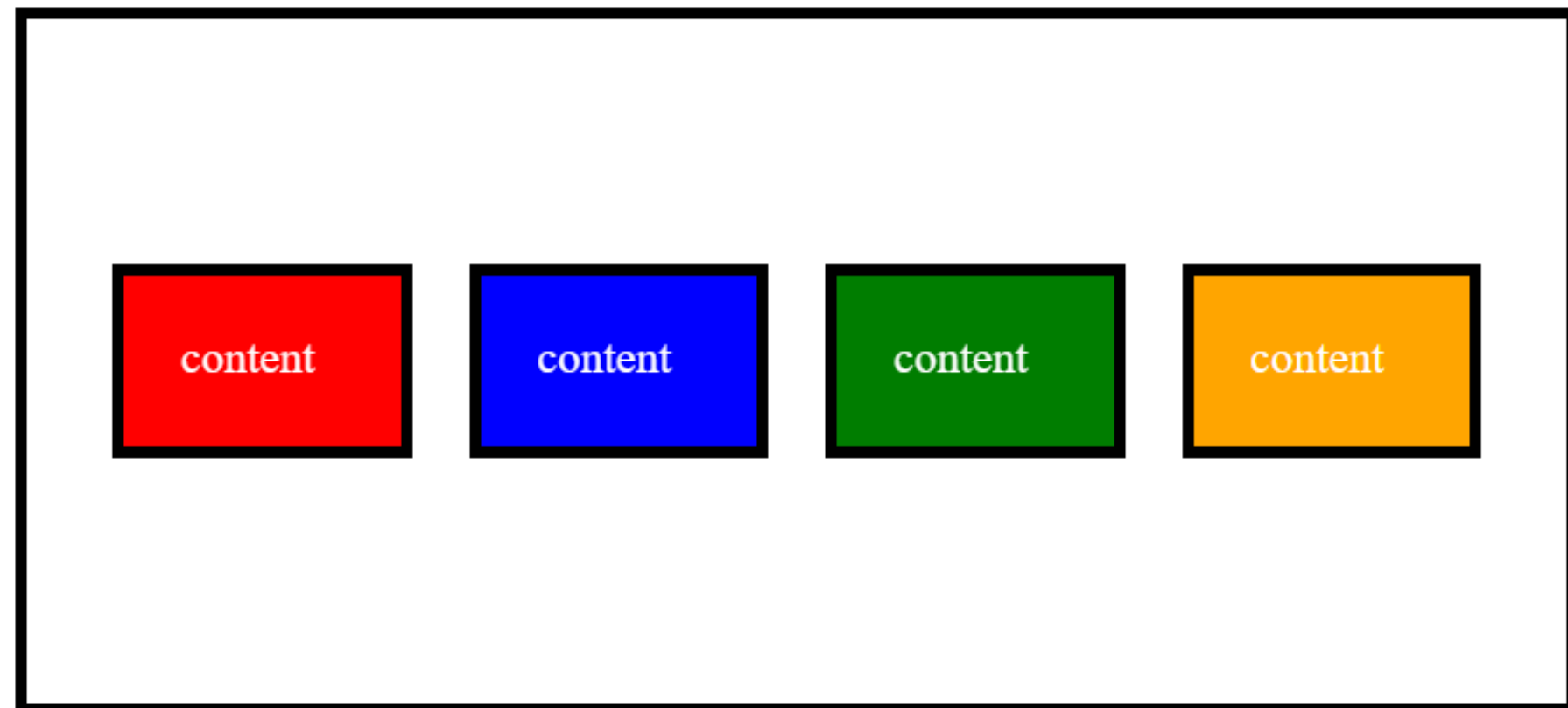
Properties

- **display: flex**
- **flex-direction**
- **justify-content**
- **align-items**

Example:

```
#flexbox {  
  display: flex;  
  flex-direction: row;  
  align-items: center;  
}
```

Flexbox



GRID LAYOUT



Properties

- **display: grid**
- **grid-template-columns**
- **grid-template-columns**
- **gap**

Example:

```
#grid {  
  display: grid;  
  grid-template-columns: 200px 200px;  
  grid-template-rows: auto;  
  align-content: center;  
  justify-content: center;  
  gap: 10px;  
}
```

Grid

