

CSS



HTML



CSS

Relative und absolute Positionierung

Beispiele für relative Positionierung

- › [Beispiel 1](#)
- › [Beispiel 2](#)

Relative Positionierung

- › Für manche Anwendungsfälle reicht die Positionierung von Boxen mit Flexboxen nicht aus.
- › Beispiel: Jegliche Layouts, bei denen einzelne Boxen *unabhängig von ihren benachbarten Boxen* von ihrer Standardposition verschoben werden sollen.
- › In solchen Fällen spricht man von einer *relativen Positionierung* einer Box.

Wie baut der Browser eine Seite auf?

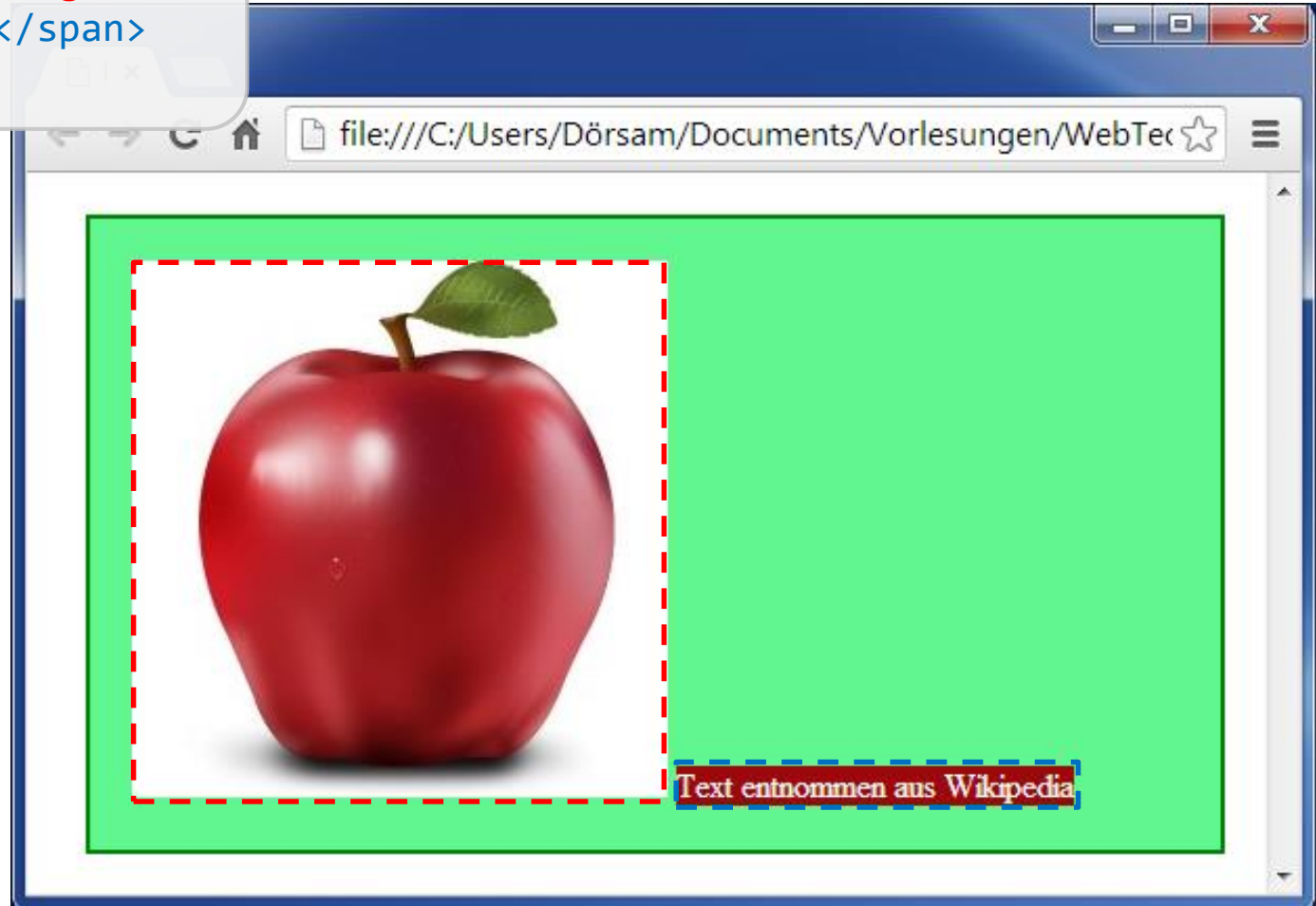
- › Der Browser berechnet zunächst die Größe und Lage der Boxen, bevor er diese mit Inhalten füllt.
- › Die Berechnung folgt den Regeln:
 1. Block-Elemente (`<p>`, `<h1>`, ...) beanspruchen die gesamte Fensterbreite und werden daher *untereinander angeordnet*.
 2. Jedes Element (Inline- / Blockelement) wird *so weit wie möglich links* und *oben* positioniert.
 3. Wenn weitere Elemente dazukommen, werden sie jeweils *rechts davon angeordnet*, und zwar so lange, bis kein Platz mehr ist.
 4. Wenn rechts kein Platz mehr ist, *rutschen sie eine Zeile tiefer und beginnen wieder ganz links*.
 5. Verschachtelte Elemente werden nach dem selben Vorgehen innerhalb der Elternbox aufgebaut.

Relative Positionierung

- › Bei der *relativen Positionierung* wird bei der Positionierung einer Box das folgende Verfahren angewendet:
 1. Der Browser berechnet die Standard-Positionierung einer Box im Browser (nach den Regeln der statischen Positionierung)
 2. Anschließend wird *jede Box, welche die Eigenschaft **position:relative** besitzt, relativ zu ihrer Standardposition horizontal oder vertikal verschoben* (Entfernung von der Standard-Randposition wird angegeben)
- › Bei der relativen Positionierung bleibt der Bereich, in dem die Box normalerweise stehen würde, leer.
- › Wird eine Box relativ positioniert, hat es *KEINE Auswirkung auf andere Boxen* auf der Seite.

Statische Positionierung

```
<div>  
  <img> ... </img>  
  <span> ... </span>  
</div>
```

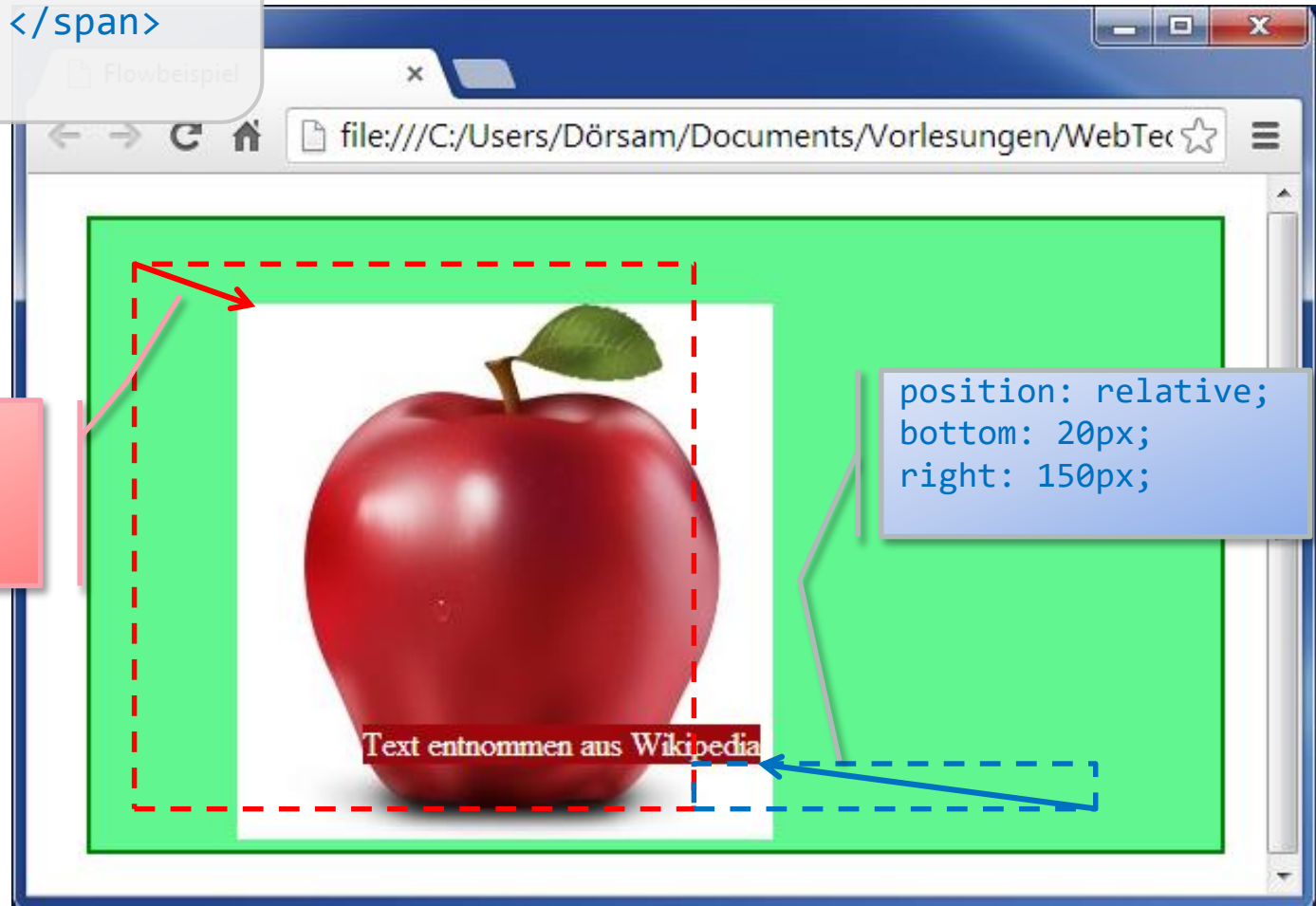


Relative Positionierung

```
<div>
  <img> ... </img>
  <span> ... </span>
</div>
```

```
position: relative;
top: 20px;
left: 50px;
```

```
position: relative;
bottom: 20px;
right: 150px;
```



Relative Positionierung: CSS-Eigenschaften

› **position:relative**

- › Angabe der Entfernung der neuen Boxen-Ränder von der Standardposition der Box:
 - › **top**: Abstand des oberen Randes
 - › **right**: Abstand des rechten Randes
 - › **bottom**: Abstand des unteren Randes
 - › **left**: Abstand des linken Randes
- › Die Werte dieser Eigenschaften können sein:
 - › Absolute Angaben mit Maßeinheit (meistens **px**)
 - › Prozentangaben (im Verhältnis zur Breite des übergeordneten Elementes)

Beispiele für absolute Positionierung

- › [Beispiel 1](#): Bildbeschriftungen
- › [Beispiel 2](#): exakte Überdeckung
- › [Beispiel 3](#): Bildergalerie
- › [Beispiel 4](#): Footer immer am unteren Rand

Absolute Positionierung

- › Auch für folgende Anwendungsfälle reicht die Positionierung von Boxen mit Flexboxen nicht aus:
 - › Elemente müssen an einer exakt vorgegeben *Position innerhalb des Browserfensters* stehen (z.B. Footer: immer ganz unten)
 - › Elemente müssen *bezogen auf andere Elemente an einer exakt vorgegebenen Position* stehen (z.B. Bilder müssen in einer Bildergalerie exakt übereinander stehen)
- › In solchen Fällen spricht man von einer *absoluten Positionierung* einer Box.

Absolute Positionierung

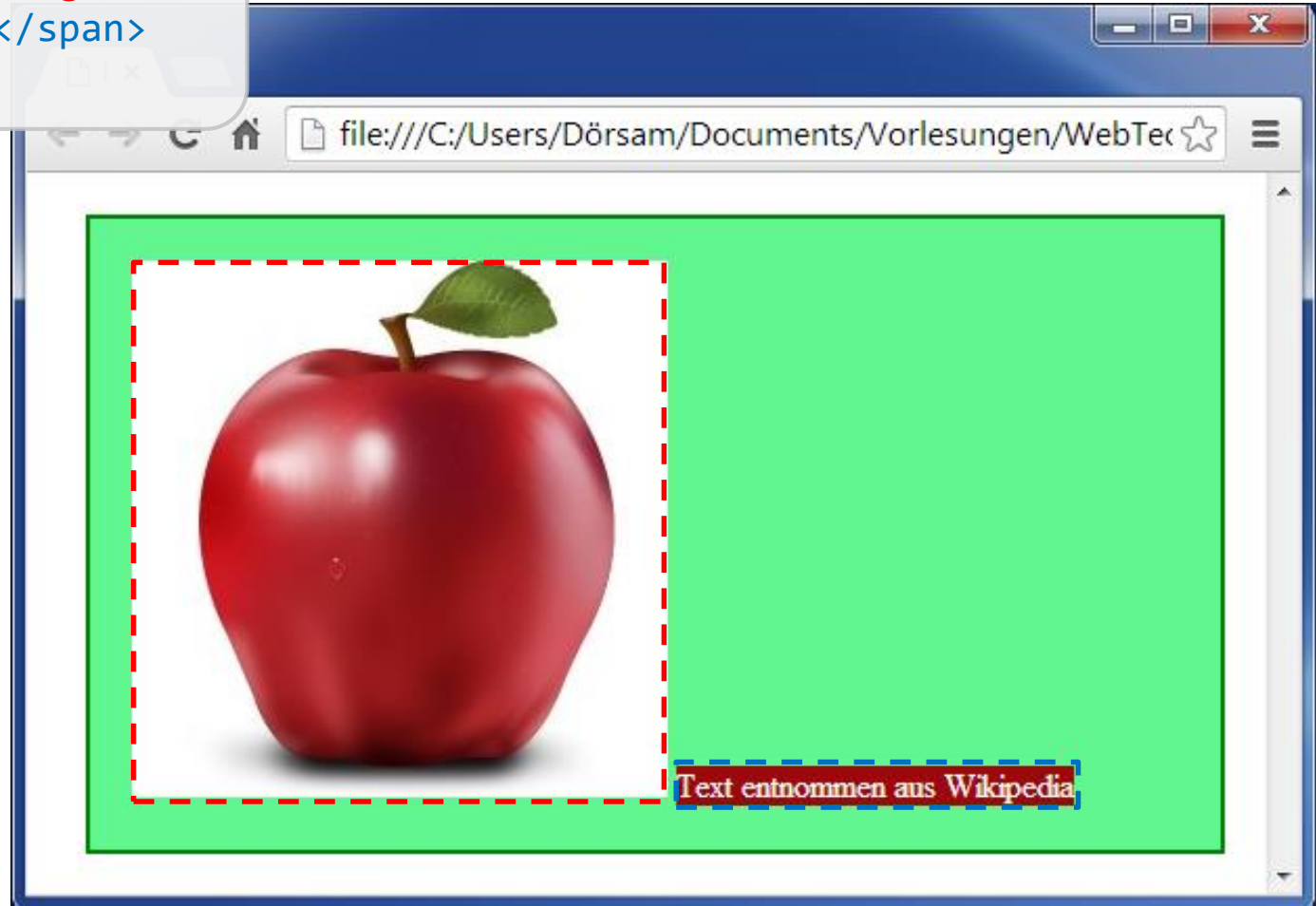
- › Bei der *absoluten Positionierung* wird die Positionierung einer Box *in Form von Abständen vom übergeordneten Container* definiert.
 - › die erste *explizit* positionierte Box in der Boxenhierarchie entlang der Eltern-Beziehung oder
 - › das Browserfenster, falls keine der Eltern-Boxen *explizit* positioniert wurde.
- › Eine Box wird *explizit positioniert*, wenn im zugehörigen CSS der Wert der Eigenschaft **position** gesetzt wurde. Der Wert selbst (**relative**, **absolute**) spielt dabei keine Rolle.

Absolute Positionierung

- › Bei der absoluten Positionierung wird eine Box *aus dem Standard-Fluss herausgenommen*.
- › Alle anderen Boxen im selben Container werden daher so angeordnet, als gäbe es die absolute Box gar nicht.

Statische Positionierung

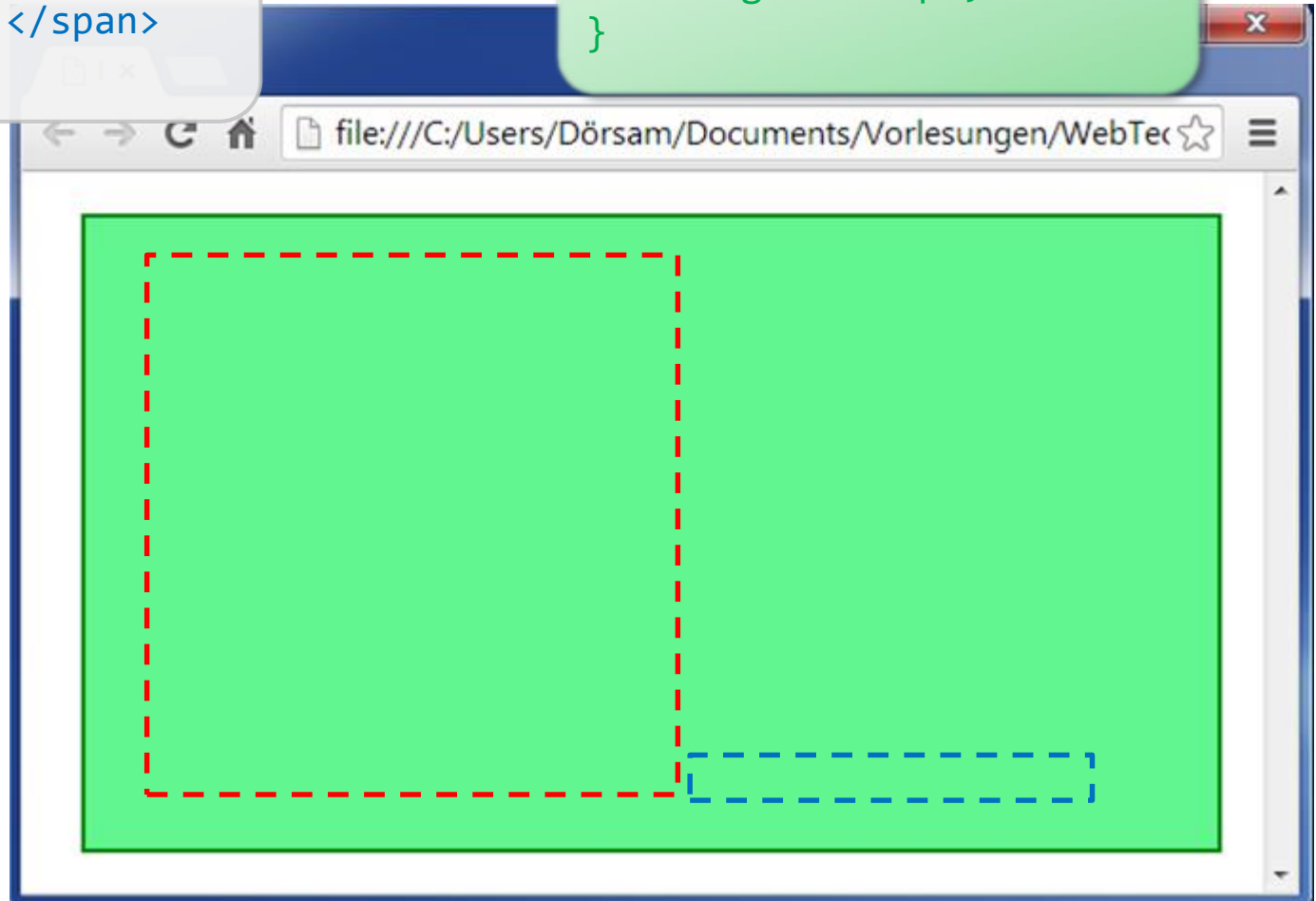
```
<div>  
  <img> ... </img>  
  <span> ... </span>  
</div>
```



Statische Positionierung ohne die absolut positionierten Boxen

```
<div>  
  <img> ... </img>  
  <span> ... </span>  
</div>
```

```
div {  
  height: 300px;  
}
```



Absolute Positionierung (in Bezug auf das Browserfenster)

```
<div>
  <img> ... </img>
  <span> ... </span>
</div>
```

```
div {
  height: 300px;
}
```

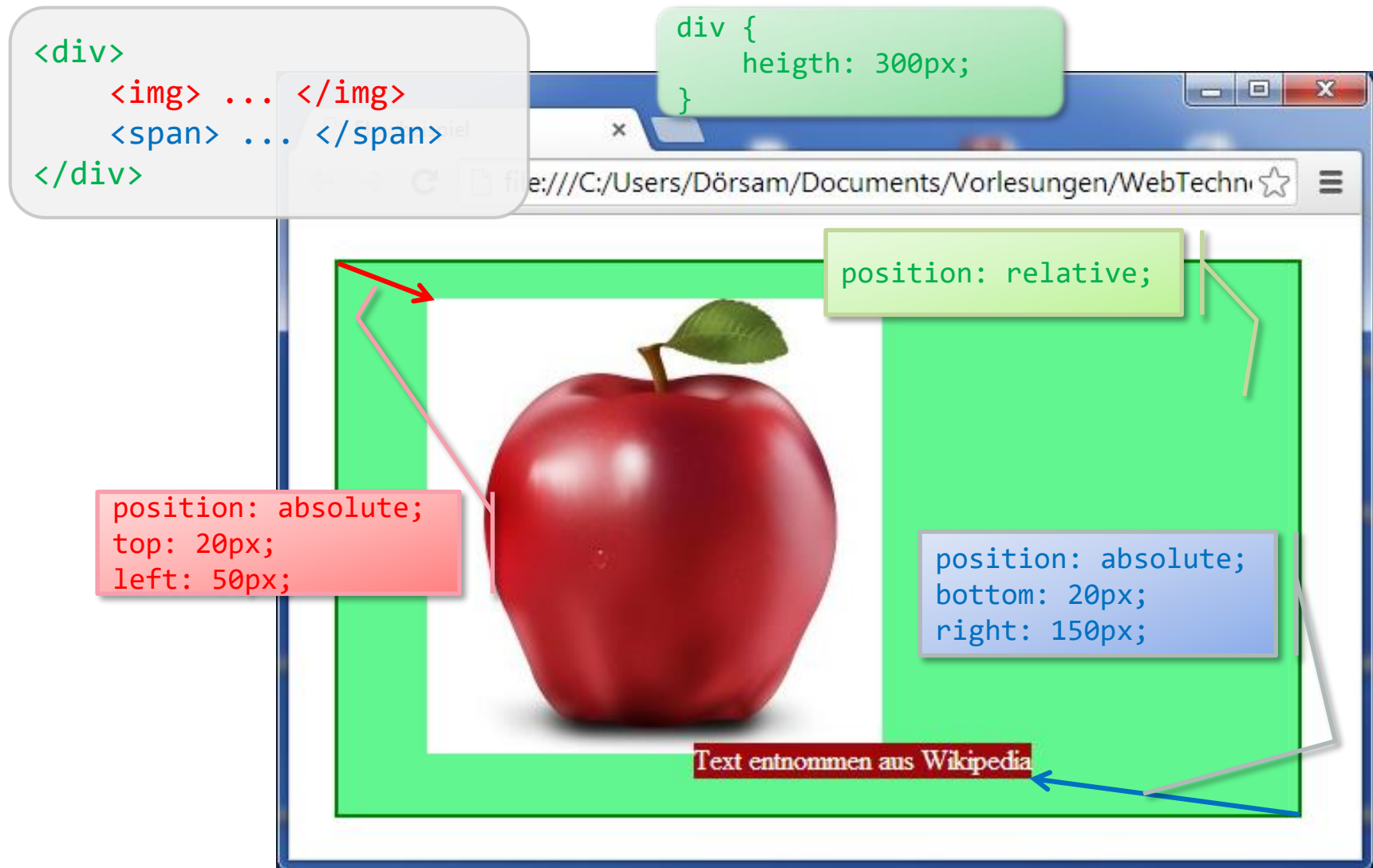
<div>-Element ohne
explizite
Positionierung

```
position: absolute;
top: 20px;
left: 50px;
```

```
position: absolute;
bottom: 20px;
right: 150px;
```

Text entnommen aus Wikipedia

Absolute Positionierung (in Bezug auf den übergeordneten Container)



Absolute Positionierung

- › CSS-Eigenschaften für die absolute Positionierung:
 - › **position: absolute**
 - › Angabe des Abstands vom jeweiligen Rand des übergeordneten Containers:
 - › **top**: Abstand vom oberen Rand
 - › **right**: Abstand vom rechten Rand
 - › **bottom**: Abstand vom unteren Rand
 - › **left**: Abstand vom linken Rand
- › Die Werte dieser Eigenschaften können sein:
 - › Absolute Angaben mit Maßeinheit (meistens **px**)
 - › Prozentangaben (im Verhältnis zur Breite des übergeordneten Elementes)

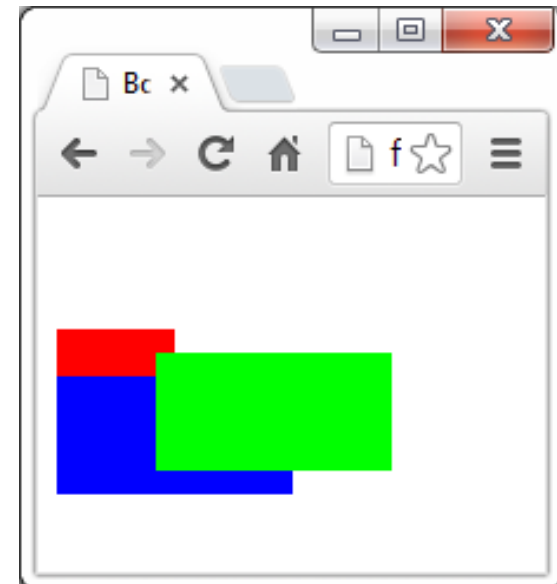
Überlappungen

- › Bei der Definition von Positionierungen kann es zu Überlappungen von Elementen kommen.
- › Standardmäßig werden alle nachfolgenden Elemente eine Ebene weiter im Vordergrund gezeichnet, d.h., das letzte Element überlappt alle vorhergehenden Elemente.
- › Mit der Eigenschaft **z-index** kann die Reihenfolge bei der Überlappung festgelegt werden:
 - › **z-index** ist eine beliebige ganze Zahl
 - › Elemente werden in der Reihenfolge ihrer **z-index** -Werte gezeichnet.
 - › Je höher die Zahl ist, desto weiter nach vorne kommt das zugehörige Element.

Standardüberlappung

```
#box1{background-color:red; width:50px; height:50px;  
      position:absolute; top:40px}  
#box2{background-color:blue; width:100px; height:50px;  
      position:absolute; top:60px}  
#box3{background-color:green; width:100px; height:50px;  
      position:absolute; top:50px; left:50px}
```

```
<body>  
  <p id="box1"></p>  
  <p id="box2"></p>  
  <p id="box3"></p>  
</body>
```

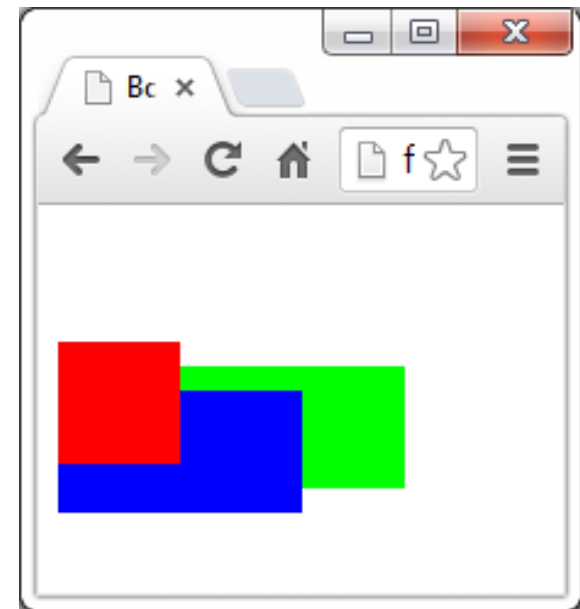


[Interaktives Beispiel](#)

Überlappung mit **z-index**

```
#box1{background-color:red; width:50px; height:50px;
      position:absolute; top:40px; z-index:100}
#box2{background-color:blue; width:100px; height:50px;
      position:absolute; top:60px; z-index:5}
#box3{background-color:green; width:100px; height:50px;
      position:absolute; top:50px; left:50px; z-index:1}
```

```
<body>
  <p id="box1"></p>
  <p id="box2"></p>
  <p id="box3"></p>
</body>
```



[Interaktives Beispiel](#)

Wichtige Eigenschaften des **z-index**

- › Die Eigenschaft **z-index** wirkt erst, wenn die Eigenschaft **position** den Wert **relative** oder **absolute** hat.
- › Der Wert **position:static** entspricht dem Standardverhalten des Browsers. Ist dieser gesetzt (explizit oder implizit), wird der Wert des **z-index** nicht ausgewertet.