

# User Interfaces 1

## CSS

Layout: positioning

# Positionering

---

- "Normale flow": browser positioneert de verschillende elementen afhankelijk van hun soort:
  - inline elementen: naast elkaar
  - blok elementen: onder elkaar
- Willen we ingrijpen in de normale flow, dan kan dat via de eigenschappen `position` en `float`

# position

---



- Mogelijke waarden:

- **static**: het blok volgt de normale flow
- **relative**: het blok wordt verschoven t.o.v. zijn standaardpositie in de normale flow

normal

- **absolute**: het blok verdwijnt uit de normale flow en wordt gepositioneerd t.o.v. de padding-rand van het *omvattende blok*

parent

- **fixed**: het blok wordt verschoven t.o.v. de rand van het browservenster

viewport

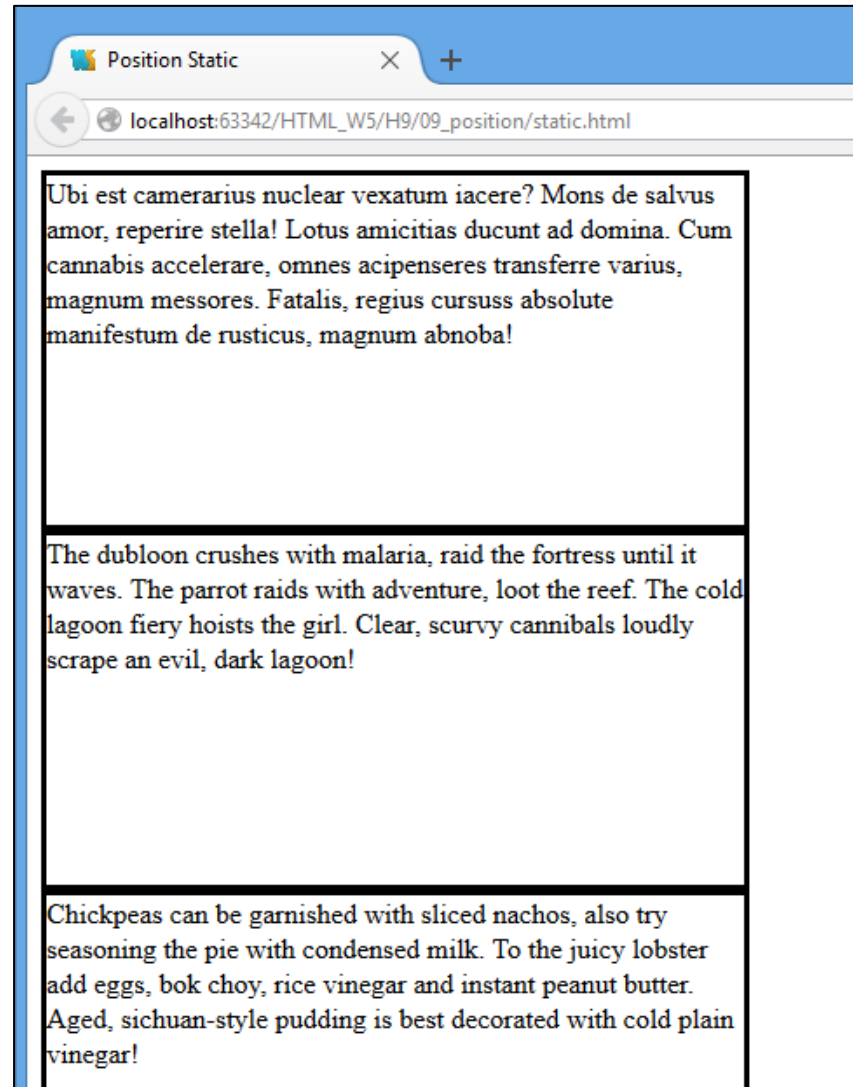
→ De eigenlijke verplaatsing gebeurt vervolgens met de eigenschappen **left**, **right**, **top** en **bottom**...

# position: static;

---

Zonder positionering:  
blokken volgen de  
normale flow...

Dit is het standaard  
(default) gedrag.



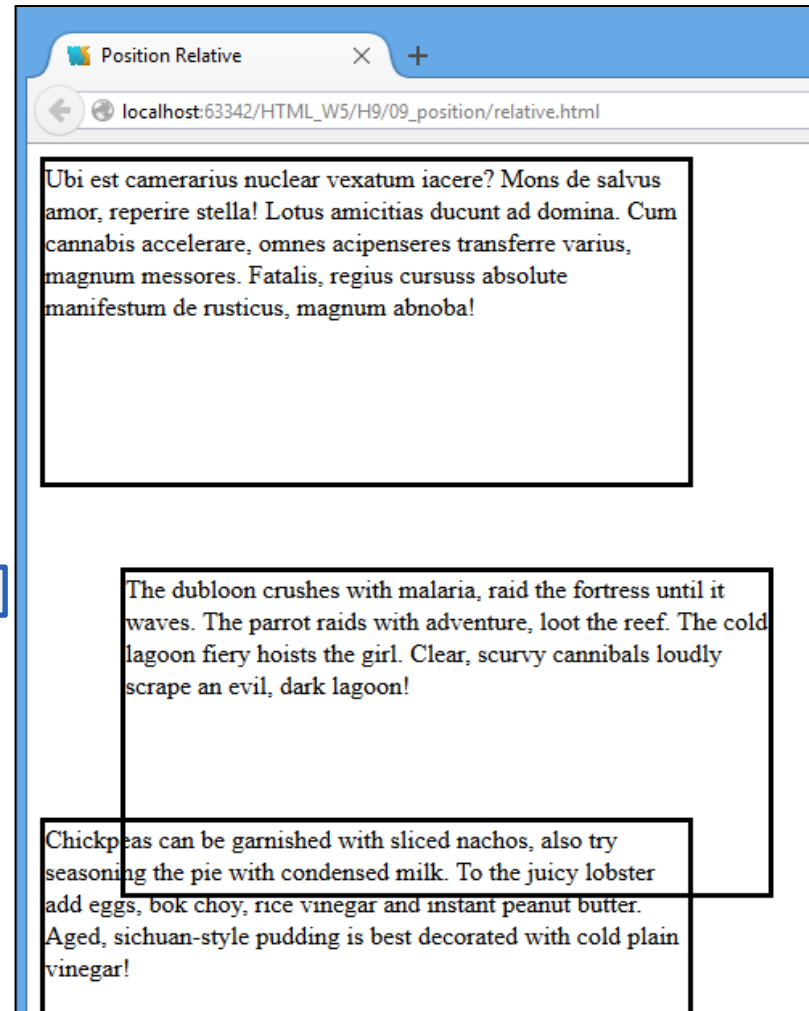
Voorbeeld: [H10/09\\_position/static](#)

# position: relative;

Het tweede blok wordt  
relatief t.o.v. positie in  
de normale flow  
verschoven:

```
section {  
  border: solid;  
  width: 400px;  
  height: 200px;  
  display: block;  
}  
  
.verschuif {  
  position: relative;  
  left: 50px;  
  top: 50px;  
}
```

Overbodig

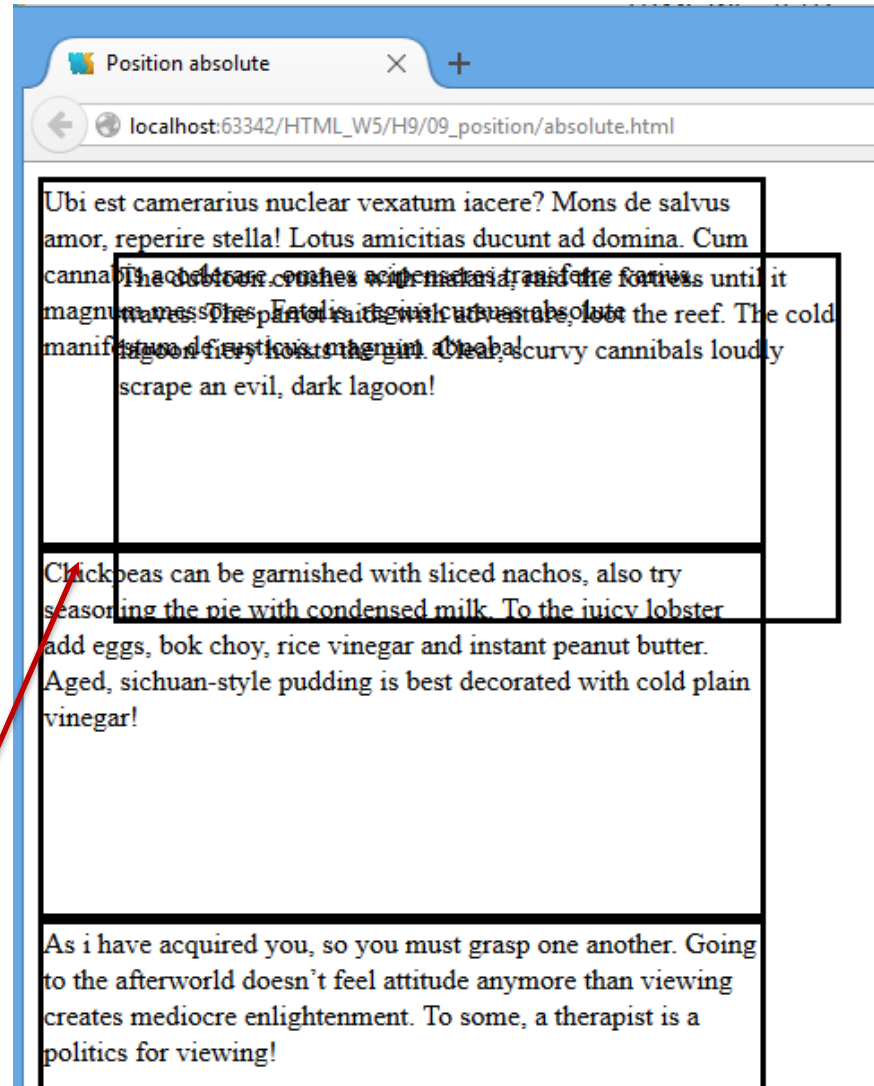


# position: absolute;

Het tweede blok  
verdwijnt uit de flow en  
wordt absoluut  
gepositioneerd t.o.v. het  
*omvattende blok*:

```
.verschuif {  
  position: absolute;  
  left: 50px;  
  top: 50px;  
}
```

Merk op: de overige  
blokken gaan tegen elkaar  
staan, het absoluut  
gepositioneerde blok is  
volledig uit de normale flow  
verwijderd!



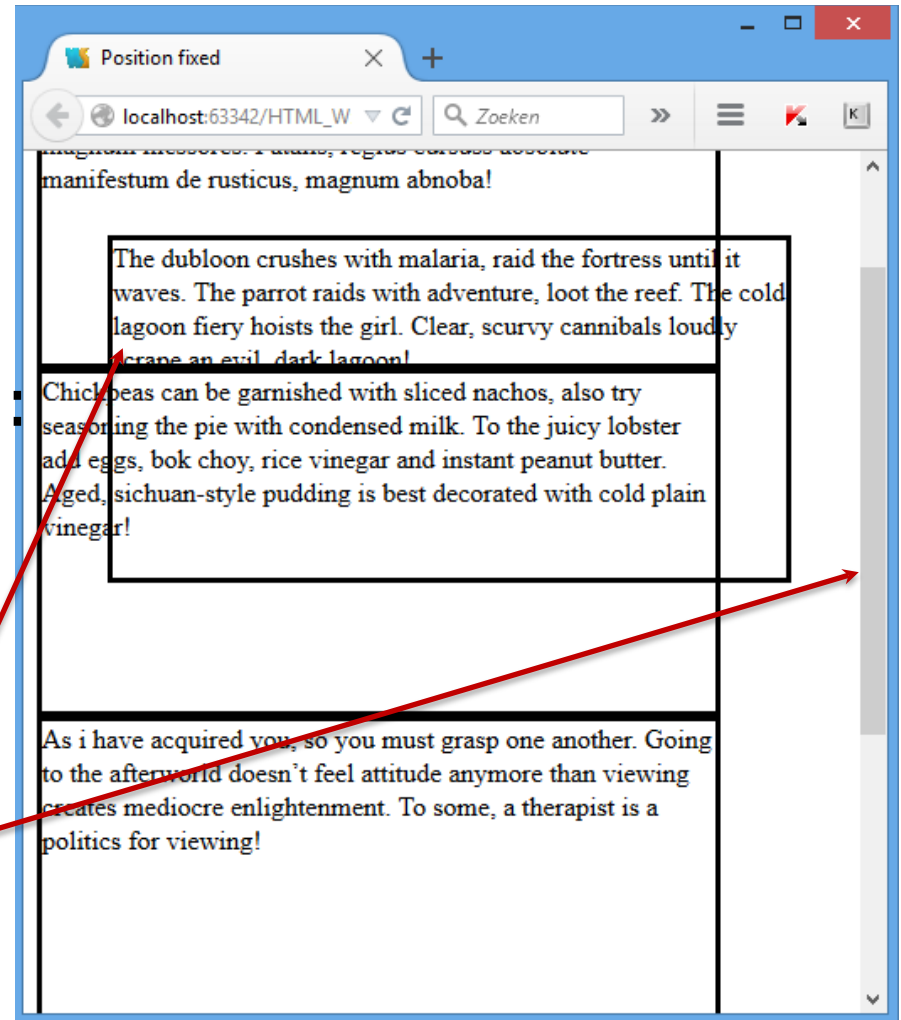
Voorbeeld: H10/09\_position/absolute

# position: fixed;

Het blok wordt vast  
gepositioneerd t.o.v. het  
browser window  
(ongevoelig voor scrolling):

```
.verschuif {  
  position: fixed;  
  left: 50px;  
  top: 50px;  
}
```

Als je scrollt blijft het  
blok nog steeds staan!



## Het *omvattend* blok?

---

- Absoluut gepositioneerde blokken worden gepositioneerd t.o.v. het omvattend ("containing") blok: dit is *de eerste parent die niet static gepositioneerd is* (dus onder andere **relative** of **absolute**).
- Een voorbeeld maakt veel duidelijk!





# Voorbeeld omvattend blok

```
.container section {  
    width: 100px;  
    height: 100px;  
    background-color: red;  
}
```

Voorbeeld:

HTML:

```
<body>  
<h1>Positioneringsdemo</h1>  
<article class="container">  
    <section id="id1">  
    </section>  
    <section id="id2">  
    </section>  
    <section id="id3">  
    </section>  
    <section id="id4">  
    </section>  
</article>  
</body>
```

Container is niet static  
en dus het omvattende  
blok voor de sections. Ze  
worden bijgevolg t.o.v.  
de container geplaatst

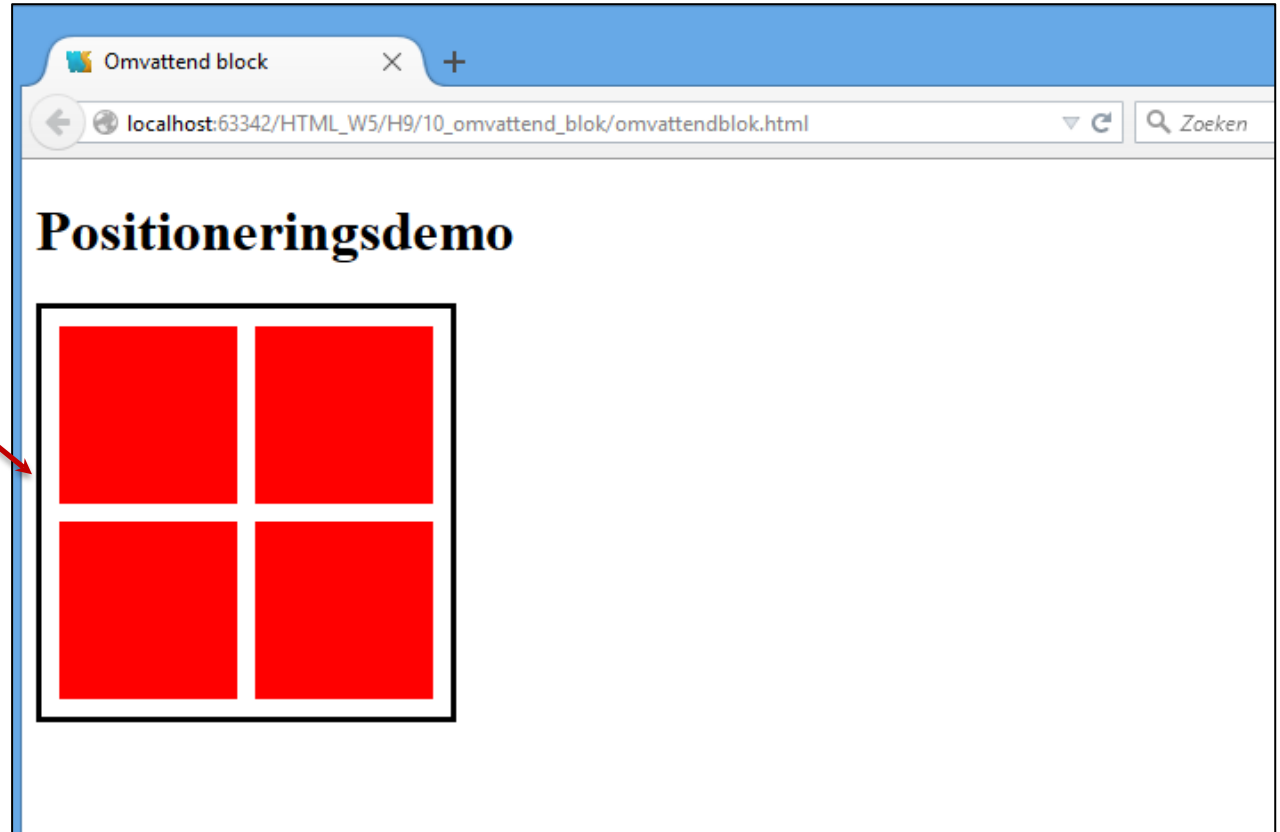
CSS:

```
.container {  
    position: relative;  
    width: 230px;  
    height: 230px;  
    border: solid;  
}  
#id1 {  
    position: absolute;  
    left: 10px;  
    top: 10px;  
}  
#id2 {  
    position: absolute;  
    right: 10px;  
    top: 10px;  
}  
/*... enzovoort ...*/
```

# Voorbeeld omvattend blok

---

De 4 sections staan  
gepositioneerd t.o.v. de  
container (met zwarte  
rand)



Voorbeeld: H10/10\_omvattend\_block

# Voorbeeld omvattend blok

Voorbeeld:

HTML:

```
<body>
<h1>Positioneringsdemo</h1>
<article class="container">
  <section id="id1">
  </section>
  <section id="id2">
  </section>
  <section id="id3">
  </section>
  <section id="id4">
  </section>
</article>
</body>
```

Container is hier **wel** static en dus worden de sections gepositioneerd t.o.v. het hele browservenster!

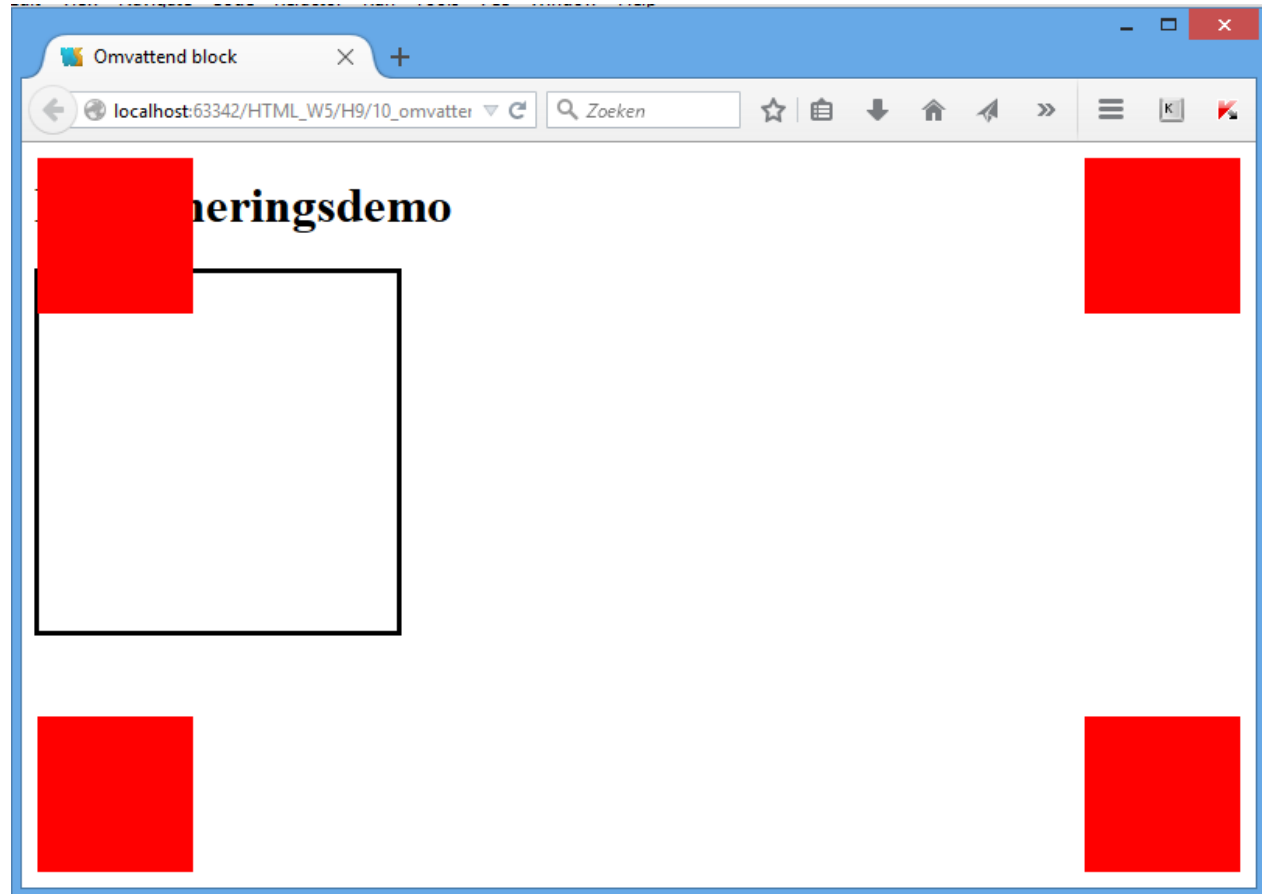
CSS:

```
.container {
  /*position: relative;*/
  width: 230px;
  height: 230px;
  border: solid;
}
#id1 {
  position: absolute;
  left: 10px;
  top: 10px;
}
#id2 {
  position: absolute;
  right: 10px;
  top: 10px;
}
/*... enzovoort ...*/
```

# Voorbeeld omvattend blok

---

De 4 sections staan  
gepositioneerd ten  
opzichte van het hele  
browservenster!



Voorbeeld: H10/10\_omvattend\_block

# float

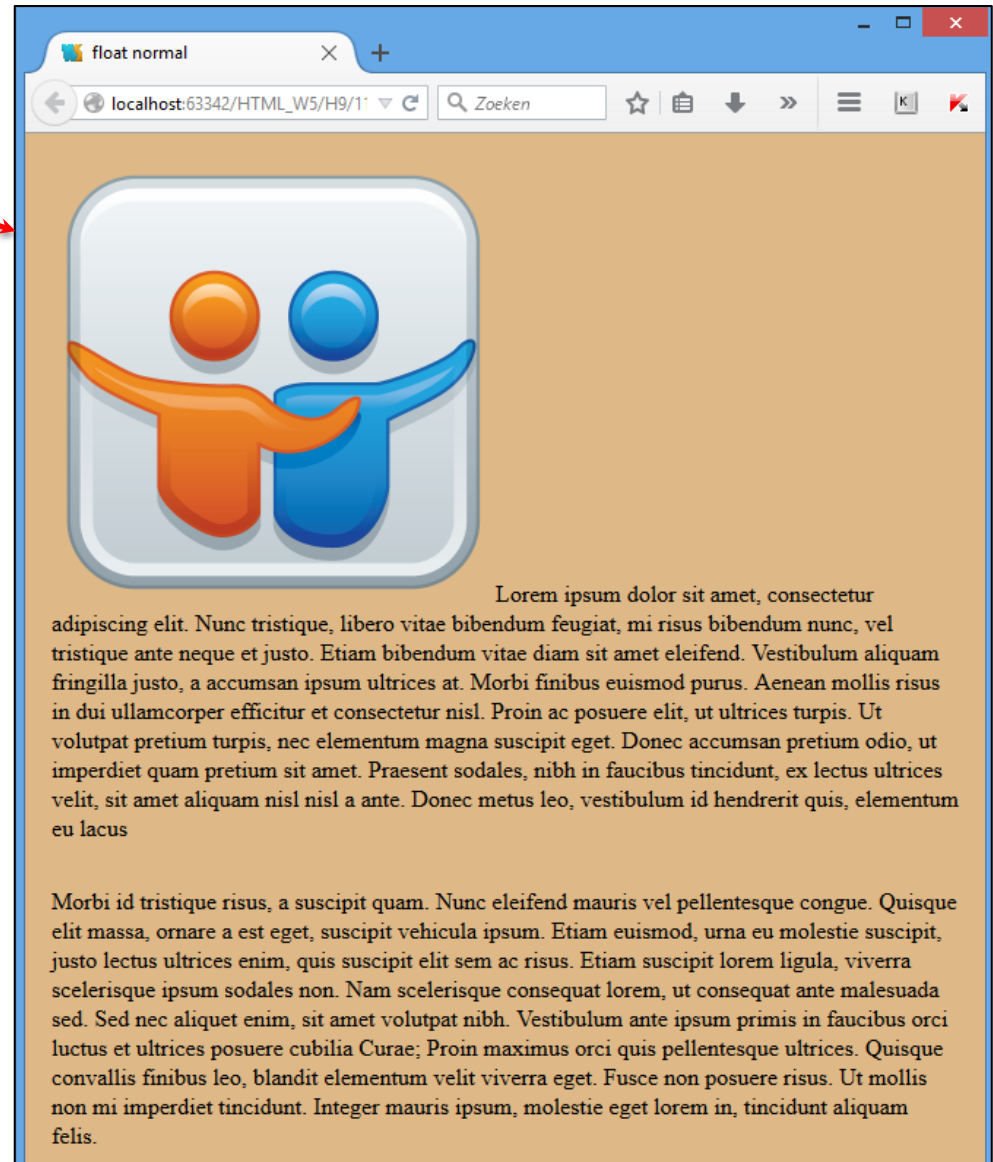
---

- Een andere (en zeer populaire) manier om de positionering aan te passen is via float.
- De eigenschap float laat een element naar links (of rechts) drijven.
- De rest van de pagina loopt met de normale flow *om de float heen*.
- Een floating element wordt automatisch een blok element:
  - Je kan de grootte en de marges, borders en padding instellen

# float bij images en tekst

Normal flow: `img` is een inline element: de tekst volgt gewoon erna...

```
body {  
  font-family: sans-serif;  
  background: burlywood;  
}  
  
section {  
  margin: 10px 0;  
  padding: 10px;  
}
```



# float bij images en tekst

Met `float:left` `img` is uit de natuurlijke flow gehaald en helemaal links geplaatst, de tekst komt er nu rechts naast...

```
img {  
  float: left;  
  margin: 0 10px 0 0;  
}
```

ofwel

```
img {  
  float: left;  
  margin-right: 10px;  
}
```



Hoe kan dit nog?

# float bij images en tekst

Met `float:right` `img` is uit de natuurlijke flow gehaald en helemaal rechts geplaatst, de tekst komt er nu links naast...

```
img {  
  float: right;  
  margin: 0 0 0 10px;  
}
```

ofwel

```
img {  
  float: right;  
  margin-left: 10px;  
}
```



Hoe kan dit nog?



# float links en rechts

```
img {  
  border: solid black 2px;  
  margin: 0 10px;  
}  
  
img.first {  
  float: left;  
}  
  
img.second {  
  float: right;  
}
```



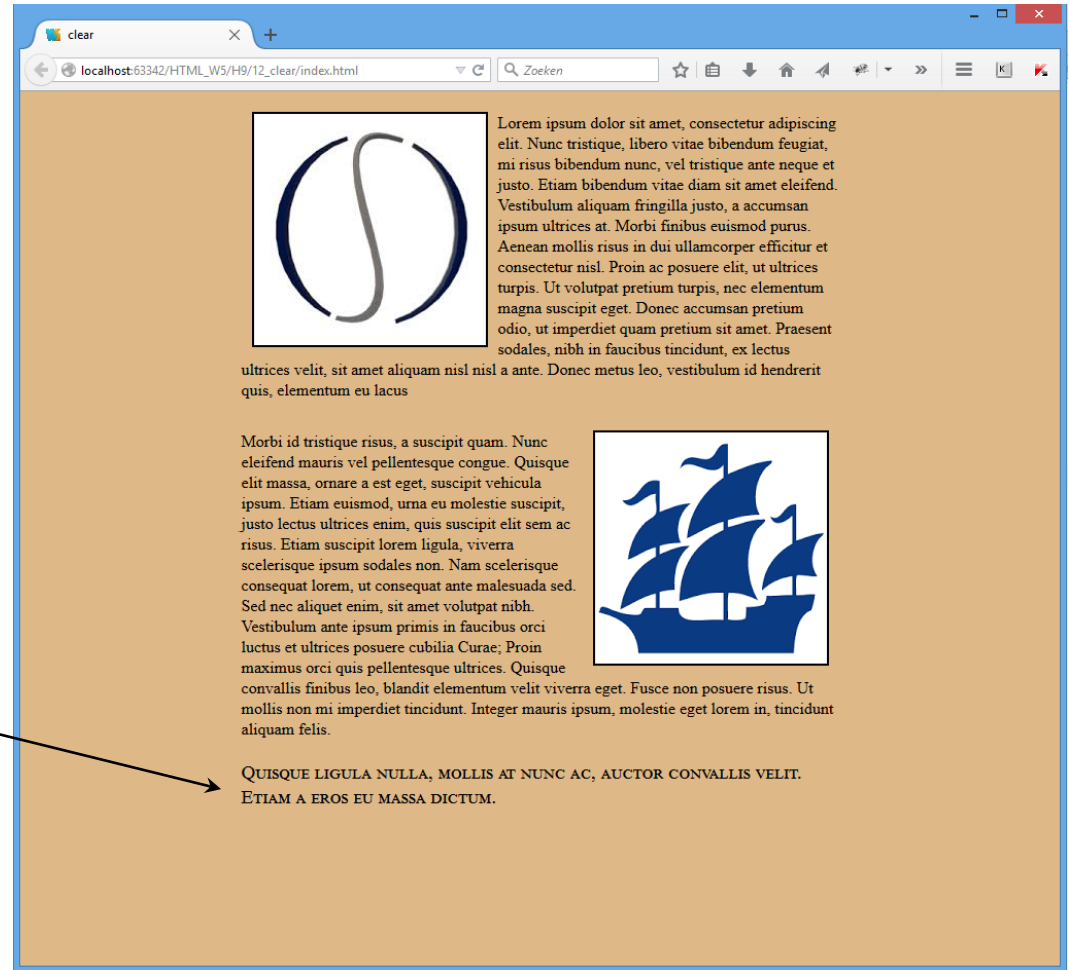
## `clear`

---

- Als je niet wilt dat een element naast een floating element komt te staan, kan dat met de eigenschap **`clear`**
  - **`clear:left`** → geen floating element aan de linkerkant
  - **`clear:right`** → geen floating element aan de rechterkant
  - **`clear:both`** → geen floating element langs beide kanten
- Typisch gebruik: een **header** of **footer**

# Voorbeeld clear

```
footer {  
  clear: both;  
  font-size: 1.2rem;  
  font-variant: small-caps;  
  margin-left: 10px;  
}
```



Voorbeeld: H10/12\_clear/left\_right

# Voorbeeld: 2 kolommen lay-out met float

```
article {  
  float: left;  
  width: 50%;  
  box-sizing: border-box;  
}  
footer {  
  clear: both;  
}  
main {  
  border: solid;  
}
```

De main heeft een border, maar doordat de twee articles gefloat zijn, zijn ze uit de normal flow getrokken en klapt de border dus dicht. Is dit een probleem voor je site, dan kan je dit oplossen met de "clearfix" oplossing: zie handboek p281

