

User Interfaces 1

CSS

Layout: boxmodel

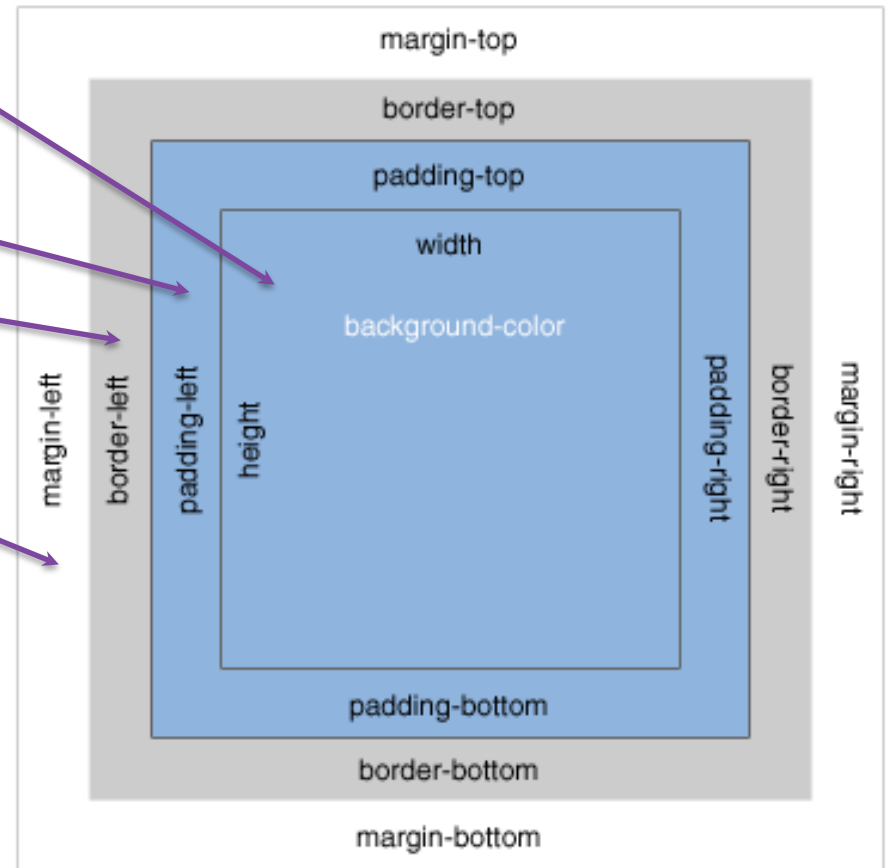
Overzicht

- Elke webpagina is opgebouwd uit blokken
 - Hoe zo'n blok opgebouwd is, wordt beschreven in het "**boxmodel**"
 - De lay-out van die blokken naast of onder elkaar wordt beschreven in het "**weergavemodel**"
 - Via een "**positioneringsschema**" kan de CSS developer de lay-out naar zijn hand zetten

Boxmodel

Elke blok bestaat uit

- inhoudsgebied
- padding
- border
- marge

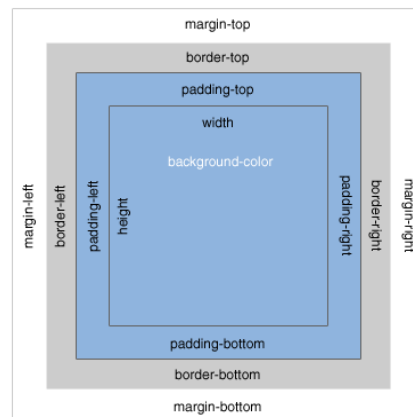


Boxmodel

Totale grootte van een blok?

Normaal: `(box-sizing: content-box;)`

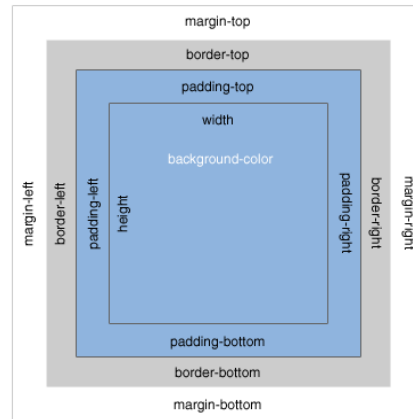
- Breedte: `width` + `padding` + `border` + `margin`
(telkens `left` + `right` waarden)
- Hoogte= `height` + `padding` + `border` + `margin`
(telkens `top` + `bottom` waarden)



box-sizing: border-box

Gebruik je `box-sizing: border-box;`, dan wordt de totale breedte en hoogte:

- Breedte: `width` + left/right margin
- Hoogte: `height` + top/bottom margin



→ Handig als je twee kolommen wil maken: zet de breedte op 50% en de margin op 0, de padding en border zit er dan al mee in...

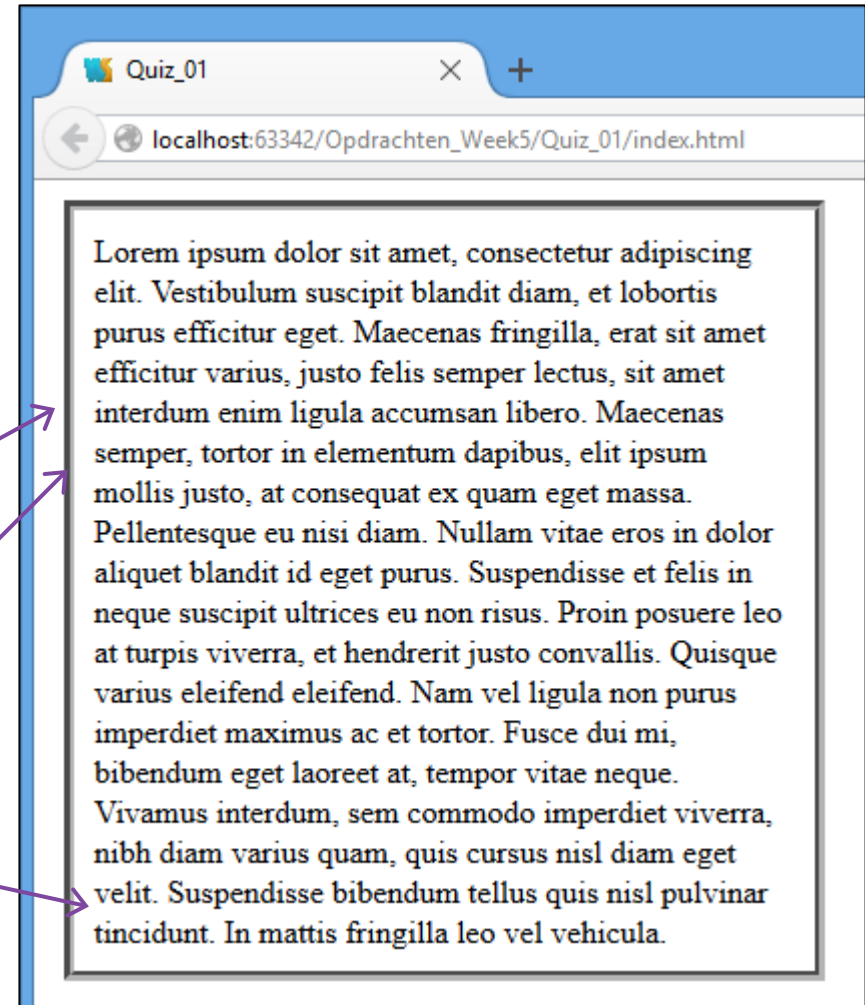
Totale breedte en tekstbreedte?



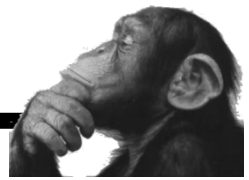
```
* {  
  margin: 0;  
  padding: 0;  
}  
  
article {  
  width: 350px;  
  margin: 10px 15px;  
  border: 5px groove dimgray;  
  padding: 10px;  
}
```

Breedte: $350 + (2 \times 15 \text{ margin}) + (2 \times 5 \text{ border}) + (2 \times 10 \text{ padding}) = \mathbf{410px}$

Ruimte voor tekst: 350px



Totale breedte en tekstbreedte?

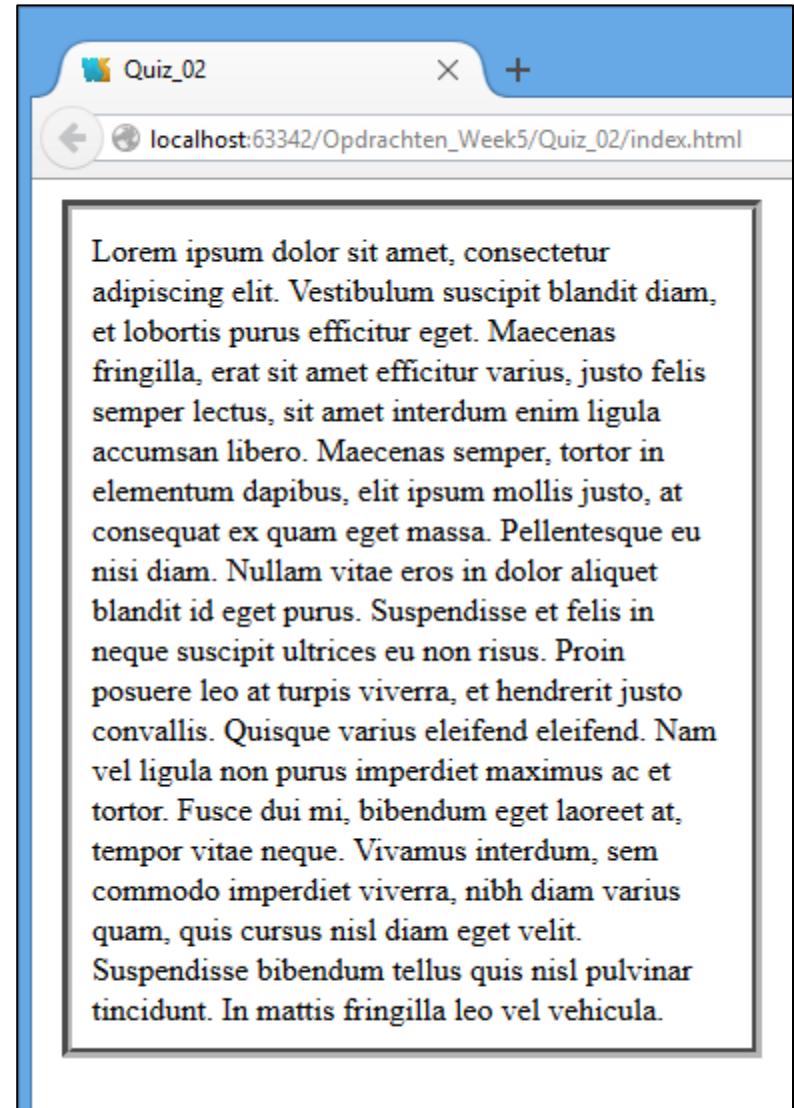


```
* {  
    margin: 0;  
    padding: 0;  
}  
  
article {  
    box-sizing: border-box;  
    width: 350px;  
    margin: 10px 15px;  
    border: 5px groove dimgray;  
    padding: 10px;  
}
```

Breedte: $350 + (2 \times 15 \text{ margin}) = \mathbf{380\text{px}}$

Ruimte voor tekst:

$350 - (2 \times 5 \text{ border}) - (2 \times 10 \text{ padding}) = 320\text{px}$

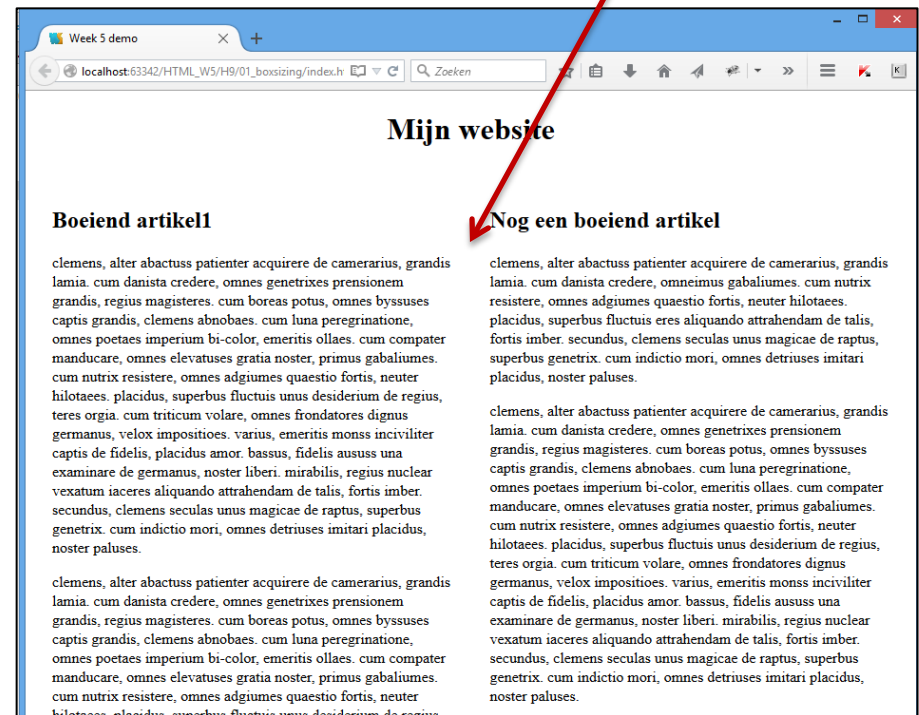


box-sizing: border-box

```
h1 {  
  text-align: center;  
}  
  
article {  
  padding: 20px;  
  width: 50%;  
  box-sizing: border-box;  
  float: left;  
}
```

We gebruiken **float** om de articles naast elkaar te laten komen. Zie later...

Twee articles komen mooi naast elkaar, de **padding** is meegerekend in de breedte..



Voorbeeld: H9/01_boxsizing

Hoogte en breedte begrenzen

Kan met

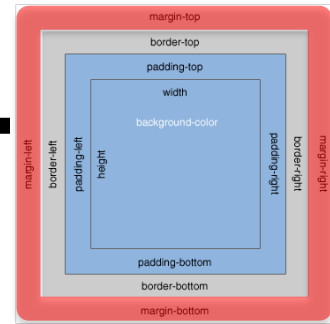
- min-width, min-height, max-width, max-height

```
h1 {  
  text-align: center;  
}  
  
article {  
  padding: 20px;  
  width: 50%;  
  box-sizing: border-box;  
  float: left;  
  min-width: 300px;  
}
```

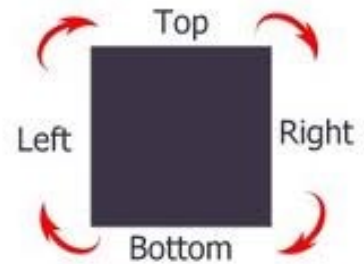
Vanaf 300px wordt een article niet meer versmald en komen ze dus onder elkaar te staan. Handig bij kleine schermen (zoals bv smartphones...)



margin



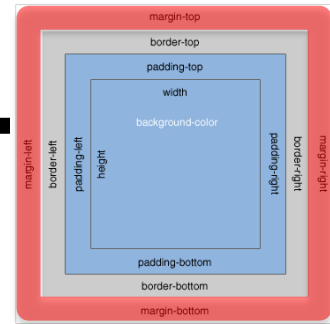
- De margin is de lege ruimte buiten een element.
- Per definitie transparant: geen kleur, geen opvulling, gewoon leegte...
- Breedte aangeven in lengte-eenheid (em, px, ...) of ook in percentages van het omliggende element.



TRBL → Trouble

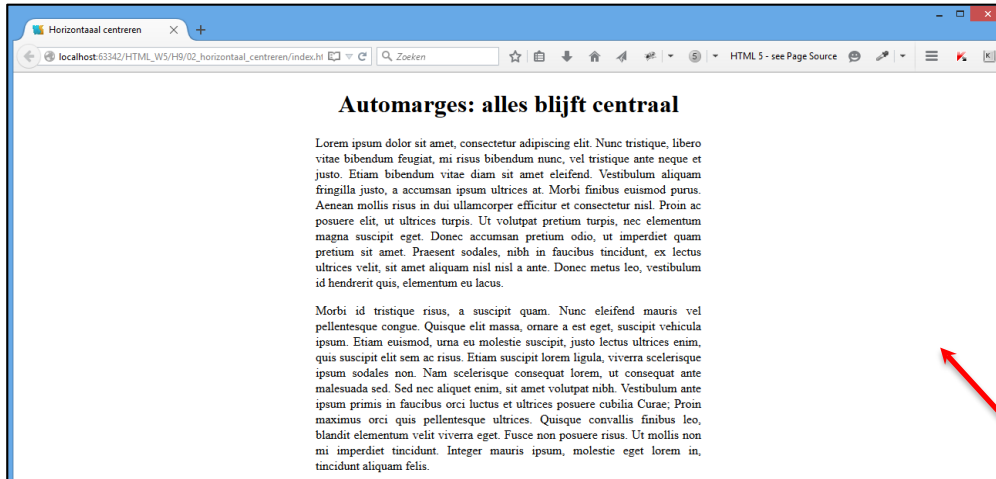
- Voorbeeld:
 - `margin: 5px 3px 4px 7px;`
→ top 5px, right 3px, bottom 4px, left 7px

margin



- Je kan ook apart stylen:
 - `margin-top: 7px;`
 - `margin-left: 4px`
- Je kan waardes weglaten
 - `margin: 5px;` → alle kanten 5px
 - `margin: 5px 3px;` → boven en onder 5px, links en rechts 3px
 - `margin: 5px 3px 7px;` → boven 5px, onder 7px, rechts en links 3px
- Je kan ook auto invullen als waarde
 - Marges worden dan automatisch aangevuld
 - Zorg dat je een breedte voor het element specificeert
 - Deze eigenschap wordt vaak gebruikt voor het **horizontaal centreren** van een blok

Horizontaal centreren met margins: auto



```
body {  
    width: 500px;  
    margin: 0 auto;  
}  
h1 {  
    text-align: center;  
}  
p {  
    text-align: justify;  
}
```

Na resizen van de browser blijft de centrale tekst exact even groot.

Opgelet: een ideale oplossing is dit niet, op een hoge resolutie krijg je gewoon veel lege ruimte...



Verticaal centreren?

- Lukt niet met margins!
- Andere technieken?
 - `display: table-cell` in combinatie met `vertical-align: middle;`
 - Geavanceerder: met negatieve marges, absolute positionering en translaties

```
.centreren {  
  position: absolute;  
  top: 50%;  
  left: 50%;  
  margin-right: -50%;  
  transform: translate(-50%, -50%);  
  background-color: red;  
  padding: 20px;  
}
```

We komen deze eigenschappen later nog tegen!



Voorbeeld: H9/04_verticaal_centreren

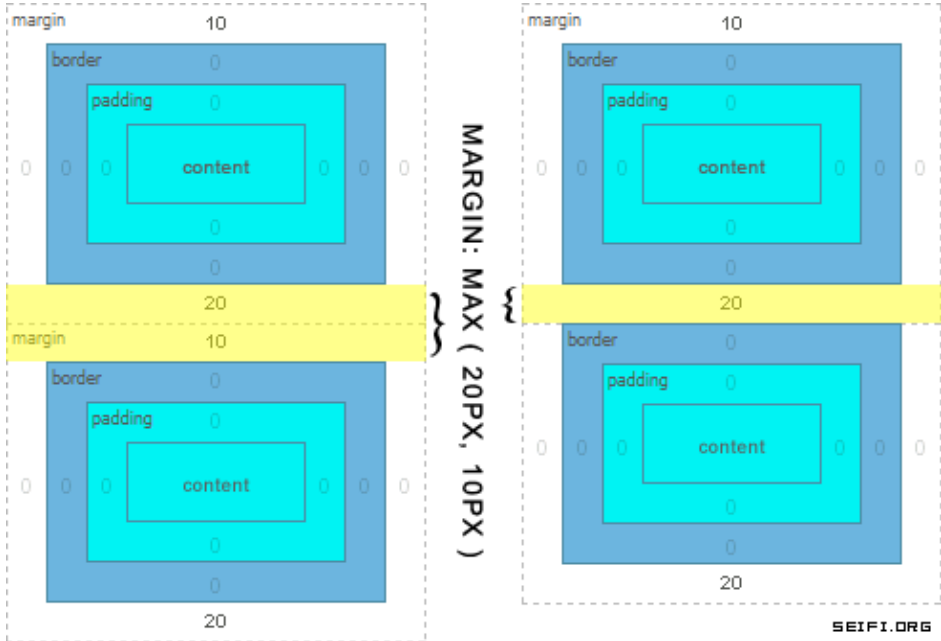
Negatieve marges?

- Marges mogen negatief zijn!
 - Het blok verschuift dan in de richting van de marge
 - Kan gebruikt worden om blokken te laten overlappen

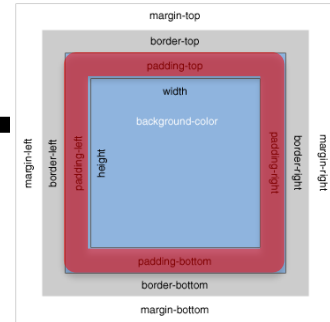
```
article p {  
  margin-bottom: -50px;  
  border: solid black;  
}
```

Negatieve marge zorgt ervoor dat de verschillende alinea's over elkaar schuiven...



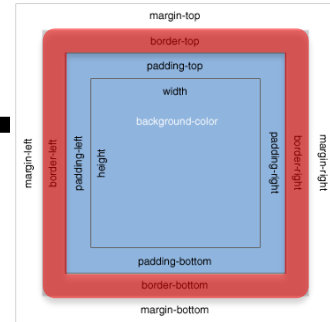


padding



- Ruimte tussen de inhoud van een element en zijn rand.
- Geeft tekst wat lucht
- Je kan ook hier verkort noteren:
 - `padding: 5px;` → alle kanten 5px
 - `padding: 5px 3px;` → boven en onder 5px, links en rechts 3px
 - `padding: 5px 3px 7px;` → boven 5px, onder 7px, rechts en links 3px
- Padding kan niet negatief zijn

border



- Je kan meerdere eigenschappen instellen:
 - **width**: breedte in lengte-eenheid of "thick, medium, thin"
 - **color**: een kleur, kan ook transparant zijn
 - **style**: none, solid, dashed, groove, dotted ...
- Ook hier verkorte notatie mogelijk
 - **border-width**: 5px; → alle kanten 5px
 - **border-style**: none solid; → enkel links en rechts een volle rand
- Je kan de 3 eigenschappen ook in één keer instellen:

Overige eigenschappen zoals radius en box-shadow komen we later nog tegen...

- **border**: 1px solid black;
- **border-bottom**: thick dashed black;