



## Les 7

Lay-out:  
Boxmodel - weergavemodel

# Content

- Boxmodel
- Marges – padding – randen
- Overflow
- Weergave model: display
- Visibility

# Boxmodel

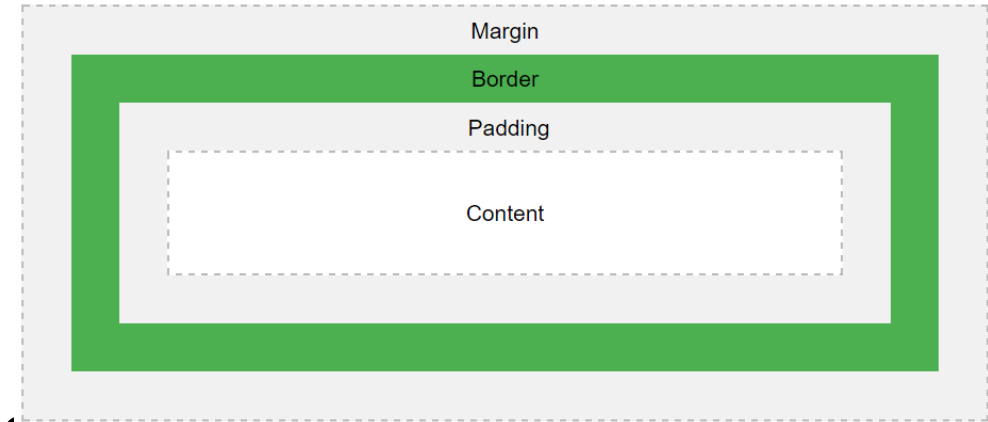
# Box

- Een html pagina is opgebouwd uit blokken. Deze blokken worden bovenop mekaar geplaatst.
- De CSS box stelt ons in staat om een uitgebalanceerde layout te maken. De opbouw van een blok wordt beschreven in het [boxmodel](#).
- Elk element op de pagina kan je zien als een box

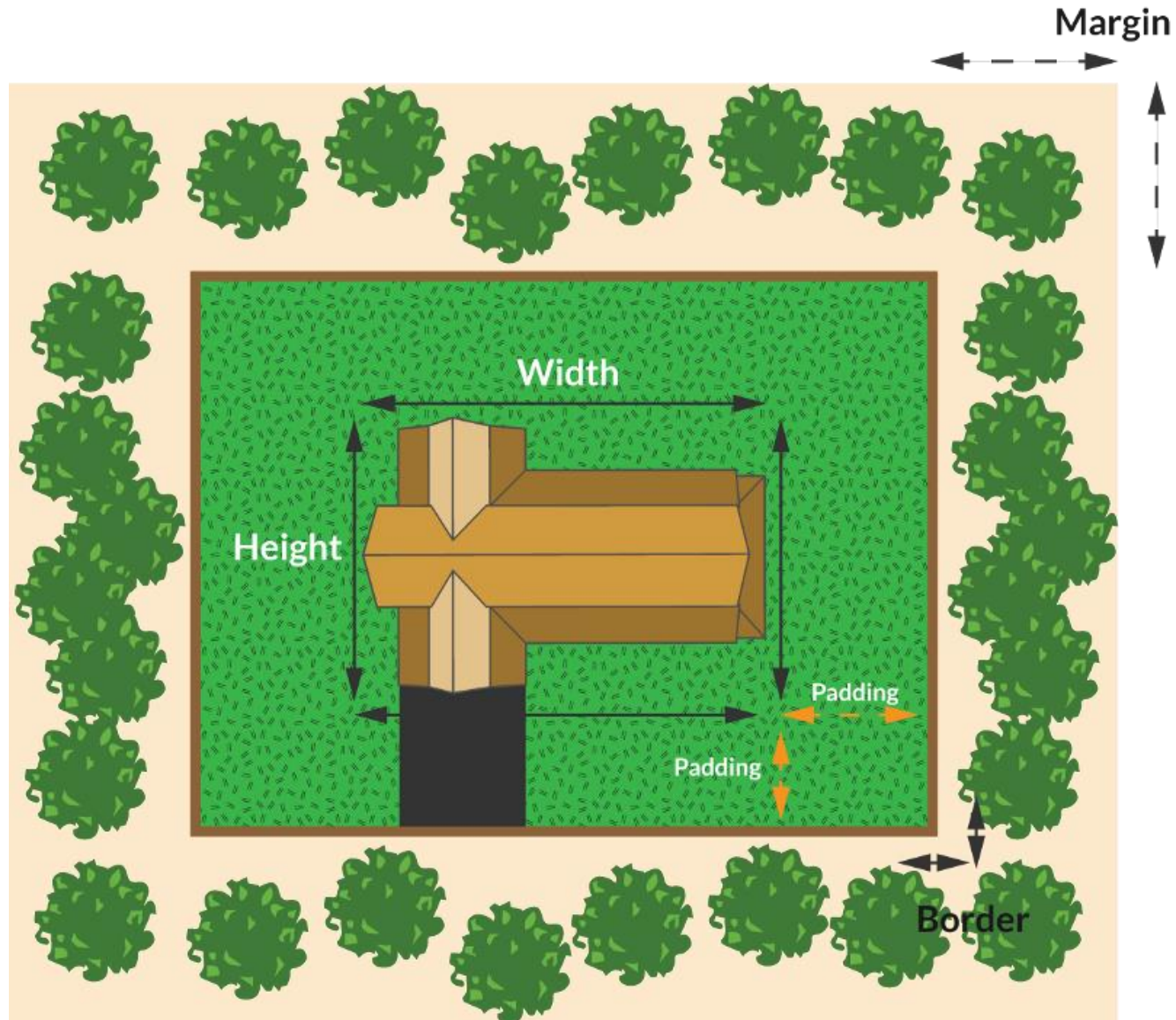


# Box

- De 4 onderdelen van een box
  - Inhoud (content): bevat tekst en afbeeldingen.  
Dit wordt bepaald door de width en height properties
  - [padding](#): ruimte rond de inhoud, zorgt ervoor dat de rand (border) niet direct aan de inhoud vasthangt.
  - [border](#): een rand rond de padding.
  - [margin](#): ruimte rond de border, aan de buitenkant.  
Deze wordt niet beïnvloed door een achtergrondkleur van het element, is volledig transparant.

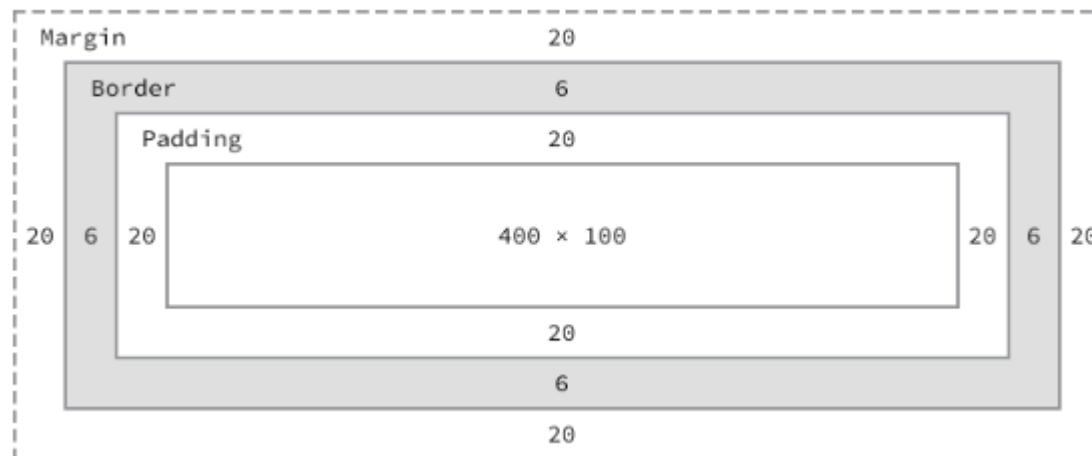


# Box



# Box

- De **totale ruimte** die een blok inneemt
  - Actual width = margin-right + border-right + padding-right + width + padding-left + border-left + margin-left
    - Width: 492px = 20px + 6px + 20px + 400px + 20px + 6px + 20px
  - Actual height = margin-top + border-top + padding-top + height + padding-bottom + border-bottom + margin-bottom
    - Height: 192px = 20px + 6px + 20px + 100px + 20px + 6px + 20px





# Box

Property [box-sizing](#) bepaalt hoe width en height door de browser worden gebruikt.

Er zijn 2 mogelijke waarden:

**box-sizing: content-box;**

De ingestelde width en height hebben enkel betrekking op de **content**.

Totale breedte box=width+padding left/right+border left/right+margin left/right.

Totale hoogte box=height+padding top/bottom+border top/bottom+margin top/bottom

**box-sizing: border-box;**

De ingestelde width en height hebben betrekking op de **content + de padding + de border**.

Totale breedte box=width+margin left/right.

Totale hoogte box=height+margin top/bottom.



# Box: width – height

- Eigenschappen: [width](#) en [height](#)
  - auto (standaardwaarde) – box is zo groot dat de inhoud erin past.
  - lengte of percentage (zie hoofdstuk 8)
  - max-content: afmeting van het grootste item in de box
  - min-content: de kleinst mogelijke afmeting zonder overloop te veroorzaken (vb. langste woord).

## Box: min-width/max-width – min-height/max-height

[min-width](#) en [max-width](#) bepalen minimale en maximale breedte.

[min-height](#) en [max-height](#) bepalen minimale en maximale hoogte.

Mogelijke waarden:

- auto: standaardwaarde bij \*-width
- none: enkel bij \*-height, tevens standaardwaarde
- lengte of percentage
- max-content
- min-content

**Margin – padding – border**

# Marges - margin

- Ruimte aan de buitenkant van een element. Gebruiken voor de creatie van **witruimte** tussen elementen
- Een marge is **altijd transparant**. We zien de **achtergrondkleur van het parent element**.
- Voor inline elementen kan de top en bottom margin niet worden ingesteld!!!
- Blokken centreren door linker- en rechtermarge op auto te plaatsen

# Marges - margin

Eigenschappen:

- [margin-top](#)
- [margin-right](#)
- [margin-bottom](#)
- [margin-left](#)
- [margin](#) (shorthand)

Een voorbeeld van een paginacontainer van 1000px breed die altijd in het midden staat:

```
<body>
  <div class="container">
    ... alle inhoud van de pagina ...
  </div>
</body>
/* CSS */
.container {
  margin: 0 auto;
  width: 1000px;
}
```

# Samengevoegde marges

- Marges worden aan onder en bovenkant samengevoegd, zodat er geen dubbele marges ontstaan.
- Wanneer de bottom margin van 1 element de top margin van een ander element overlapt, is het resultaat **NIET** de som, maar **de grootste margin**.
- Meer informatie over dit probleem: <https://css-tricks.com/what-you-should-know-about-collapsing-margins/>

# Padding

- Ruimte tussen de inhoud en de rand
- Een padding is **altijd transparant**. We zien de **achtergrondkleur van het element** waarop de padding wordt toegepast

Eigenschappen:

- [padding-top](#)
- [padding-right](#)
- [padding-bottom](#)
- [padding-left](#)
- [padding](#) (shorthand)



# Randen - border

- Van een border kunnen we 3 waarden instellen :  
breedte, kleur en stijl (kan ook voor elke zijde apart worden ingesteld border-top/right/bottom/left-width/style/color)  
[border-width](#) (px, %, em, rem, thin, medium, thick)  
[border-style](#) (none / hidden / dotted / dashed / solid / double / groove / ridge / inset / outset)  
[border-color](#) (een kleurwaarde of transparent)
- Men kan ook de drie eigenschappen combineren in een shorthand:  
[border](#): 1px solid black; (volgorde van de waarden is niet belangrijk): dit is een zwarte solid border van 1px.

# Overflow

# Overflow

- Overflow treedt op indien de content te groot wordt voor de box. Dit kan enkel als een breedte of hoogte is ingesteld.
- Wat er met de inhoud gebeurt hangt af van de waarde die wordt toegekend aan [overflow](#).

Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
Aut repellat incididunt non, rerum fuga deserunt dolore  
obcaecati qui eaque quaerat consequuntur maiores dolore  
deleniti laborum provident dicta ducimus possimus  
voluptatem quibusdam. Totam ex non expedita temporibus  
sed recusandae, facilis error explicabo assumenda dolores  
quibusdam saepe quis perspiciatis incididunt a illum mollitia  
quia harum qui necessitatibus doloreque aliquid dolorum  
cupiditate! Nihil optio ipsam provident ullam, ut  
temporibus porro illo aut nesciunt vitae saepe blanditiis,  
molestias itaque.

# **Weergave model - display**

# Weergave model – display

- Verdeling van de boxen over de pagina: lay-out.
- Delen naast elkaar plaatsen of onder elkaar plaatsen wordt bepaald door het type box.
- Het type box wordt bepaald door de eigenschap display.
- Elk element heeft een standaard display (block – inline - list-item ...), vastgelegd in de html beschrijving.
- Op dit moment is er een hele reorganisatie en vernieuwing bezig ivm de display eigenschap.

# Weergave model – display

- Er wordt onderscheid gemaakt tussen:
  - display-outside type: gedrag van boxen tov andere boxen:  
**block – inline – (run-in)**
  - display-inside type: gedrag van de afstammelingen van de box:  
**flow – flow-root – table – flex – grid – list-item**
  - display-box type: de parent maakt geen box aan:  
**none:** de children maken ook geen box aan, maw. wordt niet gerenderd  
**contents:** de children maken boxes aan, vb grid- en flex lay-out.
  - display-legacy: waarden kunnen ook gekoppeld worden:  
**inline-block – inline-grid – inline-flex**
- Interessant voorbeeld: <https://developer.mozilla.org/en-US/docs/Web/CSS/display#display-legacy>

# Visibility



# Visibility

- Een element verbergen met de eigenschap [visibility](#).  
Mogelijke waarden: visible (default) – hidden.
- Er wordt een box aangemaakt en bestaat ook in de boomstructuur.
- Het wordt niet weergegeven in de browser, maar de ruimte nodig voor de box wordt wel gereserveerd op het scherm.