

College 1 intro Client - Server

“Dat zoeken we op!”

Teruggekomen van vakantie...

- ...zet ik de computer aan.



Teruggekomen van vakantie...

- ...zet ik de computer aan...
- ...en wacht ik (soms heel lang)
- Iemand en idee wat er gebeurt bij het opstarten van je computer?



Besturingssysteem

- De software waardoor je computer taken kunt laten uitvoeren. Alleen een printplaat met chips doet niet zoveel
- Beheren en verdelen van geheugen, processortijd
- Verwerken van gegevens uit invoerapparaten of bestemd voor uitvoerapparaten
- Etc. etc.



Browser opstarten

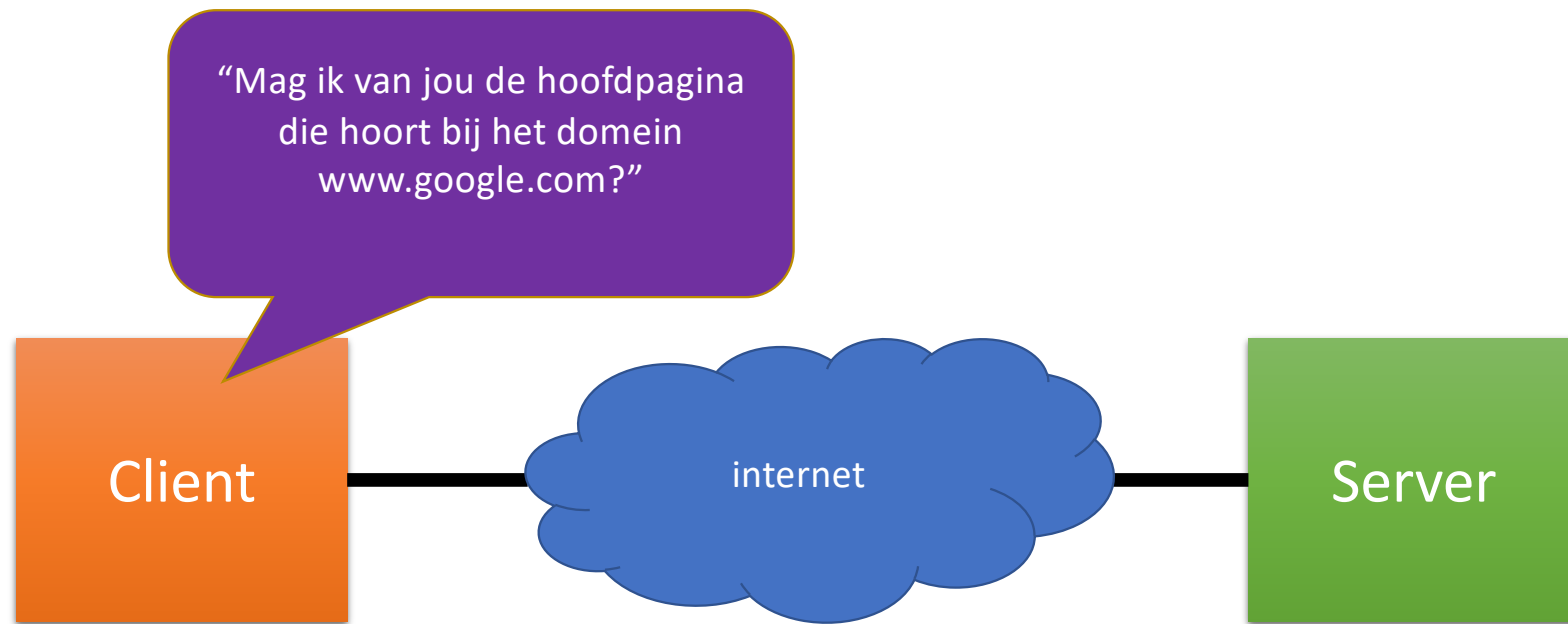
- Wat doet een browser precies?



Browser opstarten

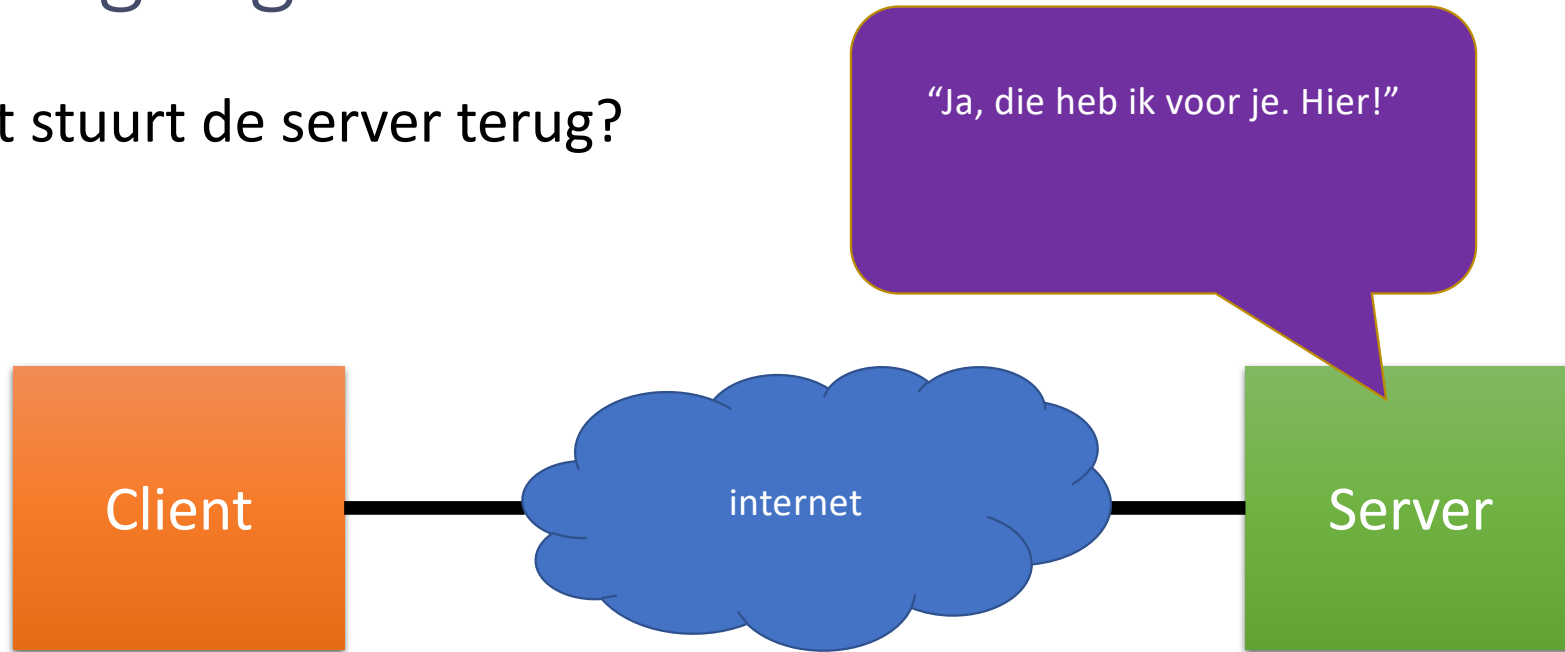
- Wat doet een browser precies?
- Verzoeken sturen naar andere computers -> servers
- Webpagina's presenteren

www.google.com



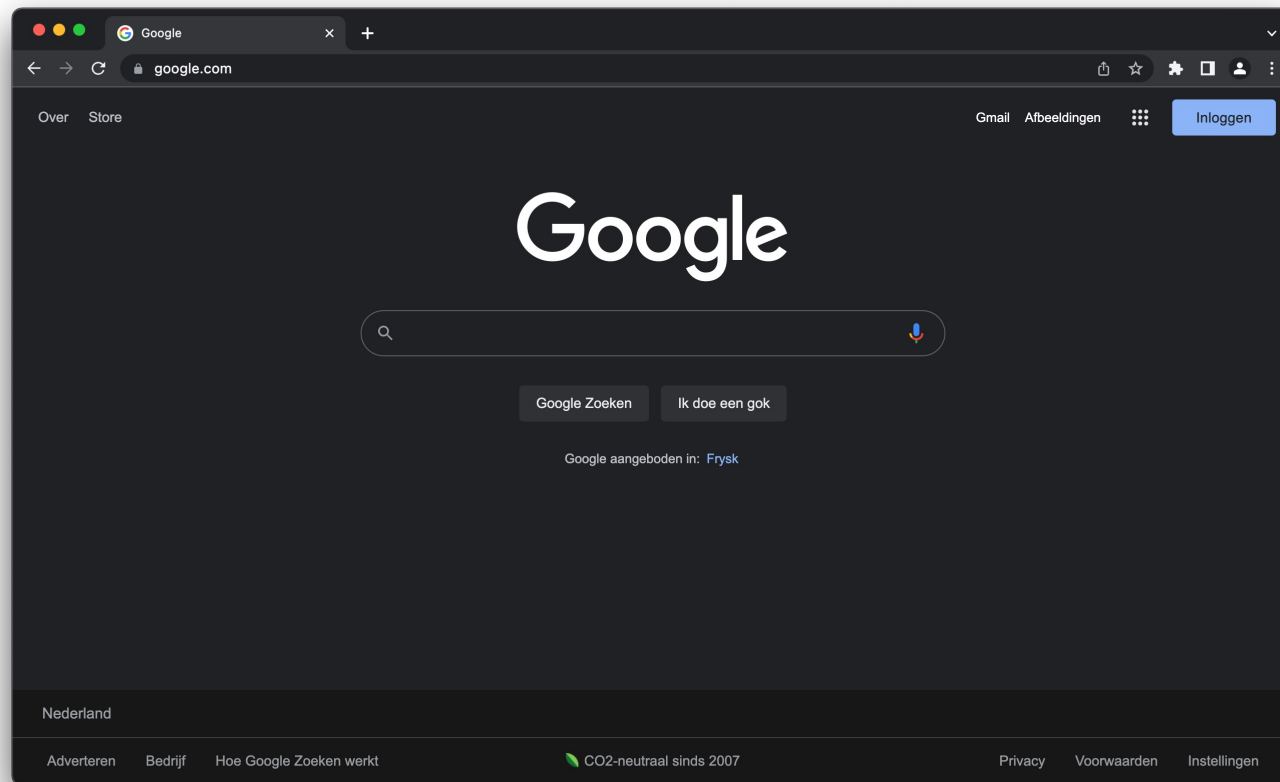
www.google.com

- Wat stuurt de server terug?



Browser bekijken

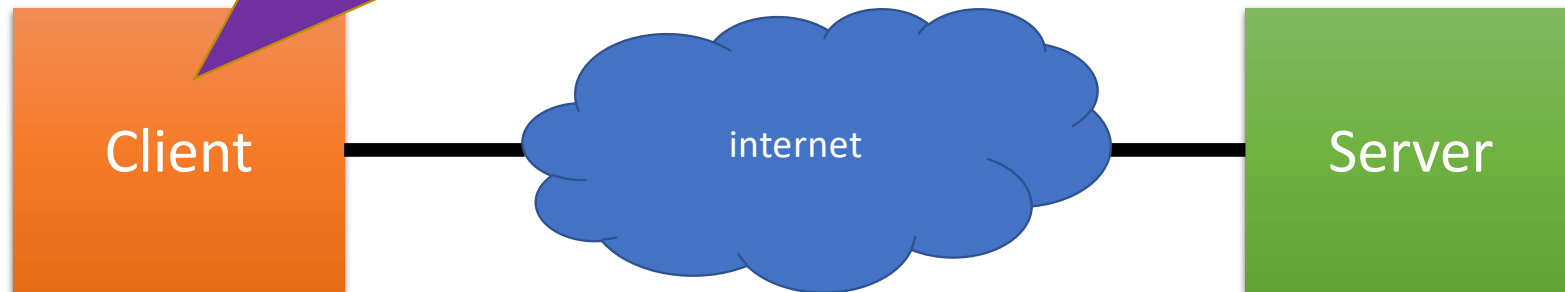
- Laten we het nu zelf doen



www.google.com

“Mag ik van jou de hoofdpagina in de map ‘search’ die hoort bij het domein www.google.com”?
Ik geef je hier extra informatie bij:

q=test



College 2 HTML en CSS

HTML: tag, attribute, value

HTML

HTML is een computertaal waarmee je de inhoud van een webpagina beschrijft.

- Tag: geeft aan wat voor inhoud een stuk webpagina bevat.
 - Opening tag: geeft begin van stuk aan
 - Closing tag (met /): geeft eind van stuk aan
- Attribute: een eigenschap die je aan een tag kunt geven
- Value: de waarde die je de eigenschap wilt geven, bijvoorbeeld een link.

HTML voorbeeld

```
<A HREF="contact.html">
Contactinformatie
</A>
```

Uitzondering: van
 en bestaan geen closing tags.

Een tag kan meer dan één attribute hebben.

HTML: Parent, Child, Sibling

```
<html>
  <head>
    <title>  Vacatures  </title>
  </head>
  <body>
    <p>  Voordelen als je html kent  </p>
    <ol>
      <li>  Leuk werk  </li>
      <li>  Lekker verdienen  </li>
    </ol>
  </body>
</html>
```

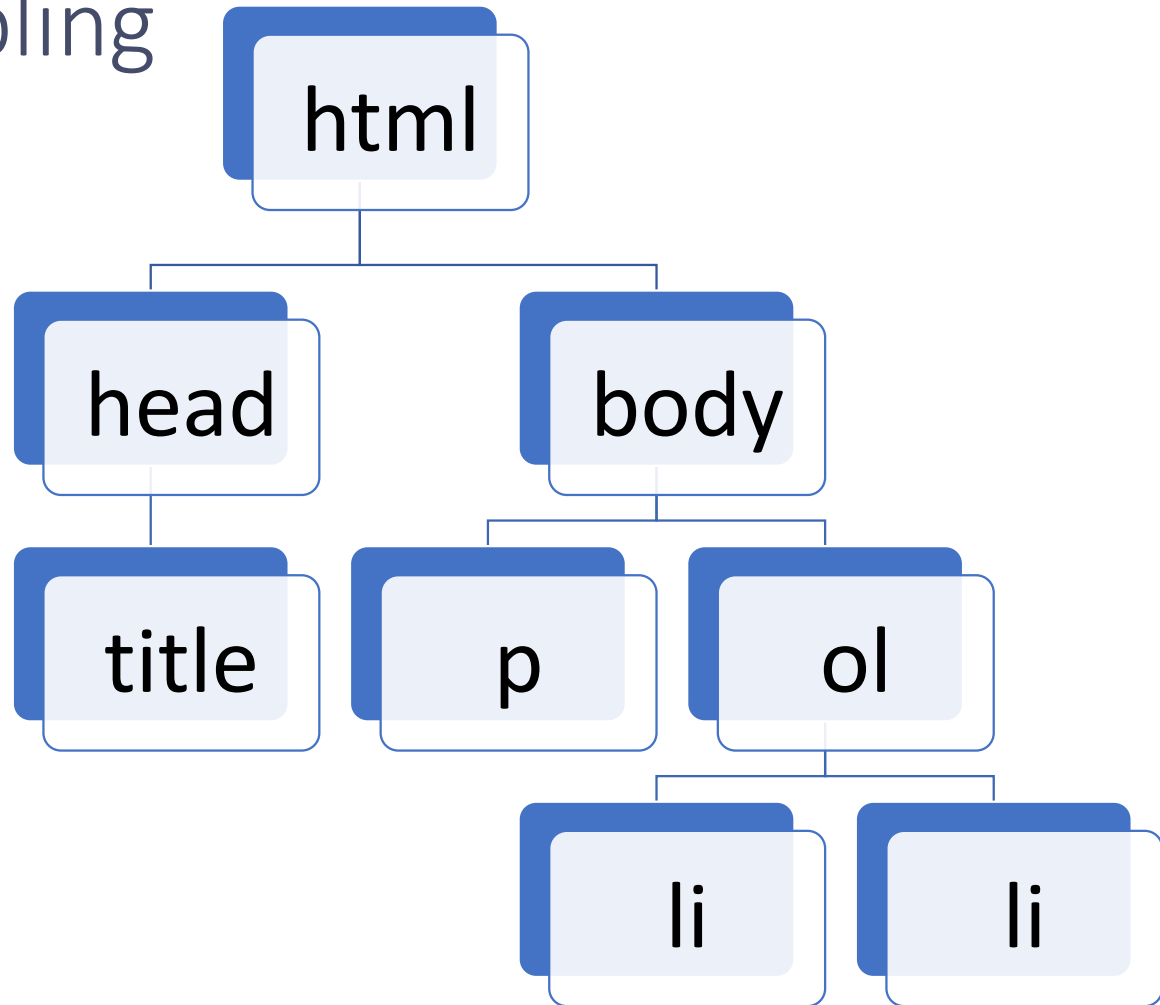
11-12-2023

Parent, Child, Sibling (stamboom)

- Je kunt Parents, Children en Sibling in een boomstructuur (stamboom) beschrijven.
- Schrijf de navolgende tekst in de vorm van een stamboom

```
<html>  <head>  <title>  Vacatures  </title>
</head>  <body>  <p>  Voordelen als je html kent
</p>  <ol>  <li>  Leuk werk  </li>  <li>  Lekker
verdiene  </li>  </ol>  </body>  </html>
```

Parent, Child, Sibling



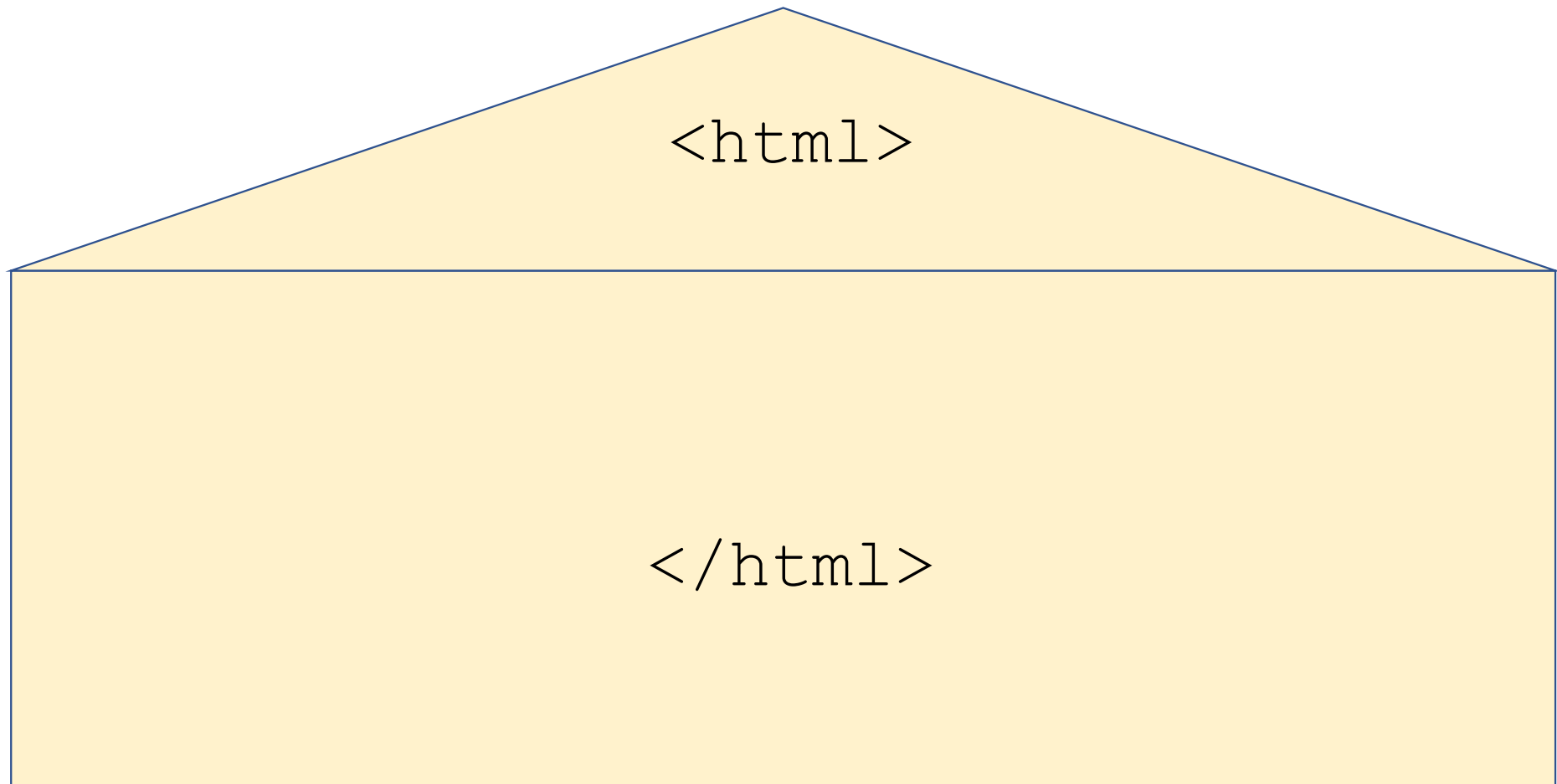
Parent, Child, Sibling (enveloppen)

Je kunt Parents, Children en Sibling met enveloppen beschrijven.

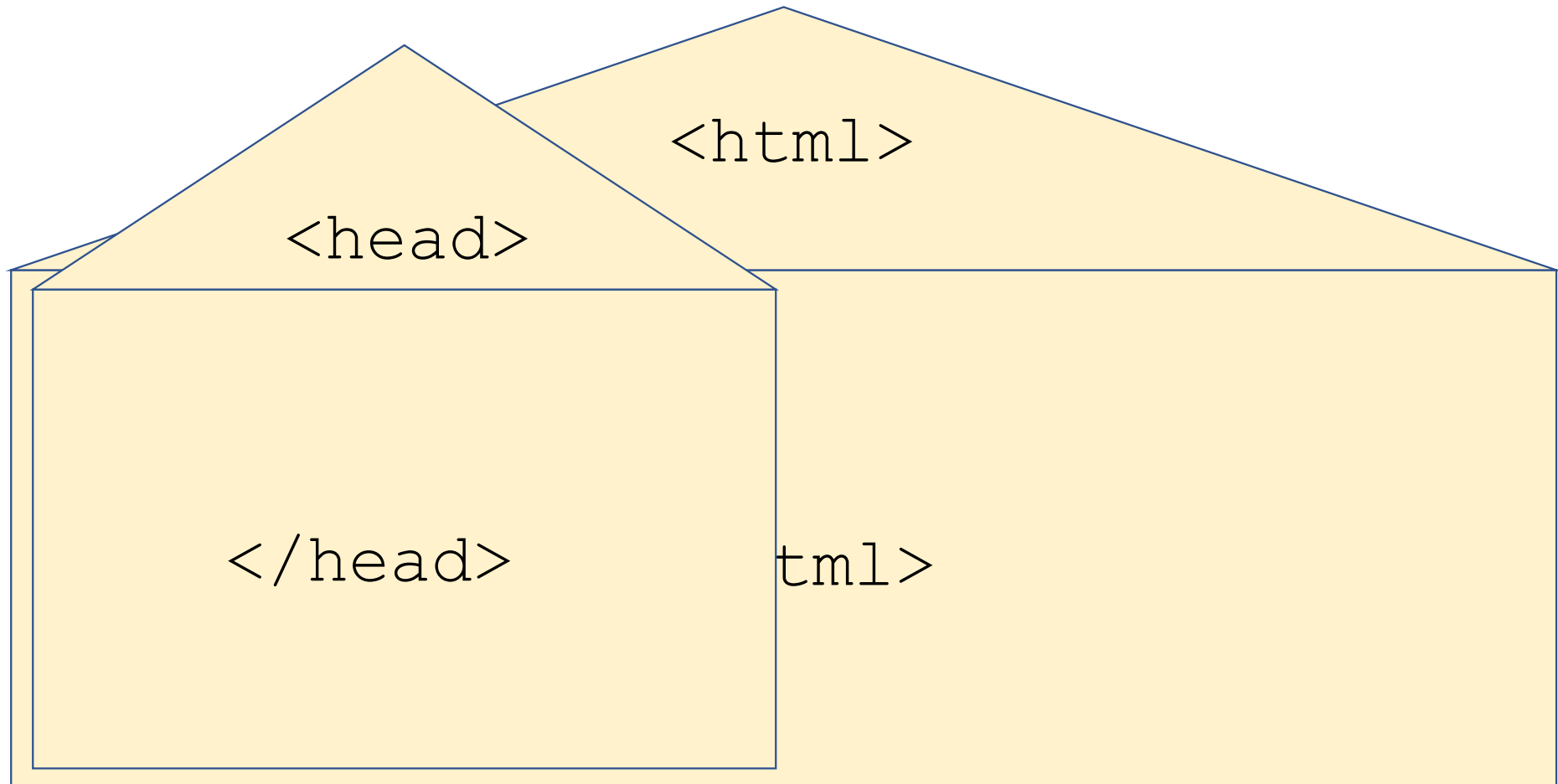
1. Knip de html-tags en de stukjes tekst hieronder los. Je heb dan 20 stukjes.
2. Pak een enveloppe: schijf de eerste opening-tag “<html>” op de voorkant en de bijbehorende closing-tag “</html>” op de achterkant. Neem een grote enveloppe.
3. Zoek de eerste tag na “<html>”, dit is het eerste child van “<html>”. Kijk of er nog meer children zijn van “<html>”. Maak voor elk child een kleine enveloppe en stop die in de grote enveloppe van hun parent.
4. Ga zo door, met steeds kleinere enveloppen, tot je bij children bent die geen parent meer zijn. Alle tags zijn dan op. De 4 losse teksten kun je in de enveloppe stoppen met de tag waar ze tussen staan.

```
<html> <head> <title> Vacatures </title> </head> <body> <p>  
Voordelen als je html kent </p> <ol> <li> Leuk werk </li> <li>  
Lekker verdienen </li> </ol> </body> </html>
```

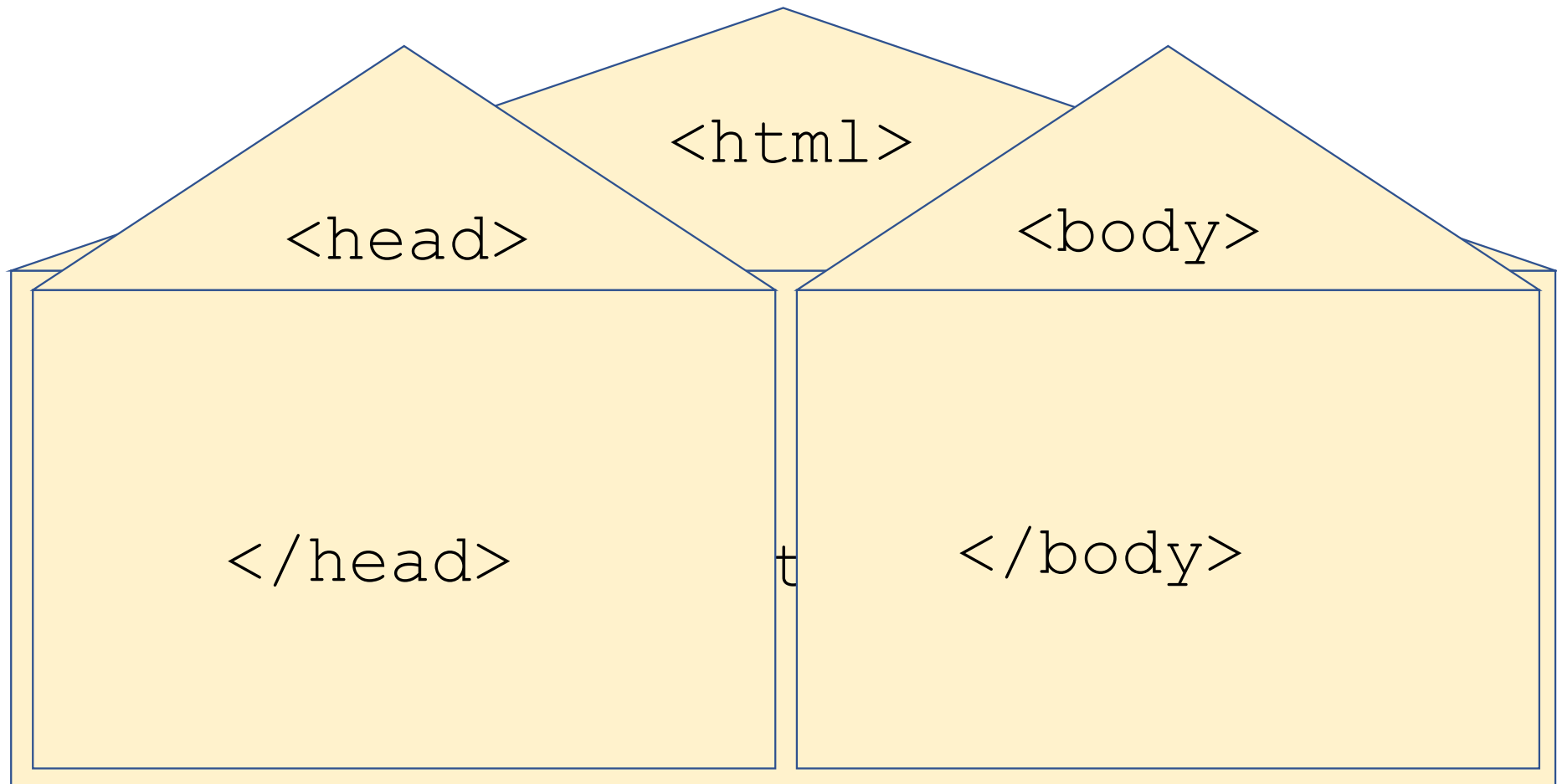
Parent, Child, Sibling



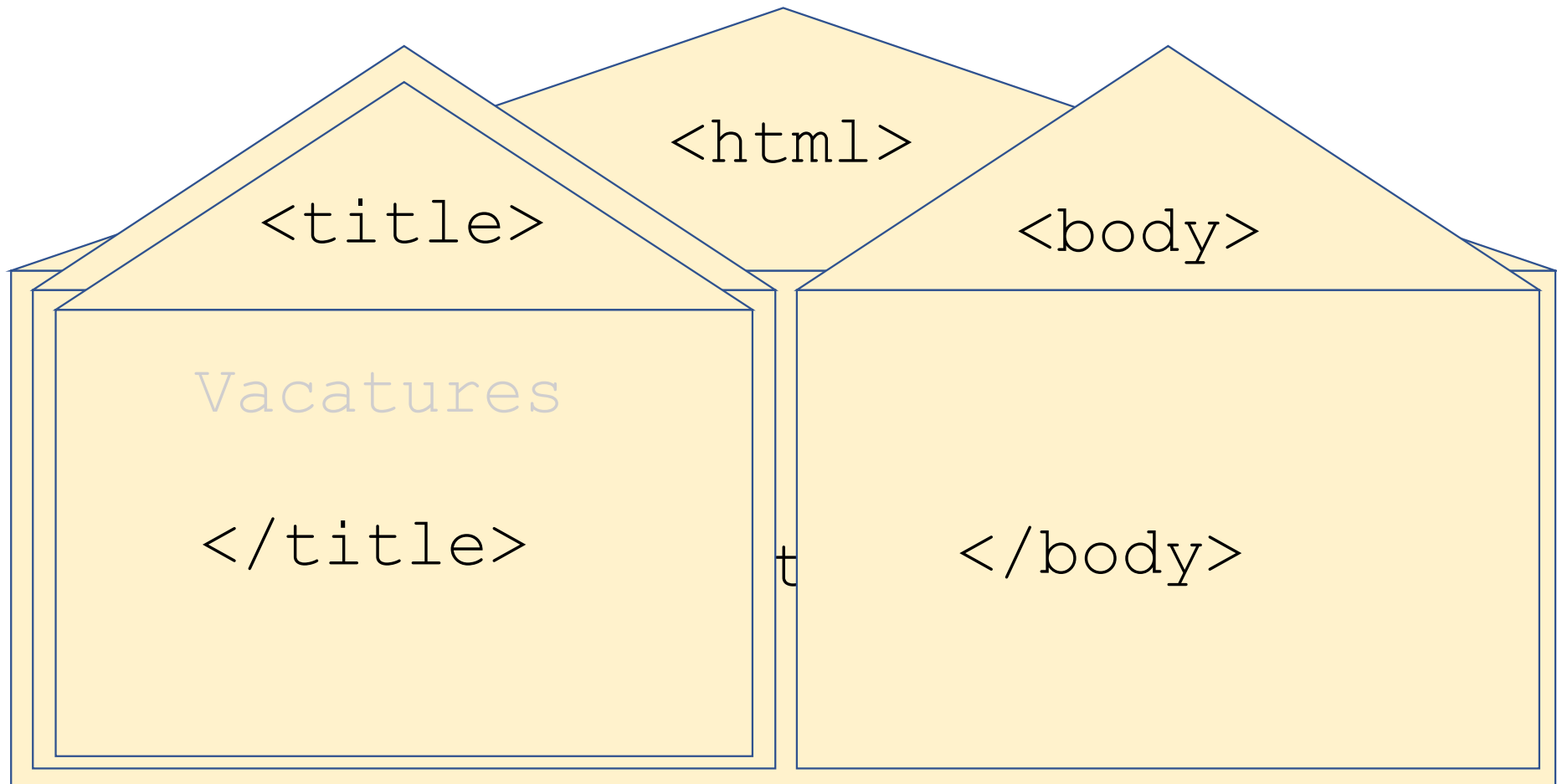
Parent, Child, Sibling



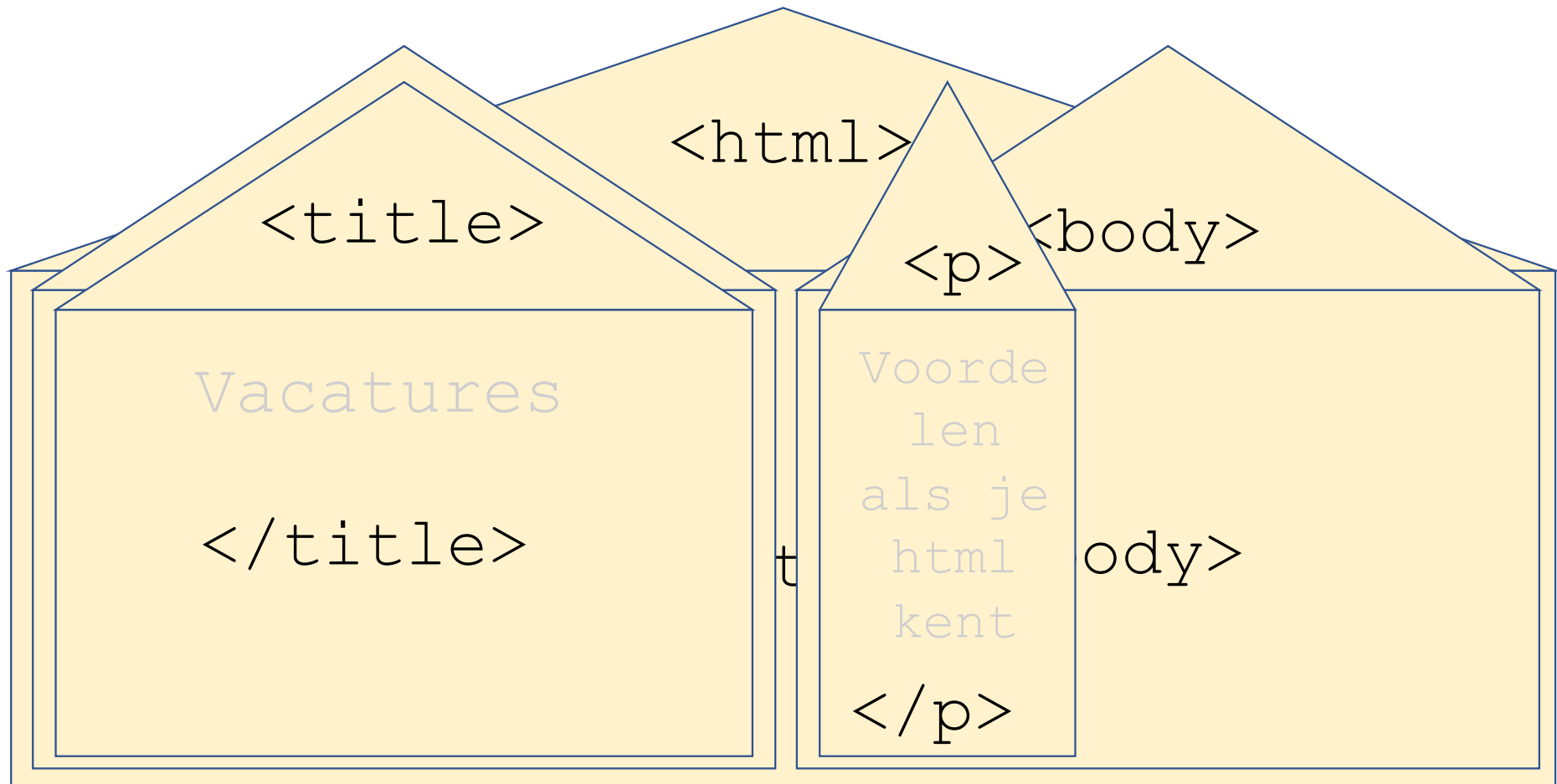
Parent, Child, Sibling



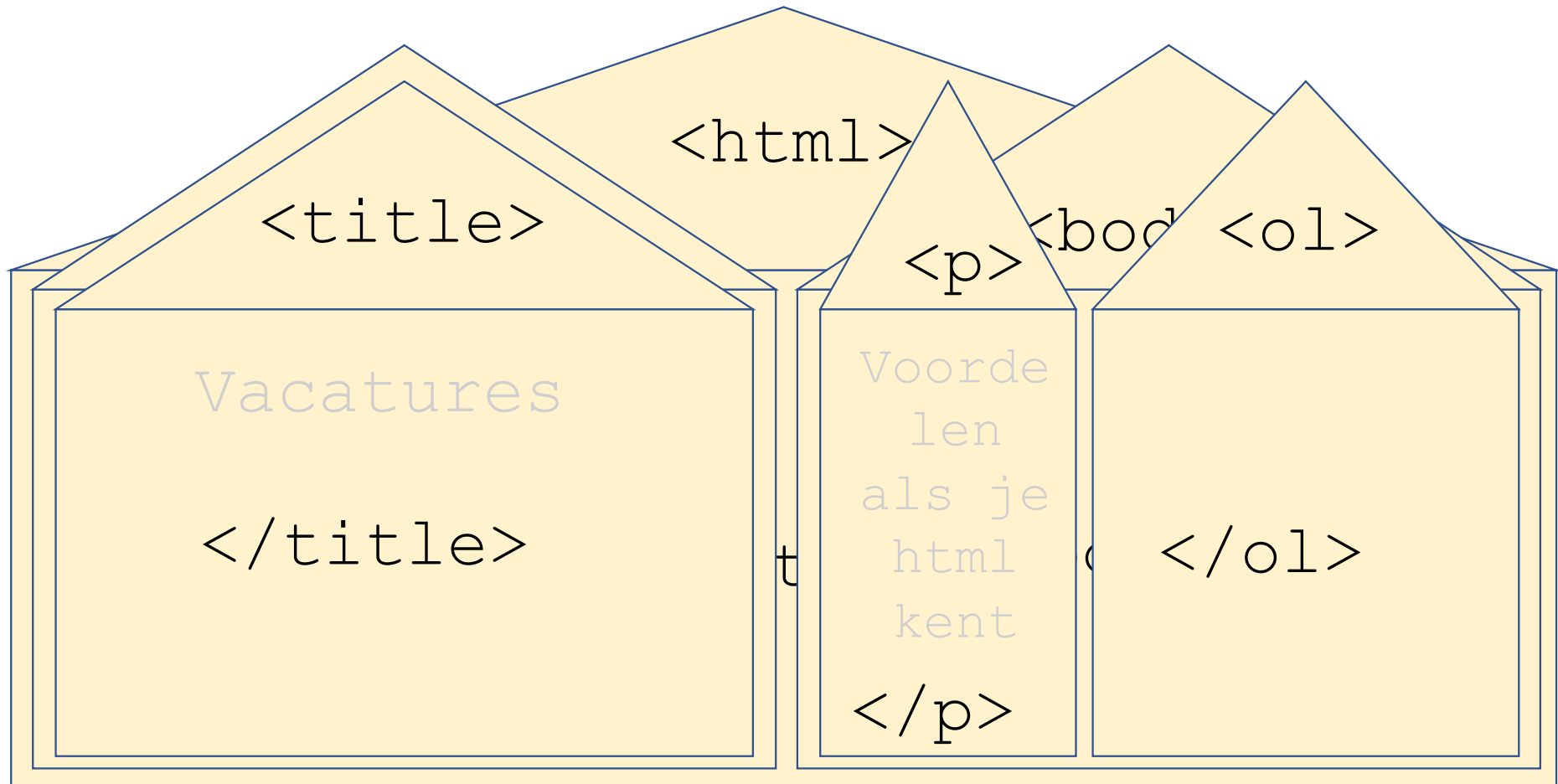
Parent, Child, Sibling



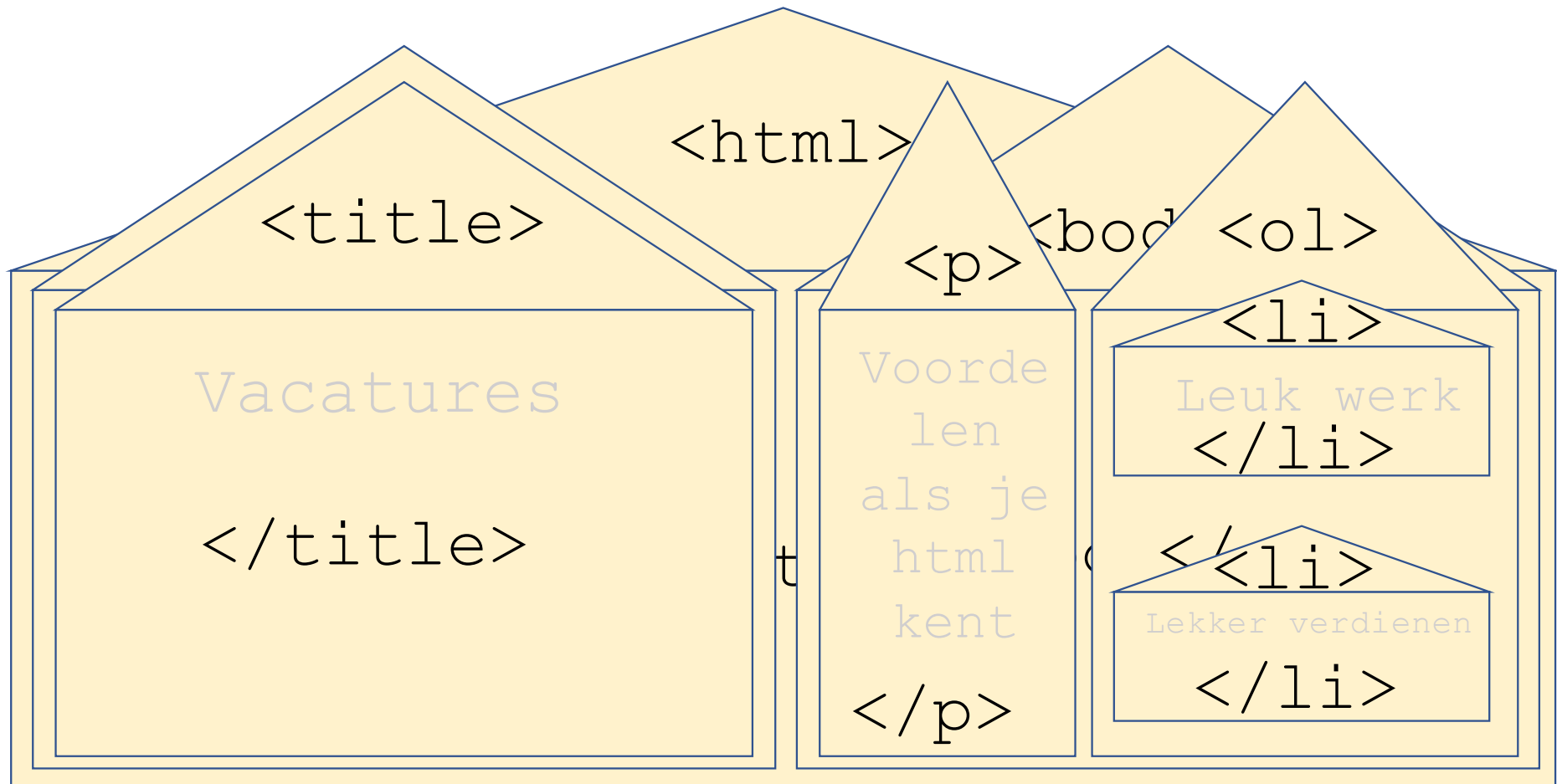
Parent, Child, Sibling



Parent, Child, Sibling



Parent, Child, Sibling



Samenvatting Parent, Child, Sibling

- Een **Parent** is een element in een programmeertaal, dat één of meerdere andere elementen (**Children**) bevat.
- Children van dezelfde Parent noemen we **Sibling**.
- Een element dat een Child is, kan weer Parent zijn van andere elementen. Dit fenomeen noemen we **Nesting**.
- Parent, Child en Sibling zijn Engelse begrippen. In het Nederlands zou je kunnen zeggen: Ouder, Kind en Zus/Broer. Binnen de informatica is gebruikelijk om de Engelse begrippen te gebruiken.

Separation of Concerns

Wat heeft spaghetti met HTML en CSS te maken?



Separation of Concerns

Separation of Concerns in de informatica betekent dat je software of hardware modulair opbouwt.

Anders gezegd: Code die je maakt moet je opdelen in onderdelen die elk een duidelijk af te bakenen functie hebben.

Door Separation of Concerns kun je:

- Code die anderen gemaakt hebben sneller te begrijpen
- Code later gemakkelijker aanpassen

Toepassing van SoC bij HTML+CSS

- HTML beschrijft de inhoud, wat er op de webpagina staat
- CSS beschrijft de opmaak, hoe het er op de webpagina uit moet zien

Scheiden van inhoud en opmaak

In moderne webpagina's worden inhoud en opmaak gescheiden.

- HTML beschrijft de inhoud, wat er op de webpagina staat
- CSS beschrijft de opmaak, hoe het er op de webpagina uit moet zien

Voordelen:

- Webpagina's die anderen gemaakt hebben kun je sneller te begrijpen
- Webpagina's kun je later gemakkelijker aanpassen

CSS selector, property, value

CSS

CSS is een computertaal waarmee je de opmaak van een webpagina beschrijft.

- Selector: wijst aan welk stuk html je wilt aanpassen, bijvoorbeeld alle tekst tussen `<h1>` en `</h1>`
- Property: geeft aan welke eigenschap je wilt aanpassen, bijvoorbeeld `color`
- Value: de waarde die je de eigenschap wilt geven, bijvoorbeeld `red`.

CSS voorbeeld

```
h1 {  
  color: red;  
}
```

CSS selector

Er zijn 3 soorten selectoren:

- Tags
- Classes
- ID's

HTML voorbeeld

```
<h1 class="title">  
rode letters gele achtergrond  
</h1>
```

```
<p ID="publish-time">  
grijs  
</p>
```

CSS voorbeeld

```
h1 {  
    color: red;  
}  
  
.title {  
    background-color: yellow;  
}  
  
#publish-time {  
    color: gray;  
}
```

CSS selector specificiteit

Als er meerdere selectoren naar hetzelfde stuk html verwijzen, dan bepaalt de meest specifieke selector hoe de layout wordt.

- ID's zijn specifieker dan Classes.
- Classes zijn specifieker dan Tags.

HTML voorbeeld:

```
<h1 class="title"
      ID="publish-time">
```

Grijze letters, gele achtergrond

```
</h1>
```

CSS voorbeeld

```
h1 {
    color: red;
}
.title {
    background-color: yellow;
}
#publish-time {
    color: gray;
}
```

HTML met CSS voorbeeld

HTML voorbeeld

```
<h1 class="title">
  Roosterwijziging
</h1>
<p>
  <div id="publish-time">
    Maandag 08:30
  </div>
</p>
<h1>
  Eerste lesuur
</h1>
<p>
  vervallen
</p>
```

CSS voorbeeld

```
h1 {
  color: green;
}
p {
  color: blue;
}
.title {
  color: red;
}
#publish-time {
  color: gray;
}
```

Kahoot - CSS

The Kahoot! logo is displayed in a large, white, bold, sans-serif font with a slight shadow effect, set against a solid purple rectangular background.

<https://play.kahoot.it/#/k/b5e2cda3-b05d-435b-a35e-cf119f3af0b7>

Praktisch voorbeeld

The screenshot shows a Replit editor interface. The top bar includes the Replit logo, the username 'vangeest', the project name 'CSS-tag-class-id', a green play button, and an 'Invite' button. The editor is split into two panes. The left pane shows the source code for 'index.html' with line numbers 9 through 21. The code defines a table with three rows of animal names. The right pane shows the rendered webview, displaying the table and the text 'Vijf dieren Five animals' below it.

```
9 <body>
10
11 <table>
12 <tr>
13   <td>aap</td><td>monkey</td>
14 </tr>
15 <tr>
16   <td>koe</td><td>cow</td>
17 </tr>
18 <tr>
19   <td>schaap</td><td>sheep</td>
20 </tr>
21 <tr>
```

Rendered output:

| | |
|--------|--------|
| aap | monkey |
| koe | cow |
| schaap | sheep |
| paard | horse |
| kat | cat |

Vijf dieren Five animals

<https://replit.com/@vangeest/CSS-tag-class-id#index.html>

Praktische tips

- Probeer dubbele code te voorkomen
- Keep it simple / niet te complex
- Gebruik nette layout, consistente naamgeving, commentaar waar nodig

Op die manier:

- Zorg je dat het leesbaar is voor anderen
(of voor jezelf als je er na een week weer meer verder moet)
- Maak je het systeem flexibel
(gemakkelijk aanpasbaar, ook voor wijzigingen die je nu nog niet kent)



Boxen

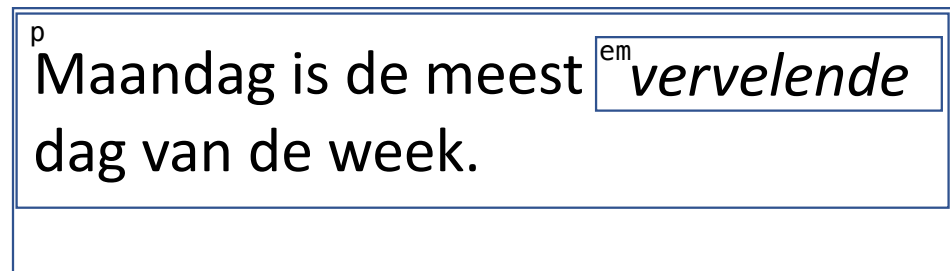
College 3

Boxmodel & Float

Boxen

- Alle HTML-elementen zijn een box
- HTML / CSS kent inline elementen en block elementen.
 - Een block element is een onzichtbare rechthoek op het scherm waarbinnen content wordt weergegeven.
 - Inline elementen staan in een tekstregel van een block element.

body



```
<body>  
  <p>  
    Maandag is de meest  
    <em>vervelende</em>  
    dag van de week.  
  </p>  
</body>
```

Inline en block elementen

Inline elementen

- Elementen worden op 1 regel geplaatst
- ``, ``, ``, ...



Dit is een korte tekst om te laten weten dat jullie allemaal geweldig zijn!

Block elementen

- Blokken worden onder (of naast) elkaar geplaatst
- `<h1>`, `<p>`, `<div>`, ...



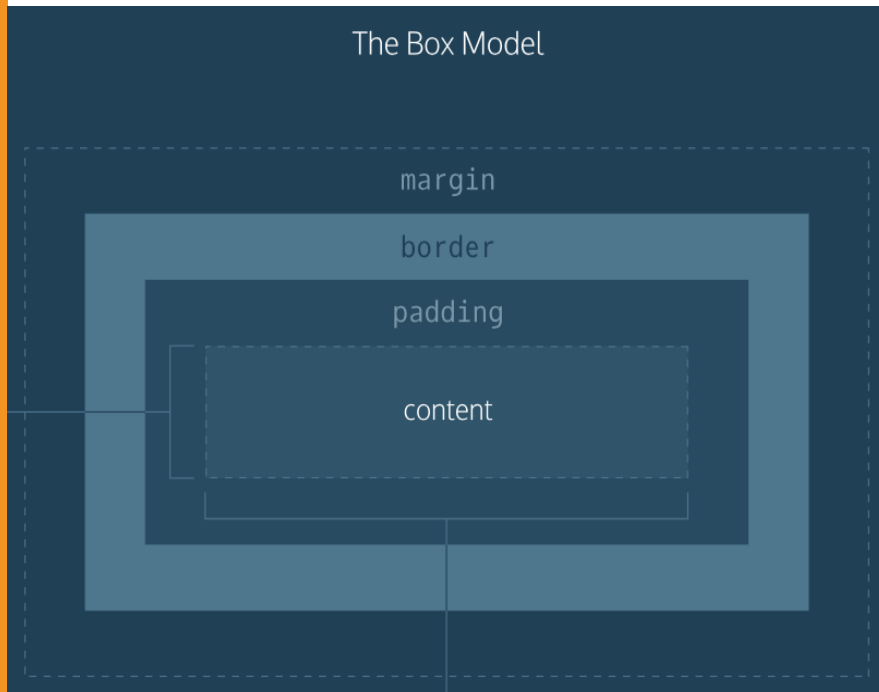
Dit is een korte tekst om te laten weten dat jullie allemaal geweldig zijn!

Op block-elementen kun je alle CSS-properties die bij het box-model horen toepassen.

Boxen

- Alle HTML-elementen zijn een box
- HTML / CSS kent inline elementen en block elementen.
 - Inline elementen staan in een regel.
 - Een block element is een onzichtbare rechthoek op het scherm waarbinnen een stukje html wordt weergegeven.
- *Children* zijn kleine boxen binnen de box van hun *Parent*

Layout van boxen



Een box bestaat uit 4 gebieden

- Content = Inhoud, plaats voor tekst in de box
- Padding = afstand tussen tekst en rand van de box
- Border = rand van de box
- Margin = leeg gebied, afstand met andere boxen

- Het uiterlijk van alle box onderdelen kun je veranderen

CSS-properties voor boxen

Enkele voorbeelden:

- De box moet bovenaan 10 pixels ruimte tot de box erboven hebben

```
margin-top: 10px;
```

- De rand moet paars gestippeld zijn en 5 pixels breed

```
border-style: dotted;  
border-color: purple;  
border-width: 5px;
```

```
border: 5px dotted purple;
```

- links en rechts moet de tekst 15 pixels ruimte tot de rand hebben

```
padding-top: 0px;  
padding-bottom: 0px;  
padding-left: 15px;  
padding-right: 15px;
```

```
padding: 0 0 15 15px;
```

```
padding: 0 15px
```

CSS-properties voor afmeting van boxen

Enkele voorbeelden:

- `width: 100px;`
Breedte 100 pixels
- `width 50%;`
Breedte 50% van de parent box
- `height: 200px;`
Hoogte 200 pixels
- `box-sizing: content-box;`
De opgegeven breedte en hoogte gaan over content, exclusief border+padding (dit is de standaard instelling).
- `box-sizing: border-box;`
De opgegeven breedte en hoogte gaan over content+border+padding.

Tip: reset standaard padding, margin en box-sizing

- Zonder CSS heeft body in de meeste browsers standaard een margin van 8px aan alle kanten
- Meest begrijpelijk als alle elementen standaard geen padding en margins hebben.
- Border-box voelt het meest natuurlijk aan.
- ```
* { margin: 0; padding: 0; box-sizing: border-box; }
```



# Raar gedrag (goed om te weten)

- Inline elementen: alleen horizontale margins hebben effect:
- Inline elementen: padding werkt, maar de regel loopt gewoon door
- Block elementen: bij twee blokken die onder elkaar staan en richting elkaar margin hebben, wordt alleen de grootste margin genomen

# Generieke elementen: span

- Soms wil je een stukje tekst **anders** opmaken (en zijn `<em>`, `<strong>` etc. niet voldoende)
- Daarvoor is `<span>`:

```
<p>
 Soms wil je een stukje tekst anders opmaken
</p>
```

```
span.opvallend {
 color: green;
}
```

# Generieke elementen: span

- Geef je geen CSS-layout aan `<span>`, dan zie je niets bijzonders:  
Soms wil je een stukje tekst anders opmaken

```
<p>
 Soms wil je een stukje tekst anders opmaken
</p>
```

- Span is een generiek inline element. Je moet helemaal zelf aangeven wat de opmaak is

# Generieke elementen: div

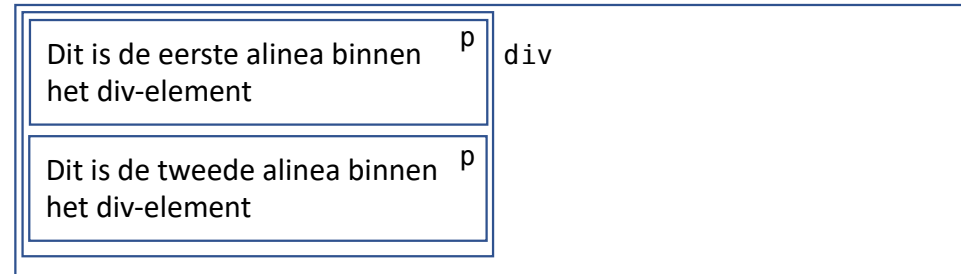
- Soms wil je block elementen onderbrengen / groeperen in een gezamenlijke parent met een bepaalde opmaak, zoals breedte of positie.
- Daarvoor is `<div>`
- Je kunt je een div voorstellen als een 'hulprechthoek'

# Generieke elementen: div

```
<body>
 <div class="halve-breedte">
 <p>
 Dit is de eerste alinea binnen het
 div-element.
 </p>
 <p>
 Dit is de eerste alinea binnen het
 div-element.
 </p>
 </div>
</body>
```

```
div.halve-breedte {
 width: 50%;
}
```

body



# Andere generieke block elementen

- Vroeger was div het enige generieke block element
- Voor bijv. zoekmachines gaf dit weinig betekenis. Een div met de hoofdtekst is veel belangrijker dan een div met de navigatiebalk
- Daarom nieuwe elementen die net zo werken als een div, maar meer betekenis geven:
  - `<header>`
  - `<footer>`
  - `<main>`
  - `<nav>`
  - `<article>`
  - `<aside>`
  - ...



# Layout met float

Block elementen op de juiste plek zetten met behulp van float

# Verticale ordening

- Block elementen worden standaard altijd onder elkaar gezet. Ook als de breedte minder is dan de breedte van de parent.



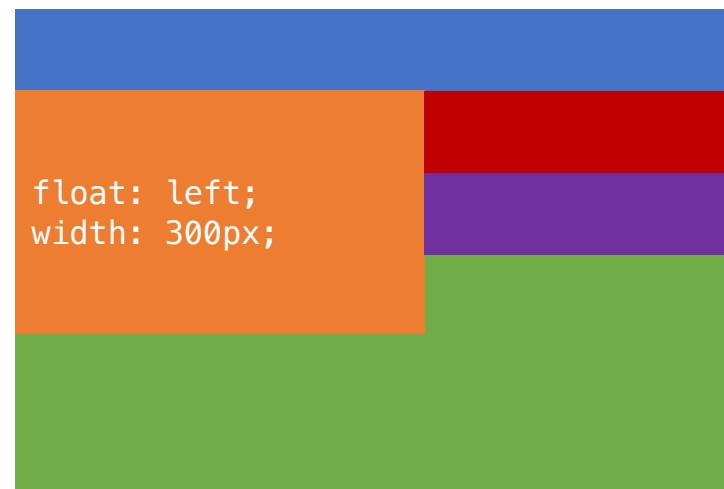


# Horizontale ordening

- Wil je block-elementen naast elkaar krijgen, dan gebruik je daarvoor float

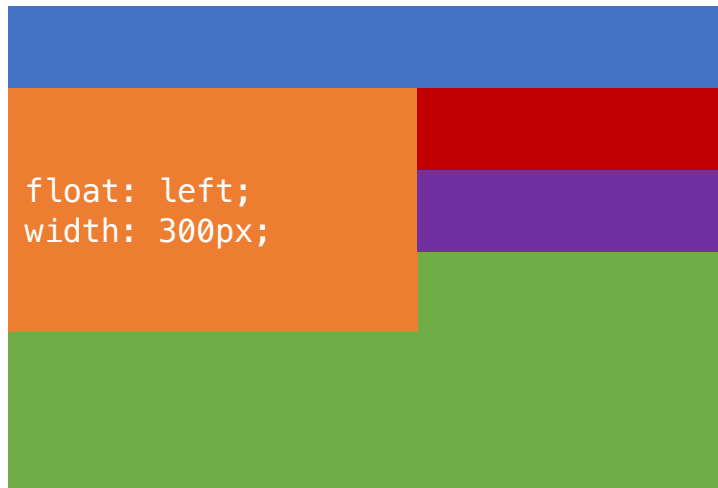


met drie block-elementen na het oranje blok

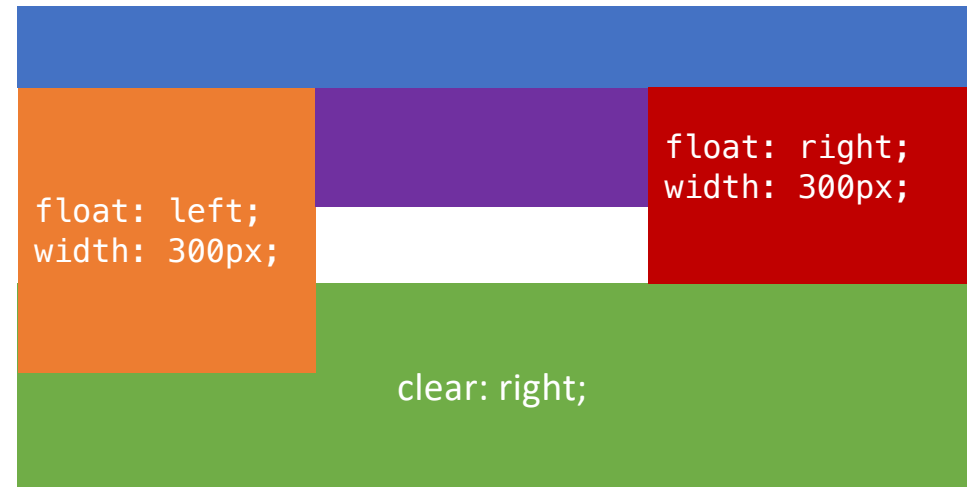
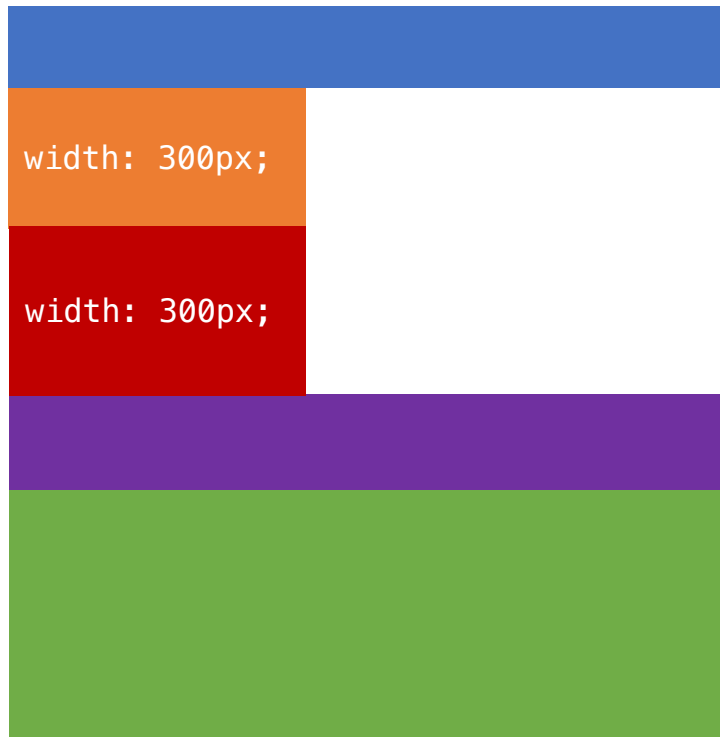


# Horizontale ordening

- Wil je dat een block element weer een 'normale' positie krijgt, dan gebruik je clear.

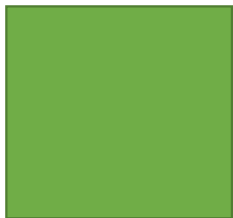


# Float left en float right



# Float gebruiken bij plaatjes

## Plaatje zonder float



Dit is een korte tekst om te laten weten dat jullie allemaal geweldig zijn!

## Plaatje met float

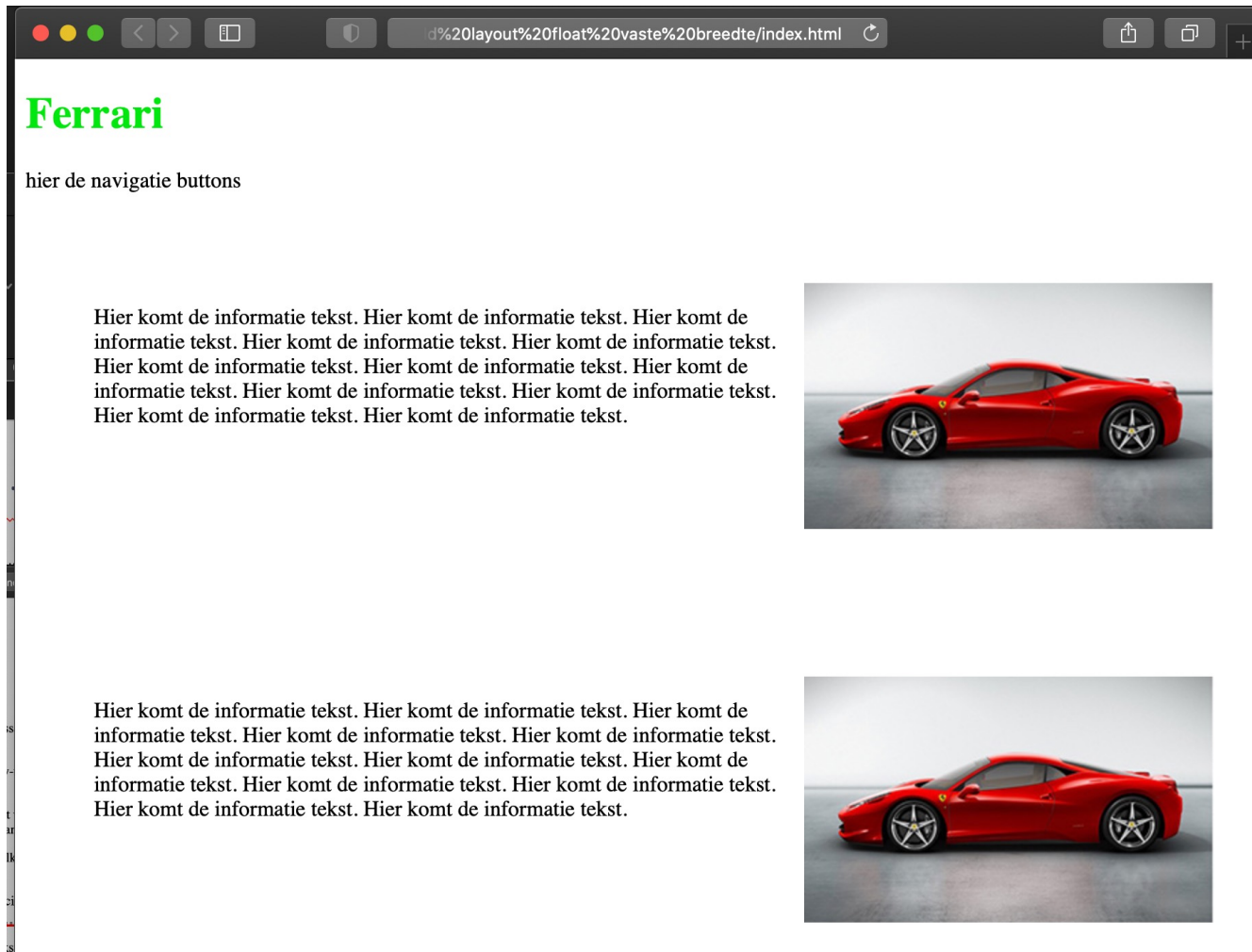


Dit is de eerste paragraaf.

Dit is de tweede paragraaf, die is iets langer.

Dit is de volgende paragraaf na clear.

# Layout maken met float



11-12-2023





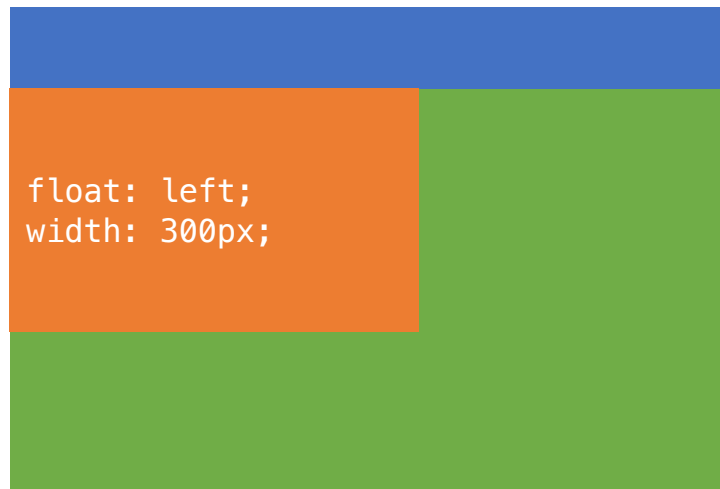
# Lay-out – deel 2



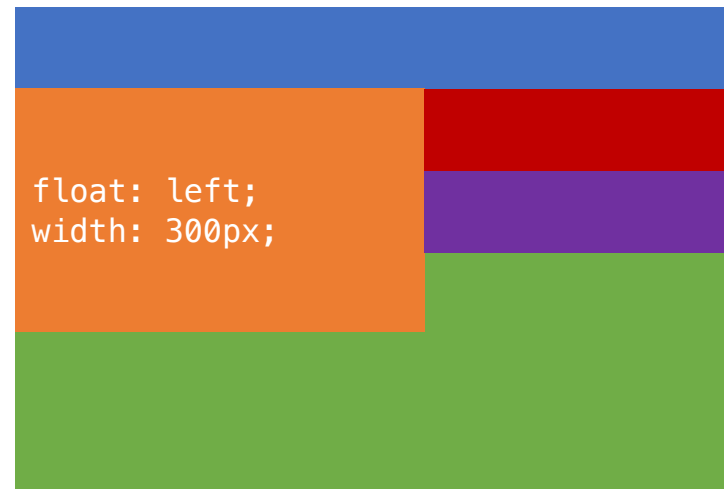
# College 4 Layout

# Scherper uitgelegd:

- Vorige week:

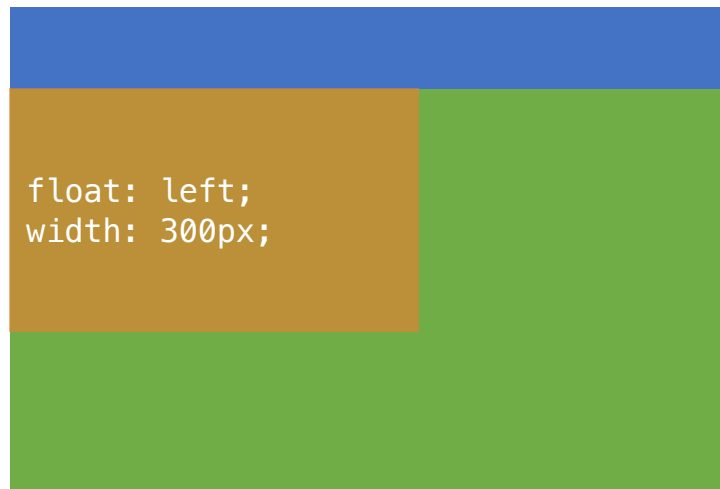


met drie block-elementen na het oranje blok

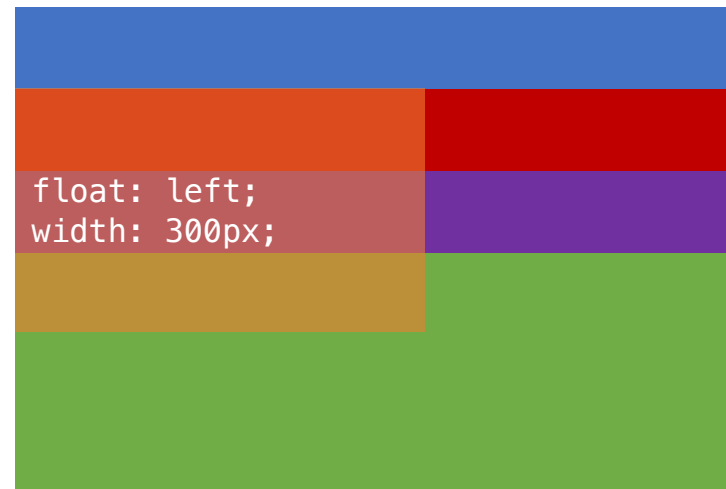


# Scherper uitgelegd:

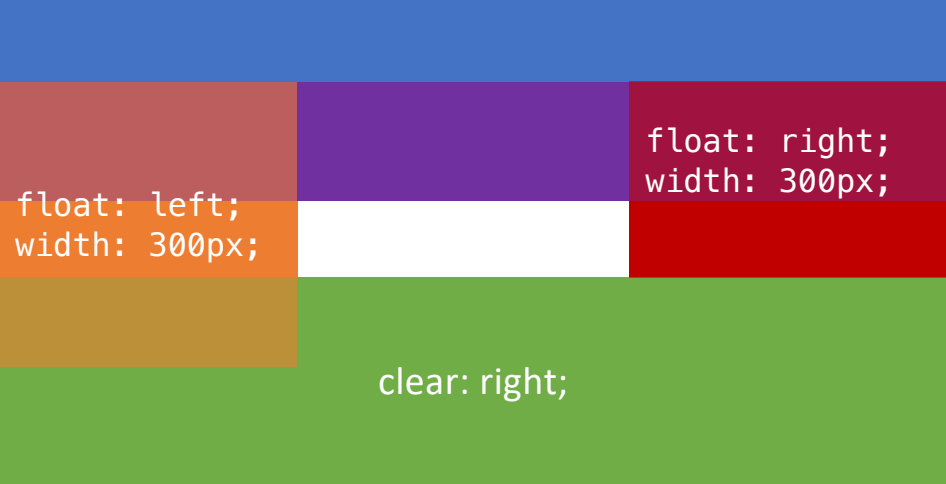
- Een float valt buiten de 'normale' top – down opbouw. Die gaat gewoon verder met niet-'floatende' block elementen:



met drie block-elementen na het oranje blok



# Evenzo:



# Centeren

- Je kunt twee dingen centreren:
  - content in een box
  - een block element in zijn parent box



← `width: 500px;`  
`margin: 0 auto;`

← `text-align: center;`

# Meer info

- <https://https://internetingishard.com/>

# College 5 eye catchers met CSS

# Meer met CSS

Voorbeelden:

1. Background
2. Meer met border
3. Transition (met :hover)
4. Animation
5. Slideshow (met animation en opacity)

De gebruikte code-voorbeelden staan op

<https://github.com/emmauscollege/4HV-webdesign-voorbeelden>

Meer informatie over de gebruikte properties staat op

<https://www.w3schools.com/cssref/>



# Tot slot

De getoonde voorbeelden

- zijn **geen** toetsstof
- helpen je **wel** een mooie praktische opdracht te maken
  
- **niet de bedoeling:**  
kopiëren (leer je niks van)
- **wel de bedoeling:**  
zorgen dat je ze begrijpt en met die kennis zelf iets maken  
(je leert unieke websites maken voor de hobby of voor veel monnie)