

Op welke manier bepaalt de HTML-code toch gedeeltelijk de layout?

Dit gebeurt door de volgorde, hiërarchie en de manier waarop elementen worden genest binnen de code.

Opdracht 1

Inline:

01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20

Block:

01
02
03
04
05
06
07
08
09

Inline-block:

01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18
19 20

Voor welke waarden worden de opgegeven width en height genegeerd?

Inline-block en block

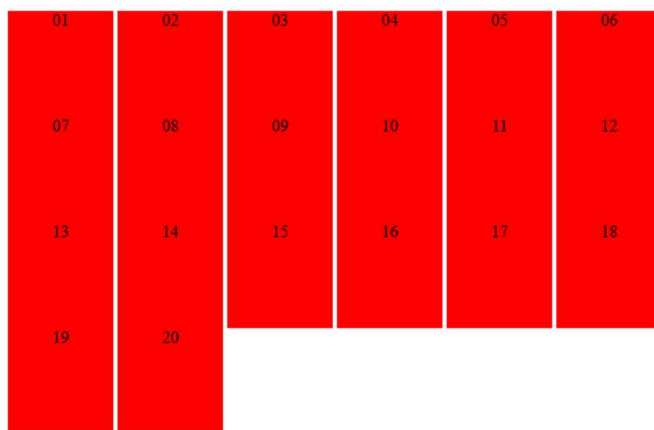
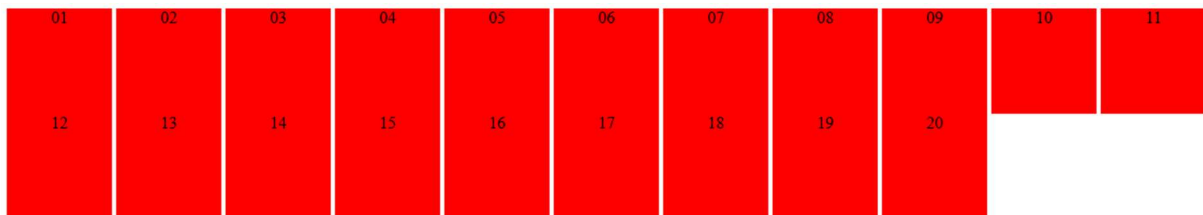
Voor welke waarden worden de elementen als woorden op een pagina gezet?

Inline en inline-block

Opdracht 2

Stel `display:inline-block` voor de elementen met class `area`, en bekijk opnieuw 'voorbeeld1.html'.
Vanwaar komen de lege ruimtes tussen blokken die naast elkaar staan?

Dit is standaard bij inline of inline-block, je kan het zien als een soort van spaties tussen de woorden/elementen. HTML beschouwt witruimte tussen inline of inline-block elementen als renderbare ruimte.



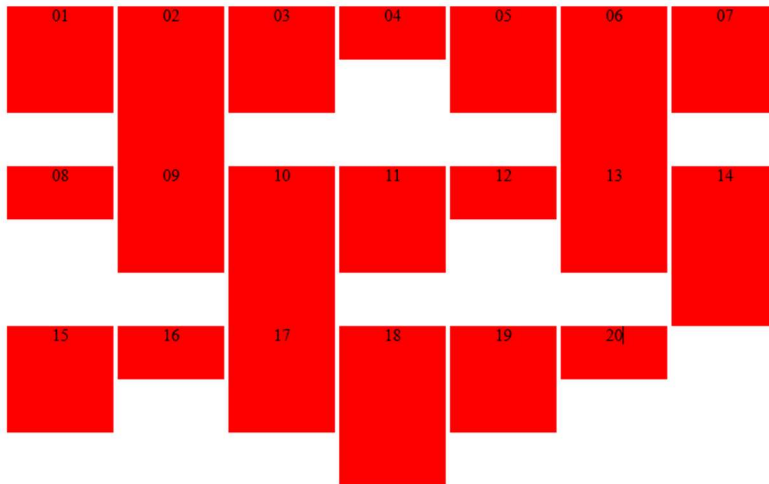
Opdracht 3

Waarom staat de tekst in de blokken horizontaal gecentreerd?

```
Door de regel: div {  
  text-align:center;  
}
```

In de CSS en omdat de tekst een div is.

Ga na dat een property `vertical-align:center` geen effect heeft. Waarom eigenlijk niet?



```
div {  
  text-align:center;  
  vertical-align: center;  
}
```

De reden waarom `vertical-align: center` niet werkt op een `<div>`-element, komt doordat de eigenschap `vertical-align` alleen van toepassing is op **inline** of **inline-block** elementen en niet op **block-level** elementen zoals een `<div>`.

Ga na dat `vertical-align:middle` er niet voor zorgt dat de tekst verticaal gecentreerd wordt. Wat doet `vertical-align` dan wel?

```
div {  
  text-align:center;  
  vertical-align:middle  
}
```



De `div`'s worden verticaal gecentreerd.

Zoek op google of er een simpele manier is om tekst verticaal te centreren in een element.

- Wat is de 'simpele' manier om een één enkele regel tekst verticaal te centreren? (tip: line-height)

Als je de line-height even groot zet als de hoogte van de container rond de tekst.

- Welke truc wordt vaak gebruikt om meerdere regels verticaal te centreren? (tip: table cell)

Voor meerdere regels tekst wordt vaak gebruikgemaakt van de CSS-eigenschappen `display: table` en `display: table-cell` met `vertical-align: middle`. Dit creëert een structuur die lijkt op een tabelcel waar de inhoud verticaal gecentreerd is.

In 'styles.css' staan twee verschillende CSS regels die de height instellen van een element met `class="area tall"`. Waarom krijgt de regel voor class 'tall' voorrang?

Als er twee CSS-regels zijn voor `.area.tall` en `.tall`, krijgt de regel `.tall` voorrang omdat deze meer specifiek is. De selector `.tall` heeft namelijk een directere match met de HTML-class dan een gecombineerde selector zoals `.area.tall`, afhankelijk van hoe de stijlen zijn toegepast. Als de specificiteit gelijk is, is de volgorde in de CSS belangrijk, waarbij de laatst gedefinieerde regel prioriteit krijgt.

Opdracht 4

01	02	03	04	05	een twee drie vier vijf zes zeven acht negen tien elf twaalf dertien veertien vijftien zestien zeventien achttien negentien twintig										06	07	08	09	10
11	een twee drie vier vijf zes zeven acht negen tien elf twaalf dertien veertien vijftien zestien zeventien achttien negentien twintig										12	13	14	15	16	een twee drie vier vijf zes zeven acht negen tien elf twaalf dertien			
veertien vijftien zestien zeventien achttien negentien twintig					17	18	19	20											

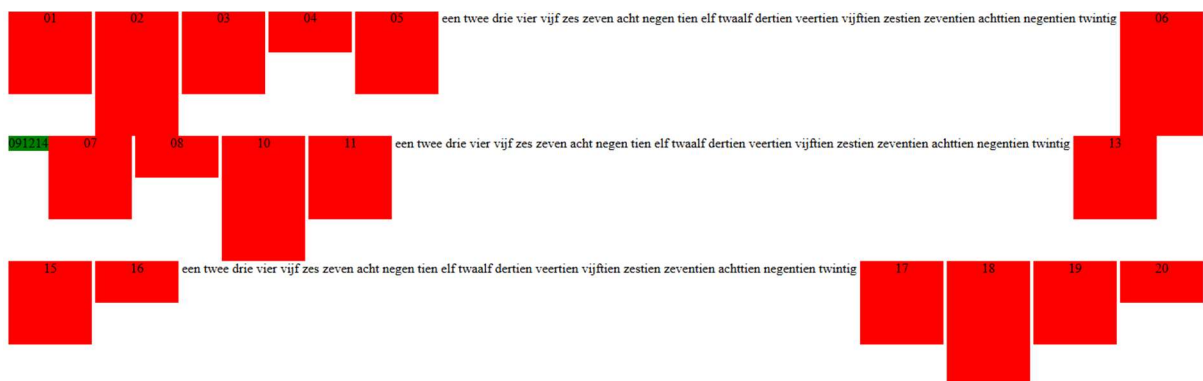
01	02	03	04	05	een twee drie vier vijf zes zeven acht negen tien elf twaalf dertien veertien vijftien zestien zeventien achttien negentien twintig										06	
07	08	09	10	11	een twee drie vier vijf zes zeven acht negen tien elf twaalf dertien veertien vijftien zestien zeventien achttien negentien twintig										12	
13	14	15	16	een twee drie vier vijf zes zeven acht negen tien elf twaalf dertien veertien vijftien zestien zeventien achttien negentien twintig										17	18	
19	20															

01	02	03	04	05	een twee drie vier vijf zes zeven acht negen tien elf twaalf dertien veertien vijftien zestien									
zeventien achttien negentien twintig					06	07	08	09	10	11	een twee drie vier vijf zes zeven acht negen			
tien elf twaalf dertien veertien vijftien zestien zeventien achttien negentien twintig						12	13	14	15	16	een twee drie			
vier vijf zes zeven acht negen tien elf twaalf dertien veertien vijftien zestien zeventien achttien negentien twintig								17	18	19	20			

Geef elementen 09, 12 en 14 de 'floaty' class.

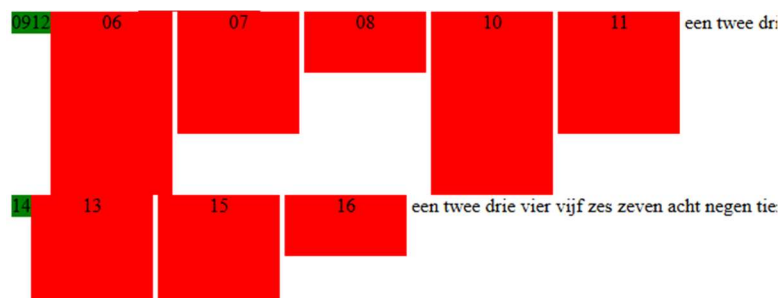
```
<html>
<head>
  <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
  <div class="area">01</div>
  <div class="area tall">02</div>
  <div class="area">03</div>
  <div class="area short">04</div>
  <div class="area">05</div>
  een twee drie vier vijf zes zeven acht negen tien elf twaalf dertien veertien vijftien zestien zeventien achttien negentien twintig
  <div class="area tall">06</div>
  <div class="area">07</div>
  <div class="area short">08</div>
  <div class="floaty">09</div>
  <div class="area tall">10</div>
  <div class="area">11</div>
  een twee drie vier vijf zes zeven acht negen tien elf twaalf dertien veertien vijftien zestien zeventien achttien negentien twintig
  <div class="floaty">12</div>
  <div class="area">13</div>
  <div class="floaty">14</div>
  <div class="area">15</div>
  <div class="area short">16</div>
  een twee drie vier vijf zes zeven acht negen tien elf twaalf dertien veertien vijftien zestien zeventien achttien negentien twintig
  <div class="area">17</div>
  <div class="area tall">18</div>
  <div class="area">19</div>
  <div class="area short">20</div>
</body>
</html>
```

- Ga na dat de floaty elementen naar links verplaatst werden 'in hun regel'.
- Ga na dat er geen grote ruimte achterblijft tussen elementen 08 en 10 waar element 09 stond.



- Ga na dat 12 en 14 niet altijd netjes naast elkaar staan, het hangt er maar vanaf of ze in dezelfde regel staan.

- Ga na dat er geen kleine ruimte is tussen de groene floaty elementen en het rode element rechts ernaast, maar wel tussen de rode elementen onderling.



Geef element 17 de 'break' class.

```
een twee drie vier vijf zes zeven  
<div class="break">17</div>  
acht negen tien elf twaalf dertien veertien vijftien zestien zeventien achttien negentien twintig
```

- Ga na dat dit helemaal geen invloed heeft qua positionering, er wordt niks 'gecleared'.



Opdracht 5

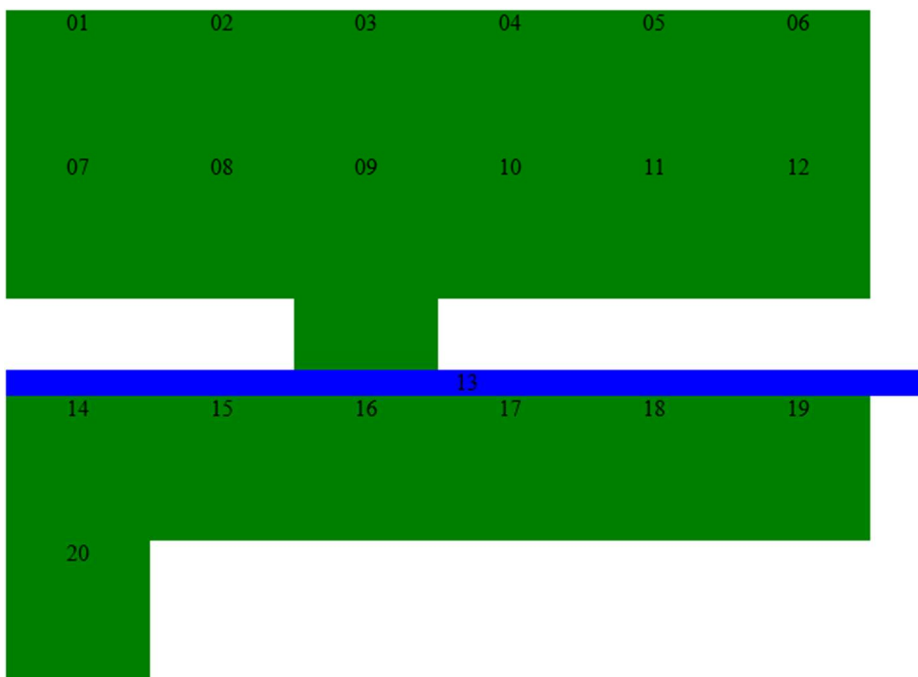
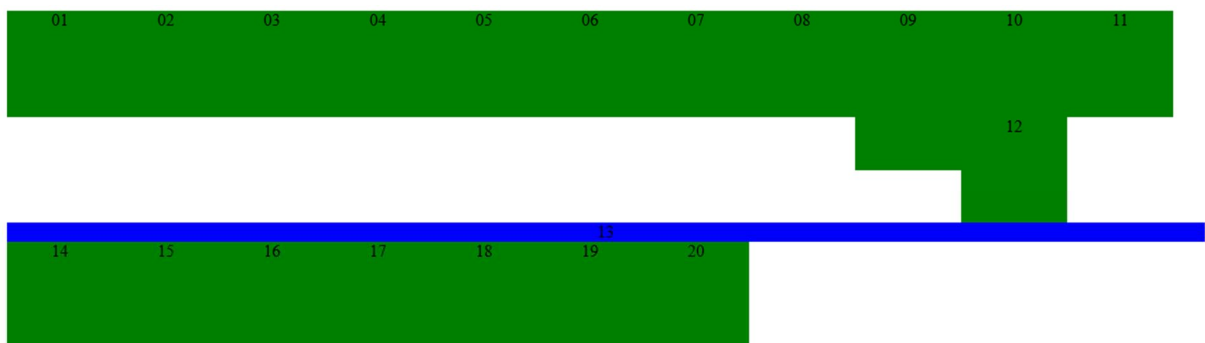
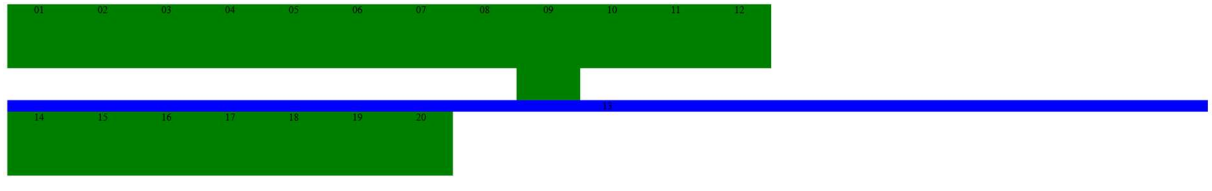
Open 'voorbeeld4.html' in de browser. Resize het browser venster (horizontaal). Ga na dat er naast element 09, 2 'regels' van blokken geplaatst worden als het venster breed genoeg is.



Geef element 13 de 'break' class.

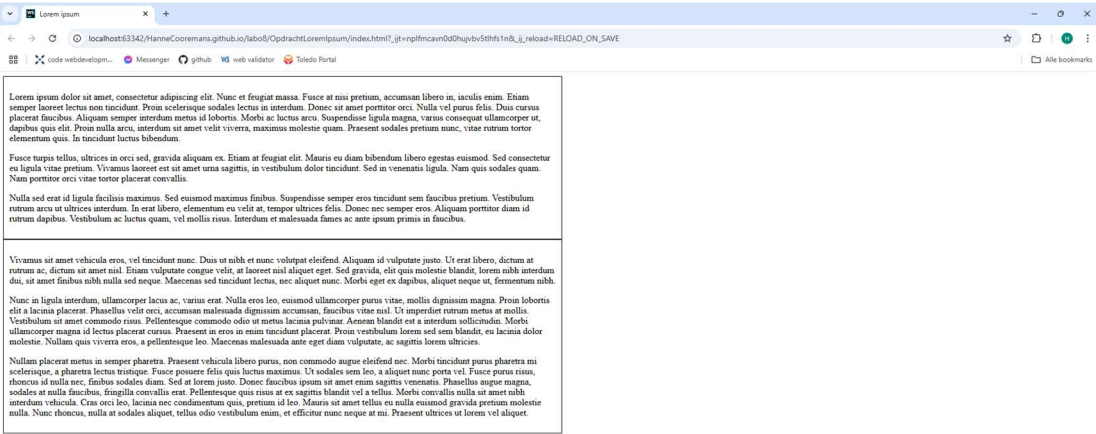
```
<div class="break">13</div>
```

Resize je browser opnieuw. Ga na dat element 13 altijd op een nieuwe 'regel' begint en wel lager dan element 09.

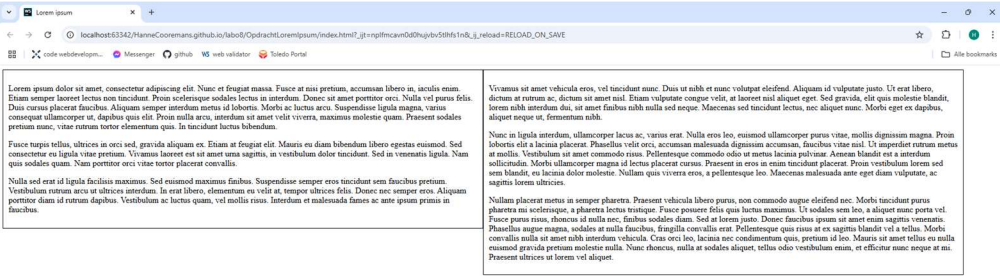


Opdracht Lorem ipsum

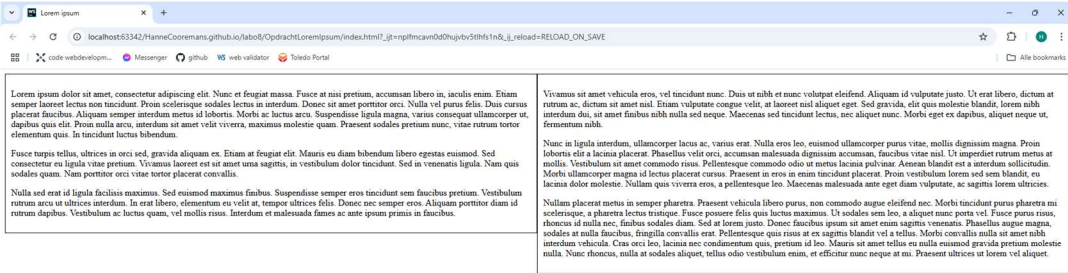
```
section{
  width: 50%;
  float:left;
  border: 1px solid black;
  padding:10px;
}
```



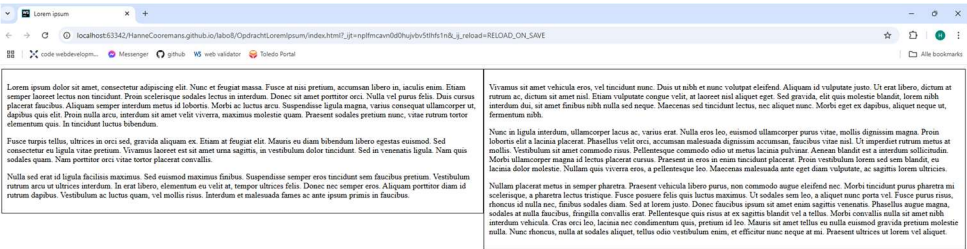
```
section{
  width: 47%;
  float:left;
  border: 1px solid black;
  padding:10px;
}
```

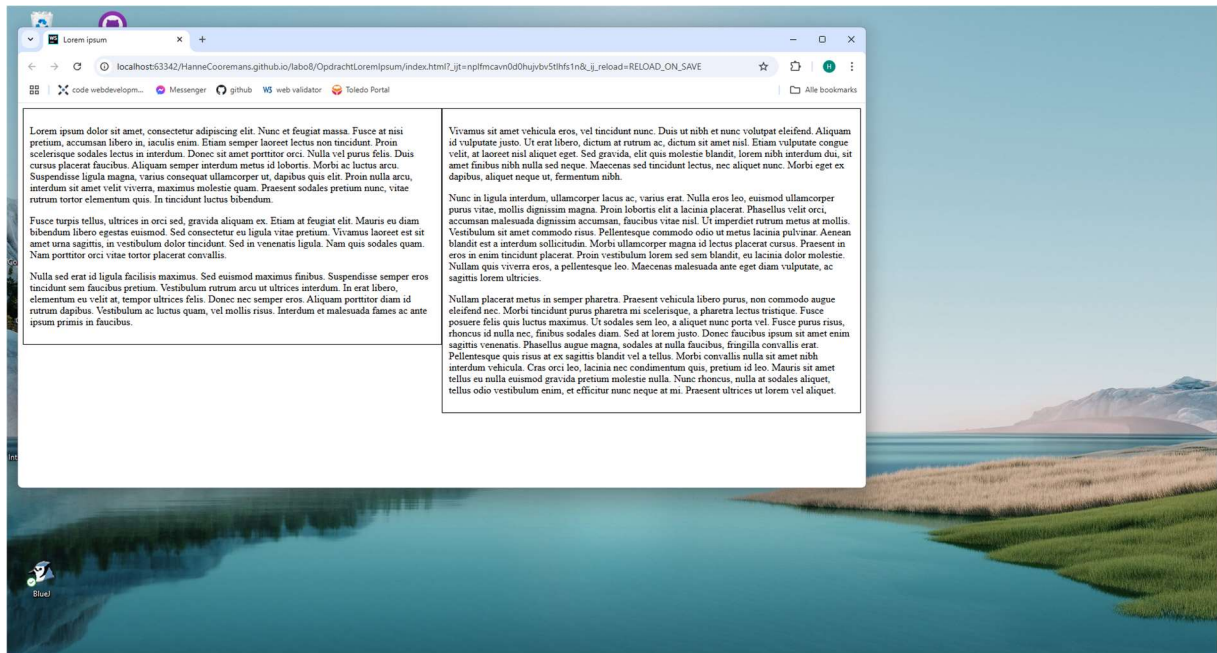


```
section{
  width: 930px;
  float:left;
  border: 1px solid black;
  padding:10px;
}
```



```
section{
  ! box-sizing: border-box;
  width: 50%;
  float:left;
  border: 1px solid black;
  padding:10px;
}
```





Opdracht Lorem ipsum extra space

```
section{
  width: 50%;
  float:left;
}
p{
  margin:10px;
  border: 1px solid black;
  padding:10px;
}
```