

# CSS DEEL 4

## Opdracht 1

### **:block**



Wanneer ik de display property verander naar :block, komen de rode blokjes onder elkaar terecht, telkens op een nieuwe regel dus.

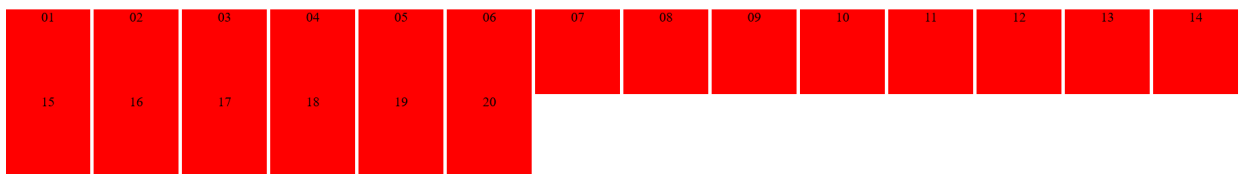
De .tall en .short worden niet genegeerd.

### **:inline**



Wanneer ik het wijzig naar inline verandert de lay-out naar kleine rode blokjes op een rij. Hier worden de .tall en .short genegeerd en houdt het geen rekening met de width en height.

### **:inline-block**

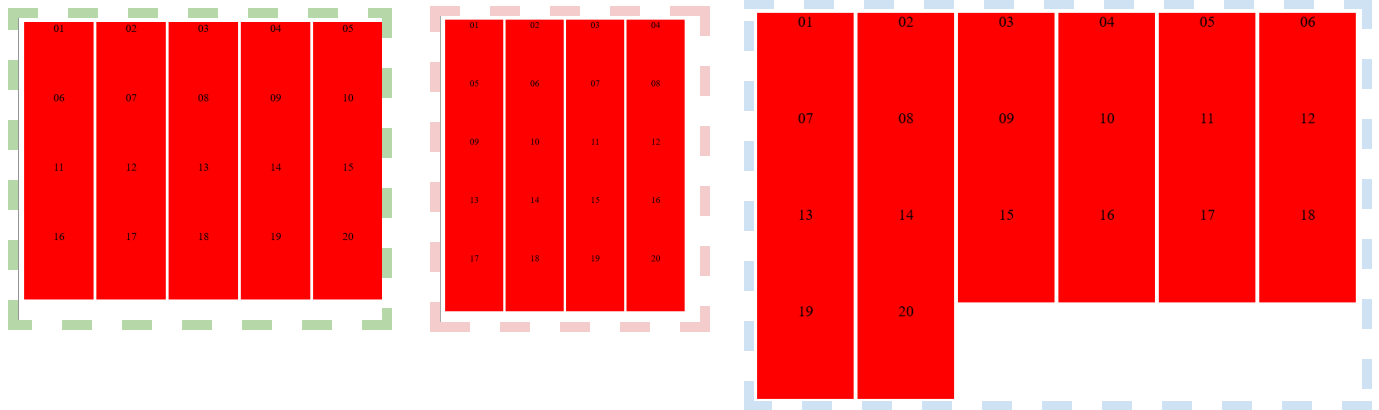


Inline-block zorgt ervoor dat de blokjes groter worden en recht onder elkaar komen als de pagina marge is opgebruikt.

Hier worden de width en height niet genegeerd.

## Opdracht 2

Re-size voorbeelden:



Deze witruimtes worden veroorzaakt door de witruimte in je HTML-code (zoals spaties, tabs, en nieuwe regels).

## Opdracht 3

**Waarom staat de tekst in de blokken horizontaal gecentreerd?**

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

Door de

**De eigenschap vertical-align werkt alleen op:**

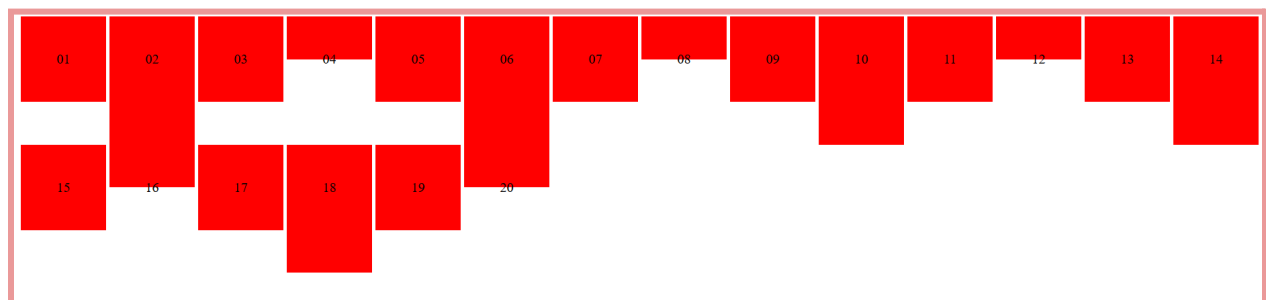
- Inline-elementen.
- Elementen die inline-block zijn ingesteld (zoals de .area-blokken), maar alleen in relatie tot andere inline en inline-block elementen in dezelfde regel.
- 

**Waarom zorgt vertical-align: middle; niet voor verticale centrering van tekst?**

De eigenschap vertical-align: middle; beïnvloedt de uitlijning van een inline-element, dus niet de tekst.

**Hoe kun je tekst verticaal centreren?**

Met line-height



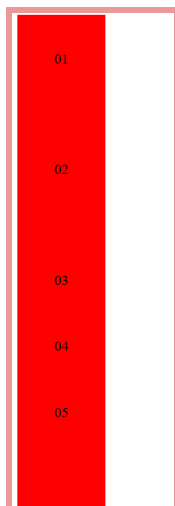
```
.area {
  display: inline-block;
  width: 100px;
  height: 100px;
  background: red;
  text-align: center;
  line-height: 100px;
}
```

Line-height code

**Voor meerdere regels tekst?**

**Met: display: table-cell**

Hierbij werkt vertical-align ook.



```
.area {
  display: table-cell;
  width: 100px;
  height: 100px;
  background: red;
  text-align: center;
  vertical-align: middle;
}
```

Table-cell + vertical-align

**Waarom krijgt de CSS-regel voor .tall voorrang?**

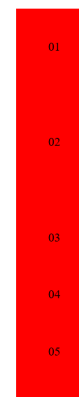
1. Specificiteit: Als één van de regels een selector heeft met een hogere specificiteit (zoals .area.tall in plaats van .tall), krijgt deze voorrang.
2. Als beide selectors dezelfde specificiteit hebben, krijgt de regel die later in het stylesheet staat voorrang.

## VERMOEDENS TESTEN:

```
✓ .area.tall {  
    height: 200px;  
}  
✓ .tall {  
    height:150px;  
}
```



```
13  
14  
15 ✓ .tall {  
16     height:150px;  
17 }  
18
```



Geen verandering.

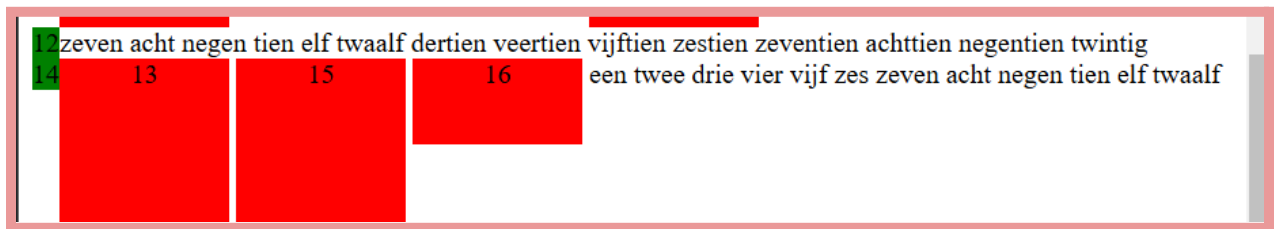
## Opdracht 4

**.floaty { float: left; background: green; }**

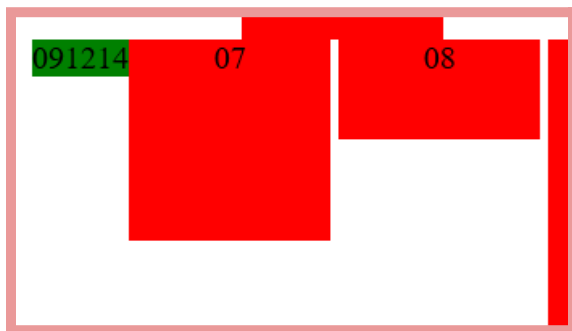
👍 Ga na dat de floaty elementen naar links verplaatst werden 'in hun regel'.

👍 Ga na dat er geen grote ruimte achterblijft tussen elementen 08 en 10 waar element 09 stond.





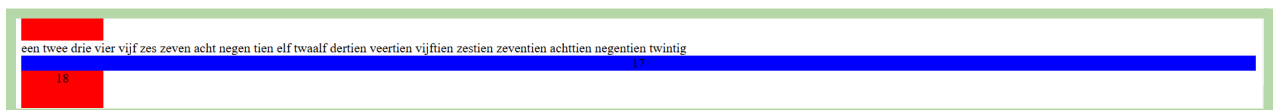
👍 Ga na dat 12 en 14 niet altijd netjes naast elkaar staan, het hangt er maar vanaf of ze in dezelfde regel staan.



👍 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**

```
<div class="break">17</div>
```



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

## Opdracht 5

