**Verslag CSS deel 4**

**Opdracht 1**

Bij display: block worden de opgegeven width en height genegeerd.

A picture containing calendar

Description automatically generatedText

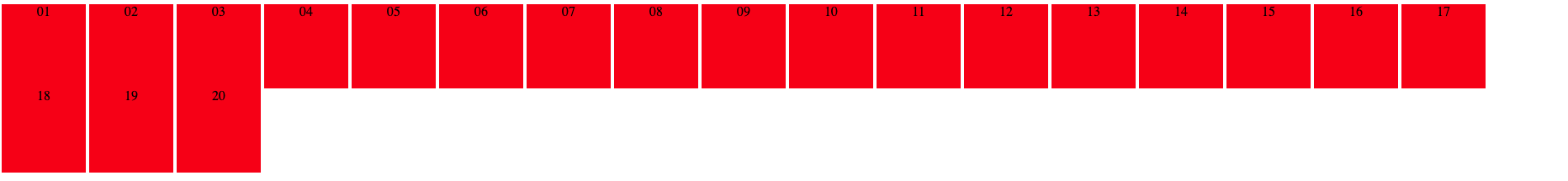
Description automatically generated

Bij display: inline worden alles naast elkaar geplaatst, dus worden ook.

Text

Description automatically generated

Bij display:inline-block worden elementen zoals op de pagina gezet.



**Opdracht 2**

Dit heeft niks te doen met margin, border, of padding, maar heeft te doen met html document. Dit zijn zoals de spaties tussen de worden.

**Opdracht 3**

* Waarom staat de tekst in de blokken horizontaal gecentreerd?

Door de class .area

display:inline-block;

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

Vertical-align heeft enkel effect op inline elementen positioneren van zijn parent element.

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

Chart, waterfall chart

Description automatically generated

Vertical-align:middle – positioneert alle elementen verticaal in het midden in vergelijking met de grootste element op dit rij (line).

* Wat is de ‘simpele’ manier om een één enkele regel tekst verticaal te centreren?

line-height: dit is goed om enkele regels verticaal te centreren.

* Welke truc wordt vaak gebruikt om meerdere regels verticaal te centreren?

In table cells kunnen wij centreren met: display: table; vertical-align: middle;

Anderskan gebruiken flex-box property: display:flex; align-items: center; justify-content: center;

* 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?

Wij hebben algemene classe met area die specifeerd voor alle regels die hebben dit word area. Maar als wij gebruiken de tweede word in de class area, kan ik zien, dat dit neemt de belangrijker rol, dus dit betekent zo’n ‘tagging’ heeft meer punten en is sterker. MAAR! Ik heb juist getest, als ik die regel met tall zet in voor met regel area, dan dit werkt niet. Dus mijn conclusie is de laatste regel wint.

**Opdracht 4**

* Ga na dat de floaty elementen naar links verplaatst werden 'in hun regel'.
* .floaty{  
   float:left;  
   background:green;  
  }

**Chart, waterfall chart

Description automatically generated**

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

Ik moest eigenlijk niks doen, de grote ruimte is vanzelf weg.

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

.separate {  
 margin-right: 3px;  
}

* 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.

.separate {  
 margin-right: 3px;  
}

**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.

Timeline

Description automatically generated with medium confidence

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

Graphical user interface

Description automatically generated

**2.2 Opdracht Lorem ipsum**

* Je eerste idee is wellicht om elke section een width van 50% te geven. Probeer dit uit. Je zult merken dat de kolommen niet naast elkaar, maar onder elkaar staan. Waarom is dit zo?

Omdat wij een border en padding hebben van 1px en 10px, dus wij hebben al meer van 50% van hele ruimte gebruikt met 1 section en met 2 sections boven de 100%.