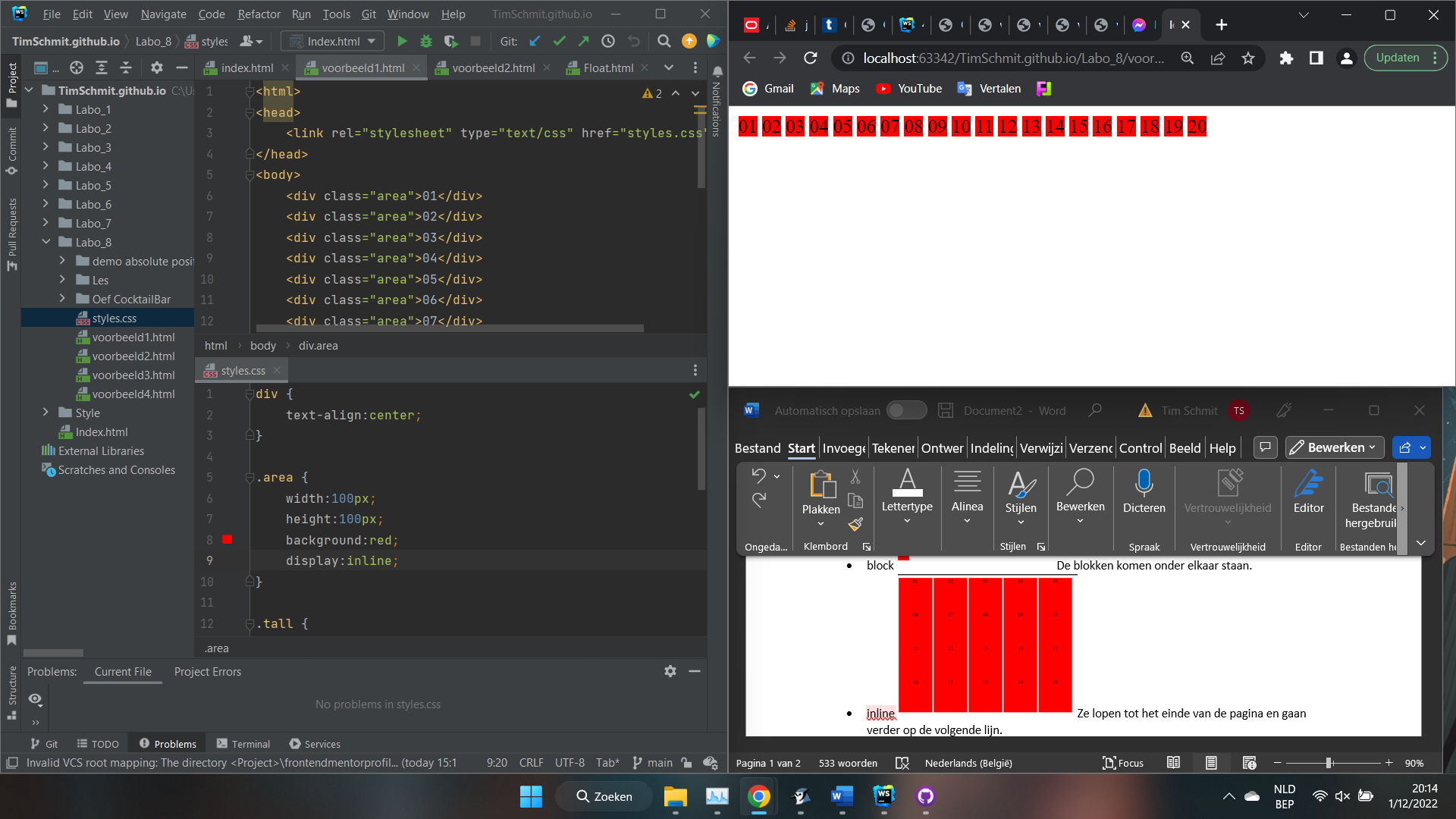
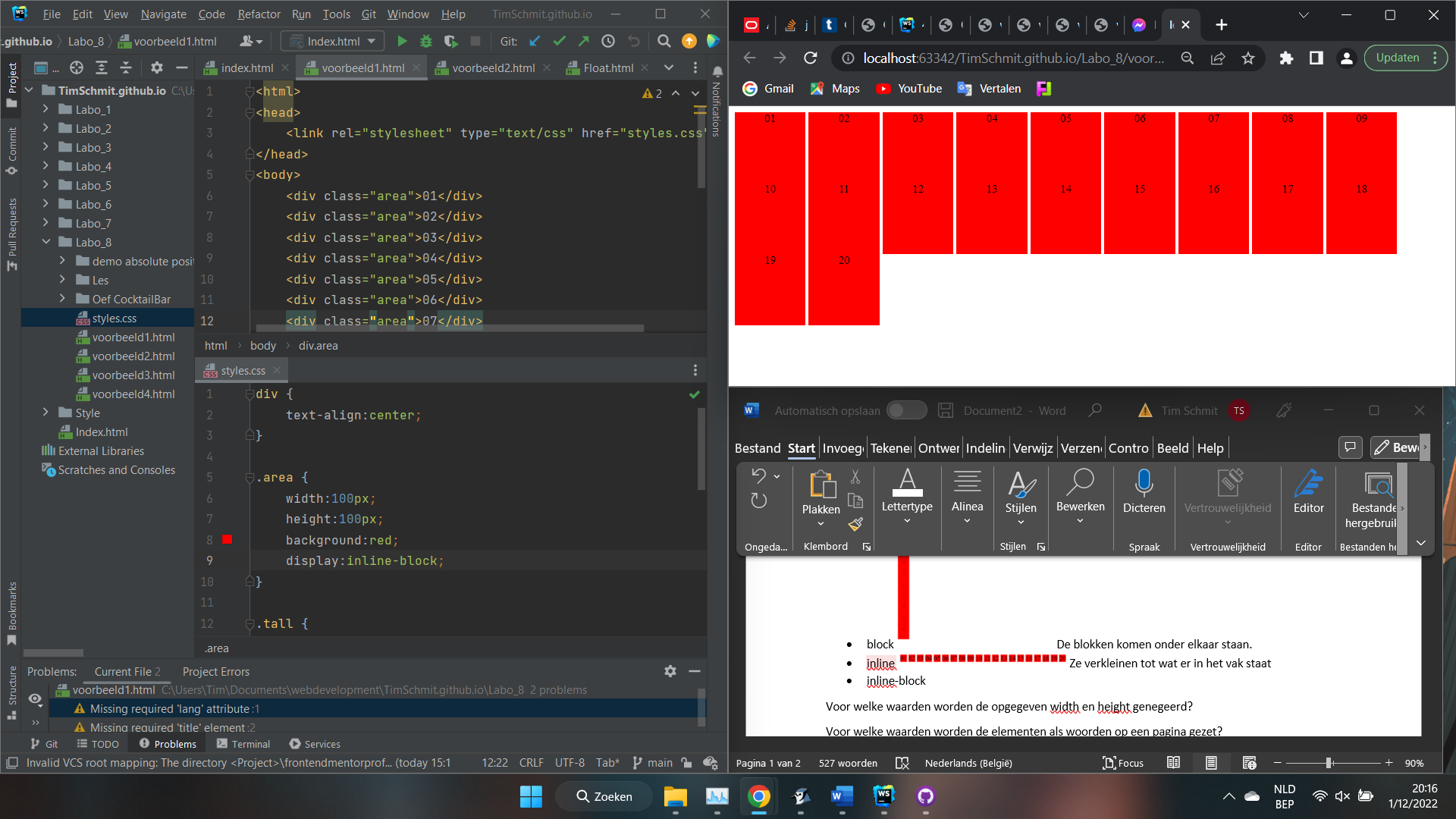
**Opdracht 1**

Unzip 'CSS deel 3 voorbeelden positionering.zip' en open 'voorbeeld1.html' in je browser. Open de HTML en CSS bestanden in je editor.

Ga na wat het effect is van het veranderen van de display property voor .area naar achtereenvolgens:

* block Afbeelding met tekst, schermafbeelding, monitor, scherm

  Automatisch gegenereerde beschrijvingDe blokken komen onder elkaar staan.
* inline  Ze verkleinen tot wat er in het vak staat
* inline-block versprijdt zich tot aan het einde van de pagina en verspringt dan naar een volgende lijn

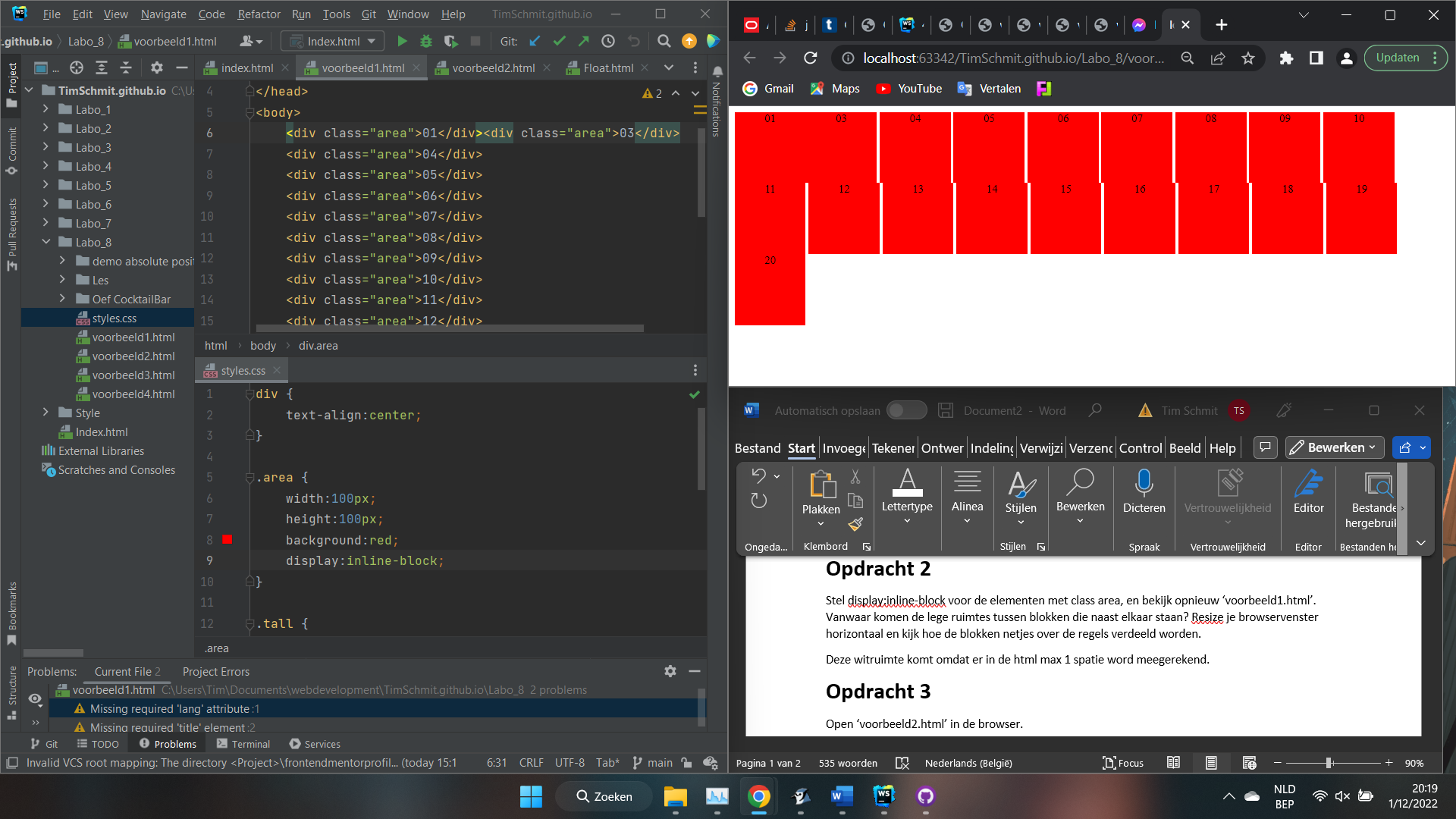
Voor welke waarden worden de opgegeven width en height genegeerd?

Grootte van de vierkanten

**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? Resize je browservenster horizontaal en kijk hoe de blokken netjes over de regels verdeeld worden.

Deze witruimte komt omdat er in de html max 1 spatie word meegerekend.



**Opdracht 3**

Open ‘voorbeeld2.html’ in de browser.

Waarom staat de tekst in de blokken horizontaal gecentreerd?

div {  
 text-align:center;  
}

Ga na dat een property vertical-align:center geen effect heeft. Waarom eigenlijk niet?   
Omdat vertical-align:center niets met de tekst doet

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

Alignt de blokken zodat ze verticaal in het midden staan

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

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?

**Opdracht 4**

Open 'voorbeeld3.html' in de browser. Bekijk nogmaals dat alle elementen als in een tekstverwerker gezet worden als je het browservenster vergroot en verkleint.

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

* Ga na dat de floaty elementen naar links verplaatst werden 'in hun regel'.
* Afbeelding met tekst, elektronica, schermafbeelding, scherm

  Automatisch gegenereerde beschrijving
* 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.

* Ga na dat dit helemaal geen invloed heeft qua positionering, er wordt niks 'gecleared'.
* Afbeelding met tekst, elektronica, schermafbeelding, computer

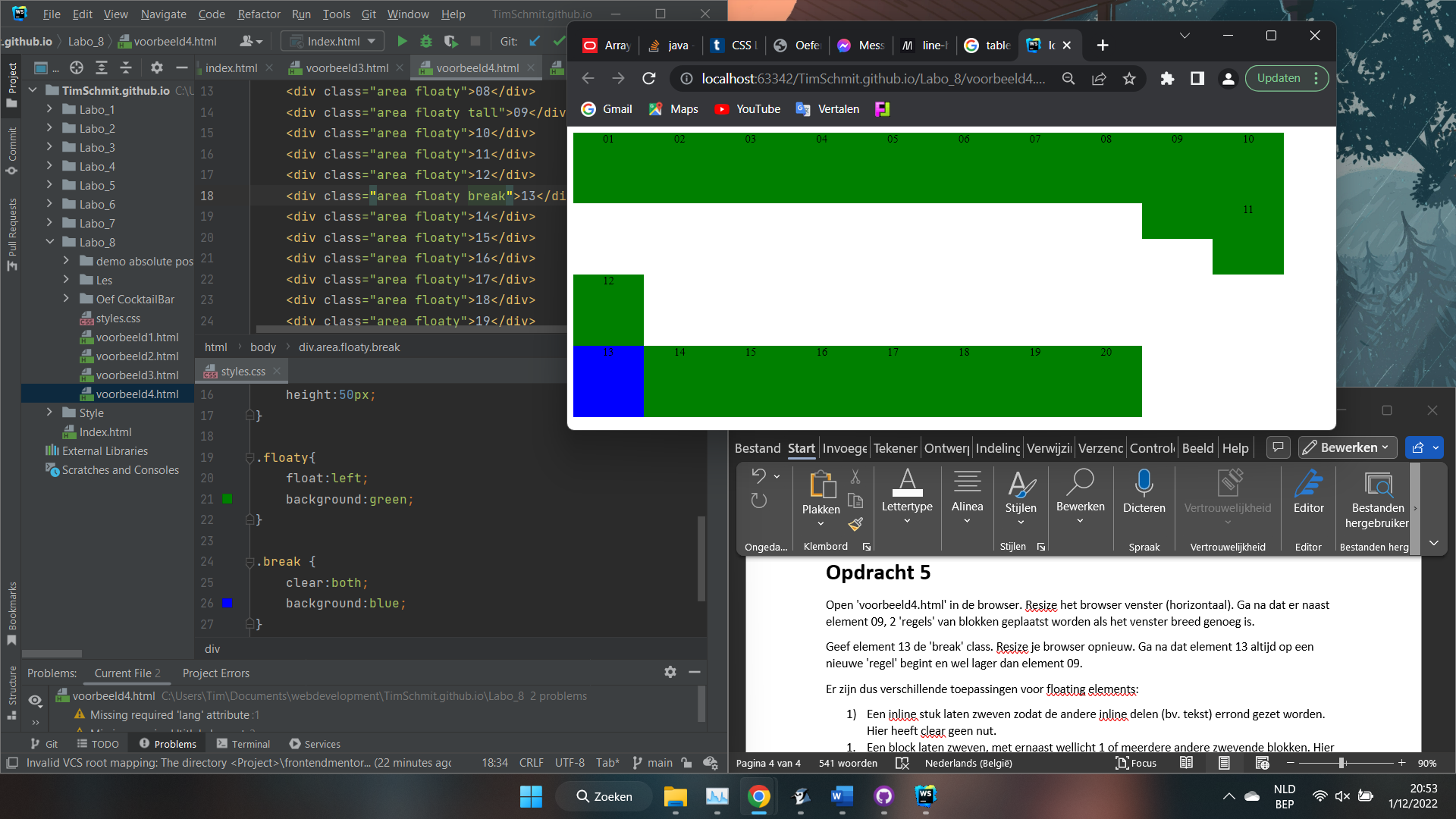
  Automatisch gegenereerde beschrijving

Het clearen van floating elementen is alleen zinvol bij display:block elementen, niet bij inline(-block) elementen!

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



Er zijn dus verschillende toepassingen voor floating elements:

1. Een inline stuk laten zweven zodat de andere inline delen (bv. tekst) errond gezet worden. Hier heeft clear geen nut.
2. Een block laten zweven, met ernaast wellicht 1 of meerdere andere zwevende blokken. Hier kan clear nuttig zijn voor blokken die erna komen.

**Opdracht Lorem ipsum**

Lorem ipsum is een fake Latijnse tekst die vaak gebruikt wordt voor layout doeleinden. Op de volgende pagina kun je zoveel van deze tekst genereren als je nodig hebt alsook de achterliggende geschiedenis nalezen: <http://www.lipsum.com/>.

Maak een HTML5 pagina met 2 lorem ipsum sections van voldoende lengte. Maak een layout waarbij elke section een eigen kolom vormt (dus 2 kolommen naast elkaar). Elke kolom dient de helft van de breedte in beslag te nemen, met een border van 1px en een padding van 10px.

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? Probeer onderstaande twee pogingen uit en ga na dat ze niet het gewenste resultaat opleveren:

* De width op bv. 47% in stellen, maar dit werkt niet goed.
  + Extra lege ruimte rechts op grote schermen.
  + Kolommen onder elkaar op kleine schermen.
* De width op de helft van de beschikbare breedte in pixels in stellen, bv. 550px, maar dan is het niet exact de helft als het browservenster een ander formaat heeft.

In CSS3 is er een betere mogelijkheid: zoek op het internet naar de 'box-sizing' property en probeer hiermee een goede oplossing te bekomen waarbij de width van elke kolom netjes op 50% is ingesteld.

**Opdracht Lorem ipsum extra space**

We willen nu een lege ruimte van 10px toevoegen rond de kolommen:

Merk op dat er tussen de kolommen een dubbele lege ruimte voorkomt. We zouden dit kunnen wegwerken zodat de ruimte tussen de kolommen eveneens slechts 10px is maar voor de eenvoud laten we het zo.

Hint: als je een element met expliciete width en "box-sizing: border-box" ook nog een margin geeft, zal dat element breder uitvallen dan de opgegeven width! De box-sizing is immers t.o.v. de border en de margin ligt daar nog omheen. Bedenk echter dat de lege ruimte rondom een element niet noodzakelijk van het element zelf afkomstig moet zijn, maar net zo goed bij een ancestor of descendant element kan horen!

**Opdracht Cocktail bar**

Maak de 'Cocktail bar' opdracht.