# Opdracht: roll the dice…

## Doelstelling:

Het aanpassen van een HTML-pagina met daarin CSS en Javascript code zodat deze weer naar behoren werkt.

## Opdracht:

In deze opdracht begin je met een webpagina waar een dobbelsteen op wordt getoond. Deze dobbelsteen doet nog niets, het is de bedoeling dat je deze laat rollen door op de knop te drukken. Aan jou de taak om uit te zoeken hoe je dit voor elkaar krijgt.

## Benodigd Materiaal:

* Een PC / Mac met daarop geinstalleerd een code / tekst editor (notepad ++ / sublime / ultra-edit).
* Een webbrowser (Chrome / Firefox / Edge )
* Het internet (om dingen op te zoeken)

## Het stappenplan

De benodigde bestanden welke je nodig hebt om de opdracht te maken kun je terugvinden op de USB-stick. Als eerste is het verstandig om de inhoud van de USB-stick ergens op de harde schijf van je computer te kopieren. Je mag zelf weten waar naartoe maar zorg er wel voor dat je deze folder op een later moment terug kunt vinden.

Tip: maak een nieuwe folder aan op je computer en kopieer daar de inhoud van de USB-stick naar toe.  
  
Voordat je nu begint met de opdrachten is het eerst zaak de bestanden in de gekopieerde map te bekijken. Verzeker je ervan de de volgende bestanden op de harde schijf van je computer staan:

### │ dice.html

### │ RollTheDice\_Opdracht\_VMBO\_traject.docx

### └───[Images]

### 1.jpg

### 2.jpg

### 3.jpg

### 4.jpg

### 5.jpg

### 6.jpg

### round\_dice\_1.png

### round\_dice\_2.png

### round\_dice\_3.png

### round\_dice\_4.png

### round\_dice\_5.png

### round\_dice\_6.png

Het bestand [dice.html] is het bestand waar code in staat. De code die hier wordt gebruikt is HTML in combinatie met CSS (Cascading StyleSheets) en Javascript.

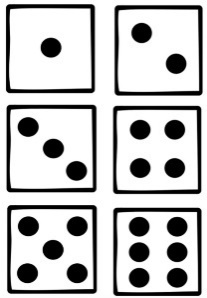
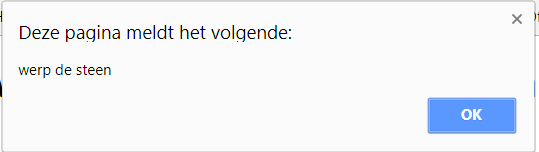
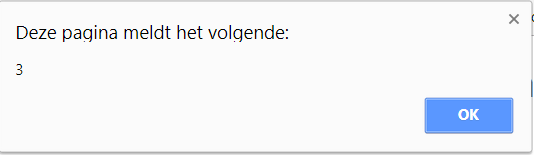
De werking van dit bestand kun je eenvoudig bekijken via een webbrowser. Afhankelijk van de standaard browser op jouw computer wordt door te dubbelklikken op het [dice.html] bestand deze weergegeven in Chrome, Firefox of Edge. Wordt een andere browser gestart dan dien je Windows te vertellen dat een van laatstgenoemde browsers als standaard moet worden gekozen.

De broncode van het bestand kun je ook inzien, open dit bestand met een texteditor zoals bijvoorbeeld Notepad ++ en bekijk de diverse onderdelen.  
  
Als je goed oplet zie je tussen de diverse code onderdelen drie secties:

1. CSS-code tussen de zg. <style> ... </style> tags[[1]](#footnote-1)
2. HTML-code tussen de <body> … </body> tags
3. Javascript tussen de <script> … </script> tags

De HTML-code in de body wordt gebruikt om de *structuur* van je pagina te bepalen, de CSS-code wordt gebruikt om je pagina een mooie styling te geven terwijl de Javascript onderdelen zorgen voor lociga zoals het stoppen van de dobbelsteen en de keuze van het nummer.

### Level 1 - opdrachten:

1. Het eerste onderdeel van deze opdracht is de dobbelsteen zelf. Zoals je nu ziet zijn de zijden van de dobbelsteen voorzien van groene ronde afbeeldingen. Dit is niet de bedoeling. Maak van deze afbeeldingen normale dobbelsteen afbeeldingen.   
     
     
     
     
   Zoek in de code de onderdelen op waar de afbeeldingen van de dobbelsteenzijden worden gezet.   
     
   Tip: De dobbelsteen nummers zijn afbeeldingen (zoals jpg of png) kijk in de webfolder van het project in de Images folder. Het gaat hier om de styling dus moet je zoeken tussen de CSS-code
2. Onder de dobbelsteen zie je een grote knop staan, wanneer je nu op deze knop drukt gebeurd er niets.   
     
   Zoek in de HTML-code het onderdeel op waar de knop wordt neergezet. Dit onderdeel moet worden aangepast. Je moet iets toevoegen dat reageert op het klikken van je muis op de knop. Dit doe je met het *attribuut*[[2]](#footnote-2) **onclick**. Dit attribuut moet dus worden toegevoegd aan het element wat verantwoordelijk is voor de knop.   
     
   Voeg het attribuut toe binnen het juiste HTML element met daarin de volgende waarde:   
     
   onclick="alert('werp de steen');"  
     
   Let op dat je exact de bovenstaande code overneemt. Haakjes, hoofdletters, kleine letters, dubbele en enkele quotes. Het is allemaal belangrijk dat het klopt.  
     
   Voer de code uit en zie wat er gebeurd. Doe dit door je HTML-bestand op te slaan en deze opnieuw in je webbrowser te **herladen met F5.**  
     
   
3. Omdat een *attribuut* meestal *een waarde* krijgt moet het stukje Javascript, welke nu zorgt voor de *alertbox* met het bericht "werp de steen", worden vervangen. Om dit voor elkaar te krijgen moet je weten dat er in de code een functie is gemaakt die zorgt voor het stoppen van de steen op een bepaald nummer. Deze functie heet: StopDice(). Zoek in het <script> element de code voor deze functie op en vervang op de plek van de *alert* uit de vorige oefening, de verwijzing naar deze functie.   
     
   onclick="…"  
     
   Let ook goed op de ronde haakjes die hier moeten worden gebruikt, zet je geen getal tussen deze ronde haken dan stopt de dobbelsteen altijd op "1".  
     
   Tip: zoek op internet naar de HTML *onclick* functie en hoe hiermee een Javascript functie kan worden uitgevoerd.
4. Wanneer de vorige opdracht is gelukt zul je merken dat de dobbelsteen altijd op nummer "1" stopt. Dit is natuurlijk niet de bedoeling. Vul maar eens tussen de haakjes van StopDice() een ander nummer in. Als het goed is zal de dobbelsteen stoppen op dat nummer.   
     
   onclick="StopDice(2);" // dit zal de dobbelsteen op 2 tot stilstand brengen.  
     
   Omdat je bij een dobbelsteen natuurlijk niet van tevoren weet wat er gegooid gaat worden zul je een zg. random (willekeurig) nummer moeten kiezen. Dit kan met de functie getRandomNr() die ook tussen de <script> tags staat.   
     
   Vervang de StopDice() functie in het onclick attribuut maar eens voor alert(getRandomNr())en kijk wat er gebeurd. Als je alles goed hebt ingevuld zal via een alertbox steeds een ander nummer tussen de 1 en 6 worden getoond.   
     
     
   
5. In de vorige oefening heb je gezien dat het resultaat van de getRandomNr() functie naar de *alert box* wordt gebracht. Zorg er nu voor dat dit random resultaat naar de *StopDice()* functie wordt gebracht.   
     
   Vervang hier wederom de *alert* functie uit de vorige opdracht voor de *StopDice()* functie en plaats binnen de ronde haakjes de aanroep naar de *getRandomNr()* functie. Je brengt dus het resultaat uit de *getRandomNr()* functie in de *StopDice()* functie.

Het je alles goed gemaakt dan stopt nu je dobbelsteen op een random nummer na iedere druk op de knop.

### Level 2 - opdrachten:

Nu je de dobbelsteen hebt aangepast en kunt laten stoppen op een willekeurig getal is het de bedoeling dat we **twee** dobbelstenen gaan maken. Omdat we de onderdelen voor de eerste dobbelsteen al hebben kun je er betrekkelijk eenvoudig twee van maken:

1. Zoek in de HTML-code de elementen op welke de 6 zijden van de dobbelsteen bevatten. Tip: zoek naar het <div> element met als class: *cube*. Heb je deze gevonden kopieer dit deel inclusief de sub-elementen en zet dit eronder. Wanneer je dit goed hebt gedaan en je het resultaat in de webbrowser bekijkt zie je als het goed is nog steeds één dobbelsteen staan.   
     
   Waarom zie je geen twee dobbelstenen?   
     
   Het antwoord daarop is eenvoudig: *ze staan over elkaar heen*. Je zult ze dus moeten verplaatsen ten opzichte van elkaar om ze beide zichtbaar te maken. Dit ga je doen in de volgende opdracht.
2. In de vorige stap heb je de HTML van de dobbelsteen gekopieerd zodat er nu twee dobbelstenen in je HTML staan. Ze hebben beide de class *cube* welke ervoor zorgt dat de styling in CSS gelijk is. Je moet er dus voor zorgen dat iedere dobbelsteen van een aparte CSS style wordt voorzien. Om dit te realiseren zul je het div HTML-element een *id* moeten geven. Met een *id* in HTML kun je enkele elementen benoemen en stylen via bijvoorbeeld CSS. Maak voor iedere dobbelsteen <div> een *id* aan en geef deze als waarde respectievelijk cube1 en cube2. Hoe je een *id* aanmaakt voor een HTML-element zul je op moeten zoeken op internet. Gebruik als zoek woorden: *id attribute html*
3. Wanneer de dobbelstenen zijn voorzien van een eigen *id* kun je ze via CSS een style gaan geven. Deze styling moet bovenop de reeds bestaande styling, informatie over de positie gaan bevatten. Binnen de <style> tags bevindt zich zoals eerder gemeld, de CSS stylesheets. Om een *id* aan een CSS style te koppelen gebruik je de *hashtag* (= #) met daarachter de naam van de *id*. Dus in ons geval: #cube1 en #cube2. Zet achter iedere selector een accolade openen en sluiten { … }. Tussen deze accolades komt de styling te staan.  
     
    #cube1 { css styling voor cube1 }   
    #cube2 { css styling voor cube2 }   
     
   Let op: plaats deze style selectors direct **na** de class selector: .cube  
     
   Je kunt de <div> met dobbelstenen naar links of rechts verplaatsen met de *left* eigenschap. Gebruik de *left* om de eerste steen 100 pixels naar links te schuiven en de andere 100 pixels naar rechts. Weet je niet hoe dit moet? Zoek op internet de juiste schrijfwijze en zoek naar: *css left property***Tip:** Gebruik je de *left* eigenschap om HTML-elementen te verplaatsen dan ga je naar links met een negatief getal, naar rechts met een positief getal.  
     
   Als het goed is heb je nu beide dobbelstenen 200 pixels uit elkaar gezet. Test dit in je webbrowser.
4. Rol je nu de dobbelstenen dan merk je dat ze beide op hetzelfde nummer stoppen. Leuk als je veel dubbel wilt gooien, maar niet echt realistisch natuurlijk. We gaan iets verzinnen om iedere dobbelsteen te laten stoppen op zijn eigen nummer.

We gaan de functie *StopDice()* aanpassen. Deze functie is namelijk verantwoordelijk voor het toewijzen van de juiste animatie aan de hand van het random gekozen nummer welke de *getRandomNr()* functie aanmaakt.

Nu werkt deze functie als volgt:  
  
de variabele *s* krijgt alle informatie binnen over de stylesheets binnen deze HTML pagina (= document.Stylesheets) . Iedere style selector heeft een eigen index nummer. De .wrap selector is index: 0 de .cube selector index 1 en de door ons nieuw aangemaakte #cube1 en #cube2 hebben respectievelijk index 2 en 3.gekregen. Kijk maar eens goed in het gedeelte waarde CSS stylesheets staan in het HTML document. Je zult hier deze volgorde in herkennen.

Je kunt ook de Inspector van de webbrowser openen en het resultaat van de variable s naar de console printen. Hierin zitten tenslotte alle stylesheets. Doe dit met de console.log(s) functie. Plaats deze functie in je code op die plek waar de variabele 's' bekend is. Controleer vervolgens in de inspector de waarde van de Stylesheets eigenschap door HTMLdocument op te klappen.

Gebruik je Firefox of Chrome dan open je de inspector met Ctrl-Shift i.

Als alles klopt dan kun je in je code zoeken naar de regels waar nu index 1 wordt aangeroepen (cssRules[1].style.animation). Kopieer deze regel zodat je er twee regels van kunt maken voor de styles van #cube1 en #cube2. Dit zijn als het goed is index 2 en index 3.

Om nu iedere dobbelsteen een ander nummer te geven kun je de randomfunctie twee keer uit laten voeren. Dit is het eenvoudigst door simpelweg een tweede argument aan de StopDice() functie toe te voegen. Nu is dat er nog eentje. Maak hiervan bijvoorbeeld function StopDice(a,b) { … }. Pas ook in de functie de a en b toe op de plek waar nog de 'n' staat.  
  
In de onclick van de knop is het nu nog wel belangrijk om twee maal de random functie aan te roepen. Doe dit door onclick="StopDice(getRandomNr(), getRandomNr());"

Wanneer je alles goed hebt gedaan draaien er twee dobbelsteen en stoppen ze beide op verschillende getallen..

1. Een tag in een HTML-pagina is een item tussen de < > tekens. Dit worden ook wel angle brackets genoemd. [↑](#footnote-ref-1)
2. Een attribuut in HTML zijn de onderdelen binnen de HTML-tags waaraan meestal een waarde wordt gegeven. <input type="button">. Hier is *type* het attribuut met de waarde "*button*". [↑](#footnote-ref-2)