

HTML CSS JS extra

Les 13:

CSS shapes, CSS animaties



Onderwerpen les:

- CSS shapes
- CSS combineren shapes (karakters maken)
- CSS animation met keyframes
- CSS battle

Leerdoelen

- ✓ Aan het einde van de dag kun je verschillende operatoren gebruiken in SQL
- ✓ Aan het einde van de dag weet je wat joins zijn en kun je deze gebruiken in queries

CSS shapes

Simple rectangle and squares

Rechthoeken en vierkanten zijn makkelijk te maken met een width, height (px, %, rem, em) en background-color.



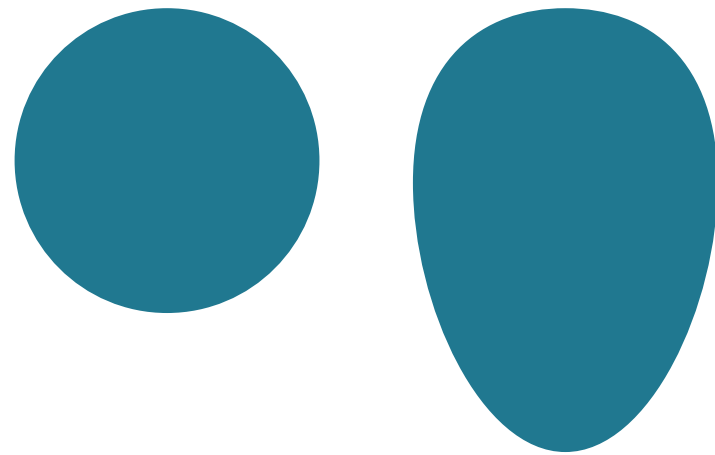
```
.rectangle{  
  width: 700px;  
  height: 200px;  
  background-color: #207890;  
}
```

```
.square{  
  width: 200px;  
  height: 200px;  
  background-color: #207890;  
}
```

CSS shapes

Elipses

Ovalen en ronde vormen kunnen we maken met de border radius.



De radius wordt toegepast op de hele achtergrond ook als het element geen border heeft.

Met de slash '/' kun je ook de radius opgeven waarmee de hoek wordt afgerond. (maken niet perfecte afronding)

```
.circle {  
  width: 200px;  
  height: 200px;  
  background-color: #207890;  
  border-radius: 9999em;  
}  
.egg {  
  ..  
  border-radius: 50% 50% 50% 50% / 60% 60% 40% 40%;  
}
```

CSS shapes

Triangles

Een triangle maak je met de borders. 1 border geef je een kleur 1 of 2 andere geef je een transparent achtergrond.

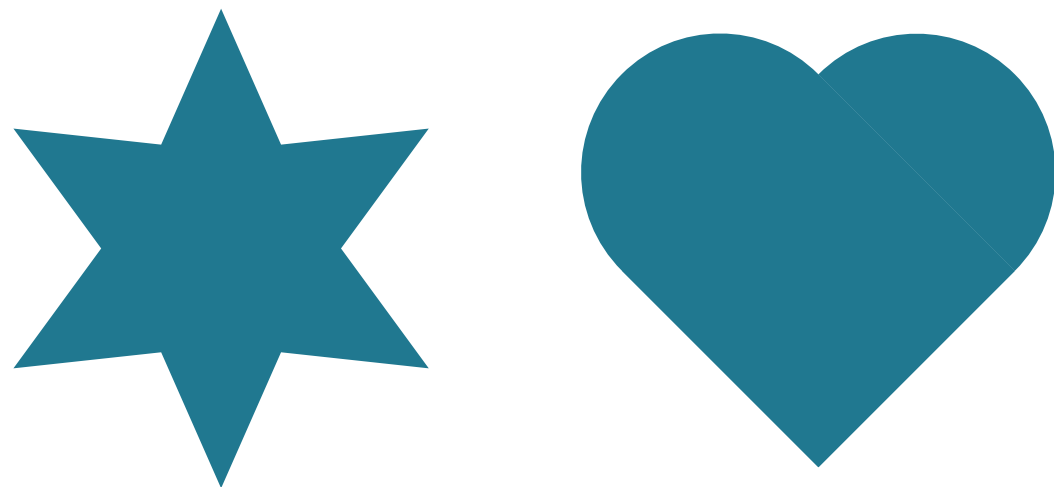


```
.triangle {  
  border-left: 50px solid transparent;  
  border-right: 50px solid transparent;  
  border-top: 100px solid red;  
}
```

CSS shapes

Combine shapes met before en of after

Met before of after (pseudo elements) kunnen we content toevoegen als bijvoorbeeld een extra vorm. Hiermee kun je onder andere een 6 hoekige ster maken (2 driehoeken over elkaar)

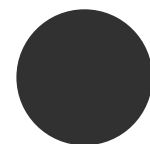
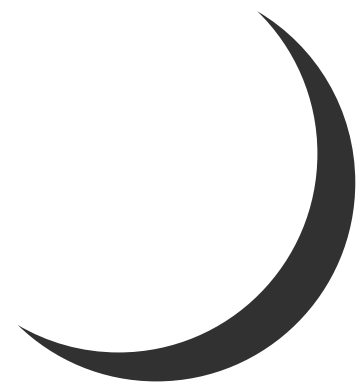


```
.star {  
  width: 0;  
  height: 0;  
  border-left: 50px solid transparent;  
  border-right: 50px solid transparent;  
  border-bottom: 100px solid red;  
  position: relative;  
}  
.star::after {  
  width: 0;  
  height: 0;  
  border-left: 50px solid transparent;  
  border-right: 50px solid transparent;  
  border-bottom: 100px solid red;  
  position: absolute;  
  content: "";  
  top: 30px;  
  bottom: -50px;  
}
```

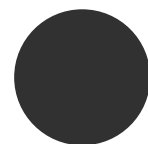
CSS shapes

Shapes maken met dropshadow

Het is ook mogelijk om een vorm te maken met een schaduw. Dit kan een zachte maar ook harde schaduw zijn. Bijvoorbeeld een 2e oog voor het dupliceren van de 1e. Of een transparante voorgrond die de schaduw bedekt zodat je onder andere een maan kan maken.



origineel



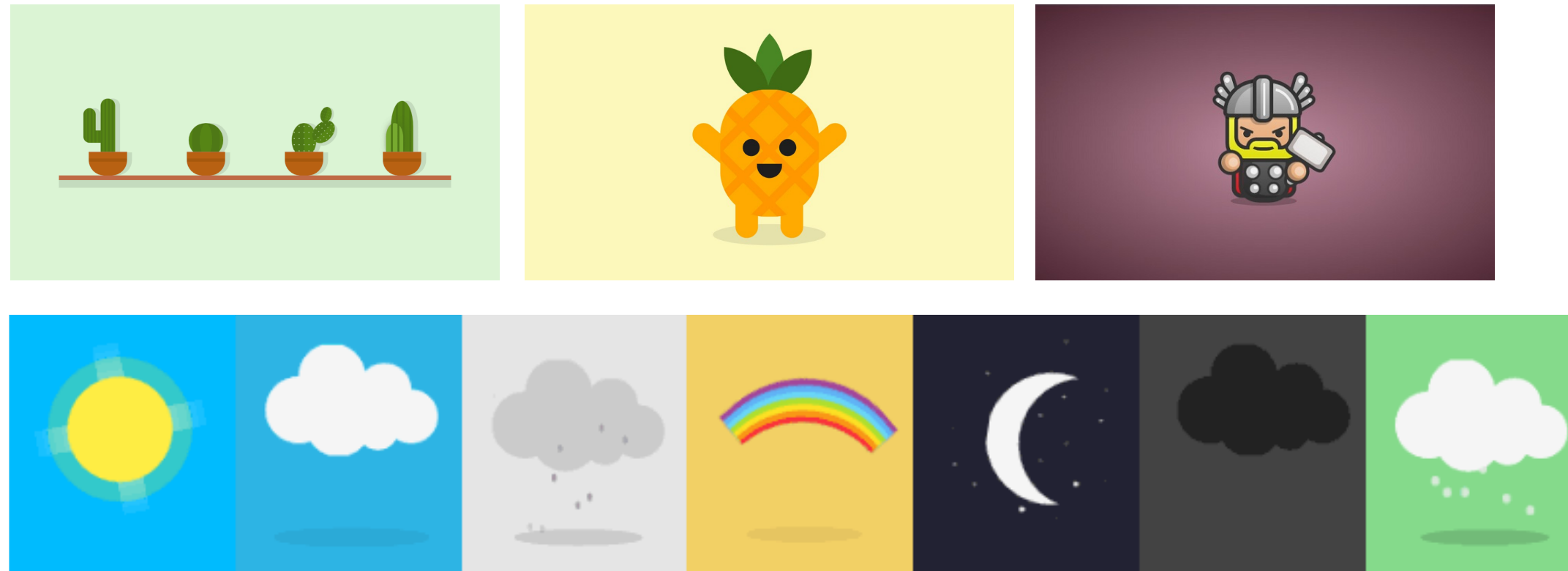
schaduw

```
.maan{  
  width: 80px;  
  height: 80px;  
  border-radius: 50%;  
  box-shadow: 15px 15px 0 0 red;  
}
```


CSS art (combineren shapes)

Maak kunst met CSS

Nu je hebt gezien hoe je vormen kunt bouwen. Kun je deze technieken gebruiken op toffe illustraties/karakters/icons/kunst te maken.

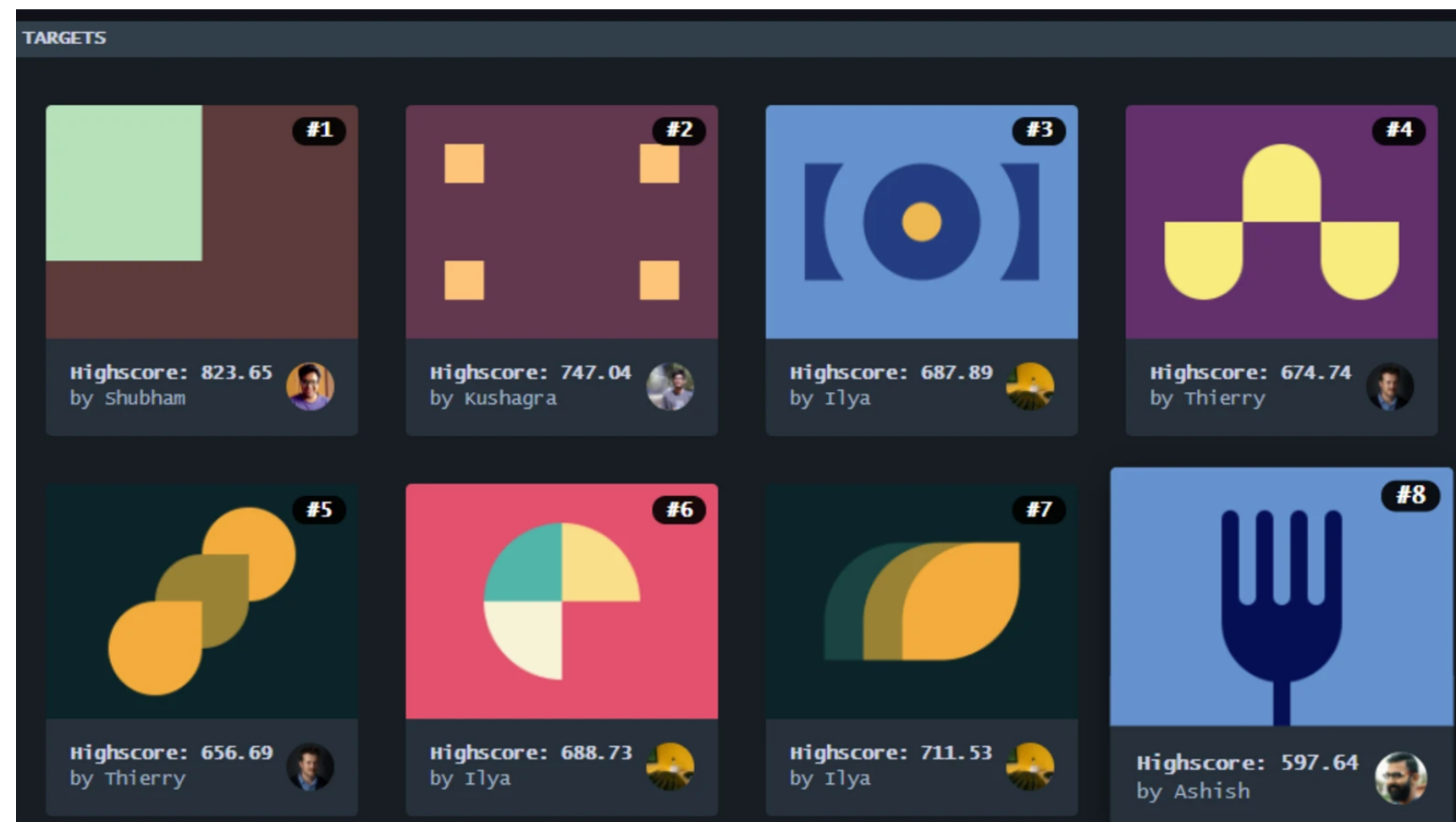


CSS art (combineren shapes)

CSS battle

Een leuke manier om je CSS art skills te testen is op cssbattle.dev

| <https://cssbattle.dev/>



CSS animation

keyframes

@keyframes kun je gebruiken voor het maken van animaties. Hier kun je met from to een start en eindpunt stylen. Ook kun je in percentages aangeven wat er veranderd van een element.

De @keyframe krijgt een naam, deze kun je aanroepen in een CSS rule van een element. Het element dat de animatie aanroept krijgt het attribuut animation met daarin de: naam, duration, timing function, delay, iteration count en direction (niet allemaal verplicht). Dit kun je ook als losse attributen toevoegen. Daarnaast heb je nog andere attributen die je kunt gebruiken voor je animaties.

| https://www.w3schools.com/css/css3_animations.asp

CSS shapes

Leer meer shapes

- | https://www.w3schools.com/howto/howto_css_shapes.asp
- | <https://css-tricks.com/the-shapes-of-css/>

CSS animation

keyframes

```
.myelement {  
  width: 200px;  
  height: 200px;  
  animation: myanimation 5s infinite;  
}  
  
@keyframes mymove {  
  0% {top: 0px; background: red; width: 100px;}  
  100% {top: 200px; background: yellow; width: 300px;}  
}
```

Opdracht

CSS art

