HoGent

BEDRIJF
EN
ORGANISATIE

Webapplicaties 1

Hoofdstuk 11 – CSS animatie

HoGent

CSS advanced

- Transform
  - Scale
  - Rotate
  - Skew
  - Translate
- Transition
- Animation

Transform

Transform – Transition – Animation

- Transform kan objecten (elementen) transformeren:
  - groter-kleiner maken (scale)
  - draaien (rotate)
  - vervormen (skew)
  - verplaatsen (translate)
- Transition is een korte animatie tussen twee toestanden van een element. Voorbeelden: zichtbaar worden van een element, uitschuiven van een menu, …
- Een animatie is een opeenvolging van verschillende transitions.

Support

- https://robots.thoughtbot.com/transitions-and-transforms
- Browser support
  - https://caniuse.com/#feat=transforms2d
  - https://caniuse.com/#feat=css-transitions
  - https://caniuse.com/#feat=css-animation
- Oudere/mobile browsers ondersteunen, maar nood aan prefix
  - -webkit-
  - -ms-
- https://autoprefixer.github.io/

Support

- Schakel de autoprefixer functie aan in de Visual Studio Code settings (File – Preferences – Setting)
- Zoek naar: liveSassCompile.settings.autoprefix
- Autoprefixer functie in Live Sass Compiler
  - Bepaal browser support voor de autoprefixer en de prefixen zullen automatisch gegenereerd worden bij het compileren van SCSS naar CSS.
  - https://github.com/ai/browserslist

```
// Automatically add vendor prefixes to
unsupported CSS properties (e. g.
transform -> -ms-transform). Specify what
browsers to target with an array of
strings (uses [Browserslist]
(https://github.com/ai/browserslist)).
Pass `null` to turn off.
// Default is `null`
"liveSassCompile.settings.autoprefix":
null,
```

Place your settings here to overwrite the Default Settings.

```
1  {
2      "workbench.colorTheme": "Default Light+",
3      "window.zoomLevel": 0,
4      "liveServer.settings.donotShowInfoMsg": true,
5      "liveSassCompile.settings.autoprefix": ["> 1%","last 5 versions"]
6  }
```

2D transforms

- In de volgende slides worden er verschillende voorbeelden van 2D transforms getoond.
- We gaan daarbij uit van de onderstaande code [en zullen er in de volgende slides code aan toevoegen]
- Vendor-prefixes zullen gegenereerd worden in de CSS door de autoprefixer (-webkit- voor Safari/Chrome –moz voor Firefox)

```
<body>
    <div>ONE</div>
    <div>TWO</div>
    <div>THREE</div>
</body>
```

```
div {
    display: inline-block;
    padding: 1em;
    margin: 1em;
    background-color: mediumorchid;
    color: #fff;
}
```

scale()

- Met behulp van scale() kan je de grootte van een element aanpassen, al naargelang de waarde van de parameters voor de breedte en de hoogte. Wordt er maar 1 parameter opgegeven, dan is die van toepassing op zowel de breedte als de hoogte

```
&:nth-child(2) {
    transform: scale(1.5);
}
```

```
&:nth-child(3) {
    transform: scale(2.5);
}
```

scale()

- Nog 2 voorbeelden

```
&:nth-child(3) {
    transform-origin: left;
    transform: scale(2.5);
}
```

De positie van het getransformeerde element wordt aangepast.
Mogelijke waarden zijn
horizontaal: left-right-center of percentages
verticaal: top-bottom-center of percentages

```
&:nth-child(3) {
    transform: scale(1.5,3);
}
```

De breedte is 1.5 keer de oorspronkelijke breedte. De hoogte is 3 keer de oorspronkelijke hoogte

rotate()

- Met behulp van rotate() kan je een element roteren over een opgegeven aantal graden.
  - Gebruik positieve waarden om in wijzerzin te roteren
  - Gebruik negatieve waarden om tegenwijzerzin te roteren

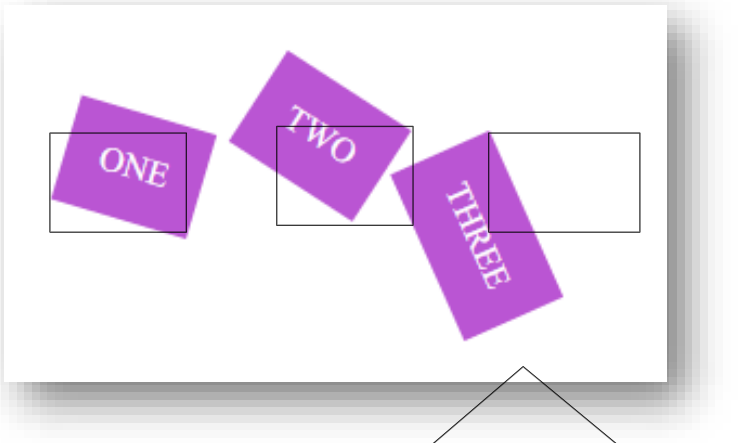
```scss
&:nth-child(1){
    transform: rotate(16.5deg);
}

&:nth-child(2) {
    transform: rotate(33deg);
}

&:nth-child(3) {
    transform: rotate(66deg);
}
```

```scss
&:nth-child(1){
    transform: rotate(-16.5deg);
}

&:nth-child(2) {
    transform: rotate(-33deg);
}

&:nth-child(3) {
    transform: rotate(-66deg);
}
```

rotate()

- Je kan gebruik maken van transform-origin om het punt waarrond gedraaid wordt, te wijzigen

```
&:nth-child(1){
    transform-origin: bottom right;
    transform: rotate(16.5deg);
}

&:nth-child(2) {
    transform-origin: top right;
    transform: rotate(33deg);
}

&:nth-child(3) {
    transform-origin: top right;
    transform: rotate(66deg);
}
```

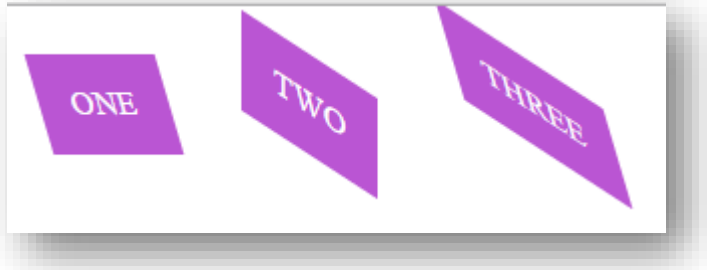


In dit voorbeeld werden de oorspronkelijke posities van de 3 div's toegevoegd, zodat het gemakkelijker is om te zien dat elk van de 3 div's rond een ander punt roteert.

skew()

- Met behulp van skew() (=scheef trekken) kan je een vorm van perspectief creëren.
  - skewX() wordt gebruikt voor horizontaal skewing
  - skewY() wordt gebruikt voor verticaal skewing

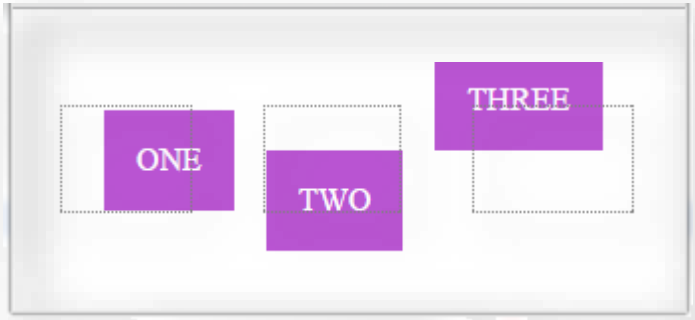
```
&:nth-child(1){
    transform: skewX(16.5deg);
}

&:nth-child(2) {
    transform: skewY(33deg);
}

&:nth-child(3) {
    transform: skew(16.5deg, 33deg);
}
```



translate()

- Met behulp van translate() (=verschuiven) kan je een element verplaatsen.
  - translateX() wordt gebruikt voor de horizontale verschuiving
  - translateY() wordt gebruikt voor de verticale verschuiving

```
&:nth-child(1){
    transform: translateX(20px);
}

&:nth-child(2) {
    transform: translateY(20px);
}

&:nth-child(3) {
    transform: translate(-20px, -30px);
}
```



In dit voorbeeld werden de oorspronkelijke posities van de 3 div's toegevoegd om de verschuiving beter te illustreren

Transitions

- Een transitie is een kleine animatie tussen 2 statussen van een element, zoals het activeren van een drop-down menu of het sluiten van een pop-up venster.
- In plaats van de elementen onmiddellijk te laten verschijnen of onmiddellijk te laten verdwijnen, kan het menu bijvoorbeeld langzaam verschijnen of kan het pop-up venster langzaam verdwijnen.
- Transities kunnen gemakkelijk gerealiseerd worden met CSS. Maak daarbij bijvoorbeeld gebruik van de pseudo-class **:hover**

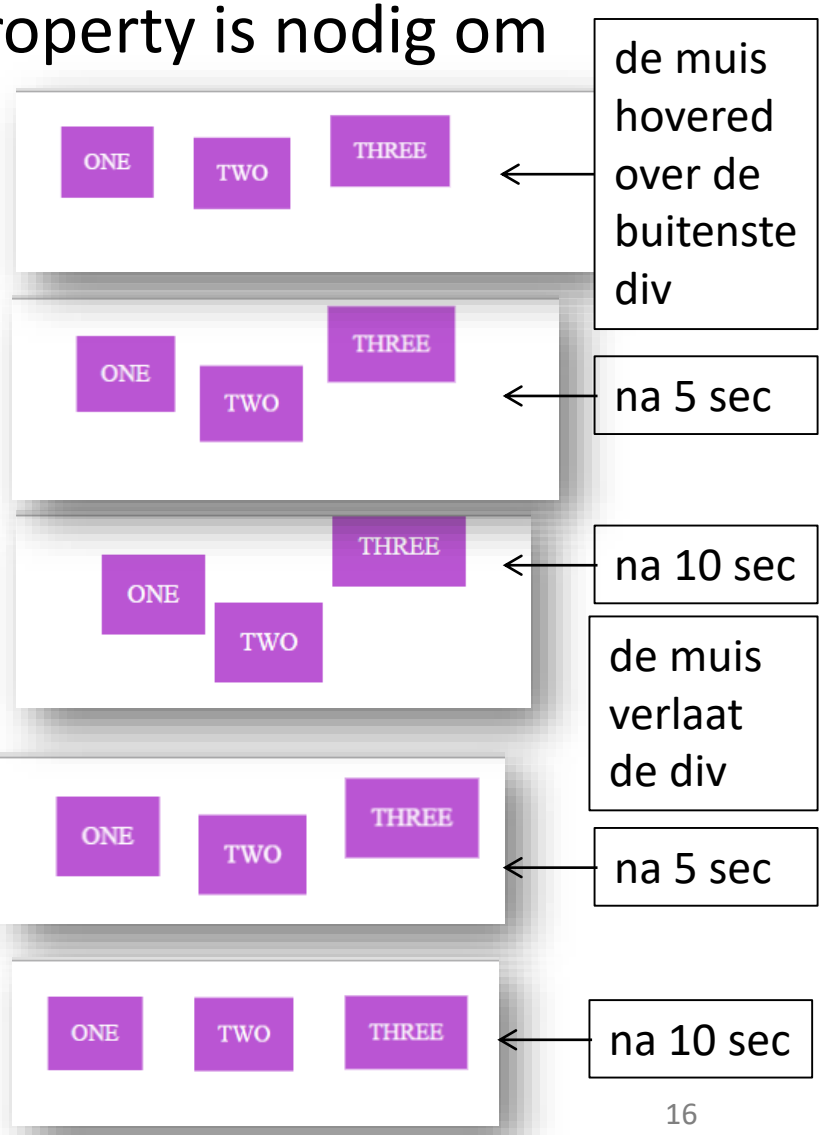
Transitions

- Transition is een korte notatie die bestaat uit enkele onderdelen
  - Transition: [property] [duration] [timing-function] [delay]
- Standaard heeft transition de volgende waarde
  - all 0s ease 0s
  - Dit betekent dat alle properties een transitie krijgen gedurende 0 sec met de ease-functie, met andere woorden alles gebeurt onmiddellijk
- Daarom moet minstens de transition-duration ingesteld worden voor we een transitie kunnen zien

transition-duration

- Enkel de transition-duration property is nodig om transition te implementeren:

```
body {
    div {
        transition-duration: 10s;
    }
    &:hover div {
        &:nth-child(1) {
            transform: translateX(20px);
        }
        &:nth-child(2) {
            transform: translateY(20px);
        }
        &:nth-child(3) {
            transform: translate(-20px, -30px);
        }
    }
}
```

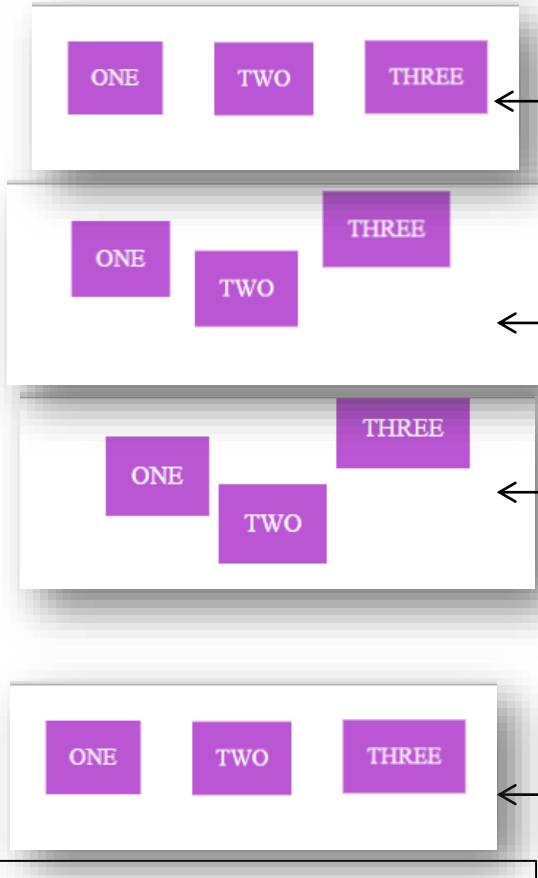


Wanneer transition-duration is ingesteld op de default status van het element, dan wordt deze tijdsduur zowel voorwaarts (bij het binnengaan) als achterwaarts (bij het verlaten) toegepast

transition-duration

- Enkel de transition-duration property is nodig om transition te implementeren:

```
body {
    &:hover div {
        transition-duration: 10s;

        &:nth-child(1) {
            transform: translateX(20px);
        }
        &:nth-child(2) {
            transform: translateY(20px);
        }
        &:nth-child(3) {
            transform: translate(-20px, -30px);
        }
    }
}
```



de muis hovered over de buitenste div

na 5 sec

na 10 sec

de muis verlaat de div

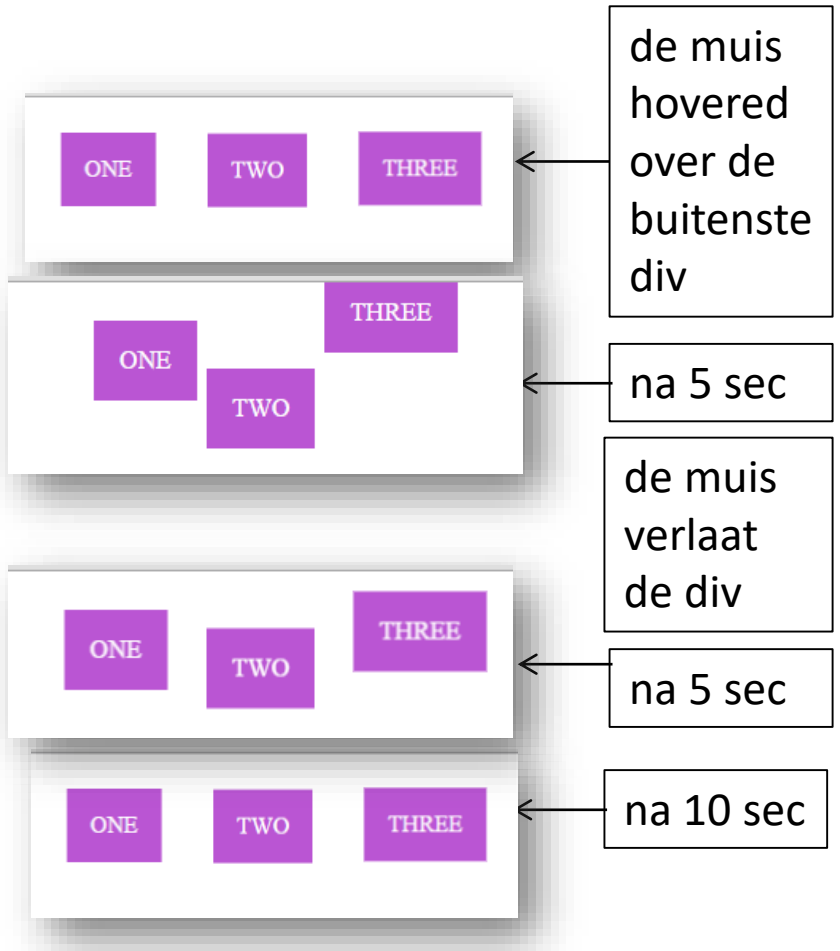
onmid-delijk daarna

Wanneer transition-duration is ingesteld op de hover status van het element, dan wordt deze tijdsduur enkel voorwaarts (bij het binnengaan) toegepast. Bij het verlaten keert het element onmiddellijk terug naar zijn begintoestand

transition-duration

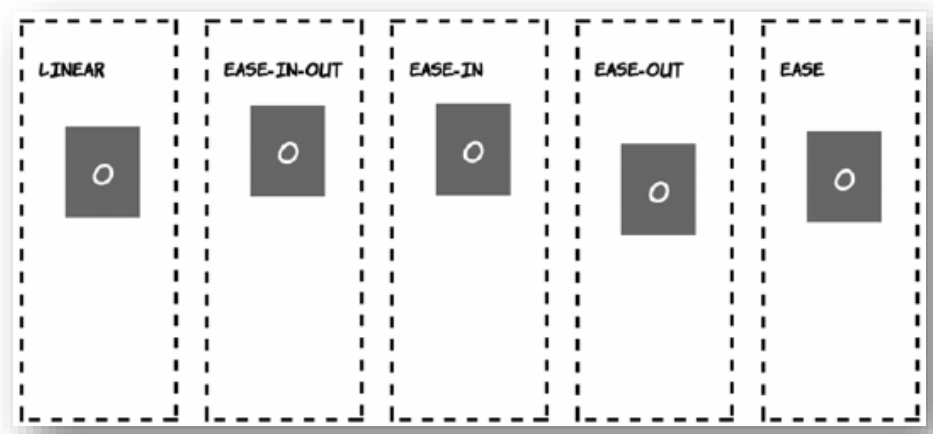
- Je kan een andere transition-duration instellen op de default én op de hover status

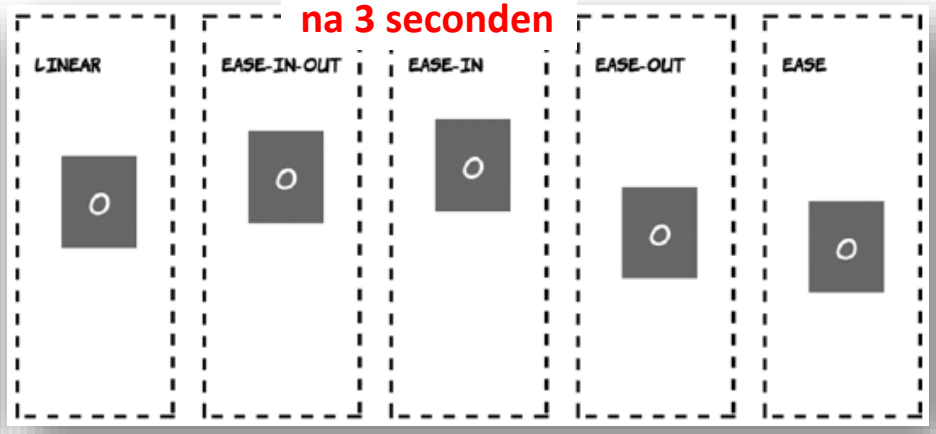
```
body {
    div {
        transition-duration: 5s;
    }
    &:hover div {
        transition-duration: 10s;

        &:nth-child(1) {
            transform: translateX(20px);
        }
        &:nth-child(2) {
            transform: translateY(20px);
        }
        &:nth-child(3) {
            transform: translate(-20px, -30px);
        }
    }
}
```

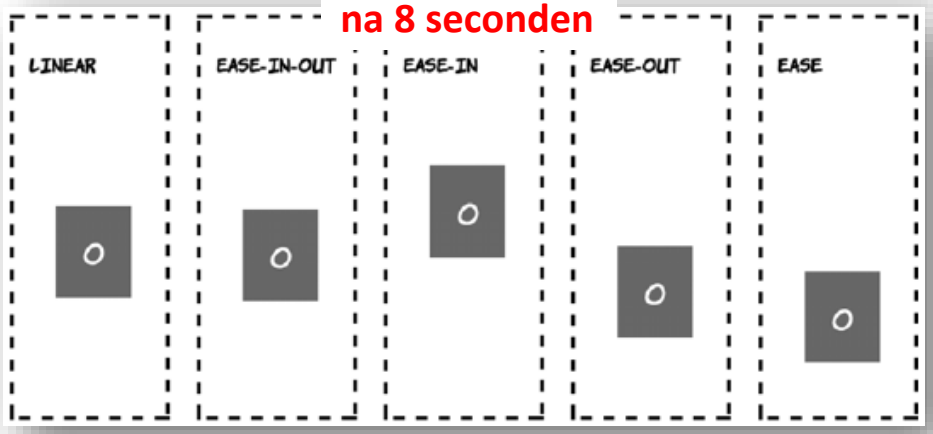


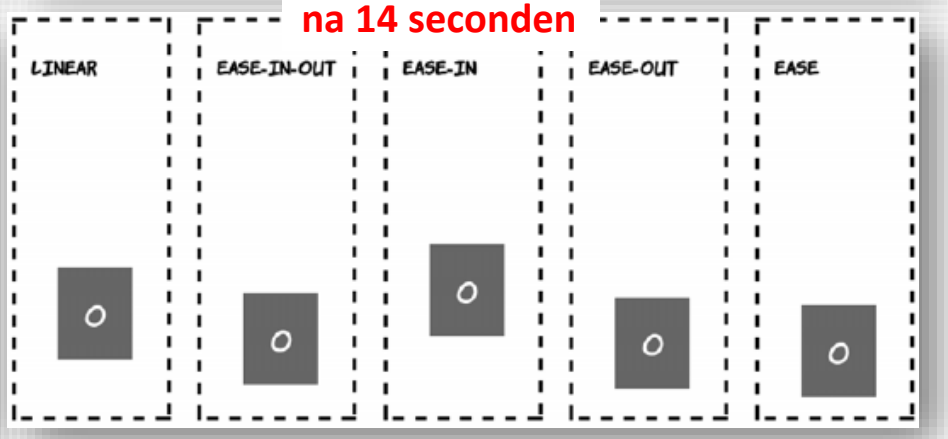
transition-timing-function

- Een transitie verloopt standaard lineair in de tijd. Andere mogelijke waarden zijn: ease-in-out, ease-in, ease-out, ease. ease-out is de snelste, gevolgd door ease





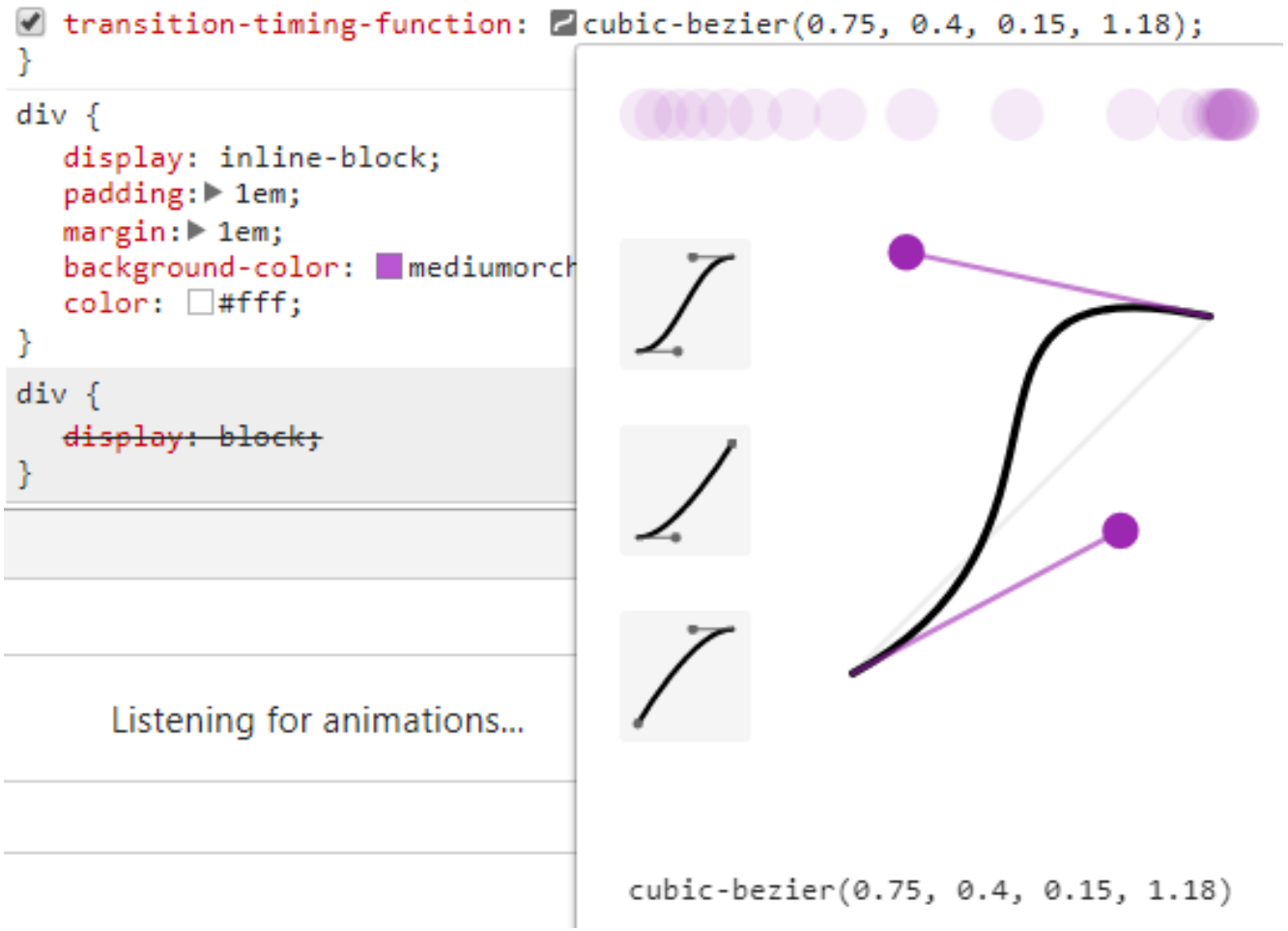




transition-timing-function

- Meer geavanceerde timing functions kunnen gemaakt worden aan de hand van een cubic-Bezier functie
  - cubic-bezier(0.75, 0.4, 0.15, 1.18);
- Deze kan gegenereerd worden met de Chrome Dev Tools
  - Zet een standaard transition-timing-function zoals ease-in
  - Open de Chrome Dev Tools en selecteer het element met de transition
  - Klik op de transition-timing-function in het CSS gedeelte van Chrome Dev Tools
  - Pas de curve aan waarmee de transition-timing-function omschreven wordt
  - Een cubic-bezier functie wordt gegenereerd

transition-timing-function



transition-property

- Met behulp van transition-property kan je verschillende transities toepassen op 1 element (gescheiden door een ,)

```
body {
    div {
        position: relative;
        top: 0px;
        transition-property: top, transform;
        transition-duration: 10s, 20s;
        transition-timing-function: ease-out, linear;
    }
    &:hover div {
        &:nth-child(1) {
            top: 75px;
            transform: scale(2);
        }
```

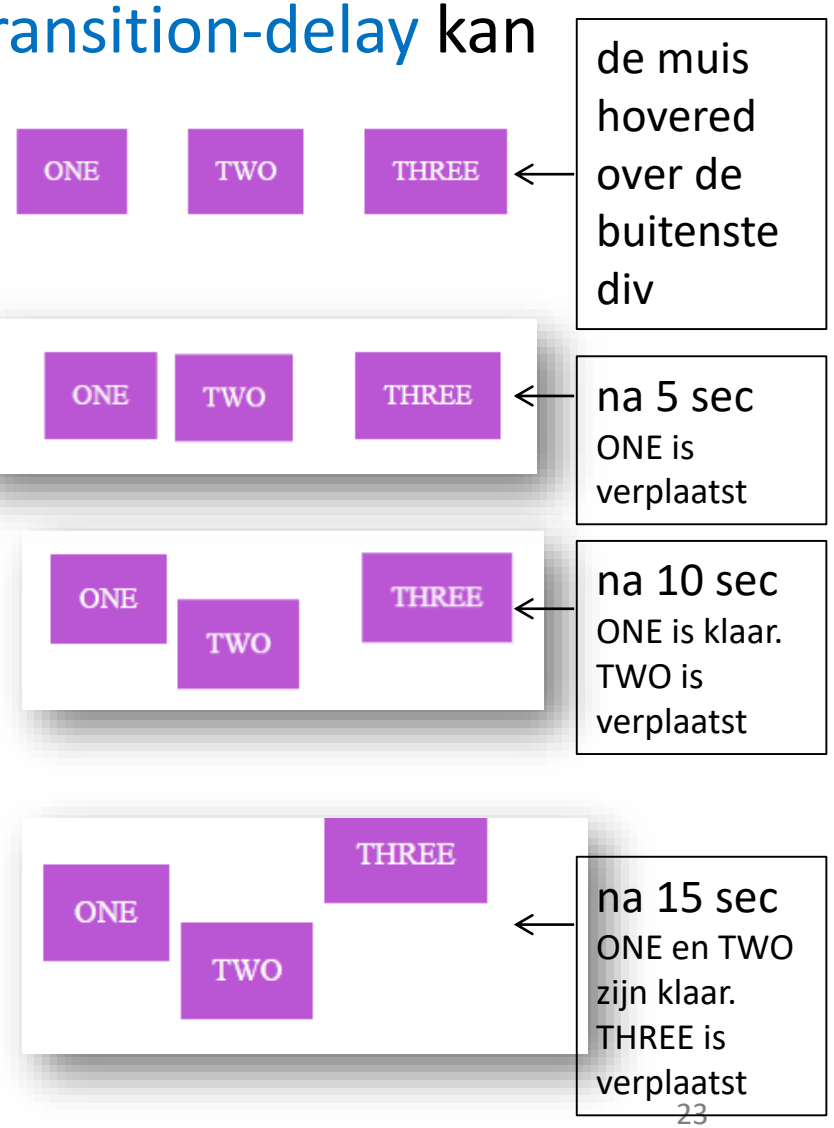
Er zijn 2 transities van toepassing op de div

- top / 10 s / ease-out: verplaatst de top van 0px naar 75px in 10 seconden. Deze transitie is dus afgerond na 10 sec.
- transform / 20s / linear: maakt de div 2 keer zo groot in 20 seconden. Deze transitie is dus pas afgerond na 20 sec.

transition-delay

- Met behulp van de property transition-delay kan je een wachttijd instellen vooraleer de transitie begint

```
body {
    div {
        transition-duration: 10s;
    }
    &:hover div {
        &:nth-child(1) {
            transform: translateX(30px);
        }
        &:nth-child(2) {
            transform: translateY(30px);
            transition-delay: 5s;
        }
        &:nth-child(3) {
            transform: translate(-30px, -30px);
            transition-delay: 10s;
        }
    }
}
```



Voorbeelden

- Mondrian Clock Pure CSS : http://codepen.io/slyka85/pen/xgRmpa

Animation

Animation

- Een animatie is een opeenvolging van verschillende transitions.
- Animatie wordt bepaald door keyframes

```
@keyframes [name] {
    from {
        [styles];
    }
    to {
        [styles];
    }
}
```

```
@keyframes bounce {
    from {
        top: 50px;
        transform: scale(1);
    }
    25% {
        top: 200px;
        transform: scale(2);
    }
    50% {
        top: 50px;
        transform: scale(1);
    }
    75% {
        top: 200px;
        transform: scale(2);
    }
    to {
        top: 50px;
        transform: scale(1);
    }
}
```

- Kies bij voorkeur properties die transities kunnen zijn:
  https://www.w3.org/TR/css3-transitions/#properties-from-css-

Animation

- Vervolgens kunnen we met behulp van de ‘animation’ property de keyframes laten afspelen op een bepaald element dat we willen animeren

```
.element {
    animation: [name] [duration] [timing-function] [delay]
    [iteration-count] [direction] [fill-mode] [play-state];
}
```

- Animation-name is een referentie naar de keyframes
- Animation-duration zet de duur van de totale animatie

```
div {
    left: 50px;
    position: relative;

    animation-name: bounce;
    animation-duration: 20s;
    animation-iteration-count: infinite;
}
```

- Animation-iteration-count stelt in hoe vaak de animatie herhaald moet worden. Dit kan een vaste waarde zijn zoals 3 of bijvoorbeeld infinite

Animation

- Voorbeeld

Animation

- We kunnen de animatie nog verder gaan bepalen met extra properties
- Animation-delay zegt hoelang er gewacht moet worden tot de animatie mag beginnen
- Animation-timing-function vertelt welke “flow” de animatie heeft, gelijklopend met transition
- Animation-fill-mode bepaalt hoe de transitie moet beginnen en eindigen: bijvoorbeeld op het einde van de animatie terug naar originele staat (none) of blijven op de eind staat (forwards)

Voorbeelden animation

- CSS Only Girl running : http://codepen.io/renduh/pen/mRraOd
- https://daneden.github.io/animate.css/
- https://codepen.io/anon/pen/YEJQEq