

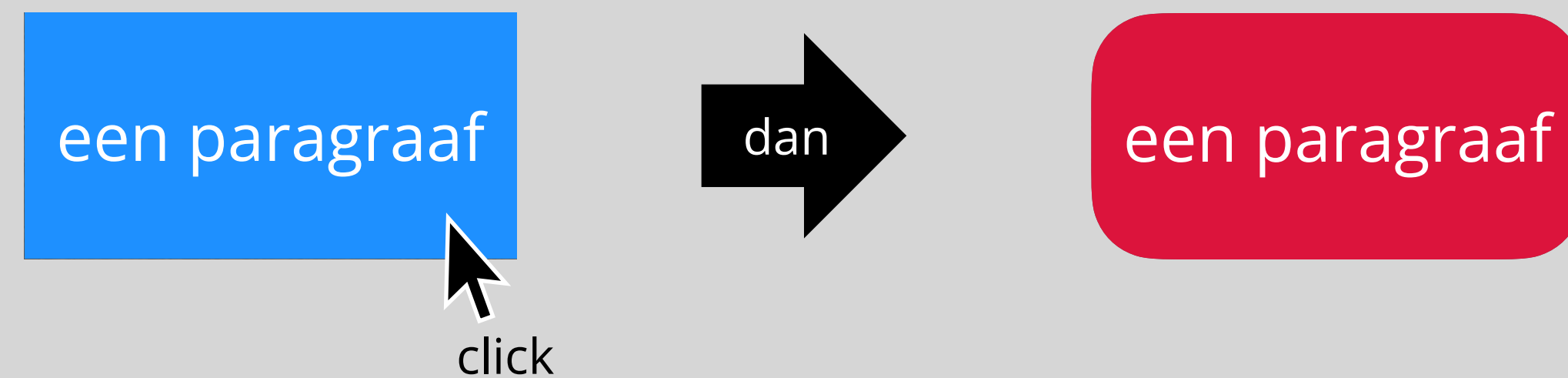
CSS challenge #5

# JS 3-STAP INTRO

# Recap JS 3-stap 1/5

1. `querySelector`
2. `addEventListener`
3. `classList`

- en dan CSS



## De HTML

```
<p>een paragraaf</p>
```

## De CSS

```
p {  
  width: 10em;  
  color: White;  
  background-color: DodgerBlue;  
}
```

## De JS

# Recap JS 3-stap 2/5

## 1. `querySelector`

2. `addEventListener`
3. `classList`

- en dan CSS



De eerste stap is het selecteren van het element dat de interactie start. Het advies om daarvoor altijd `document.querySelector()` te gebruiken. Dan kun je dezelfde selecteren gebruiken als in CSS.

Het geselecteerde element wordt opgeslagen in een variabele.

### De HTML

```
<p>een paragraaf</p>
```

### De CSS

```
p {  
  width: 10em;  
  color: White;  
  background-color: DodgerBlue;  
}
```

### De JS

```
deP = document.querySelector("p");
```

# Recap JS 3-stap 3/5

1. `querySelector`

**2. `addEventListener`**

3. `classList`

- en dan CSS



De tweede stap is het geselecteerde element iets laten doen als er een actie op uitgevoerd wordt.

Het laten luisteren naar kliks gebeurt met `onclick`.

Er zijn ook andere acties, bijv: `ondblclick`, `onkeydown`, `ondragstart`...

Het 'iets laten doen', gebeurt met een functie (hier `ietsDoen`). De naam van de functie mag je zelf bedenken.

## De HTML

```
<p>een paragraaf</p>
```

## De CSS

```
p {  
  width: 10em;  
  color: White;  
  background-color: DodgerBlue;  
}
```

## De JS

```
deP = document.querySelector("p");  
  
deP.addEventListener("click", ietsDoen);
```

# Recap JS 3-stap 4/5

1. `querySelector`
2. `addEventListener`
3. **`classList`**

- en dan CSS



De derde stap is om in de functie een class toe te voegen aan het element in de HTML waar iets mee moet gebeuren.

In het voorbeeld wordt - als er op de paragraaf geklikt wordt - de class “anders” aan de paragraaf in de HTML toegevoegd.

Nb. De class kan aan hetzelfde element worden toegevoegd als waarop de actie is uitgevoerd (zoals hier in het voorbeeld), maar dat hoeft niet (dat gaan we in oefening mee aan de slag).

## De HTML

```
<p class="anders">een paragraaf</p>
```

## De CSS

```
p {  
  width: 10em;  
  color: White;  
  background-color: DodgerBlue;  
}
```

## De JS

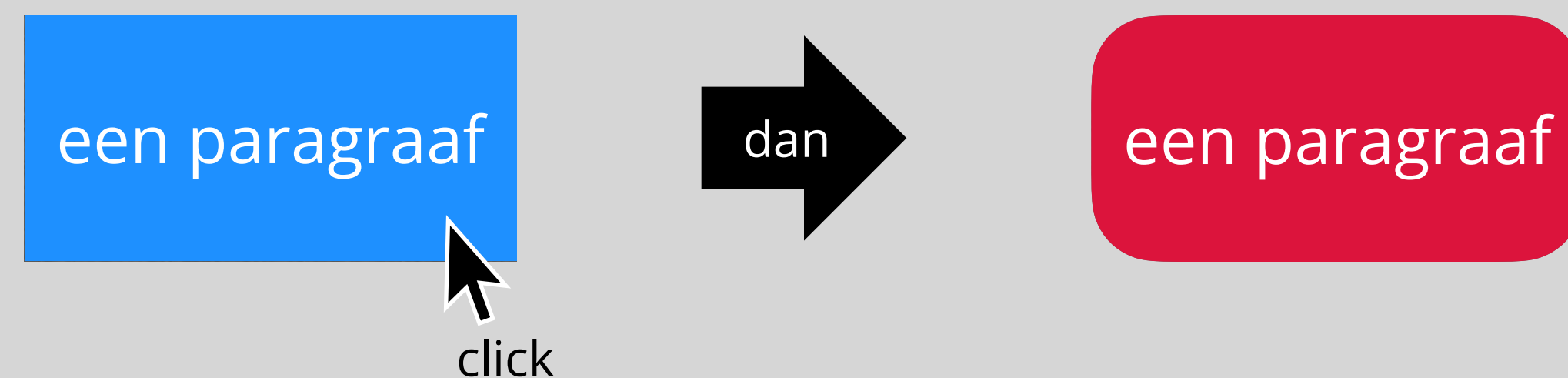
```
deP = document.querySelector("p");  
  
deP.addEventListener("click", ietsDoen);  
  
function ietsDoen() {  
  deP.classList.add("anders");  
}
```

Na 't klikken wordt een class aan de HTML toegevoegd. Dat hoeft je dus niet zelf te doen. Dat doet je JS dynamisch voor je.

# Recap JS 3-stap 5/5

1. `querySelector`
2. `addEventListener`
3. `classList`

- en dan CSS



De 'extra' class in de HTML kan in de CSS gebruikt worden om het element aan te passen.

## De HTML

```
<p class="anders">een paragraaf</p>
```

## De CSS

```
p {  
  width: 10em;  
  color: White;  
  background-color: DodgerBlue;  
}  
  
p.anders {  
  background-color: Crimson;  
  border-radius: 1.5em;  
}
```

## De JS

```
deP = document.querySelector("p");  
  
deP.addEventListener("click", ietsDoen);  
  
function ietsDoen() {  
  deP.classList.add("anders");  
}
```

# transition in 1 slide

In 't voorbeeld → Met de CSS property transition kan ingesteld worden dat de verandering van 'normaal' naar 'anders' 1 seconde duurt\*.



\*De kleur fade van blauw via paars naar rood. De afronding fade van geen afronding naar een afronding van 1.5em.

Voeg de transition toe aan de 'normale' status van het element in de CSS. Er kan veel meer als je dat interessant vindt. Alles weten:

[css-tricks.com/almanac/properties/t/transition/](https://css-tricks.com/almanac/properties/t/transition/)

## De HTML

```
<p class="anders">een paragraaf</p>
```

## De CSS

```
p {  
  width: 10em;  
  color: White;  
  background-color: DodgerBlue;  
  
  transition: 1s;  
}  
  
p.anders {  
  background-color: Crimson;  
  border-radius: 1.5em;  
}
```

## De JS

```
deP = document.querySelector("p");  
  
deP.addEventListener("click", ietsDoen);  
  
function ietsDoen() {  
  deP.classList.add("anders");  
}
```

ИИТВО  
123-21АБ

С22 сүгүлүгү #2