

Uitleg over acc en curr in reduce()

In deze regel:

```
-----  
const totaal = this._cijfers.reduce((acc, curr) => acc + curr, 0);  
-----
```

wordt de reduce methode gebruikt om alle cijfers in de array _cijfers op te tellen.

Wat zijn acc en curr?

-
- acc (accumulator): Dit is de tussentijdse optelling van alle vorige elementen in de array.
 - curr (current value): Dit is het huidige element uit de array dat opgeteld wordt bij acc.

Hoe werkt het stap voor stap?

Stel dat _cijfers de volgende waarden heeft:

```
this._cijfers = [8, 7.5, 6];
```

Dan gebeurt dit intern in de reduce() methode:

Iteratie	acc (huidige som)	curr (huidig cijfer)	Nieuwe acc
----------	-------------------	----------------------	------------

Start	0	8	$0 + 8 = 8$
1e loop	8	7.5	$8 + 7.5 = 15.5$
2e loop	15.5	6	$15.5 + 6 = 21.5$

Na de reduce() bewerking is de totaalsom van de cijfers dus 21.5.

Vervolgens wordt het gemiddelde berekend met:

return Number((totaal / this._cijfers.length).toFixed(2));

Wat betekent:

- totaal / this._cijfers.length -> $21.5 / 3 = 7.17$
- .toFixed(2) -> Houdt twee decimalen over, dus '7.17' als string.
- Number(...) -> Zet de string '7.17' terug om naar een number.

Conclusie

- acc begint als 0 en verzamelt de som van de cijfers.
- curr is het huidige cijfer dat wordt toegevoegd aan acc.
- reduce() zorgt ervoor dat we niet met een for-loop door de array hoeven te gaan.

Wil je dit in de praktijk testen met een voorbeeld?