

## Taakgroep: Javascript Callbacks

### Context

Je bent bekend met het aanroepen van functies na elkaar. Zoals in onderstaande voorbeeld:

```
ingredientenVerzamelen();
```

```
dinerKoken();
```

De functie `dinerKoken` wordt pas uitgevoerd wanneer `ingredientenVerzamelen` volledig is uitgevoerd. Dit betekent dat de rest van het programma geblokkeerd is. In dit level gaan we op zoek naar een oplossing waarbij we callback functions gebruiken. De callback function wordt aangeroepen zodra de eerste function klaar. Dit is noodzakelijk omdat in de game die je ontwikkeld de server wordt aangeroepen, dit kost tijd en dit mag niet ten koste gaan van de gebruikerservaring.

### Taak: Callback oefening

In deze opdracht implementeer je een eenvoudige callback function.

#### Aanpak

Eerst implementeer je de functions `ingredientenVerzamelen` en daarna `dinerKoken`. Vervolgens wijzig je de functies zodat er sprake is van een callback.

- Maak een project aan in Webstorm.
- Maak een `index.html` en voeg een `<script>` tag toe.
- Neem onderstaande code over in je pagina:

```
function ingredientenVerzamelen() {  
  console.log('Ingredienten verzameld.')}  
  
function dinerKoken() {  
  console.log('Lekker roeren in de soep');  
}  
  
ingredientenVerzamelen();  
dinerKoken();
```

- Controleer in de browser op de output in de console aanwezig is.
- In werkelijkheid duurt het verzamelen van ingredienten enige tijd. Wijzig daarom de function `ingredientenVerzamelen` in onderstaande code:

```
function ingredientenVerzamelen() {
  setTimeout( function(){
    console.log('Ingredienten verzameld.');
```

- Controleer de output in de console. Je zult zien dat de output is: 'Lekker roeren in de soep' 'Ingredienten verzameld.'

### Conclusies:

Het programma wordt niet meer geblokkeerd: asynchroon programma verloop. Dit is een voordeel wanneer een functie lang(er) duurt.

Er moet een oplossing komen om te reageren op een function die niet blokkeert (async function).

- Door function `dinerKoken` mee te geven als parameter aan de function `ingredientenVerzamelen` kan deze zelf bepalen wanneer de function `dinerKoken` wordt uitgevoerd. Neem onderstaande code over en probeer deze uit in de browser:

```
function ingredientenVerzamelen(callbackFunction) {
  setTimeout( function(){
    console.log('Ingredienten verzameld.');
```

```
function dinerKoken() {
  console.log('Lekker roeren in de soep');
}
```

```
ingredientenVerzamelen(dinerKoken);
```

**Conclusie** Een callback kan gebruikt worden om te reageren op een async function.

## Ondersteunende informatie

### Eenvoudige tutorial over callbacks

What the heck is a callback