

# Console en dialogen

## Console API

Het `console` [🔗](#) object geeft toegang tot de debugging console van de browser (Firefox: Web Console, Chrome: Console tab in DevTools). Het `console` object kan overal binnen JavaScript aangesproken worden, omdat dit een **globaal object** is.

```
1 console.log('Greetings Earthlings! We \u{1F47e} come in peace.');
```

De bovenstaande code resulteert in de volgende output: `Greetings Earthlings! We 🌍 come in peace .`

Het `console` object kan gebruikt worden in iedere JavaScript applicatie gebruikt worden, bijv.: Node.js, website ...

## Methoden

Een methode is een actie die we kunnen uitvoeren op één of meerdere eigenschappen van een object aan te passen. Met de methoden uit het `console` object kunnen we de output naar de console uit een browser manipuleren.

De meest gebruikte methode is de `log()` [🔗](#) methode. Deze methode print een boodschap naar de browser console. De boodschap kan tekst of een object zijn. Om de output meer betekenis te geven kunnen we gebruik maken van de methoden `info()`, `warning()` en `error()`. Afhankelijk van het type webbrowser wordt de tekst in de console voorafgegaan door een icoon.

```
1 const person1 = {
2   firstName: 'Philippe',
3 }
4 const person2 = {
5   firstName: 'Evelien',
6 }
7 console.log('Person ', person1, ' will teach webpgm-1 and ', 'person ', person2, ' shall')
```

Meerdere objecten kunnen tegelijkertijd in de console getoond worden al dan niet in combinatie met tekst.

```
1  const msgInfo = 'This is just information!';
2  const msgWarning = 'Try not to use the var keyword!';
3  const msgError = 'Can\'t fetch data from the server!';
4  console.info(msgInfo);
5  console.warn(msgWarning);
6  console.error(msgError);
```

`info`, `warn` en `error` geven een visuele indicatie van het bericht.

```
1  console.error('[%s] Can\'t fetch the data from the server %c%s', new Date().toLocaleString(),
```

Door middel van string substitutie kunnen we strings ( `%s` ) en objecten ( `%o` of `%O` ) injecteren in de boodschappen. Met `%c` kunnen we zelfs opmaak voorzien voor het bericht in de Console.

## Dialogen

Alle dialoogvensters in JavaScript hebben het `window` object nodig om te kunnen functioneren of aangeroepen te worden. Dit betekent dat een HTML document steeds aanwezig moet zijn aangevuld met een `<script>` element met daarin de JavaScript-code of via een gekoppeld JavaScript-bestand. Hierdoor kan het `window` object en de gerelateerde eigenschappen en methoden aangesproken worden.

**Dialogen** (*Eng. dialog boxes*) zijn **modal windows**. Dit betekent dat de gebruiker de toegang wordt ontzegd tot de onderliggende applicatie totdat het dialoogvenster wordt gesloten.

### alert

Met de `alert()` [🔗](#) methode uit het `window` object kunnen we een **alert dialog** implementeren met tekstuele inhoud en een OK knop. Het alert dialoogvenster wordt voornamelijk gebruikt wanneer de gebruiker een erkenning moet geven aan een bepaald bericht.

```
1  window.alert('Greetings earthlings!');
2  alert('Greetings Earthlings!');
```

Het **bericht** (*Eng. message*) is optioneel. Indien een object aanwezig is als parameter i.p.v. een string, zal (indien een `toString()` methode aanwezig is) de string representatie van het object weergegeven worden, of het object zelf.

```
1  const person = {
2    firstName: 'Philippe',
3    toString: function() {
4      return this.firstName;
5    }
6  }
7  alert(person);
```

## prompt

Met de `prompt()` methode kunnen we een dialoogvenster weergeven met een input tekstveld en een optionele boodschap. De ingegeven tekst wordt **teruggegeven** (*Eng. return*) aan de JavaScript-code. Het resultaat is dus altijd een stringwaarde of een `null` waarde.

```
1  const height = prompt('What is your height?', 'ex.: 1.86');
2  console.log(height);
```

De tweede parameter van de `prompt()` methode geeft een standaard of default waarde aan het input tekstveld. Drukt de gebruiker op de knop “**Annuleren**” (*Eng. cancel*), dan zal een `null` waarde teruggegeven worden. Schrijven we niks in het veld resulteert dit in een lege string `""` na het drukken op de “OK” knop.

Wensen we een numerieke waarde te bewaren in de variabele `height`, dan moeten we de string converteren naar een **Number**.

```
1  let height = prompt('What is your height?', 'ex.: 1.81');
2  height = Number(height);
3  console.log('Datatype of height is %s with the value %f', typeof height, height);
```

## confirm

Met de `confirm()` methode kunnen we een dialoogvenster weergeven met twee knoppen “OK” en “Annuleren” en een optionele boodschap. Deze methode geeft `true` (“OK” knop) of `false` (“Annuleren” knop of sluiten van

venster) terug.

```
1  if (confirm('Aanvaard je de cookie van onze website?')) {  
2      console.log('Cookie aanvaard!');  
3  } else {  
4      console.log('Cookie verworpen!');  
5  }
```

[← Grammatica en datatypen](#)

[Operatoren →](#)