Functie declaratie

**Een functie declaratie geeft een naam aan een functie variabele zonder deze toe te kennen aan een variabele. Functie declaraties beginnen met het sleutelwoord function. Ze gelijken op de tradionele manier waarop functies in procedurele en objectgeoriënteerde talen gedeclareerd worden.**

Het sleutelwoord function

Identieke stukjes script kunnen dikwijls op meerdere plaatsen gebruikt worden. Het is veel efficiënter om er een functie van te maken. Eventuele wijzigingen moeten maar op één plaats aangebracht worden, zodat de code gemakkelijker te onderhouden is.

Het gebruik van functies veronderstelt twee dingen:

1. de functie eerst definiëren
2. de functie aanroepen

Het definiëren van een functie doe je zo:

function sayHelloWorld() {

alert(‘Hello, world’);

}

Gebruik het gereserveerde woord function, gevolgd door de naam van de functie (in het voorbeeld: sayHelloWorld) en de twee haakjes "(" en ")". Tussen die haakjes kunnen éé of meer parameters staan. Daarna volgt het codeblok waarin staat wat er moet gebeuren als de functie wordt aangeroepen. Het codeblok staat tussen accolades "{" en "}". Deze accolades worden ook gebruikt om twee of meer opdrachten samen te voegen tot een "blok" dat meerdere malen moet worden uitgevoerd in een herhalingslus. Bij het definiëren van functies zijn de accolades echter altijd verplicht, ook als de functie slechts één enkele opdracht bevat.

Om dat functie aan te roepen schrijf je de naam gevolgd door ronde haken:

sayHelloWorld();

Bij het aanroepen van een functie gebruiken we dus de naam ervan, gevolgd door de twee ronde haakjes, maar zonder het gereserveerde woord function.

De functie uit dit voorbeeld is niet erg interessant: er wordt alleen maar een tekst op het scherm getoond, maar er wordt geen waarde teruggegeven. In een taal als Pascal of Visual Basic zouden we dit een *procedure*noemen in plaats van een *functie*. In C is een functie die geen waarde retourneert van het type void. Als een functie daarentegen wel een waarde retourneert is de aanroep van zo'n functie zo:

antwoord = sayHelloWorld();

We kennen aan de variabele antwoord de waarde toe die door de functie wordt geretourneerd. Maar onze voorbeeldfunctie geeft geen waarde terug. Na bovenstaande aanroep zal variabele antwoord dan ook de waarde undefined hebben. We moeten dus een opdracht aan de functie toevoegen waarmee een waarde wordt geretourneerd, bijvoorbeeld:

return "Ok";

Ook return is een gereserveerd woord binnen JavaScript. Het mag alleen gebruikt worden binnen een functie.

function sayHelloWorld()

{

alert(‘Hello, world’);

return "Ok";

}

Je kan de functie uitproberen in de console:

console.log(sayHelloWorld());

Parameters

Als er sprake is van een functie met een parameter, wil dat zeggen dan een waarde (n) aan de functie wordt aangeboden als invoer. Functies kunnen één of meerdere parameters hebben. In dat geval worden de parameters van elkaar gescheiden door middel van komma's, bijvoorbeeld: mijnfunctie(a,b,c).

We herschrijven de functie van hierboven en geven er de naam aan saySomething. Deze functie kan om het even wat zeggen. Wat er gezegd wordt is afhankelijk van datgene wat als argument wordt meegegeven.

function saySomething(something)

{

alert(something);

return "Ok";

}

Ingebouwde functies

JavaScript kent ook enkele ingebouwde functies die we kunnen aanroepen zonder dat we ze eerst zelf hoeven te definiëren. Een voorbeeld daarvan is de functie parseInt. We geven een string als parameter mee aan de functie en krijgen hetzij een geheel getal, hetzij de waarde NaN (not a number) terug. Bijvoorbeeld: parseInt("3") geeft als resultaat het getal 3, net als parseInt("3Q").

Zodra de functie een teken tegenkomt dat niet kan worden geïnterpreteerd als deel van een geheel getal, wordt alles vanaf dat teken tot aan het einde van de string genegeerd. Zo geeft parseInt("3.14") het resultaat 3, want een "decimal point" (onze komma) kan geen deel uitmaken van een geheel getal, en parseInt("-3.14") geeft het resultaat -3. Kan het eerste teken van de string direct al niet deel uitmaken van een geheel getal, dan krijgen we het resultaat NaN terug.

Een vergelijkbare ingebouwde functie is parseFloat, alleen kunnen we daar ook een getal mèt een decimal point terugkrijgen. Zo geeft parseFloat("-3.14") de waarde -3.14 terug, en parseFloat("3q") de waarde 3.

JavaScript bevat ook twee functies om strings te kunnen gebruiken waarin tekens anders dan letters of cijfers voorkomen. De functie escape zet een niet-alfanumeriek teken om naar de notatie "%xx", waarbij xx de ASCII-waarde van dat teken is. Zo geeft escape(" ") het resultaat "%20". Voor tekens die wèl alfanumeriek zijn, geeft deze functie het teken zelf als resultaat terug, dus escape("a") geeft als resultaat "a". De functie unescape doet precies het tegenovergestelde: unescape("a") geeft als resultaat "a", en unescape("%20") geeft als resultaat " ". Deze functies zijn van belang om niet-alfanumerieke tekens van de ene web-pagina aan de andere door te kunnen geven.