11. JSON

## Inleiding

**JSON** staat voor **J**ava**S**cript **O**bject **N**otation Het is een **tekstformaat** voor het opslaan en transporteren van gegevens en is "zelfbeschrijvend" en gemakkelijk te begrijpen.

## JSON Voorbeeld

Dit voorbeeld is een JSON-tekenreeks:

{"name":"John", "age":30, "car":null}  
Copy

Het definieert een object met 3 eigenschappen:

* naam
* leeftijd
* auto

Elke sleutel (key) heeft een waarde. Als u de JSON-String parseert met een JavaScript-programma, hebt u toegang tot de gegevens als object:

let personName = obj.name;   
let personAge = obj.age;  
Copy

## Kenmerken

* JSON staat voor **J**ava**S**cript **O**bject **N**otation
* JSON is een lichtgewicht formaat voor gegevensuitwisseling
* JSON is platte tekst geschreven in JavaScript-objectnotatie
* JSON wordt gebruikt om gegevens tussen computers te verzenden
* JSON is taalonafhankelijk

De JSON-syntaxis is afgeleid van JavaScript-objectnotatie, maar de JSON-indeling is alleen tekst. Code voor het lezen en genereren van JSON bestaat in veel programmeertalen.

## Waarom JSON gebruiken?

Het JSON-formaat is syntactisch vergelijkbaar met de code voor het maken JavaScript-objecten. Hierdoor is een JavaScript programma kan eenvoudig JSON-gegevens converteren naar JavaScript-objecten.

Omdat het formaat alleen tekst is, kunnen JSON-gegevens eenvoudig tussen computers worden verzonden, en gebruikt door elke programmeertaal.

JavaScript heeft een ingebouwde functie voor het converteren van JSON-tekenreeksen naar JavaScript-objecten:

JSON.parse()

JavaScript heeft ook een ingebouwde functie voor het converteren van een object naar een JSON-tekenreeks:

JSON.stringify()

U kunt zuivere tekst van een server ontvangen en deze als JavaScript-object gebruiken.

U kunt een JavaScript-object naar een server verzenden in pure tekstindeling.

U kunt met gegevens werken als JavaScript-objecten, zonder ingewikkelde parseren en vertalen.

## Gegevens opslaan

Bij het opslaan van gegevens moeten de gegevens een bepaald formaat hebben en ongeacht waar u ervoor kiest om ze op te slaan, *tekst* is altijd een van de wettelijke indelingen.

JSON maakt het mogelijk om JavaScript-objecten als tekst op te slaan.

## Syntaxis

De JSON-syntaxis is een subset van de JavaScript-syntaxis. JSON-syntaxis is afgeleid van javascript-objectnotatiesyntaxis:

* Gegevens bevinden zich in naam/waardeparen key=value
* Gegevens worden gescheiden door komma's ,
* Krullende haakjes houden voorwerpen vast {}
* Vierkante haken bevatten arrays []

## JS Objecten

Omdat JSON-syntaxis is afgeleid van JavaScript-objectnotatie, is er weinig extra's software is nodig om met JSON binnen JavaScript te werken. Met JavaScript kunt u een object maken en gegevens toewijzen aan het, zoals dit:

person = {name:"John", age:31, city:"New York"};  
Copy

U hebt toegang tot een JavaScript-object als volgt:

// returns John   
person.name;  
Copy

Het kan ook als volgt worden geopend:

// returns John   
person["name"];  
Copy

Gegevens kunnen als volgt worden gewijzigd:

person.name = "Gilbert";  
Copy

Het kan ook als volgt worden gewijzigd:

person["name"] = "Gilbert";  
Copy

## JS-arrays als JSON

Op dezelfde manier waarop JavaScript-objecten kunnen worden geschreven als JSON, kunnen JavaScript-arrays ook geschreven worden als JSON. U zult later in deze zelfstudie meer te weten komen over objecten en arrays.

## JSON Bestanden

* Het bestandstype voor JSON-bestanden is ".json"
* Het MIME-type voor JSON-tekst is "application/json"