JavaScript JSON

JSON stands for **J**ava**S**cript **O**bject **N**otation.

JSON is a **plain text format** for storing and transporting data.

JSON is similar to the syntax for creating JavaScript objects.

JSON is used to **send**, **receive** and **store data**.

Example

```
'{"name":"John", "age":30, "car":null}'
```

The example above defines an object with 3 properties:

- "name"

- "age"
- "car"

Each property has a value:

- "John"
- 30
- null

Why JSON?

- JSON is make it easy to send and store data between computers
- JSON is text only and language independent *****

***** The syntax is derived from JavaScript object syntax, but JSON is text only.

Code for reading and generating JSON data can be written in any programming language.

The JSON format was originally specified by Douglas Crockford.

JSON and JavaScript

The JSON format is syntactically identical to the code for creating JavaScript objects.

Because of this, a JavaScript program can easily convert JSON data into native JavaScript objects.

JavaScript has a built in function for converting JSON strings into JavaScript objects:

`JSON.parse()`

JavaScript also has a built in function for converting an object into a JSON string:

`JSON.stringify()`

You can receive pure text from a server and use it as a JavaScript object.

You can send a JavaScript object to a server in pure text format.

You can work with data as JavaScript objects, with no complicated parsing and translations.

Storing Data

When storing data, the data has to be a certain format, and regardless of where you choose to store it, *text* is always one of the legal formats.

JSON makes it possible to store JavaScript objects as text.

JSON Example

Text that defines an employees object with an array of 3 employee objects:

```
{
"employees":[
  {"firstName":"John", "lastName":"Doe"},
  {"firstName":"Anna", "lastName":"Smith"},
  {"firstName":"Peter", "lastName":"Jones"}
]
}
```

If you parse the JSON string with a JavaScript program, you can access the data as an object:

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

JSON Data - A Name and a Value

JSON data is written as name/value pairs, just like JavaScript object properties.

A name/value pair consists of a field name (in double quotes), followed by a colon, followed by a value:

```
"firstName":"John"
```

JSON names require double quotes. JavaScript names do not.

JSON Objects

JSON objects are written inside curly braces.

Just like in JavaScript, objects can contain multiple name/value pairs:

```
{"firstName":"John", "lastName":"Doe"}
```

JSON Arrays

JSON arrays are written inside square brackets.

Just like in JavaScript, an array can contain objects:

```
"employees":[
  {"firstName":"John", "lastName":"Doe"},
  {"firstName":"Anna", "lastName":"Smith"},
  {"firstName":"Peter", "lastName":"Jones"}
]
```

In the example above, the object "employees" is an array. It contains three objects.

Each object is a record of a person (with a first name and a last name).

Converting a JSON Text to a JavaScript Object

A common use of JSON is to read data from a web server, and display the data in a web page.

For simplicity, this can be demonstrated using a string as input.

First, create a JavaScript string containing JSON syntax:

```
let text = '{ "employees" : [' +
'{ "firstName":"John" , "lastName":"Doe" },' +
'{ "firstName":"Anna" , "lastName":"Smith" },' +
'{ "firstName":"Peter" , "lastName":"Jones" } ]}';
```

Then, use the JavaScript built-in function `JSON.parse()` to convert the string into a JavaScript object:

```
const obj = JSON.parse(text);
```

Finally, use the new JavaScript object in your page:

Example

```
<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML =
obj.employees[1].firstName + " " + obj.employees[1].lastName;
</script>
```