

06. JSON Arrays.

6.1. Arrays as JSON Objects

Example:

```
[ "Ford", "BMW", "Fiat" ]
```

Arrays in JSON are almost the same as arrays in JavaScript.

In JSON, array values must be of type string, number, object, array, boolean or null.

In JavaScript, array values can be all of the above, plus any other valid JavaScript expression, including functions, dates, and undefined.

6.2. Arrays in JSON Objects

Arrays can be values of an object property:

Example:

```
{  
  "name": "John",  
  "age": 30,  
  "cars": [ "Ford", "BMW", "Fiat" ]  
}
```

6.3. Accessing Array Values

You access the array values by using the index number:

Example:

```
x = myObj.cars[0];
```

6.4. Looping Through an Array

You can access array values by using a for-in loop:

Example:

```
for (i in myObj.cars) {  
    x += myObj.cars[i];  
}
```

Or you can use a for loop:

Example:

```
for (i = 0; i < myObj.cars.length; i++) {  
    x += myObj.cars[i];  
}
```

6.5. Nested Arrays in JSON Objects

Values in an array can also be another array, or even another JSON object:

Example:

```
myObj = {  
    "name": "John",  
    "age": 30,  
    "cars": [  
        { "name": "Ford", "models": [ "Fiesta", "Focus", "Mustang" ] },  
        { "name": "BMW", "models": [ "320", "X3", "X5" ] },  
        { "name": "Fiat", "models": [ "500", "Panda" ] }  
    ]  
}
```

To access arrays inside arrays, use a for-in loop for each array:

Example:

```
for (i in myObj.cars) {  
  x += "<h1>" + myObj.cars[i].name + "</h1>";  
  for (j in myObj.cars[i].models) {  
    x += myObj.cars[i].models[j];  
  }  
}
```

6.6. Modify Array Values

Use the index number to modify an array:

Example:

```
myObj.cars[1] = "Mercedes";
```

6.7. Delete Array Items

Use the delete keyword to delete items from an array:

Example:

```
delete myObj.cars[1];
```