

JSON—Javascript object notation

Any string which looks similar to javascript object is called as JSON

To convert Javascript object into JSON format use `JSON.stringify(ob)`

To convert JSON into javascript object use `JSON.parse(ob)`

Promises in javascript

Promise Object Properties

It's an object which has 2 functions, `resolve` and `reject`.

A JavaScript Promise object can be in state:

Pending

Fulfilled

Rejected

The Promise object supports two properties: `state` and `result`.

While a Promise object is "pending" (working), the result is undefined.

When a Promise object is "fulfilled", the result is a value.

When a Promise object is "rejected", the result is an error object.

<code>myPromise.state</code>	<code>myPromise.result</code>
------------------------------	-------------------------------

"pending"	undefined
-----------	-----------

"fulfilled"	a result value
-------------	----------------

"rejected"	an error object
------------	-----------------

You cannot access the Promise properties `state` and `result`.

You must use a Promise method to handle promises.

```
myPromise.then(  
  function(value) { /* code if successful */ },  
  function(error) { /* code if some error */ }  
);
```

`Promise.then()` takes two arguments, a callback for success and another for failure.

Both are optional, so you can add a callback for success or failure only.

Example

```
function myDisplayer(some) {
```

```
document.getElementById("demo").innerHTML = some;  
}
```

```
let myPromise = new Promise(function(myResolve, myReject) {  
  let x = 0;
```

```
// The producing code (this may take some time)
```

```
  if (x == 0) {  
    myResolve("OK");  
  } else {  
    myReject("Error");  
  }  
});
```

```
myPromise.then(  
  function(value) {myDisplayer(value);},  
  function(error) {myDisplayer(error);}  
);
```