**1. JSON string to object**

const json = '{"result":true, "count":42}';

const obj = JSON.parse(json);

* json is a **string** that represents a valid JSON object.
* {"result": true, "count": 42}
* JSON.parse(json) converts the string into a **JavaScript object**:
* { result: true, count: 42 }

**2. Console Output**

console.log(json);

console.log(obj);

**Output:**

{"result":true, "count":42}

{ result: true, count: 42 }

* The first console.log(json) prints the raw JSON string.
* The second console.log(obj) prints the parsed JavaScript object.

**3. Converting a JS object to a JSON string**

console.log(JSON.stringify({ x: 5, y: 6 }));

* { x: 5, y: 6 } is a simple JS object.
* JSON.stringify() converts it to:
* {"x":5,"y":6}

**4. Stringifying boxed primitives (objects created with new)**

console.log(JSON.stringify([new Number(3), new String('false'), new Boolean(false)]));

**Important point:**

* JavaScript has **primitive types** and **boxed (wrapper) objects**:
  + 3 is a number (primitive)
  + new Number(3) is an object wrapping the number
  + Same for new String('false') and new Boolean(false)

**Behavior of JSON.stringify() with boxed primitives:**

* It **automatically converts** boxed objects to their **primitive values**.

So the array:

[ new Number(3), new String('false'), new Boolean(false) ]

Gets stringified to:

[3, "false", false]

**5. What gets ignored during stringification**

console.log(JSON.stringify({ x: [10, undefined, function(){}, Symbol('')] }));

This is very important and subtle.

* JSON supports only the following data types: **string, number, object, array, true, false, null**
* Unsupported types like:
  + undefined
  + functions
  + symbols  
    …are **ignored** during JSON.stringify().

In this object:

{ x: [10, undefined, function(){}, Symbol('')] }

* x is an array.
* That array contains:
  1. 10 → valid
  2. undefined → ignored (replaced with null inside arrays)
  3. function(){} → ignored (replaced with null inside arrays)
  4. Symbol('') → ignored (replaced with null inside arrays)

So JSON.stringify() turns it into:

{"x":[10,null,null,null]}

Note: In **objects**, undefined, functions, and symbols are removed completely. In **arrays**, they are replaced with null.

**✅ Summary**

| **Code Expression** | **Output** | **Explanation** |
| --- | --- | --- |
| JSON.parse('{"result":true, "count":42}') | { result: true, count: 42 } | Converts JSON string to JS object |
| JSON.stringify({ x: 5, y: 6 }) | '{"x":5,"y":6}' | Converts JS object to JSON string |
| JSON.stringify([new Number(3), new String('false'), new Boolean(false)]) | '[3,"false",false]' | Boxed primitives converted to their values |
| JSON.stringify({ x: [10, undefined, function(){}, Symbol('')] }) | '{"x":[10,null,null,null]}' | Invalid values replaced with null in arrays |