

Javascript

# console API



Omkesh B. kendre



uiacademy.in

## Javascript

*console.assert(assertion)*

**console.assert()** logs errors if false;  
silent if true.

```
> const errorMsg = "the # is not even";
  for (let number = 0; number <= 1; number++) {
    console.log(`the # is ${number}`);
    console.assert(number % 2 === 0, "%o",
      { number, errorMsg });
  }
the # is 0
the # is 1
✖ ▶ Assertion failed: ▶ {number: 1, errorMsg: 'the # is not even'} i
          errorMsg: "the # is not even"
          number: 1
          ▶ [[Prototype]]: Object
< undefined
>
```



## Javascript

*console.clear()*

**console.clear()** clears browser console;  
terminal console has no effect.

*Console was cleared*

---

↳ undefined



## Javascript

*console.count()*

**console.count()** logs call count;  
tracks function calls.

```
> for (let num = 0; num <= 1; num++) {  
  console.count();  
}
```

```
default: 1
```

```
default: 2
```



## Javascript

### *console.debug()*

**console.debug()** outputs message at debug log level if configured.

```
> const variable = "example";
  console.debug("Debugging variable:", variable);
Debugging variable: example
```



## Javascript

### *console.dir()*

**console.dir()** displays interactive list  
of object properties.

```
> const myObject = { name: "John", age: 30, city: "New York" };
  console.dir(myObject);
  ▼ Object [i]
    age: 30
    city: "New York"
    name: "John"
  ► [[Prototype]]: Object
```



## Javascript

*console.dirxml()*

**console.dirxml()** displays an XML/HTML-like tree structure.

```
> const htmlElement = document.querySelector('body');
  console.dirxml(htmlElement);

  ▼<body data-new-gr-c-s-check-loaded="14.1145.0" data-gr-ext-:
    ►<script>...</script>
    ►<div id="root">...</div>
    ►<script type="application/json" id="hydration">...</script>
      <div id="a11y-status-message" role="status" aria-live="pol
        ns text" style="border: 0px; clip: rect(0px, 0px, 0px, 0px);
          x; overflow: hidden; padding: 0px; position: absolute; wid
        </body>
```



## Javascript

*console.error()*

**console.error()** logs error  
message to the console.

```
> console.error("This is an error message.");
```

```
✖ ▶ This is an error message.
```



## Javascript

### *console.group()*

**console.group()** groups messages in an expandable inline block.

```
> console.group("Group Example");
  console.log("Message 1");
  console.log("Message 2");
  console.groupEnd();
```

#### ▼ Group Example

Message 1

Message 2



## Javascript

### *console.table()*

**console.table()** displays data  
in tabular form.

```
> console.table(["apples", "oranges", "bananas"]);
```

(index)	Value
0	'apples'
1	'oranges'
2	'bananas'



## Javascript

### *console.trace()*

**console.trace()** logs stack trace  
to the console.

```
> function thirdFunction() {  
    console.trace();  
}  
  
function secondFunction() {  
    thirdFunction();  
}  
  
function firstFunction() {  
    secondFunction();  
}  
  
firstFunction();  
▼ console.trace  
  thirdFunction @ VM29:2  
  secondFunction @ VM29:6  
  firstFunction @ VM29:10  
  (anonymous) @ VM29:13
```



## Javascript

*console.warn()*

**console.warn()** logs warning message to the console.

```
> console.warn("This is a warning message.");
```

 ▶ This is a warning message.

# Did you like the post? follow for more!

---



Omkesh B. Kendre



uiacademy.in

