

JavaScript: Reading, Changing Text and Style

A short DOM tutorial for beginners (v4.1)

1. The basic idea

JavaScript can interact with a webpage. It can read information from HTML elements and it can write or change information on the page.

```
<p id="message">Hello!</p>

<script>
let element = document.getElementById("message");
</script>
```

The variable `element` now refers to the HTML element with the id `message`. Once an element is selected, JavaScript can read from it or change it.

2. Reading and writing data

The property you use depends on the type of HTML element.

Element	Read	Write / change
<input>, <output>, ...	<code>.value</code>	<code>.value = "..."</code>
<input>, <output>, ...	<code>.value</code>	<code>.value = "..."</code>
<p>, <div>, , ... (text only)	<code>.textContent</code>	<code>.textContent = "..."</code>
<p>, <div>, , ... (html content)	<code>.innerHTML</code>	<code>.innerHTML = "..."</code>

Remember: Use `.value` for input/output elements. Other form controls such as `<select>` and `<textarea>` also have a value.

Normal text elements such as `<p>`, `<div>` and `<span>` use `.textContent`.

Reading from an input

```
let name = document.getElementById("name").value;
```

Writing to an output

```
document.getElementById("result").value = "Hello";
```

Reading or changing normal text

```
let text = document.getElementById("message").textContent
document.getElementById("message").textContent = "Welcome!" . text;
```

3. Changing text

Use `textContent` when you want to work with normal text:

```
document.getElementById("message").textContent = "Welcome!";
```

Use `innerHTML` only when you deliberately want to insert HTML tags:

```
document.getElementById("message").innerHTML = "<b>Welcome!</b>";
```

4. Changing CSS styles

Use `.style` followed by the CSS property you want to change.

```
let box = document.getElementById("box");

box.style.backgroundColor = "lightblue"; // css: background-color
box.style.color = "navy";
box.style.fontSize = "24px"; // css: font-size
box.style.border = "2px solid black";
```

CSS properties containing a hyphen are written in camelCase in JavaScript:

background-color	→	backgroundColor
font-size	→	fontSize
border-radius	→	borderRadius

5. Reminder: functions and parameters

Functions are not a special part of the DOM. You already know functions from earlier JavaScript work. They are included here as a useful reminder because buttons often call functions.

Example with one parameter

```
<button onclick="setColor('red')">Red</button>

<script>
function setColor(color) {
    document.getElementById("message").style.color = color;
}
</script>
```

The value 'red' is passed to the function and stored in the parameter color.

Example with two parameters

```
<button onclick="changeMessage('Warning!', 'red')">Warning</button>

<script>
function changeMessage(text, color) {
    let message = document.getElementById("message");
    message.textContent = text;
    message.style.color = color;
}
</script>
```

The first parameter changes the text; the second parameter changes the color.

6. Block and inline elements

HTML elements are normally displayed either as blocks or inline elements.

Type	Examples	Behaviour
Block	<div>, <p>, <h1>	Usually starts on a new line and takes available width.
Inline	, 	Stays inside the current line of text.

Block and inline elements behave differently: **block elements can normally use properties such as width, height, margin, and padding, while inline elements mainly follow the flow of the text and do not react to width and height in the same way.**

JavaScript can also change the display type. The most usefull showing or hiding an element:

```
element.style.display = "none";    // hide
element.style.display = "block";   // show as a block
element.style.display = "inline";  // show inline
```

7. Other useful changes

Change an image

```
document.getElementById("photo").src = "photo2.jpg";
```

Change a link

```
document.getElementById("link").href = "https://example.com";
```

Add or remove a CSS class

```
document.getElementById("box").classList.add("active");
document.getElementById("box").classList.remove("active");
document.getElementById("box").classList.toggle("active");
```

8. Quick reference

The examples below assume that an element has already been selected:

```
let el = document.getElementById("box");
```

Content and values

What you want to do	JavaScript	Example
Read input/output value	el.value	let x = el.value;
Set input/output value	el.value = ...	el.value = "42";
Read normal text	el.textContent	let x = el.textContent;
Change normal text	el.textContent = ...	el.textContent = "Hello";
Change HTML content	el.innerHTML = ...	el.innerHTML = "Hello";

Colors and background

What you want to do	JavaScript	Example
Text color	<code>el.style.color</code>	<code>el.style.color = "red";</code>
Background color	<code>el.style.backgroundColor</code>	<code>el.style.backgroundColor = "yellow";</code>
Background image	<code>el.style.backgroundImage</code>	<code>el.style.backgroundImage = 'url("bg.jpg")';</code>

Text and fonts

What you want to do	JavaScript	Example
Font size	<code>el.style.fontSize</code>	<code>el.style.fontSize = "24px";</code>
Font weight	<code>el.style.fontWeight</code>	<code>el.style.fontWeight = "bold";</code>
Font family	<code>el.style.fontFamily</code>	<code>el.style.fontFamily = "Arial";</code>
Text alignment	<code>el.style.textAlign</code>	<code>el.style.textAlign = "center";</code>
Text decoration	<code>el.style.textDecoration</code>	<code>el.style.textDecoration = "underline";</code>

Size and spacing

What you want to do	JavaScript	Example
Width	<code>el.style.width</code>	<code>el.style.width = "200px";</code>
Height	<code>el.style.height</code>	<code>el.style.height = "100px";</code>
Padding	<code>el.style.padding</code>	<code>el.style.padding = "20px";</code>
Margin	<code>el.style.margin</code>	<code>el.style.margin = "10px";</code>

Borders and shape

What you want to do	JavaScript	Example
Border	<code>el.style.border</code>	<code>el.style.border = "2px solid black";</code>
Rounded corners	<code>el.style.borderRadius</code>	<code>el.style.borderRadius = "15px";</code>

Visibility and display

What you want to do	JavaScript	Example
Hide element	<code>el.style.display</code>	<code>el.style.display = "none";</code>
Show as block	<code>el.style.display</code>	<code>el.style.display = "block";</code>
Show inline	<code>el.style.display</code>	<code>el.style.display = "inline";</code>
Opacity	<code>el.style.opacity</code>	<code>el.style.opacity = "0.5";</code>

Images, links and classes

What you want to do	JavaScript	Example
Change image	<code>el.src</code>	<code>el.src = "photo.jpg";</code>
Change link	<code>el.href</code>	<code>el.href = "https://example.com";</code>
Change alternative text	<code>el.alt</code>	<code>el.alt = "A photo";</code>
Add CSS class	<code>el.classList.add()</code>	<code>el.classList.add("active");</code>
Remove CSS class	<code>el.classList.remove()</code>	<code>el.classList.remove("active");</code>
Toggle CSS class	<code>el.classList.toggle()</code>	<code>el.classList.toggle("active");</code>

9. Remember

- JavaScript can read information from the page and write/change information on the page.
- Select an element with `document.getElementById("id")`.
- Use `.value` for input/output values.
- Use `.textContent` for normal text.
- Change styles with `.style.property = "property"`.
- Functions are normal JavaScript; they are simply useful for reacting to buttons and reusing code.
- Block elements normally start on a new line; inline elements stay in the current line.