

JavaScript im Browser - CheatSheet

Um den Benutzer eine Meldung zu signalisieren oder einfach nur Text ausgeben zu lassen, gibt es

```
alert(<Text>);
```

Mit JavaScript ist es möglich, eine Webseite dynamisch zu verändern. Der Einstiegspunkt dazu ist das `document` Objekt. Das liefert Funktionen, um existierende HTML-Elemente zu suchen oder neue hinzuzufügen

```
document.querySelector(<Such-Wert>)
```

- Gibt das erste Element zurück, das `<Such-Wert>` entspricht
- `<Such-Wert>` kann ein Tag-Name, eine Class oder eine Id sein

```
document.createElement(<Tag>)
```

- Erstellt ein neues Element vom Type `<Tag>` und gibt es zurück

Auf konkretes Element kann man Funktionen anwenden, die das Element verändern.

```
element.innerHTML = <Wert>
```

- Das setzt den Wert zwischen einem öffnenden und schließenden Tag
- Beispiel:
`const element = <h1>;`
`element.innerHTML = "Überschrift";`
ist wie `<h1>Überschrift</h1>`

```
element.appendChild(<anderes Element>)
```

- Fügt ein neues Element am vor dem schließenden Tag von `element` ein
- Beispiel:
`const element = ;`
`const child = <p>;`
`element.appendChild(child);`
ist wie `<p></p>`

```
element.setAttribute(<Attribut>, <Wert>)
```

- Setzt ein `<Attribut>` von `element` auf den Wert `<Wert>`
- Beispiel:
`const element = <h1>;`
`element.setAttribute("class", "border");`
ist wie `<h1 class="border"></h1>`

Manche HTML-Elemente ermöglichen das Registrieren von Event-Handlern. Ein Event-Handler ist Code, der genau dann ausgeführt werden soll, wenn ein bestimmtes Ereignis eintritt.

```
element.addEventListener("<Ereignis>", () => {  
    <Aktion>  
});
```

Das kann z. B. das Klicken eines Buttons sein.

```
button.addEventListener("click", () => {  
    alert("Der Button wurde geklickt");  
});
```