

Wenn man einen Element-Selektor z. B. `document.getElementById()` öfters benötigt, kann man ihn als Objekt vereinbaren. Damit erspart man sich viel Schreibarbeit und der Code wird übersichtlicher.

JS

```
var name = document.getElementById();
```

```
<h1>Österreich-Quiz</h1>
<p>Wie heißt die Hauptstadt von Österreich?
<input type="text" id="Hauptstadt" ></p>
<p><button onClick="aufloesen();">Auflösen</button></p>

<script>
function aufloesen() {
    var antwort = document.getElementById("Hauptstadt");

    if (antwort.value == "Wien") {
        antwort.style.backgroundColor = 'lime';
        antwort.style.border = '3px solid green';}
    else {antwort.style.backgroundColor = 'red';
        antwort.style.color = 'white';
        antwort.style.textDecoration = 'line-through';}}
</script>
```



Das vereinbarte Objekt funktioniert nicht nur mit `.style` sondern auch mit allen anderen JS Befehlen, die auf das Element zugreifen.
Im Beispiel oben: `antwort.value`



Ein im HTML Element gesetztes Eventattribut mit dem Wert `"(this)"` (z. B. `<input onClick="this" id="dasHier" >`) übergibt das Element als Objekt an eine JavaScript-Funktion (z. B. `function anzahl(eingabe);`)



Im Beispiel wird eine Funktion ausgeführt, sobald der Fokus (`onFocus`) auf das `<input>` Element gesetzt wird. Wenn man das Input-Feld verlässt (`onBlur`) wird eine andere Funktion ausgeführt. Es ändert sich jedes Mal der Rahmen.

Die JS Funktionen `hervor()`; `raus()`; übernehmen den Selektor vom HTML Element und greifen auch wieder auf dieses zu.

```
<input type="text" id="ant1" onFocus="hervor(this)"
        onBlur="raus(this)" placeholder="Vorname" ><br>

<input type="text" id="ant2" onFocus="hervor(this)"
        onBlur="raus(this)" placeholder="Nachname" ><br>

<input type="text" id="ant4" onFocus="hervor(this)"
        onBlur="raus(this)" placeholder="eMail" ><br>

<script>
function hervor(eingabefeld) {
    eingabefeld.style.border = '3px solid blue';}

function raus(dasFeld) {
    dasFeld.style.border = '1px solid gray';}
</script>
```