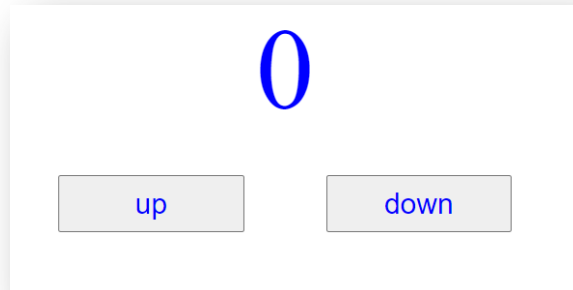


L1_1 Beispiel „Zähler“



Teil 1: HTML-Code / CSS-Regeln

1. Erstellen Sie den HTML-Code zur Darstellung des Zählers und der Schaltflächen.

```
<div>0</div>

<input type="button" value="up">
<input type="button" value="down">
```

2. Erstellen Sie CSS-Regeln.

```
body {
    text-align: center;
}
```

Notwendige Ergänzung in HTML-Seite

```
.zaehler {
    color: blue;
    font-size: 96px;
}

input {
    color: blue;
    font-size: 24px;
    width: 30%;
    padding: 8px;
    margin: 32px;
}
```

→ `<div class="zaehler">0</div>`

Teil 2: Ereignisbehandlung

Klickt man auf die Schaltflächen, geschieht nichts. Wir wollen nun auf das Klick-Ereignis reagieren und eine Javascript-Funktion aufrufen.

1. Schaltflächen um einen Eventhandler erweitern:

```
<input onclick="upwards()" type="button" value="up">
```

Ein Klick auf die mit „up“ beschriftete Schaltfläche ruft die Javascript-Funktion „**upwards()**“ auf. Die Funktion muss im nächsten Schritt erstellt werden.

Hinweis: Ein Überblick weiterer Ereignistypen finden Sie unter https://www.w3schools.com/jsref/dom_obj_event.asp.

2. Funktionen erstellen

Zur Implementierung des Javascript-Codes muss in den Script-Bereich gewechselt werden.

Das Grundgerüst einer Funktion in Javascript folgt folgender Syntax:

```
function funktionsname() {  
  
}
```

Die Funktion „**upwards()**“ soll folgende Logik implementieren:

Funktion: upwards
Initialisierung: zahl = ZahlAusSeiteErmitteln
Zuweisung: zahl=zahl+1
Ausgabe: NeuerZaehlerwert

```
function upwards() {  
  //a) aktueller Zahlenwert ermitteln  
  zahl = ???  
  //b) Zahlenwert um eins erhöhen  
  zahl=zahl+1;  
  //c) neuen Zahlenwert setzen  
  ??? = zahl;  
}
```

- a) Um den aktuellen Zahlenwert zu ermitteln müssen Ergänzungen im HTML-Code durchgeführt werden. Ein HTML-Element lässt sich mit Javascript am einfachsten ansprechen, wenn dies ein id-Attribut besitzt. Ergänzen wir das Element, welches die Zahl darstellt mit der ID „z“.

```
<div id="z" class="zaehler">0</div>
```

Mit der Anweisung `document.getElementById("z")` können wir das Element-Objekt zu dem HTML-Element ermitteln. Wir können nun alle Eigenschaften und Methoden dieses Objektes nutzen, um das zugehörige HTML-Element anzusprechen bzw. zu manipulieren. Eine Auflistung der Eigenschaften und Methoden finden Sie unter https://www.w3schools.com/jsref/dom_obj_all.asp. Die Eigenschaft „`innerText`“ liefert uns den Inhalt des Elementes. Im obigen Beispiel wäre dies die 0. Konkret sieht dies so aus:

```
element = document.getElementById("z")
elementinhalt= element.innerText
```

Da die Eigenschaft „`innerText`“ immer eine Zeichenkette liefert, müssen wir den so ermittelten Zahlenwert noch in eine Zahl konvertieren. Dazu steht uns die Funktion „`parseInt(...)`“ zur Verfügung (https://www.w3schools.com/jsref/jsref_parseint.asp).

```
zahl = parseInt(elementinhalt)
```

- b) Die so ermittelte Zahl kann nun durch einfache Addition einer 1 erhöht werden:

```
zahl = zahl+1
```

- c) Neuen Zahlenwert setzen

Hier greifen wir erneut auf das Element-Objekt zu, welches dem HTML-Element zugeordnet ist, das den Zahlenwert umschließt. Das Objekt haben wir bereits in Schritt a) ermittelt und in der Variablen `element` vorgehalten. Um den Inhalt zu ändern, können greifen wir nun schreibend auf die Eigenschaft `innerText` zu und weisen den neuen Zahlenwert zu:

```
element.innerText= zahl;
```

Die Realisierung der Schaltfläche zur Verminderung des Wertes erfolgt analog!

	Dynamische Webseiten mit Javascript Informationsmaterial	
--	--	--

Erweiterungen:

1. Erweitern Sie die Webanwendung so, dass der Zähler nur Zahlen zwischen -10 und +10 annehmen kann.
2. Erweitern Sie die Webanwendung so, dass der Zähler mit einem roten Hintergrund hinterlegt ist, wenn sich der Zahlenwert im negativen Bereich befindet und grün, wenn es sich um eine positive Zahl handelt. Bei einem Zahlenwert von 0 erfolgt keine Einfärbung.