











Web Apps

Javascript –
Verknüpfung mit HTML

JS Extensions

INSTALLED 28

-  **[Deprecated] Debugger for Chrome**
Debug your JavaScript code in the Chrome bro...
Microsoft
-  **Better Comments**
Improve your code commenting by annotating ...
Aaron Bond
-  **Bookmarks** 288ms
Mark lines and jump to them
Alessandro Fragnani
-  **Cobalt2 Theme Official**
Official theme by Wes Bos.
Wes Bos
-  **colorize** 12606ms
A vscode extension to help visualize css colors i...
kamikillerto

-  **ESLint** 348ms
Integrates ESLint JavaScript into VS Code.
Microsoft
-  **Fold Plus**
Additional fold commands
dakara
-  **indent-rainbow** 505ms
Makes indentation easier to read
oderwat
-  **JS-CSS-HTML Formatter**
Format ,prettify and beautify JS, CSS, HTML cod...
lonify
-  **Live Server** 94214ms
Launch a development local Server with live relo...
Ritwick Dev Reload Required

Position von Javascript

Inline(pro
Tag)

In Header

Extern in
JS-Datei

Syntax vom Aufruf

Element

Ereigniss

<element event='some JavaScript'>

<element event="some JavaScript">

Einlesen und Auslesen von HTML-Informationen

A simple calculator

Add two numbers

Zahl 1:

Zahl 2:

index.js

```
function calcSum() {  
    number1 = document.getElementById("zahl1").value  
    number2 = document.getElementById("zahl2").value  
    sum = Number(number1) + Number(number2);  
    document.getElementById("summe").innerHTML = 'The sum is: ' + sum;  
}
```

Eigenschaften für den Inhalt des Objektes

In <body> von index.html

```
<h2>A simple calculator</h2>  
<p>Add two numbers</p>  
<label for="zahl1">Zahl 1: </label>  
<input id="zahl1" type="text" size="12" value="1" />  
<p></p>  
<label for="zahl2">Zahl 2: </label>  
<input id="zahl2" type="text" size="12" value="2" />  
<p>  
<button onclick="calcSum()">Calculate sum</button>  
<p id="summe"></p>
```

Identifikation des Objektes mit „id“

Funktion, welche beim Anklicken

ausgelöst wird

Debuggen von JS - Generelles

- ✓ „debugger“ als Zeile im Quelltext
- ✓ `alert(„Test“);` in JS-Code als 1. Zeile oder in der Funktion
- ✓ Chrome DevTools mit F12 öffnen

Debuggen von JS - DevTools

Haltepunkt setzen/löschen

The screenshot shows the Chrome DevTools interface. The 'Sources' panel on the left displays a JavaScript function `calcSum()` with a red dot indicating a breakpoint at line 5. The 'Debugger' panel on the right shows the state of the program at the breakpoint. It includes a 'Watchlist' with variables `number1` and `number2`, a 'Breakpoints' list with the active breakpoint at `index.js:5`, a 'Scope' section showing local variables like `this` and `Global`, and a 'Callstack' showing the current function `calcSum` and the event `onclick`.

Watchlist

Haltepunkte

Lokale Variablen

Aufrufstack

Debuggen von JS – DevTools II

Ausführung fortsetzen

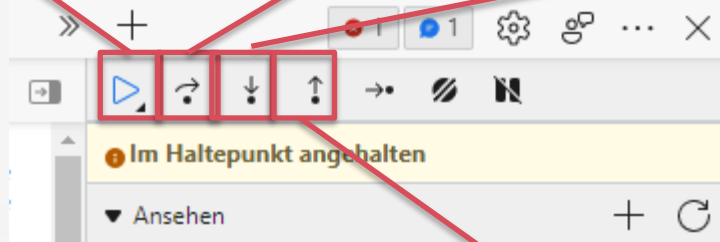
F8

Step over next function call

F10

Step into next function call

F11



Step out of current function
Shift+F11

Multicursor

Action	Key(s)
Insert cursor	ALT + Click
Insert cursor above / below	CTRL + ALT + ↑ / ↓
Undo last cursor operation	CTRL + U
Insert cursor at end of each line selected	SHIFT + ALT + I
Select current line	CTRL + L
Select all occurrences of current selection	CTRL + SHIFT + L
Select all occurrences of current word	CTRL + F2
Expand selection	SHIFT + ALT + →
Shrink selection	SHIFT + ALT + ←
Column (box) selection	SHIFT + ALT + (drag mouse)
Column (box) selection	CTRL + SHIFT + ALT + (arrow key)
Column (box) selection page up/down	CTRL + SHIFT + ALT + PgUp / PgDn