

Codieren von Algorithmen mit JavaScript und HTML

HTML Grundgerüst

Dokumententyp.

- html
- xml
- pdf
- ...

```
1 <!DOCTYPE html>
2
3 <html lang="de">
4   <head>
5
6   </head>
7
8   <body>
9
10
11 </body>
12
13
14 </html>
```

Zuweisung der Sprache

Im Kopfbereich meta Angaben

Im Rumpf der Inhalt der Seite

Das HTML Dokument

Der Kopfbereich



```
5 <head>
6
7   <title>Der Titel des Dokuments</title>
8   <meta charset="utf-8">
9   <script src="JavaScriptDokument.js" defer></script>
10  [ <link rel="stylesheet" href="StylesheetDokument.css"> ]
11
12 </head>
```

Utf-8 ist der westeuropäische Zeichensatz

Optional. Gestaltung der Elemente (Farbe, Schrift, ..)

Dateiname mit relative Pfadangabe. Z. B.:

- berechnung.js liegt im gleichen Ordner wie die HTML Datei
- ../berechnung.js liegt einen Ordner oberhalb der HTML Datei
- ../../berechnung.js zwei Ordner oberhalb
- javascript/berechnung.js liegt im Untersordner von der HTML Datei
- ../javascript/berechnung.js liegt von der HTML Datei aus einen Ordner höher und dort in einem Untersordner

```
<script src="JavaScriptDokument.js" defer></script>
```

defer: JS Dokument wird versetzt, nach dem lesen der HTML Datei vom Browser gelesen.

Die Elemente im Rumpf zur Textstrukturierung

```
<h3>Berechnung des Sparguthabens</h3>
```

Überschriften der Kategorie 1 bis 6 entsprechend **<h1>** bis **<h5>**

```
<p>
  Hier steht text innerhalb eines Absatzes.
</p>
```

p paragraph gleich Absatz

Dieser Text steht in der selben Zielwert

Ein Zeilenumbruch im Texteditor bewirkt noch keine neue Zeile. Möchte man eine neue Zeile so muss dies explizit mit **
** angegeben werden.

Dieses Tag ermöglicht **
** die Anzeige eines Texts in einer neuen Zeile.

Seltener:

```
<ul>
  <li>Listenelement</li>
  <li>Listenelement</li>
</ul>

<ol>
  <li>1. Listenelement</li>
  <li>2. Listenelement</li>
</ol>
```

Listen bzw. Aufzählungen als **** unsortierte Liste oder als **** ordered list in der Form:

1. Erstens
2. Zweitens
3. ...

Wird häufig gefragt:

Hyperlink mit dem Attribut **target="blank"**, das Ziel wird in einem neuen Tab geöffnet

```
<a href="http://wiki.selfhtml.org/wiki/html/" target="blank"></a>
```

```

```

Bild mit relativer Pfadangabe

```
<table>
  <thead>
    <tr>
      <td></td>
      <td></td>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td></td>
      <td></td>
    </tr>
  </tbody>
</table>
```

Eine Tabelle aus 2 Zeilen **<tr>** table row und jeweils 2 Zellen **<td>** table data ergibt letztendlich 2 Spalten.

Die Elemente im Rumpf zur Ein- und Ausgabe

```
<input type="text">  
<input type="number">  
<input type="checkbox">
```

Eingabefelder. (3 Beispiele)

```
<input type="button" value="Drücke mich!">  
<button>Drücke mich!</button>
```

Button – Knopf, um eine Aktion auszulösen

```
<output></output>
```

Ausgabefeld *)

*) Ausgabefelder können an sich alle HTML Elemente sein. Man kann auch ein `<p>` oder z. B ein `<h6>` zur Ausgabe benutzen. JavaScript kann sogar jedes Element neu erzeugen. Zum Beispiel wenn man eine größere Datenmenge berechnet, so könnte man diese in einer Tabelle ausgeben, die man erst passend zu dem Zeitpunkt generiert, wenn das Ergebnis feststeht.

Genauer – die **id**:

```
<input type="number" id="eingabe_1">  
<input type="number" id="eingabe_2">  
  
<button id="knopf"></button>  
  
<output id="ausgabe"></output>
```

Jedes HTML Element kann eine `id` haben.
Alle Elemente, die vom JavaScript aus benutzt werden sollen benötigen diese `id`.

Einige typischen Attribute für Eingabefelder:

Eingabefeld für Zahlen

Über die `id` auf das Element zugreifen

```
<input type="number" id="eingabe" value="100" step="0.5">
```

Vorbelegung mit einem Wert

Schrittweite für den Drehregler

Typische Objekte und Methoden in JavaScript

`document.getElementById("element")` Über den Namen der **ID** erhält man Zugriff auf das benötigte Element.

```
document.getElementById("element").value ;
```

Mit **value** bekommt man den Wert aus dem Eingabefeld. Der Wert ist ohne weitere Angabe eine Zeichenkette, string

```
document.getElementById("element").valueAsNumber ;
```

Benötigt man statt Zeichen eine Zahl, so wäre dies eine Möglichkeit.

```
var zinsatz = document.getElementById("zinsatz").valueAsNumber;
```

Hier wird ein Wert aus einem Eingabefeld einer Variablen übergeben. Bei längeren Programmen ist es sinnvoll den Variablen sinnvolle Namen zu geben.

Eingabe lesen

```
document.getElementById("ergebnis").value = ergebnis;
```

Oder ein Ergebnis wird ausgegeben.

Ausgabe

```
document.getElementById("element").addEventListener() ;
```

Ein Element lässt sich mit einer **Aktion** ausstatten. Typischerweise ist das der Button.

```
document.getElementById("element").addEventListener("click", funktionsname) ;
```

Der Button reagiert auf einem Mausklick *), einfach "**click**". Der 2. Parameter gibt an, was nach dem click gemacht werden soll: eine Funktion mit diesem Namen wird aufgerufen.

*) Nicht nur Buttons können Aktionen ausführen, im Prinzip alle Elemente. Und statt einem click gibt es viele andere Events, z. B.: "dblclick", "keypress", "keydown", "mousedown", "mouseleave",

Programmierwerkzeuge

Funktionen

- Funktionen kapseln **kleine logische Einheiten**.

```
function berechne() {  
}
```

- Sie haben einem selbst gewählten, möglichst bezeichnenden Namen.
- Sie werden aufgerufen.

- Ihnen **können Werte übergeben werden**.

```
function berechne(kapital, zinssatz) {  
    kapital = kapital + kapital * zinssatz/100;  
    return kapital;  
}
```

```
function berechne(kapital, zinssatz) {  
    kapital = kapital + kapital * zinssatz/100;  
    document.getElementById("ausgabe").value = kapital;  
}
```

- Sie manipulieren etwas **oder geben etwas zurück**.

- Ein gleichwertiger Begriff für Funktion ist Methode.
- Man kann sie für sich hernehmen und zu einem größerem zusammenfassen.

Variable

- Variablen sind **Platzhalter für Werte**.
- Die Werte können sein: **Zahlen** (ganze Zahlen, Dezimalzahlen, ...), **Zeichenketten**, Wahrheitswerte (boolean), Arrays, Objekte.
- Ihnen wird typischerweise eine Speicheradresse zugeordnet.
- Variablenamen sollten selbstbezeichnet sein und müssen aus Zeichen oder Zeichenketten bestehen und dürfen nicht nur aus Zahlen bestehen wie `var 12;`

```
var zinssatz;  
var zinssatz = 3.1;  
var zinssatz = document.getElementById("zinssatz").valueAsNumber;  
var kapital = berechne(zinssatz, kapital);  
var wort_1 = "Hello";  
var wort_2 = "Welt!";  
document.getElementById("ausgabe").value = wort_1 + wort_2;  
var zahl_1 = 6;  
var zahl_2 = 7;  
document.getElementById("ausgabe").value = zahl_1 + zahl_2;  
var richtig = true;  
var falsch = false;  
var unter16 = document.getElementById("checkbox_unter_16").checked;
```

Grundsätzliches zur Syntax u. Schreibweise

- Jeder Befehl wird mit einem Semikolon/Strichpunkt **;** abgeschlossen.
- Alles, was zu einer Funktion oder zu einer Bedingung gehört befindet sich innerhalb der geschweiften Klammern **{ ... }**.
- Das Programm sollte **übersichtlich, verständlich** und **leicht** zu lesen sein. Die Namen sollten **sinnvoll** sein.

Bedingungen

Die typische Bedingung: **wenn** <-> **ansonsten**, sind Verzweigungen im Programm.

In den Klammern hinter dem **if**, oder dem **else if** steht ein Wahrheitswert der **richtig (true)** oder **falsch (false)** sein kann; oder aber der Ausdruck ergibt eine eindeutig richtige oder falsche Überprüfung. Siehe Beispiele.

```
var unter16 = document.getElementById("checkbox_unter_16").checked;

if( unter16 ){
    true.etwas;
}
else {
    true.etwas.anderes;
}
```

```
var wert = 10;

if( wert < 30 ){
    true.etwas;
}
if( wert < 20 ){
    true.etwas.anderes;
}
```

Ein interessanter Unterschied zwischen mehreren *if* **oder** *else if*. Beides ist möglich. Was jedoch wird hier wohl abgearbeitet: das erste *etwas* oder das *andere*?

```
var wert = 10;

if( wert < 30 ){
    true.etwas;
}
else if( wert < 20 ){
    true.etwas.anderes;
}
```

Schleifen

Schleifen ermöglichen, eine Anweisung beliebig oft zu wiederholen.

Die **for**-Schleife

```
for( var i = 0; i < laufzeit; i++ ){  
    spareinlage = spareinlage + spareinlage * zinssatz/100;  
}
```

Bei der for-Schleife steht die Anzahl der Wiederholungen in der Angabe – den Klammern (): Variable i fängt bei 0 an, erhöht sich bei jedem Schleifendurchlauf um 1 (i++), und läuft solange i kleiner einem festen Wert ist.

Die **while**-Schleife

```
while( spareinlage < sparziel ){  
    spareinlage = spareinlage + spareinlage * zinssatz/100;  
}
```

Die Laufzeitbedingung ist bei der while-Schleife ähnlich wie bei den Bedingungen if angegeben. Solange diese Bedingung wahr/true ergibt, wird die Schleife ausgeführt. Wichtig ist, einen Wert in der Schleifenausführung so zu verändern, dass die Bedingung der while-Schleife irgendwann falsch/false ist, damit diese zum Ende kommt.

Merkregel für: wann **while** \leftrightarrow **for**?

- Ist bekannt, wie oft die Schleife wiederholt werden muss, nimmt man die **for**-Schleife.
- Weiß man nicht wie oft die Schleife ausgeführt wird, nimmt man die **while**-Schleife.

(Anmerkung: theoretisch wären die for-Schleifen durch while-Schleifen ersetzbar)

Berechnet das Quadrat einer Zahl

```
<!DOCTYPE html>
<html lang="de">
  <head>
    <meta charset="utf-8">
    <title>Das Quadrat einer Zahl</title>
    <script src="quadrat.js" defer></script>
  </head>
  <body>
    <h3>Berechnung der Quadratzahl</h3>
    <p>
      | Hier kannst du das Quadrat einer Zahl berechnen.
    </p>
    <input type="number" id="eingabe"> Zahl
    <br>
    <button id="berechne">Berechne</button>
    <br>
    Das Quadrat lautet:
    <output id="ausgabe"></output>
  </body>
</html>
```

```
document.getElementById("berechne").addEventListener("click", berechne);
function berechne() {
  var eingabe = document.getElementById("eingabe").value;
  var ergebnis = eingabe * eingabe;
  document.getElementById("ausgabe").value = ergebnis;
}
```

Programmablauf

- ① Der Button reagiert auf einen **Mausklick** und ...
- ② ... führt die Funktion 'berechne() { ... }' aus.
- ③ Die Benutzereingabe wird in eine **Variable** gelesen.
- ④ Das Herzstück, der **Algorithmus**, die Berechnung.
Hier nur kurz die Multiplikation der Eingabezahl mit sich selbst (eben das Quadrat). Das Ergebnis wird in eine Variable geschrieben.
- ⑤ Das **Ergebnis** wird dem Benutzer **angezeigt**.

Rechnung

```
<!DOCTYPE html>

<html lang="de">

  <head>
    <meta charset="utf-8">
    <title>Rechnung</title>
    <script src="rechnung.js" defer</script>
  </head>

  <body>

    <h3>Erstellung einer Rechnung</h3>

    <p>
      | Berechnet den Rechnungswert nach Stundenaufwand und Stundensatz.
    </p>

    <br>
    <input type="number" id="stunden" value="10" step="0.5"> Stundenzahl
    <br>
    <input type="number" id="stundensatz" value="13" step="0.5"> Stundensatz
    <br>
    <input type="number" id="mwst" value="19"> MwSt Satz
    <br><br>

    <button id="berechne">Berechne</button>
    <br><br>

    Die Rechnungssumme lautet: <output id="ausgabe"></output>

  </body>

</html>
```

```
document.getElementById("berechne").addEventListener("click", berechne);
```

```
function berechne() {

  var stunden      = document.getElementById("stunden").value;
  var stundensatz  = document.getElementById("stundensatz").value;
  var mwst         = document.getElementById("mwst").value;

  var ergebnis = stunden * stundensatz * ( 1 + mwst/100 );

  document.getElementById("ausgabe").value = ergebnis;

}
```

Erstellung einer Rechnung

Berechnet den Rechnungswert nach Stundenaufwand und Stundensatz.

10	Stundenzahl
13	Stundensatz
19	MwSt Satz

Berechne

Die Rechnungssumme lautet:

BMI – Body Mass Index

```
<!DOCTYPE html>
<html lang="de">
  <head>
    <meta charset="utf-8">
    <title>BMI</title>
    <script src="bmi.js" defer</script>
  </head>
  <body>
    <h3>Berechnung des Body-Mass-Index</h3>
    <p>Gib Deine Körpergröße in Meter und Dein Gewicht in Kilogramm an:</p>
    <input type="number" id="gewicht" value="60" step="1"> Gewicht [kg].
    <br>
    <input type="number" id="groesse" value="1.7" step="0.1"> Größe [m].
    <br>
    Unter 16 Jahren: <input type="checkbox" id="alter">
    <br>
    <button id="berechne">Berechne</button>
    <br>
    Dein berechneter BMI lautet: <output id="bmi"></output>
    <br>
    Bemerkung: <output id="bemerkung"></output>
  </body>
</html>
```

Berechnung des Body-Mass-Index

Gib Deine Körpergröße in Meter und Dein Gewicht in Kilogramm an:

Gewicht [kg].

Größe [m].

Unter 16 Jahren:

Dein berechneter BMI lautet:

Bemerkung:

```
document.getElementById("berechne").addEventListener("click", berechneBMI);

function berechneBMI() {
  var groesse = document.getElementById("groesse").valueAsNumber;
  var gewicht = document.getElementById("gewicht").valueAsNumber;
  var bmi;

  bmi = gewicht / (groesse * groesse);

  document.getElementById("bmi").value = bmi.toFixed(2);

  bewertung(bmi);
}

function bewertung(bmi) {
  var bemerkung = "";
  var unter16 = document.getElementById("alter").checked;

  if( unter16 ){
    if( bmi < 17 ) {
      bemerkung = "Dieser Wert liegt unterhalb des Normgewichts.";
    }
    else if( bmi < 25 ) {
      bemerkung = "Der Wert liegt im Normbereich."
    }
    else {
      bemerkung = "Der Wert liegt im Bereich Übergewichtig."
    }
  }
  else {
    if( bmi < 18.5 ) { ...
    }
    else if( bmi < 25 ) { ...
    }
    else { ...
    }
  }

  document.getElementById("bemerkung").value = bemerkung;
}
```

Der kleine Gauß

```
<!DOCTYPE html>

<html lang="de">

  <head>
    <meta charset="utf-8">
    <title>Der Kleine Gauß </title>
    <script src="gauss.js" defer></script>
  </head>

  <body>
    <h3>Der Kleine Gauß</h3>
    <p>
      Es wird die Summe der ersten n natürlichen Zahlen berechnet.
    </p>
    <p>
      Formel:<i>  $0 + 1 + 2 + 3 + \dots + n = n(n+1)/2$ </i>
    </p>
    <input type="number" id="summeBis"> Summe bis
    <br>
    <button id="berechne">Berechne</button>
    <br>
    Die berechnete Summe lautet:
    <output id="ergebnis"></output>
  </body>
</html>
```

```
document.getElementById("berechne").addEventListener("click", berechne);

function berechne(){

  var summeBis = document.getElementById("summeBis").valueAsNumber;
  var ergebnis = 0;

  for(var i = 0; i <= summeBis; i++){
    ergebnis = ergebnis + i;
  }

  document.getElementById("ergebnis").value = ergebnis;
}
```

Der Kleine Gauß

Es wird die Summe der ersten n natürlichen Zahlen berechnet.

Formel: $0 + 1 + 2 + 3 + \dots + n = n(n+1)/2$

Summe bis

Die berechnete Summe lautet: 5050

Spareinlage mit Zinseszins

```
<!DOCTYPE html>

<html lang="de">

  <head>
    <title>Zinsberechnung</title>
    <meta charset="utf-8">
    <script src="zinsen.js" defer</script>
  </head>

  <h3>Berechnung des Guthabens mit Zinseszins</h3>

  <input type="number" id="kapital" value="1000" step="100"> Geldanlage
  <br>
  <input type="number" id="zinssatz" value="3" step="0.1"> Zinssatz
  <br>
  <input type="number" id="laufzeit" value="5"> Laufzeit
  <br><br>
  <button id="berechne"> Berechne</button>
  <br><br>
  Dein Guthaben beträgt nach dieser Zeit:
  <output id="ergebnis"></output>

</html>
```

```
document.getElementById("berechne").addEventListener("click", berechne);

function berechne(){

  var kapital = document.getElementById("kapital").valueAsNumber;
  var zinssatz = document.getElementById("zinssatz").valueAsNumber;
  var laufzeit = document.getElementById("laufzeit").valueAsNumber;

  for( var i = 0; i < laufzeit; i++){
    kapital = kapital + kapital * zinssatz / 100;
  }

  document.getElementById("ergebnis").value = kapital.toFixed(2);
}
```

Berechnung des Guthabens mit Zinseszins

1000	Geldanlage
3	Zinssatz
5	Laufzeit

Berechne

Dein Guthaben beträgt nach dieser Zeit: 1159.27

Spareinlage mit Zielwertberechnung

```
<!DOCTYPE html>

<html lang="de">

  <head>
    <title>Zielwert</title>
    <meta charset="utf-8">
    <script src="ziewert.js" defer</script>
  </head>

  <body>

    <h3>Berechnung des Sparguthabens</h3>

    <input type="number" id="kapital" value="100" step="100"> Geldanlage
    <br>
    <input type="number" id="zinssatz" value="3" step="0.1"> Zinssatz
    <br>
    <input type="number" id="zielwert" value="1000"> Zielwert
    <br><br>
    <button id="berechne"> Berechne</button>

    <p>
      Du benötigst dafür:
      <i><output id="ergebnis"></output></i> Jahre.
    <br>
    Und hast nach dieser Zeit:
    <i><output id="ersparnis"></output></i> € erreicht.
    </p>

  </body>

</html>
```

```
document.getElementById("berechne").addEventListener("click", berechne);

function berechne(){

  var kapital = document.getElementById("kapital").valueAsNumber;
  var zinssatz = document.getElementById("zinssatz").valueAsNumber;
  var zielwert = document.getElementById("zielwert").valueAsNumber;
  var laufzeit = 0;

  while( kapital < zielwert){

    kapital = kapital + kapital * zinssatz / 100;
    laufzeit ++;

  }

  document.getElementById("ergebnis").value = laufzeit.toFixed(2);
  document.getElementById("ersparnis").value = kapital.toFixed(2);

}
```

Berechnung des Sparguthabens

100	Geldanlage
3	Zinssatz
1000	Zielwert

Berechne

Du benötigst dafür: *78.00* Jahre.
Und hast nach dieser Zeit: *1003.01* € erreicht.