

---

# HTML – AG

---

## INHALT

---

---

Allgemeines .....	2
Womit kann ich Websites erstellen? .....	2
Grundgerüst von HTML.....	3
Tags und Attribute .....	3
Links .....	4
Links auf andere Seiten.....	4
Bilder.....	5
meta-Tags .....	5
Webseiten mit Stil – Inline Styles.....	6
<span> .....	6
Style global.....	7
Style im <head> .....	7
Stylesheets.....	12
Tabellen .....	13
Hilfe es bewegt sich – JavaScript .....	15
Hallo Welt!.....	15
Buttons .....	15
Rechnen per Knopfdruck .....	16
for Schleifen.....	17
Wichtige Tags.....	18

## ALLGEMEINES

---

Um eine Website zu erstellen werden wir in der AG 3 Sprachen gebrauchen.

HTML, CSS und JavaScript.

HTML erzeugt die Struktur, also das, was später zu sehen ist.

Mit CSS bearbeiten wir das Aussehen (z.B. Farbe).

Mit JavaScript wird die Seite interaktiv.

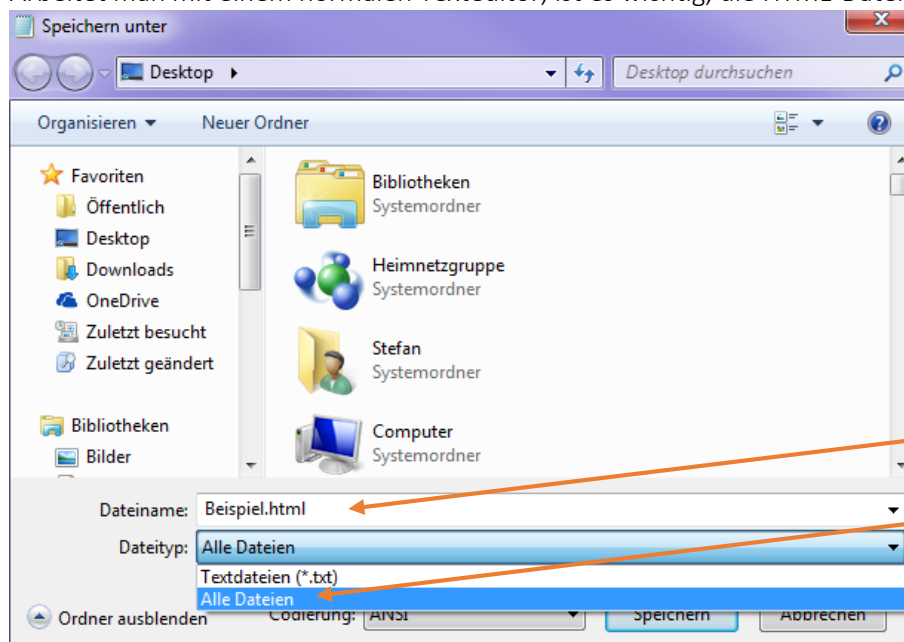
Wichtig: Man braucht in der Regel mindestens 2 verschiedene Webbrowser um mit HTML zu programmieren. HTML wird nämlich vom Webbrowser interpretiert und der Code, den man geschrieben hat, kann je nach Browser, der ihn liest, anders dargestellt werden. Zur Sicherheit sollte man seine programmierten Seiten also mit mehreren Browsern testen.

## WOMIT KANN ICH WEBSITES ERSTELLEN?

---

Um Websites zu erstellen, genügt ein einfacher Texteditor (z.B. Wordpad). In der Regel ist es aber besser einen Editor zu benutzen, der Syntaxhighlighting unterstützt. Dadurch wird der geschriebene Code etwas eingefärbt. Dies macht das Programmieren übersichtlicher.

Arbeitet man mit einem normalen Texteditor, ist es wichtig, die HTML-Datei richtig zu speichern:



Endung Dateiname:  
.html

Dateityp: Alle Dateien

## GRUNDGERÜST VON HTML Das steht immer am Anfang!

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <title>ersteSeite</title>
6
7   <meta charset="UTF-8"/>
8   <meta name="description" content=""/>
9   <meta name="author" content=""/>
10
11 </head>
12
13 <body>
14
15 </body>
16 </html>
```

öffnender Tag

Titel der Website (wird in Tabs gezeigt)

Zeichensatz UTF-8

Für die Suchmaschinen

hierhinein kommt alles, was man später auf der Website sehen soll!

schließender Tag

## TAGS UND ATTRIBUTE

HTML Tags bestehen in der Regel aus 2 Teilen, einem öffnenden und einem schließenden Tag.

Bsp.: `<p> ... </p>`

Öffnender und schließender Tag umschließen den Inhalt wie ein Klammernpaar.

Es gibt aber auch Ausnahmen.

z.B.: `<br />` oder `<img />`

Diese offenen Tags bestehen nur aus einem Teil. Erkennbar sind sie daran, dass der Slash (/) sich am Ende befindet. Sie werden dann verwendet, wenn ein Tag keinen Inhalt haben kann.

Jeder Tag kann auch Attribute besitzen. Attribute legen die Eigenschaften, die ein Tag hat, näher fest.

z.B.: `<auto marke="audi" modell="r8"> mein Auto </auto>`

Die Attribute stehen immer nur im öffnenden Tag.

### Aufgabe:

Erstelle eine erste HTML-Seite und schreibe Text in den Body-Bereich.

Speichere und sieh dir deine Website an.

## LINKS

---

```
<a href="#unten" id="oben"> link nach unten </a>
```

```
    hier steht sehr viel text
```

```
<a href="#oben" id="unten"> link nach oben </a>
```

href gibt das Ziel an, wohin man auf der Seite springen will.

id ist die Sprungmarke.

Möchte man zu einer Sprungmarke springen, muss in href das hash (#) geschrieben werden.

## LINKS AUF ANDERE SEITEN

---

```
<a href="seiteZwei.html"> Link Text </a>
```

verlinkt auf eine Seite, die im selben Ordner, wie die erste Seite ist.

```
<a href="seiteZwei.html#unten"> Link Text </a>
```

verlinkt auf eine Seite und springt dort zur Sprungmarke „unten“.

```
<a href="andereSeiten/seiteZwei.html"> link Text </a>
```

verlinkt auf eine Seite, die sich im Unterordner „andereSeiten“ befindet.

```
<a href="../seiteZwei.html"> Link Text </a>
```

verlinkt auf eine Seite, die sich in einem Ordner darüber befindet.

```
<a href="http://www..."> link Text </a>
```

verlinkt auf eine externe Internetseite.

```
<a href="seiteZwei.html" target="_blank"> Link Text </a>
```

Durch target="blank" wird beim Klicken auf den Link direkt ein neues Fenster oder ein neuer Tab geöffnet. (Was davon passiert hängt vom Browser ab).

Um eine Sprungmarke (oder Sprungziel) zu setzen, wird das Attribut „id“ verwendet.

id kann in jedem Tag gesetzt werden, nicht nur in a.

### Aufgaben:

1. Erstelle ein langes Dokument und versehe es mit Links um an bestimmte Stellen zu kommen.  
Was passiert mit der Adresszeile, wenn man auf den Link klickt?
2. Erstelle eine zweite Seite und verlinke beide von dir erstellten Seiten.  
Benutze auch hier Sprungmarken.

## BILDER

---

```

```

**img** ist der Tag um ein Bild einzufügen.

**src** gibt die Quelle an. Liegt die Quelle im selben Ordner, wie die Seite, muss man nur den Bildnamen mit der Dateiendung angeben. ansonsten auch den relativen Pfad vom Ordner der Seite aus gesehen. Man kann auch Bilder aus dem Internet verwenden: src=http://www.....

Dabei muss man aber das Urheberrecht beachten.

**height** gibt die Höhe in Pixeln an und **width** die Breite.

**alt** ist ein Alternativtext für Hörgeschädigte, oder falls das Bild durch einen Fehler nicht angezeigt wird. alt wird benötigt, damit die Seite valide ist.

alt kann aber auch leergelassen werden: alt=""

„Ein HTML-Dokument ist valide, wenn es allen Regeln für HTML entspricht: Es verwendet nur offizielle Tags und Attribute an Stellen, an denen sie erlaubt sind, es schließt alle Tags korrekt und enthält alle verpflichtenden Tags und Attribute. Valide ist für den Webentwickler koscher, halal und zuckerfrei in einem.“ (Definition aus „Schrödinger lernt HTML 5, CSS 3 und JavaScript“)

## META-TAGS

---

```
<meta charset="UTF-8"/>
<meta name="description" content="" />
<meta name="author" content="" />
```

Diese Tags kennst du schon aus dem Grundgerüst am Anfang dieser Erklärung.

charset="UTF-8" gibt den Zeichensatz an, der verwendet wird.

name="description" wird für die Suchmaschinen verwendet

name="author" gibt den Autornamen an.

Bei content="" können freie Informationen eingetragen werden.

## WEBSEITEN MIT STIL – INLINE STYLES

In diesem Kapitel wird es jetzt endlich etwas bunt. Und zwar mit CSS.  
Dafür brauchst du aber noch ein neues HTML-Tag.

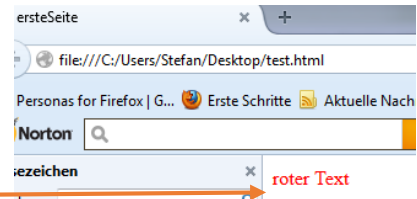
### <SPAN>

<span> ist ein Tag, das für sich allein rein gar nichts bewirkt. <span> ist NICHTS und hat keine Funktion.  
Durch das Hinzufügen von Attributen, kann es dann jedoch etwas bewirken.

Probier das aus. Schreibe in einer Datei: <span> Test </span> und schau was passiert.

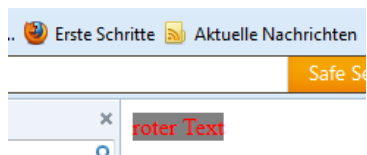
In den <span> - Tag muss man nun das style-Attribut setzen:

```
<span style="color: red;"> roter Text </span>
```



Aber da geht noch mehr:

```
<span style="color: red; background-color: #808080;"> roter Text </span>
```



Was bewirkt wohl background-color: #808080; ?

Achte beim style-Attribut genau auf die Zeichen. Jeder Doppelpunkt und jedes Semikolon (;) ist wichtig.

Farben kann man nicht nur durch die englischen Begriffe angeben, sondern auch als RGB Wert (Red, Green, Blue). Hierbei muss man aber darauf achten, dass die Zahl eine Hexadezimalzahl ist. Die ersten beiden „Ziffern“ stehen für den Rot-Wert, die nächsten beiden für den Grün-Wert und die letzten beiden für den Blau-Wert.

Such im Internet nach dem Begriff „HTML Farben“, so erhältst du als Suchergebnis verschiedene Farben und was du dafür eingeben musst.

### Aufgaben:

1. Füge ein <span> Element in deine Datei ein und passe den Text darin farblich an.  
Probiere einige Farben aus.
2. probiere das style-Attribut in anderen Tags aus.

### Wichtig:

Normalerweise verwendet man CSS NICHT auf diese Weise. Tut man dies doch, wird das als schlechter Programmierstil gewertet. Dadurch ist die spätere Internetseite auch sehr wartungsintensiv.

### Aufgabe:

Überlege dir, weshalb dieses „Inline-Styling“ unpraktisch ist!

---

## STYLE GLOBAL

---

Wenn du eine Website erstellst, dann erstellst du in der Regel nicht nur eine einzige Seite, sondern sehr viele. Wenn du nun das CSS im Code jeder einzelnen Seite in den verschiedenen `<span>`-Tags gesetzt hast, funktioniert das zwar, aber wenn du am Aussehen der gesamten Website etwas ändern willst, bekommst du Probleme. Dann musst du nämlich jede einzelne Datei nach allen `<span>`-Tags durchsuchen und diese einzeln ändern.

Um das zu vereinfachen, gibt es 2 Möglichkeiten:

1. im `<head>`-Bereich des HTML Dokuments
2. in einer eigenen Datei


---

## STYLE IM <HEAD>

---

```
<head>
  <title>ersteSeite</title>

  <style type="text/css">
  </style>
</head>
```

 hier kommt der CSS-Code hinein

In HTML 4 oder früheren Versionen muss das Attribut `style="text/css"` gesetzt werden. In HTML 5 darf es gesetzt werden, muss aber nicht.

```
9      <style type="text/css">
10         p{
11             color: green;
12         }
13
14         span{
15             color: red;
16             background-color: #808080;
17         }
18     </style>
```

Wie du sehen kannst, stehen alle CSS-Anweisungen im `<style>`-Tag.

`p` bedeutet, dass alle `<p>`-Tags betroffen sein werden, `span` bedeutet, dass alle `<span>`-Tags betroffen sein werden. Genau so funktioniert das auch mit sämtlichen anderen Tags.

`p` bzw. `span` werden „Selektoren“ genannt, weil sie dazu dienen, die entsprechenden Tags aus dem `<body>` zu selektieren (= auswählen). Diese Selektoren funktionieren in allen Browsern.

Alle Anweisungen müssen wie oben gezeigt, zwischen geschweiften Klammern `{ }` stehen.

Nach jeder Anweisung muss ein Semikolon `( ; )` stehen.

## Aufgaben

1. Entferne aus deinem Dokument überall das `style`-Attribut und verlagere die CSS Anweisungen in den `<head>`-Bereich.
2. Füge Überschriften und andere Tags ein und bearbeite sie mit CSS.

---

## EIGENSCHAFTEN FÜR DAS GESAMTE DOKUMENT FESTLEGEN

---

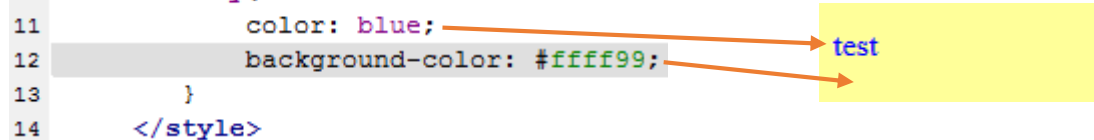
Manchmal möchte man auch Eigenschaften für das gesamte Dokument setzen, und nicht nur für einzelne Absätze. Dann benötigt man den Selektor `body`.

```
9 <style type="text/css">
10     body{
11         color: blue;
12     }
13 </style>
```

Durch den Beispiel-Code wird dafür gesorgt, dass der gesamte Text blau ist (und nicht schwarz, wie sonst üblich).

Möchte man die Hintergrundfarbe der ganzen Seite festlegen, kann man folgenden Code benutzen:

```
9 <style type="text/css">
10     body{
11         color: blue;
12         background-color: #ffff99;
13     }
14 </style>
```



Dadurch wird die Hintergrundfarbe der gesamten Seite gelb und die Schrift blau.

## Aufgabe

1. Experimentiere mit Textfarben und Hintergrundfarben.  
Sowohl für das gesamte Dokument als auch mit einzelnen Absätzen und Tags.

### ID- UND CLASS- ATTRIBUTE SELEKTIEREN

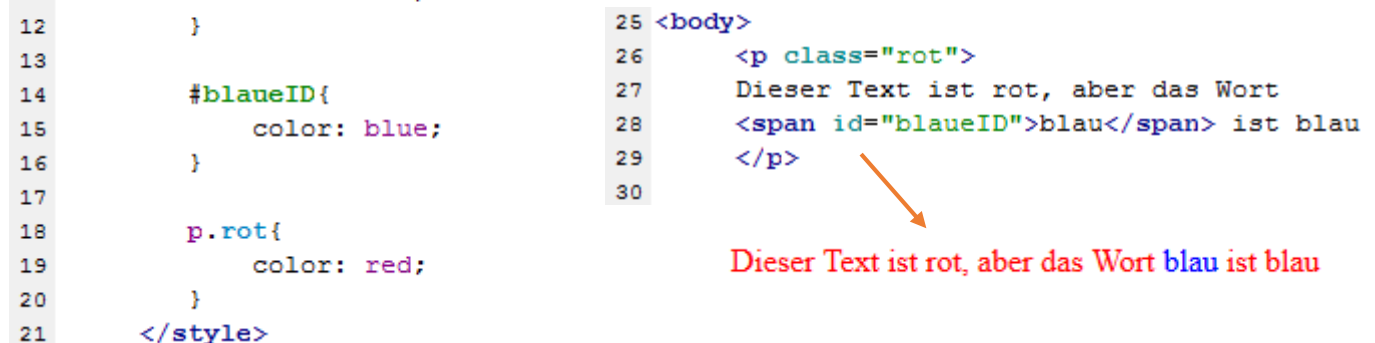
Oft möchte man aber nicht alle `<p>` oder `<span>`-Tags usw. gleich formatieren. Um nun nur ganz bestimmte Tags auszuwählen, kann man nach dem `id`-Attribut oder nach dem `class`-Attribut selektieren.

Denke daran: IDs müssen in einem Dokument immer einzigartig sein, d.h. eine bestimmte `id` darf es in einem Dokument nur ein einziges mal geben.

Dies trifft aber nicht für das `class`-Attribut zu.

```
9 <style type="text/css">
10     .rot{
11         color: red;
12     }
13
14     #blaueID{
15         color: blue;
16     }
17
18     p.rot{
19         color: red;
20     }
21 </style>
```

```
25 <body>
26     <p class="rot">
27         Dieser Text ist rot, aber das Wort
28         <span id="blaueID">blau</span> ist blau
29     </p>
30
```



`.rot` bezieht sich auf alle Tags, in denen das Attribut `class="rot"` gesetzt wurde, ganz egal ob `<p>`, `<span>`, `<h1>` oder was auch immer.

Der Punkt davor gibt an, dass es sich auf `class` bezieht.

`p.rot` bezieht sich dagegen nur auf `<p class="rot">`, also nur auf `<p>`-Tags, in denen das Attribut `class="rot"` gesetzt wurde.



#blauelD bezieht sich auf die id="blauelD"

## Aufgabe

1. Tobe dich mit class und id aus!

### MEHRERE KLASSEN AUSWÄHLEN

Manchmal ist es sinnvoll, einen bestimmten Bereich mehreren Klassen zuzuweisen. Das geht, indem man bei dem Attribut `class=""` innerhalb der Anführungszeichen alle gewünschten Klassen angibt. Diese Klassen müssen alle innerhalb desselben `class`-Attributs stehen und müssen durch ein Leerzeichen getrennt sein.

```
<style type="text/css">
  .rot{
    color: red;
  }

  .unterstrichen{
    text-decoration: underline;
  }
</style>
```

.....

```
<p><span class="rot">roter Text</span></p>
<p><span class="unterstrichen">unterstrichener Text</span></p>
<p><span class="rot unterstrichen">roter und unterstrichener Text</span></p>
```

roter Text

unterstrichener Text

roter und unterstrichener Text

### HINTERGRUNDBILDER

```
<style type="text/css">
  body {
    background-image: url(D3.jpg);
  }
</style>
```

Mit diesem Code kannst du ein Hintergrundbild auswählen, das für die gesamte Seite gilt.

Im Beispiel ist das Hintergrundbild eine jpg-Datei mit dem Namen D3 und diese Datei befindet sich im selben Ordner, wie auch die HTML-Datei.

Es ist natürlich auch möglich, ein Hintergrundbild für `<span>`, `<p>` oder alle anderen HTML-Tags zu definieren.

**Verwendest du ein Stylesheet muss die URL relativ zum Stylesheet angegeben werden!!**

Wie du aber siehst, wiederholt sich das Hintergrundbild ständig. Wenn du das nicht möchtest, kannst du das auch deaktivieren und zwar mit der Zeile:

```
background-repeat: no-repeat;
```



```
<style type="text/css">
  body {
    background-image: url(D3.jpg);
    background-repeat: no-repeat;
  }
</style>
```



Wenn du möchtest, dass sich das Bild nur von links nach rechts wiederholt, verwendest du:

```
background-repeat: repeat-x;
```

Wenn du möchtest, dass sich das Bild nur von oben nach unten wiederholt verwendest du :

```
background-repeat: repeat-y;
```

Auch die Position des Hintergrundbildes lässt sich bestimmen:

```
<style type="text/css">
  body {
    background-image: url(D3.jpg);
    background-repeat: no-repeat;
    background-position: 20px 30px;
  }
</style>
```

Diese Zeile sorgt dafür, dass das Hintergrundbild von der oberen linken Ecke aus 20 Pixel nach rechts und 30 Pixel nach unten verschoben wird.

Anstatt der genauen Angaben kannst du auch Schlüsselwörter verwenden:

- left
- center
- right
- top
- bottom

Man kann auch dafür sorgen, dass das Bild beim Scrollen mitläuft:

```
body {
  background-image: url(D3.jpg);
  background-attachment: fixed;
}
```

## LINKS BEARBEITEN

---

Auch Links können bearbeitet werden.

```
<style>
  a{
    color: red;
  }
  a:visited{
    color: green;
  }
  a:hover{
    color: blue;
  }
</style>
```

Erscheinungsbild aller Links

Erscheinungsbild aller Links, die schon einmal angeklickt wurden

Mouseover-Effekt:  
Das passiert, wenn man mit der Maus über den Link fährt.

## SCHRIFTARTEN UND -GRÖßEN

---

```
<style>
  body{
    font-size: 14px;
    font-family: Comic;
  }
</style>
```

Mit font-size: 14px; kann man die Schriftgröße anpassen.  
Mit font-family: Comic; setzt man die Schriftart fest.

## STYLESHEETS

---

```
4 <head>
5   <title>ersteSeite</title>
6
7   <link href="style.css" rel="stylesheet" />
```

Möchte man ein Stylesheet (=Style-Blatt) verwenden, muss eine neue Datei für die Styles erzeugt werden. Alles, was man im Kapitel Style im <head> (Seite 7) in das <style> - Tag geschrieben hat, kommt in eine eigene Datei. Diese Datei kann auch mit dem normalen Texteditor erstellt werden. allerdings bekommt sie als Datei-Endung .css

Der genaue Dateiname (oder auch der Pfad, falls die css-Datei in einem Unterordner ist) wird unter href im <link> - Tag angegeben.  
rel="stylesheet" ist in <link> Pflicht.

Wie du bestimmt schon gemerkt hast, steht auch <link> im <head> - Bereich!

Es ist auch möglich, mehrere <link> - Tags mit Stylesheets anzugeben und dazu auch noch zusätzlich CSS in den <head> -Bereich zu schreiben.

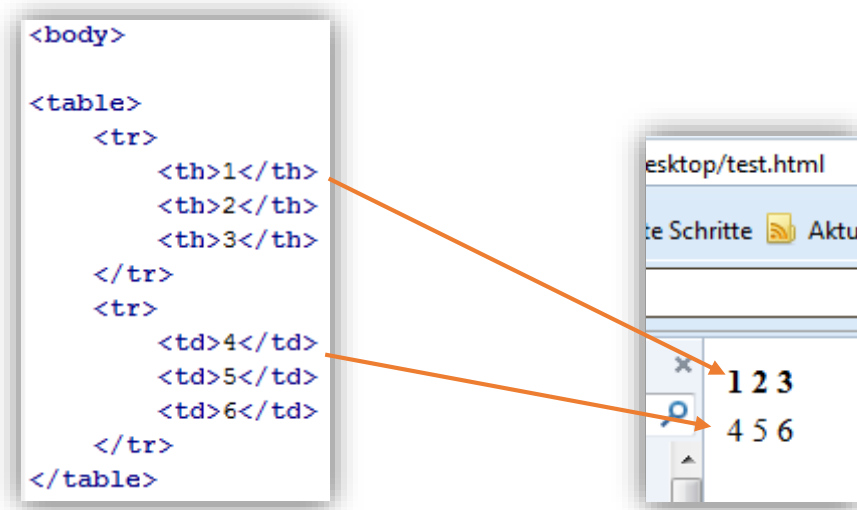
Sollte dabei etwas doppelt vorkommen, wird eine Eigenschaft aus dem Stylesheet durch eine Eigenschaft im <style>-Tag überschrieben.  
Die Eigenschaft im <style>-Tag wird durch ein style-Attribut (z.B. <span style=" ";> überschrieben.

## Aufgabe

1. Erstelle ein Stylesheet und implementiere es in dein HTML-Dokument

## TABELLEN

---



<table> wird verwendet um eine Tabelle zu erzeugen.

<tr>...</tr> erzeugt in der Tabelle eine neue Zeile. Also alles, was zwischen <tr> und </tr> steht ist in einer Zeile in der Tabelle.

Zwischen <th> und </th> oder zwischen <td> und </td> steht der Inhalt der Tabelle.  
Anstelle von 1, 2, 3 und so weiter, kann hier ganz normaler Text stehen.

Momentan sieht diese Tabelle aber noch nicht wirklich nach einer richtige Tabelle aus.

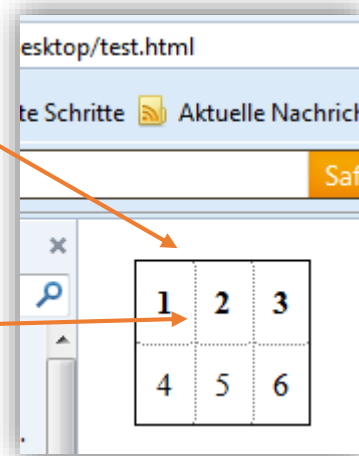
Dies muss man in CSS (also im <style>-Tag ändern.

```

<style>
  table {
    border: thin solid black;
    border-collapse: collapse;
    margin-left: 20px;
    margin-right: 20px;
    margin-top: 20px;
    margin-bottom: 20px;
    text-align: center;
  }

  td, th {
    border: thin dotted gray;
    padding: 10px;
  }
</style>

```



Nun dazu, was die einzelnen Zeilen machen:

```

<style>
  table {
    border: thin solid black;
    border-collapse: collapse;
    margin-left: 20px;
    margin-right: 20px;
    margin-top: 20px;
    margin-bottom: 20px;
    text-align: center;
  }

  td, th {
    border: thin dotted gray;
    padding: 10px;
  }
</style>

```

**border: thin solid black;**

Der Rahmen soll dünn, eine durchgezogene Linie und schwarz sein.

das kannst du alles bei border einsetzen:

#### Code-Completion

```

thin
medium
thick
none
hidden
dotted
dashed
solid
double
groove
ridge
inset
outset

```

**border-collapse: collapse;**

Das brauchst du in fast jeder Tabelle. Lösche die Zeile um zu sehen, wie die Tabelle sonst aussehen würde.

**margin-left : 20px; oder padding: 10px;**

Die Werte für margin oder padding brauchst du, um dafür zu sorgen, dass die Werte in der Tabelle nicht zu sehr gequetscht wirken.

Ändere die Werte für margin und padding, oder vertausche margin mit padding um die Unterschiede zu sehen.

### Aufgabe

1. Erstelle eine Tabelle und fülle sie mit Inhalt.
2. Übernehme den <style>-Tag in deinen Head-Bereich.
3. Ändere die Werte für margin und padding.

4. Lösche die Zeile „border-collapse: collapse;“ und sieh was passiert.
5. Ändere die Zeile border und füge andere Angaben ein. (z. B. anstatt thin schreibst du thick)
6. Ändere die Schriftfarbe deiner Tabelle.

---

## HILFE ES BEWEGT SICH – JAVASCRIPT

---

---

### HALLO WELT!

---

Das erste Programm, das man schreibt hat immer den Titel „Hallo Welt!“.

für unsere erste Bildschirmausgabe brauchen wir nicht viel.

Nur ein ganz normales HTML Dokument. Wir müssen im Body-Bereich lediglich eine kleine Zeile ergänzen: `<body onload="alert('Hallo Welt!')">`

Onload=“...” bedeutet, dass eine Funktion beim Laden der Seite aufgerufen wird (wenn das so wie hier im body-Tag steht).

Die Funktion die aufgerufen wird ist alert(“...“).

Was hier in den Anführungszeichen steht wird später als Text auf dem Bildschirm ausgegeben.

Achtung: Da sowohl für onload, als auch für alert Anführungszeichen benötigt werden, **muss** man jeweils unterschiedliche Anführungszeichen verwenden.

### Aufgabe

1. Schreibe ein Dokument, dass beim Starten den Text „Hallo Welt!“ ausgibt.

---

### BUTTONS

---

```
<body>

<input type="button" value="Tu etwas" onclick="alert('Hallo Welt!');"
      style="font-size: xx-large; font-family: comic;" />
```

Buttons kannst du ganz normal im Body-Tag einfügen.

Das Attribut type=“button“ sagt dem Programm, dass es sich eben um einen Button handelt.

value=“...” steht später auf dem Button.

onclick=“...” hier steht das, was passieren soll, wenn auf den Button geklickt wird. Im Beispiel erscheint der Text „Hallo Welt!“

In der 2. Zeile des Beispiels siehst du, wie der Button mit CSS bearbeitet wurde. Das sollte man aber eigentlich besser im <style>-Bereich machen.

Man kann auch rechnen.

### Aufgabe

1. Ersetze im Beispiel onclick="..." durch onclick="alert(2+3);"  
Was passiert?  
Achte darauf hier in der Klammer **keine** Anführungszeichen zu setzen!

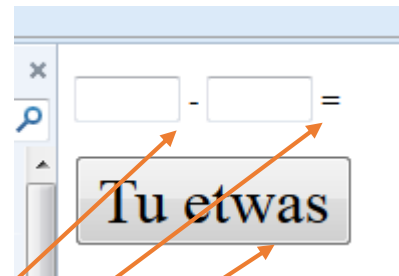
## RECHNEN PER KNOPFDRUCH

---

```
<script>
  var a = 0;
  var b = 0;
  var loesung=0;

  function addiere() {
    a = document.getElementById("zahl1").value;
    b = document.getElementById("zahl2").value;
    loesung = a - b;
    document.getElementById("loesung").innerHTML = loesung;
  }
</script>
```

....



```
<body>
<p>
<input type="text" size="5" id="zahl1" /> -
<input type="text" size="5" id="zahl2" /> =
<span id="loesung"></span>

</p>
<input type="button" value="Tu etwas" onclick="addiere();"
  style="font-size: xx-large; font-family: comic;" />
</body>
```

Durch den obigen Code werden zwei Zahlen, die in die Textfelder eingegeben werden, voneinander subtrahiert und das Ergebnis hinter das = geschrieben.

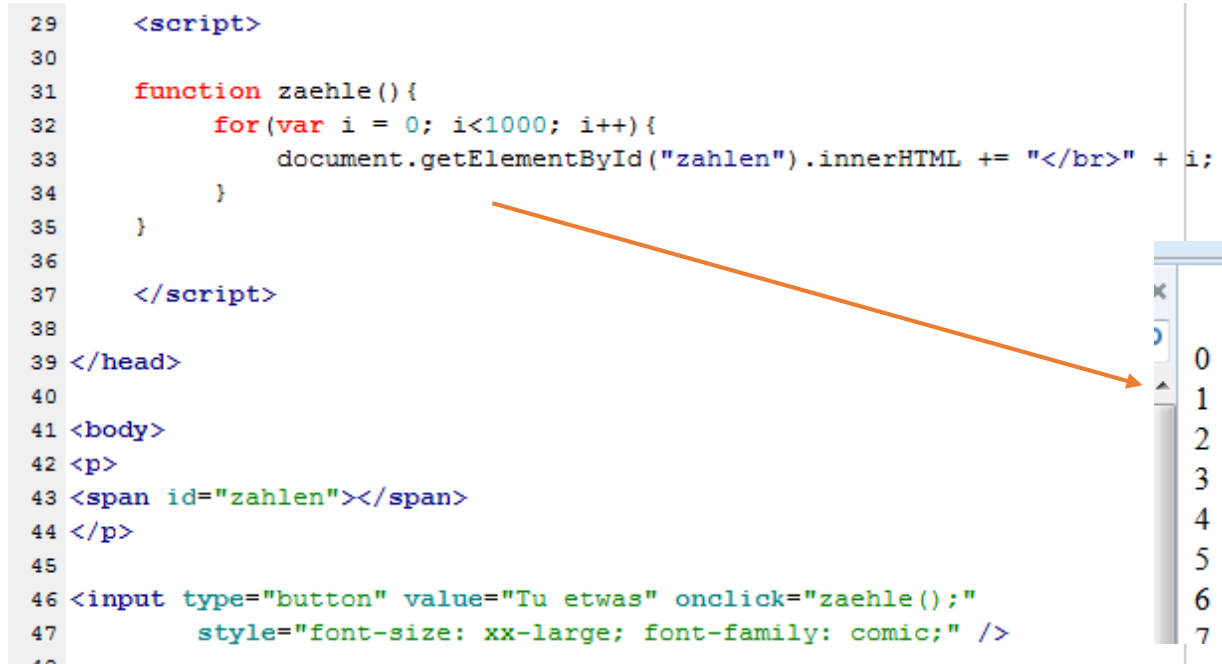
### Aufgabe

1. Übernimm den Code und probiere aus, ob es funktioniert.
2. versuche das gleiche mit einer Multiplikation und einer Division.
3. Versuche das gleiche mit einer Addition. (Was passiert? Warum? Wie kannst du das lösen?)



## FOR SCHLEIFEN

```
29 <script>
30
31 function zaehle(){
32     for(var i = 0; i<1000; i++){
33         document.getElementById("zahlen").innerHTML += "<br>" + i;
34     }
35 }
36
37 </script>
38
39 </head>
40
41 <body>
42 <p>
43 <span id="zahlen"></span>
44 </p>
45
46 <input type="button" value="Tu etwas" onclick="zaehle();"
47     style="font-size: xx-large; font-family: comic;" />
48
```



Mit dem obigen Code wird eine For-Schleife erstellt.

Sie besteht aus 3 Teilen:

for( Erzeugung einer Variablen und Festsetzung des Wertes ; Bedingung, wann die Schleife aufhört; was passiert nach jedem Schleifendurchgang)

var i = 0 erzeugt eine Variable i mit dem Wert 0.

i < 1000 Bedeutet, dass die Schleife so lange laufen wird, bis i den Wert 1000 hat.

i++ bedeutet, dass i nach jedem Durchlauf um 1 erhöht wird.

## WICHTIGE TAGS

---

<code>&lt;p&gt; ... &lt;/p&gt;</code>	Absatz
<code>&lt;h1&gt;</code> bis <code>&lt;h6&gt;</code>	Überschrift 1. Ordnung bis 6. Ordnung
<code>&lt;br /&gt;</code>	erzeugt 1 Zeilenumbruch
<code>&lt;a&gt; ... &lt;/a&gt;</code> Attribute: - href=http://www... - href="#unten" - target="_blank"	erzeugt einen Link  gibt Adresse an wohin verlinkt wird verlinkt auf eine Sprungmarke, die mit id gesetzt wurde öffnet beim Klick ein neues Fenster oder einen neuen Tab
<code>&lt;title&gt;...&lt;/title&gt;</code>	im head-Bereich: gibt Titel an
<code>&lt;img /&gt;</code> Attribute: - src="..." - alt="..." - width="..." - height="..."	erzeugt ein Bild  Quelle Alternativtext Breite Höhe
<code>&lt;span&gt;...&lt;/span&gt;</code>	- bewirkt nichts