HTML
Ein Skript für das Berufskolleg

Hermann Maier

5. Januar 2025

Aktuelle Version inklusive Quelldateien unter
https://github.com/hoerm007/HTMLSkript_KaufmBK_BW

Inhaltsverzeichnis

1 Begriffe und Grundgerüst 4

2 Umlaute und Sonderzeichen 6

3 Validieren 7

4 Textauszeichnungen 8

5 Cascading Style Sheets 10

6 Listen 12

7 Bilder 13

8 Abschnitte definieren und formatieren 15

9 Verlinkungen 17

10 Tabellen 19

11 Lösungen 20

1 Begriffe und Grundgerüst

Das Internet ist aus unserer heutigen Welt nicht mehr wegzudenken. Im WWW findet man Informationen jeder Art, man kauft ein, trifft sich und kommuniziert rund um den ganzen Globus. Das WWW unter Eingabe der URL zu nutzen, ist heute selbstverständlich geworden. Eine Webseite jedoch mit HTML zu erstellen, welche von einem Browser angezeigt werden kann, beherrschen nur wenige. Um die Grundlagen von HTML Schritt für Schritt zu erlernen, werden wir eine eigene Webseite erstellen. Wir beschränken uns auf die Grundlagen, d.h. wir werden einen einfachen Texteditor zum Erstellen verwenden anstatt einer Entwicklungsumgebung. Zum Anzeigen der Webseite werden wir einen normalen Browser wie Chrome, Firefox oder Edge verwenden.
Eine Webseite ist letztendlich eine Datei. Standardmäßig ist diese Datei in der Auszeichnungssprache HTML erstellt. Wir werden also selbst eine solche Datei erstellen. Greift man mit Hilfe eines Browsers über das Internet auf eine Seite zu, so wird diese HTML-Datei über das Internet an den eigenen PC übertragen und dann vom Browser interpretiert und dargestellt. Eine HTML-Datei kann man aber auch einfach auf dem eigenen PC erstellen, bearbeiten und betrachten.

Übung 1 **Für was steht die Abkürzung HTML und was versteht man unter dem Begriff einer Auszeichnungssprache? Die Begriffe maschinenlesbare Sprache und Tag sind hier besonders wichtig.**

Wir erstellen Schritt für Schritt unsere eigene Webseite.

Übung 2 **Erstelle deine erste Webseite.**

Erstelle in deinem Heimverzeichnis (Laufwerk mit deinem Anmeldenamen) einen Ordner *HTML*. Öffne den Editor und kopiere folgenden Text:

```
<html>

<head>

<title>Schülerseite</title>

</head>

<body>

Willkommen auf meiner Webseite!

<br>

<br>
```

```
<p>Ich bin Schüler des WG West in Stuttgart.</p>

</body>

</html>
```

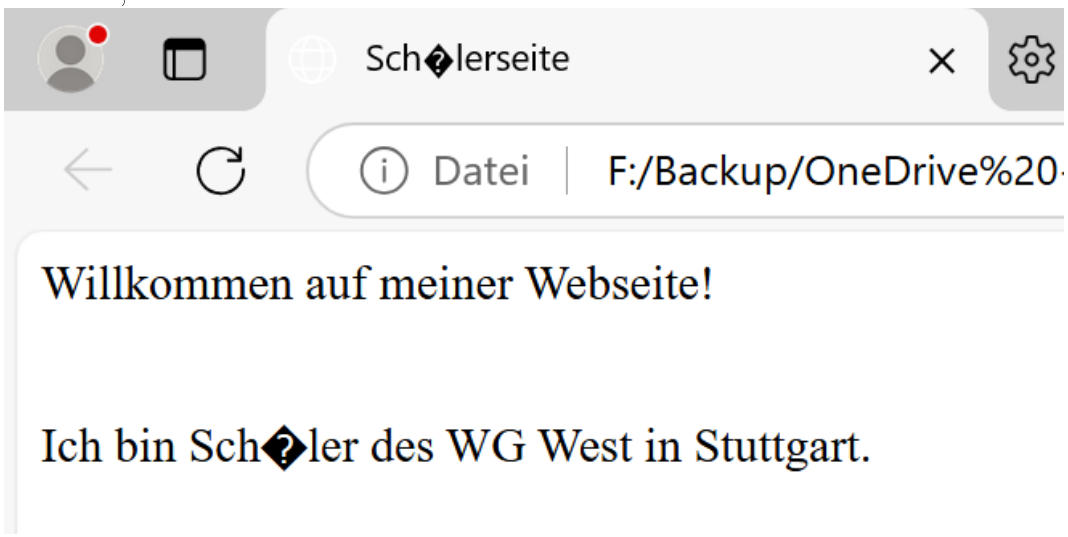
Speichere die Datei nun als *schueler.html* in deinem HTML-Ordner. (Datei→speichern unter→Dateityp auf Alle Dateien ändern.) Diese Datei kann nun mit einem Browser geöffnet werden.

Übung 3 **Recherchiere die Funktion der oben verwendeten Tags.**

Tipp: Die Webseite w3schools.com verfügt über eine Übersicht aller Tags.

2 Umlaute und Sonderzeichen

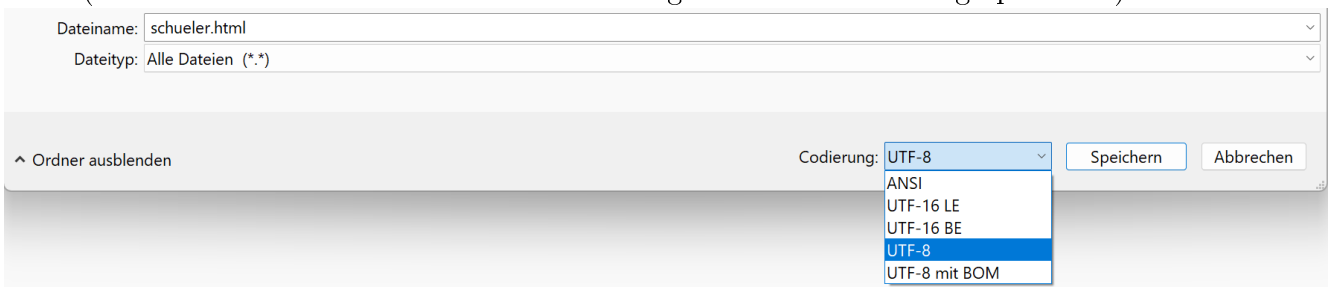
Je nach Betriebssystem und Browser kann es zu Problemen bei der Darstellung von Zeichen kommen, insbesondere bei Umlauten:



Um solche Probleme zu vermeiden, ist es empfehlenswert, die Umlaute durch bestimmte Zeichenfolgen, sogenannte Escape-Sequenzen, zu ersetzen, z.B. statt ä im Quellcode `\&auml;` zu schreiben. Im Internet lassen sich verschiedenen Listen mit diesen Zeichenfolgen finden.

Übung 4

1. Füge zwischen den head-Tags (`<head>... </head>`) den folgenden Tag ein:
   `<meta charset="utf-8">`
2. Schaue dir dann deine Seite nochmals im Browser an.
3. Ersetze alle Umlaute durch Escape-Sequenzen und verwende in Zukunft diese statt den Umlauten oder dem scharfen s.
4. Prüfe nochmals im Browser, ob die Umlaute nun alle korrekt dargestellt werden.
5. Ändere nun noch die Codierung deiner Datei auf UTF-8. Gehe dazu im Editor nochmals auf Speichern unter→Dateityp auf alle Dateien ändern und Codierung auf UTF-8. (Hinweis: Oft wird die Datei standardmäßig bereits mit UTF-8 gespeichert.)



3 Validieren

HTML existiert seit Ende der 80er und hat sich mit der Zeit stark verändert. Zum Beispiel gibt es Unmengen an Tags, die als deprecated, d.h. veraltet markiert sind. Diese Tags sollten eigentlich nicht mehr verwendet werden aber es stehen noch viele Webseiten online, die diese Tags verwenden. Die best practices, also das empfohlene Vorgehen beim Erstellen von Webseiten hat sich ebenfalls stark gewandelt. Bis heute haben Browser unterschiedliche Funktionsumfänge, z.B. ist das Format JPG für Bilder veraltet. Es gibt deutlich bessere Formate, die aber jeweils nicht von allen Browsern unterstützt werden.
Lange Rede, kurzer Sinn, HTML ist ein Sammelsurium verschiedenster Techniken, um Webseiten zu gestalten, von denen viele aus Gründen der Abwärtskompatibilität noch unterstützt werden aber nicht mehr verwendet werden sollen. Wie kann Ottonormalschüler entscheiden, ob er seine Webseite vernünftig erstellt hat? Die empfohlenen Techniken werden regelmäßig (alle paar Jahre) in einem Standard zusammengefasst, aktuell HTML 5.2. Nur, weil ein Browser eine Webseite korrekt darstellt, muss die Webseite noch lange nicht dem Standard entsprechen. Glücklicherweise gibt es Validatoren, die prüfen, ob eine Webseite dem Standard entspricht.

Übung 5 Prüfe deine Webseite (schueler.html in deinem Heimverzeichnis) mit einem Validator

Zum Beispiel W3Schools stellt unter von validator.w3.org einen Validator zur Verfügung.

Unsere Webseite entspricht also nicht dem Standard, wird aber von einem Browser (vermutlich) korrekt dargestellt.

Übung 6 Passe deine Webseite so an, dass sie dem Standard entspricht.

Ergänze am Anfang deiner Webseite noch vor dem `<html>`-Tag folgenden Tag:
`<!DOCTYPE html>`. Erweitere den `<html>`-Tag zu `<html lang="de">`
Informiere dich über die Bedeutung der eingefügten Tags und validiere deine Webseite dann nochmals.

4 Textauszeichnungen

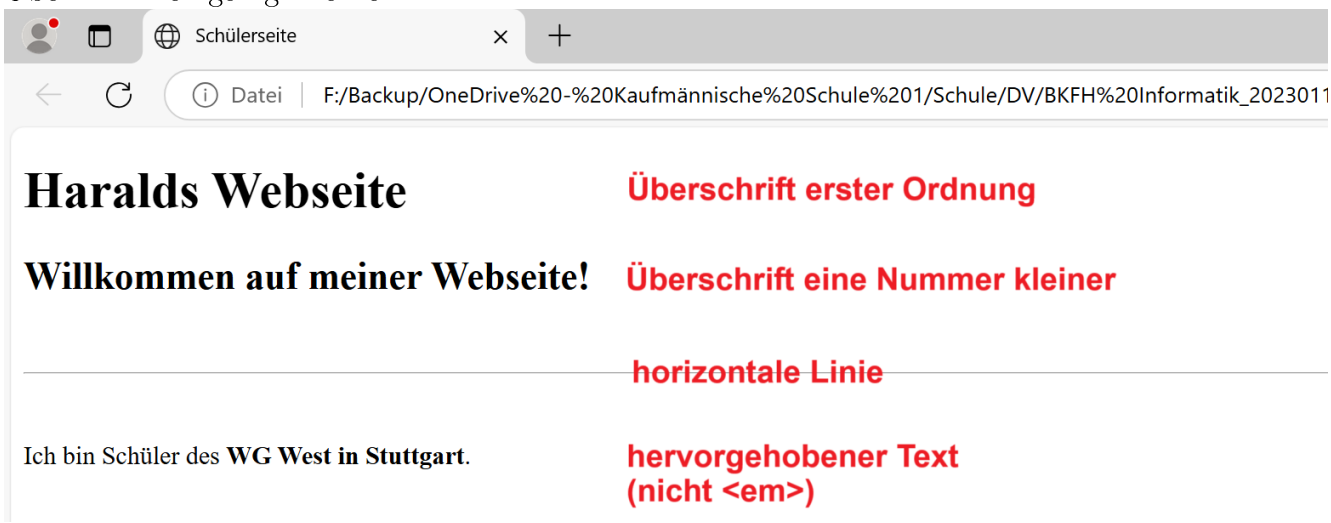
Auf Webseiten können Teile von Texten ausgezeichnet werden, um sie hervorzuheben. Hierbei wird zwischen physischen und logischen Auszeichnungen unterschieden. Bei den physischen Auszeichnungen handelt es sich lediglich um optische Auszeichnungen, so dass Texte z.B. Fett oder Kursiv dargestellt werden. Eine besondere Bedeutung haben physisch ausgezeichnete Texte jedoch nicht. Bei logisch ausgezeichneten Texten, z.B. Überschriften oder Zitaten, ist das jedoch anders. Wenn man Texte logisch auszeichnet, dann erhalten die Texte auch eine entsprechende Bedeutung.
Ein Beispiel sind die beiden Tags `<i>...</i>` und `<em>...</em>`. Beide Tags stellen den Text in der Regel kursiv dar. Das `<i>`-Tag steht für italic (deutsch kursiv). Es ist eine physische/optische Auszeichnung. Das `<em>`-Tag steht für emphasize (deutsch hervorheben/betonen) und ist eine logische Auszeichnung, die in der Regel vom Browser ebenfalls als kursiver Text dargestellt wird. Wenn beide Tags zu kursivem Text führen, wo ist der Unterschied? Der Unterschied liegt im Wesentlichen darin, wie der Computer den Text auffasst:
Die beiden Sätze `HTML ist <i>super</i>und toll` und `HTML ist <em>super</em>und toll` werden im Browser zwar gleich angezeigt. Würde man jedoch eine KI fragen was HTML ist, so würde sie beim ersten Satz sagen, dass HTML super und toll ist und beim zweiten Satz, dass HTML insbesondere super ist und außerdem auch toll.
Seit HTML 5 sollen Texte auf Webseiten nur gemäß ihrer Bedeutung logisch ausgezeichnet werden. Das ist aus verschiedenen Gründen wichtig. Dazu zählen unter anderem:

- Barrierefreiheit: Sehbehinderte bzw. blinde Menschen können die Texte auf Webseiten z.B. mit Screenreader lesen. Wenn Texte logisch ausgezeichnet werden, dann hilft das dabei, den Aufbau, die Struktur und die Inhalte der Webseite besser zu verstehen. Wenn man z.B. bei einem normalen Absatz lediglich die Schriftgröße so einstellt, dass es aussieht wie eine Überschrift, jedoch nicht als Überschrift auszeichnet, dann könnte man bei der Betrachtung am Bildschirm zwar annehmen, dass der Text eine Überschrift darstellen soll. Blinde Menschen können das jedoch nicht und für sie ist der Text ein ganz normaler Absatz. Daher ist es für sie eine große Hilfe, wenn man Texte gemäß ihrer Bestimmung auszeichnet.
- Auslesen durch Suchmaschinen: Die Aufnahme in die Suchmaschinen erfolgt in der Regel durch sogenannte Bots. Das sind Programme, die selbständig im Internet Webseiten auslesen und in den Suchmaschinenindex aufnehmen. Um die Struktur und die Inhalte einer Webseite zu erkennen und zu gewissen Suchbegriffen einzuordnen, wird unter anderem die Seite inkl. aller Texte und logischer Auszeichnungen ausgewertet. Kommt ein Begriff z.B. in einer Überschrift vor, so ist die Wahrscheinlichkeit, dass die Webseite interessante Infos zu diesem Begriff enthält größer, als wenn der Begriff nur im Fließtext vorkommt.
- Auslesen durch andere Programme: Was für Suchmaschinen gilt, das gilt auch für andere Programme wie z.B. LLMs/KIs, die Webseiten auslesen. Wenn z.B. in sozialen Netzwerken eine Webseite "geteilt"wird, dann wird häufig die Überschrift sowie ein Ausschnitt aus dem Text und evtl. ein Bild als Vorschau angezeigt. Wenn die Texte nicht entsprechend ausgezeichnet wurden, dann werden solche Vorgänge erschwert.

- Aufbau einer Suchfunktion: Viele Webseiten nehmen nach einer gewissen Zeit sehr große Ausmaße an, sodass eine Suchfunktion auf der Webseite notwendig wird. Auch hierbei ist man darauf angewiesen, dass die Texte gemäß ihrer Bestimmung logisch ausgezeichnet werden. So kann man diesen eine gewisse Gewichtung verleihen. Wenn z.B. ein Suchbegriff auf verschiedenen Seiten in der Überschrift und in den Absätzen vorkommt, dann könnte man den Suchalgorithmus so programmieren, dass die Seite, die den Begriff in der Überschrift enthält eine höhere Gewichtung erhält. Man könnte die Suche auch so eingrenzen, sodass z.B. nur in den Überschriften gesucht wird.

Übung 7 Füge deiner Webseite Textauszeichnungen hinzu.

Die Textelemente der Webseite sollen nun wie im Bild gezeigt ausgezeichnet werden. Zudem soll der spätere Betrachter wissen, um wessen Website es sich dabei handelt. Daher soll eine Überschrift eingefügt werden.



Erweitere deine Datei *schueler.html* mit dem Windows-Editor entsprechend der Vorgaben im Bild. Recherchiere dabei die notwendigen Tags.
Validiere dann deine Datei *schueler.html*.

5 Cascading Style Sheets

Es gibt zwar noch immer HTML-Tags, die zur Formatierung der Webseite dienen, wie z.B. Ändern der Schriftfarbe, jedoch sind so gut wie alle diese Tags als veraltet (deprecated) markiert. Veraltete Tags werden zwar noch unterstützt, sollten aber nicht mehr verwendet werden. In ferner Zukunft werden Browser diese Tags nicht mehr unterstützen wodurch es dann zu Darstellungsfehlern im Browser kommen wird.
Der grundlegende Style (Formatierung abgesehen von physischen Textauszeichnungen) wird über das sogenannte CSS festgelegt. CSS steht für Cascading Style Sheets, was zusätzliche Dateien impliziert. Dies ist jedoch nicht immer notwendig. CSS lässt sich auf 3 möglichen Wegen verwenden:

1. Inline, indem man ein style-Attribut innerhalb eines HTML-Tags verwendet, falls man tatsächlich nur die Formatierung dieses einen Tags ändern will.
2. Intern innerhalb einer Datei, indem man ein `<style>`-Tag innerhalb des `<head>`-Bereichs der HTML-Datei einfügt, falls man die Formatierung innerhalb dieser einen Datei ändern will.
3. Extern, indem man über ein `<link>`-Tag im `<head>`-Bereich der HTML-Datei eine externe CSS-Datei einbindet. Diese Datei kann man in verschiedenen HTML-Dateien einbinden und so die Formatierung für mehrere Webseiten zentral verwalten.

Wir werden nur die externe Variante verwenden. Die anderen Varianten sind von der Verwendung her gleich nur der Ort, an dem man die Formatierungsanweisungen angibt, ändert sich.

Übung 8

1. Erzeuge mit dem Editor eine leere Datei und speichere diese unter *style.css* im selben Ordner wie *schueler.html*. Auf die Dateiendung achten! Nicht als *style.css.txt* speichern.
2. Binde die CSS-Datei in die Schülerseite im `<head>`-Bereich von *schueler.html* ein:
   `<link href="style.css"rel="stylesheet"type="text/css">`

In der CSS-Datei kann man nun für verschiedene Tags die Formatierung ändern. Die Syntax einer CSS-Datei dazu sieht wie folgt aus:
Name des Tags (mehrere kommaseparierte Tags zulässig) { Name der zu ändernden Eigenschaft : neuer Wert der Eigenschaft; }
Will man z.B. die Farbe der größten Überschriften in Blau ändern, so fügt man folgenden Eintrag hinzu: `h1 {color: blue;}`
Man kann in den geschweiften Klammern beliebig viele Eigenschaften ändern. Die Namen und zulässigen Werte können z.B. bei w3cschools gefunden werden.

Übung 9

Nimm (über die CSS-Datei) folgende Änderungen für die bisherige Seite vor:

1. Die Hintergrundfarbe soll auf gelb gesetzt werden (informiere dich in diesem Zusammenhang über die Farbangabe mit Hilfe des RGB-Codes).
2. Alle Überschriften sollen immer Rot und unterstrichen dargestellt werden.
3. Die Überschriften ersten Grades sollen immer zentriert ausgegeben werden.

Validiere dann deine Datei *schueler.html*.

6 Listen

Damit man etwas mehr über den Webseitenersteller erfährt, soll die Webseite um zwei Listen nach folgendem Muster erweitert werden: wissen, um wessen Website es sich dabei handelt. Daher soll eine Überschrift eingefügt werden.

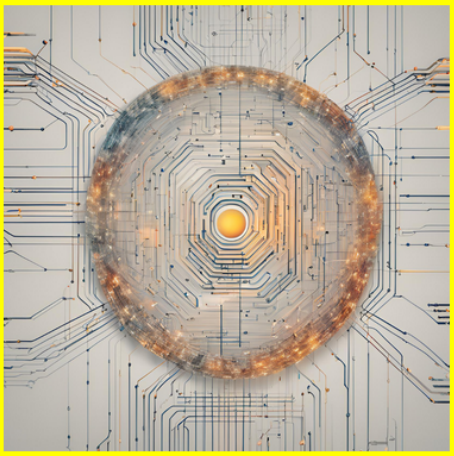
Übung 10

1. Erweitere deine *schueler.html* mit dem Windows-Editor entsprechend des obigen Bildes. Recherchiere dazu die Tags für Listen (geordnete und ungeordnete Listen).
2. Validiere dann deine Datei *schueler.html*.
3. Nimm über die CSS-Datei folgende Änderungen vor (wie auch im Screenshot zu sehen):
   - Der Aufzählungstyp für ungeordnete Listen soll nun quadratisch sein.
   - Der Aufzählungstyp für geordnete Listen sollen nun große römische Ziffern sein und die geordneten Listen sollen in Grün dargestellt werden.

7 Bilder

Eine Webseite lebt von Bildern. Um die Webseite optisch ansprechender zu machen, wird ein Bild eingefügt (Das Bild wurde mittels canva.com KI-erzeugt):

Haralds Webseite



Willkommen auf meiner Webseite!

Ich bin Schüler des **WG West in Stuttgart**.

Was ich **gut** finde:

- Informatik-Unterricht
- HTML
- CSS

Was ich *nicht* gut finde:

I. Listen
II. Ironie
III. Wiederholungen
IV. Listen

Übung 11

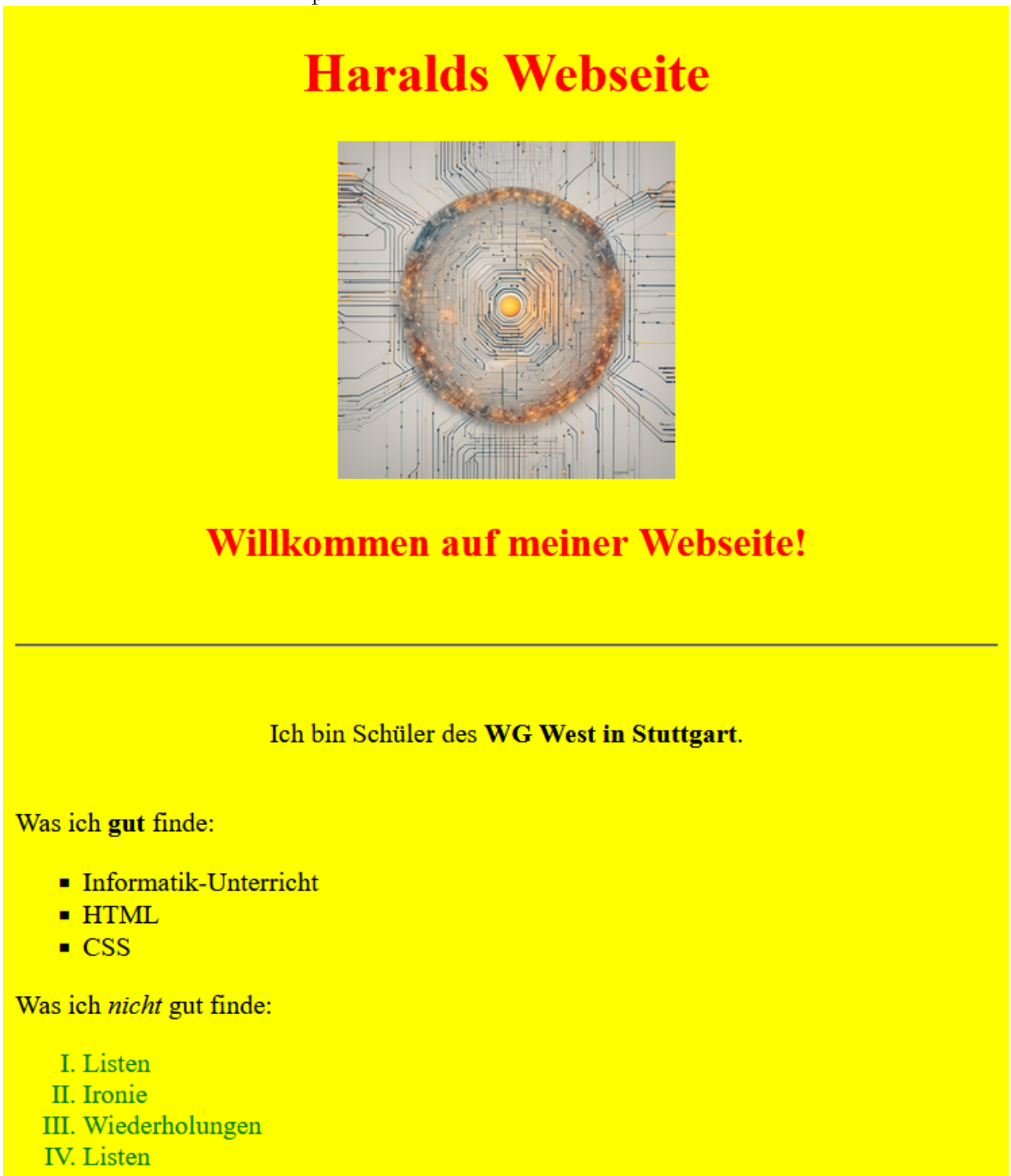
1. Lade dir Zwei Bilder aus dem Internet herunter und speichere diese in einem Ordner Bilder, der im gleichen Verzeichnis wie *schueler.html* liegt. Kurze Dateinamen machen das Einbinden leichter.

2. Binde in deiner *schueler.html* eines der beiden Bilder ein. Recherchiere dazu den Tag für das Einfügen von Bildern. Erkundige dich, welche Attribute für die Höhen- und Breitenangabe von Bildern notwendig sind und was der Alternativtext bei Bildern bedeutet.

Informiere dich über die wesentlichen Bildformate (Rasterbilder wie jpg, gif, png oder Vektorgrafiken wie svg) und deren Unterschiede. Wichtige Begriffe sind hier unter anderem Dateigröße, Komprimierung, Transparenz, Farbraum.

8 Abschnitte definieren und formatieren

Häufig will man einen ganzen Abschnitt einer Webseite gleich formatieren, z.B. wollen wir den oberen Teil der Webseite komplett zentrieren:



Dazu bietet HTML das `<div>...</div>`-Tag an. Dabei steht div für division bzw. Abschnitt auf Deutsch. Das Tag kann man um einen beliebig großen Teil des bodys der Webseite schreiben. Man kann beliebig viele solcher Abschnitte definieren und jeden solchen Abschnitt dann formatieren. Dazu erstellen wir in der CSS-Datei eine eigene Klasse.

Bisher haben wir in der CSS-Datei Eintragungen vom Typ
`tagname {attributName1:Wert; attributName2:Wert;}`. Der Syntax für eigene Klassen ist dem für Tags ganz ähnlich. Wir beginnen nun nur mit einem Punkt statt dem Namen des Tags: `.klassenName {attributName1:Wert, attributName2:Wert;}`.

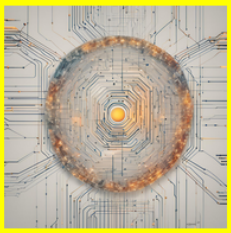
Übung 12

1. Erstelle eine eigene Klasse mit beliebigem Namen in deiner CSS-Datei mit dem Attribut `text-align` und dem Wert `center`.
2. Ändere dein *schueler.html* so ab, dass der obere Teil bis einschließlich *WG West in Stuttgart* zentriert ist. Recherchiere dazu wie man ein `<div>`-Tag mit einer eigenen CSS-Klasse verwendet.

9 Verlinkungen

Die Vernetzung von Inhalten über so genannte Links ist ein wesentliches Merkmal des WWW. Aber auch eine normale Homepage besteht in der Regel bereits aus mehreren Seiten, die mit einander verlinkt sind. Man kann aber auch auf die Seiten anderer Anbieter eine Verlinkung setzen.

Haralds Webseite



Willkommen auf meiner Webseite!

Ich bin Schüler des **WG West in Stuttgart**.

Was ich **gut** finde:

- Informatik-Unterricht
- HTML
- CSS

Was ich *nicht* gut finde:

I. Listen
II. Ironie
III. Wiederholungen
IV. Listen

Das ist meine Klasse:

Das ist mein Stundenplan.

Übung 13

1. Erstelle eine Datei *stundenplan.html* im gleichen Ordner wie *schueler.html* mit folgendem Inhalt:

```
<!DOCTYPE html>
<html lang="de">
<head>
<link href="style.css" rel="stylesheet" type="text/css">
<title>Unterricht</title>
<meta charset="utf-8">
</head>
<body>
So sieht mein Stundenplan aus:
</body>
</html>
```

2. Ergänze deine Webseite wie im Bild zu sehen (du kannst ein beliebiges Bild einbinden. Das Beispielbild wurde mittels canva.com KI-erzeugt). Hinweis: Die beiden Bilder wurden verkleinert, damit das Beispielbild nicht zu groß wird. Für das zweite Bild kannst du wieder ein beliebiges Bild aus dem Internet verwenden. Stelle eine Höhe von 400 Pixeln ein.
3. Erstelle einen Link von WG West in Stuttgart auf die Homepage des WG Wests (https://www.wg-west.de/).
4. Erstelle einen Link von Stundenplan auf *stundenplan.html*.
5. Je nach Größe der eingefügten Bilder ist die Webseite inzwischen so groß, dass nun Scrollen nach unten notwendig ist. Um direkt auf das Bild der Klasse (und weitere Informationen) zu kommen, soll eine Verknüpfung von Schüler direkt auf das Bild führen. Informiere dich über diese Art der Verknüpfung innerhalb einer Datei auf eine ID.
6. Validiere dann deine Datei *schueler.html*.

10 Tabellen

Bei der Gestaltung einer Website sollte immer die Übersichtlichkeit im Vordergrund stehen. Um Inhalte, Schrift oder Bilder strukturiert darstellen zu können, benötigt man häufig Tabellen. Mit Hilfe einer Tabelle kann man z.B. den Stundenplan übersichtlich darstellen.

Übung 14 Erstelle deinen Stundenplan in *stundenplan.html*

1. Informiere dich über Tabellen. Beachte dabei den Unterschied zwischen Tabellenüberschriften und normalen Tabellendaten.
2. Erstelle einen Beispielstundenplan (Bsp. siehe Bilder weiter unten).
3. Verbinde alle Zellen der letzten Spalte und füge eine Verlinkung zu *schueler.html* ein.

Deine Webseite sollte dann ungefähr so aussehen:

So sieht mein Stundenplan aus:

	Mo	Di	Mi	Do	Fr
1./2. Stunde	Mathe	BWL	BWL	Deutsch	Üfa
3./4. Stunde	Informatik	Wirtschaft	Englisch	Reli	Üfa
5./6. Stunde	Deutsch	GGK	SK		Üfa

Hier gehts zurück

Übung 15 Formatiere deine Tabelle mit Hilfe von CSS

1. Der Inhalt der Tabelle soll zentriert sein.
2. Die Tabelle und die einzelnen Zellen sollen schwarze Gitternetzlinien bekommen.
3. Die Tabellenüberschriften sollen kursiv sein.
4. Die Tabellenüberschriften und -inhalte sollen blau sein.
5. Die Tabelle soll zentriert und 800px breit sein.
6. Validiere dann deine Datei *stundenplan.html*.

So sieht mein Stundenplan aus:

	Mo	*Di*	*Mi*	*Do*	*Fr*
1./2. Stunde	Mathe	BWL	BWL	Deutsch	Üfa
3./4. Stunde	Informatik	Wirtschaft	Englisch	Reli	Üfa
5./6. Stunde	Deutsch	GGK	SK		Üfa
Hier gehts zurück					

11 Lösungen

Anbei Beispiele für die HTML bzw. CSS-Dateien nach jedem Kapitel. Insbesondere die CSS-Datei kann auch anders aufgebaut werden.

1. Begriffe und Grundgerüst

   schuler.html:

```
<html>
<head>
<title>Schülerseite</title>
</head>
<body>
Willkommen auf meiner Webseite!
<br>
<br>
<p>Ich bin Schüler des WG West in Stuttgart.</p>
</body>
</html>
```

2. Umlaute und Sonderzeichen

   schuler.html:

```
<html>
<head>
<title>Schülerseite</title>
<meta charset="utf-8">
</head>
<body>
Willkommen auf meiner Webseite!
<br>
<br>
<p>Ich bin Schüler des WG West in Stuttgart.</p>
</body>
</html>
```

3. Validieren

   schuler.html:

```
<!DOCTYPE html>
<html lang="de">
<head>
<title>Schülerseite</title>
<meta charset="utf-8">
</head>
```

```
<body>
Willkommen auf meiner Webseite!
<br>
<br>
<p>Ich bin Schüler des WG West in Stuttgart.</p>
</body>
</html>
```

4. Textauszeichnungen
   schuler.html:

```
<!DOCTYPE html>
<html lang="de">
<head>
<title>Sch&uuml;lerseite</title>
<meta charset="utf-8">
</head>
<body>
<h1>Haralds Webseite</h1>
<h2>Willkommen auf meiner Webseite!</h2>
<br>
<hr>
<br>
<p>Ich bin Sch&uuml;ler des <strong>WG West in Stuttgart</strong>.</p>
</body>
</html>
```

5. Cascading Style Sheets
   schuler.html:

```
<!DOCTYPE html>
<html lang="de">
<head>
<link href="style.css" rel="stylesheet" type="text/css">
<title>Sch&uuml;lerseite</title>
<meta charset="utf-8">
</head>
<body>
<h1>Haralds Webseite</h1>
<h2>Willkommen auf meiner Webseite!</h2>
<br>
<hr>
<br>
```

```
<p>Ich bin Sch&uuml;ler des <strong>WG West in Stuttgart</strong>.</p>
</body>
</html>
```

style.css

```
body {background-color: yellow;}
h1 {text-align: center;}
h1, h2 {color: red}
```

6. Listen

   schuler.html:

```
<!DOCTYPE html>
<html lang="de">
<head>
<link href="style.css" rel="stylesheet" type="text/css">
<title>Sch&uuml;lerseite</title>
<meta charset="utf-8">
</head>
<body>
<h1>Haralds Webseite</h1>
<h2>Willkommen auf meiner Webseite!</h2>
<br>
<hr>
<br>
<p>Ich bin Sch&uuml;ler des <strong>WG West in Stuttgart</strong>.</p>
<br>
Was ich <strong>gut</strong> finde:
<ul>
<li>Informatik-Unterricht</li>
<li>HTML</li>
<li>CSS</li>
</ul>
Was ich <em>nicht</em> gut finde:
<ol>
<li>Listen</li>
<li>Ironie</li>
<li>Wiederholungen</li>
<li>Listen</li>
</ol>
</body>
</html>
```

style.css

```
body {background-color: yellow;}
h1 {text-align: center;}
h1, h2 {color: red;}
ul {list-style-type: square;}
ol {list-style-type: upper-roman; color: green;}
```

7. Bilder

schuler.html:

```
<!DOCTYPE html>
<html lang="de">
<head>
<link href="style.css" rel="stylesheet" type="text/css">
<title>Sch&uuml;lerseite</title>
<meta charset="utf-8">
</head>
<body>
<h1>Haralds Webseite</h1>
<img src="Bilder/Bild.png" alt="Ein von canva.com erzeugtes KI-Bild" height
   ="200">
<h2>Willkommen auf meiner Webseite!</h2>
<br>
<hr>
<br>
<p>Ich bin Sch&uuml;ler des <strong>WG West in Stuttgart</strong>.</p>
<br>
Was ich <strong>gut</strong> finde:
<ul>
<li>Informatik-Unterricht</li>
<li>HTML</li>
<li>CSS</li>
</ul>
Was ich <em>nicht</em> gut finde:
<ol>
<li>Listen</li>
<li>Ironie</li>
<li>Wiederholungen</li>
<li>Listen</li>
</ol>
</body>
</html>
```

8. Abschnitte definieren und formatieren

schuler.html:

```
<!DOCTYPE html>
<html lang="de">
<head>
<link href="style.css" rel="stylesheet" type="text/css">
<title>Sch&uuml;lerseite</title>
<meta charset="utf-8">
</head>
<body>
<div class="meineErsteKlasse">
<h1>Haralds Webseite</h1>
<img src="Bilder/Bild.png" alt="Ein von canva.com erzeugtes KI-Bild" height
   ="200">
<h2>Willkommen auf meiner Webseite!</h2>
<br>
<hr>
<br>
<p>Ich bin Sch&uuml;ler des <strong>WG West in Stuttgart</strong>.
</div>
<br>
Was ich <strong>gut</strong> finde:
<ul>
<li>Informatik-Unterricht</li>
<li>HTML</li>
<li>CSS</li>
</ul>
Was ich <em>nicht</em> gut finde:
<ol>
<li>Listen</li>
<li>Ironie</li>
<li>Wiederholungen</li>
<li>Listen</li>
</ol>
</body>
</html>
```

style.css

```
    body {background-color: yellow;}
h1 {text-align: center;}
h1, h2 {color: red;}
```

```
ul {list-style-type: square;}
ol {list-style-type: upper-roman; color: green;}

.meineErsteKlasse {
    text-align: center;
}
```

9. Verlinkungen
   schuler.html:

```
<!DOCTYPE html>
<html lang="de">
<head>
<link href="style.css" rel="stylesheet" type="text/css">
<title>Sch&uuml;lerseite</title>
<meta charset="utf-8">
</head>
<body>
<div class="meineErsteKlasse">
<h1>Haralds Webseite</h1>
<img src="Bilder/Bild.png" alt="Ein von canva.com erzeugtes KI-Bild" height
   ="200">
<h2>Willkommen auf meiner Webseite!</h2>
<br>
<hr>
<br>
<p>Ich bin <a href="#SchuelerBild">Sch&uuml;ler</a> des <strong><a href="
   https://www.wg-west.de/">WG West in Stuttgart</a></strong>.
</div>
<br>
Was ich <strong>gut</strong> finde:
<ul>
<li>Informatik-Unterricht</li>
<li>HTML</li>
<li>CSS</li>
</ul>
Was ich <em>nicht</em> gut finde:
<ol>
<li>Listen</li>
<li>Ironie</li>
<li>Wiederholungen</li>
<li>Listen</li>
```

```
</ol>
Das ist meine Klasse:
<br>
<img id="SchuelerBild" src="Bilder/Klasse.png" alt="Ein von canva.com
   erzeugtes KI-Bild meiner Klasse" height="400"><br>
Das ist mein <a href="stundenplan.html">Stundenplan</a>.
</body>
</html>
```

stundenplan.html

```
<!DOCTYPE html>
<html lang="de">
<head>
<link href="style.css" rel="stylesheet" type="text/css">
<title>Unterricht</title>
<meta charset="utf-8">
</head>
<body>
So sieht mein Stundenplan aus:
</body>
</html>
```

10. Tabellen

stundenplan.html:

```
<!DOCTYPE html>
<html lang="de">
<head>
<link href="style.css" rel="stylesheet" type="text/css">
<title>Unterricht</title>
<meta charset="utf-8">
</head>
<body>
So sieht mein Stundenplan aus:<br>
<table>
<tr>
<th></th>
<th>Mo</th>
<th>Di</th>
<th>Mi</th>
<th>Do</th>
<th>Fr</th>
```

```
</tr>
<tr>
<td>1./2. Stunde</td>
<td>Mathe</td>
<td>BWL</td>
<td>BWL</td>
<td>Deutsch</td>
<td>Üfa</td>
</tr>
<tr>
<td>3./4. Stunde</td>
<td>Informatik</td>
<td>Wirtschaft</td>
<td>Englisch</td>
<td>Reli</td>
<td>Üfa</td>
</tr>
<tr>
<td>5./6. Stunde</td>
<td>Deutsch</td>
<td>GGK</td>
<td>SK</td>
<td></td>
<td>Üfa</td>
</tr>
<tr>
<td colspan="6">Hier gehts <a href="schueler.html">zur&uuml;ck</a></td>
</tr>
</table>
</body>
</html>
```

style.css

```
body {background-color: yellow;}
h1 {text-align: center;}
h1, h2 {color: red;}
ul {list-style-type: square;}
ol {list-style-type: upper-roman; color: green;}
table, th, td {border: 1px solid black;}
th {font-style: italic;}
td, th {text-align: center; color:blue;}
```

```
table {width:800px;}

.meineErsteKlasse {
    text-align: center;
}
```