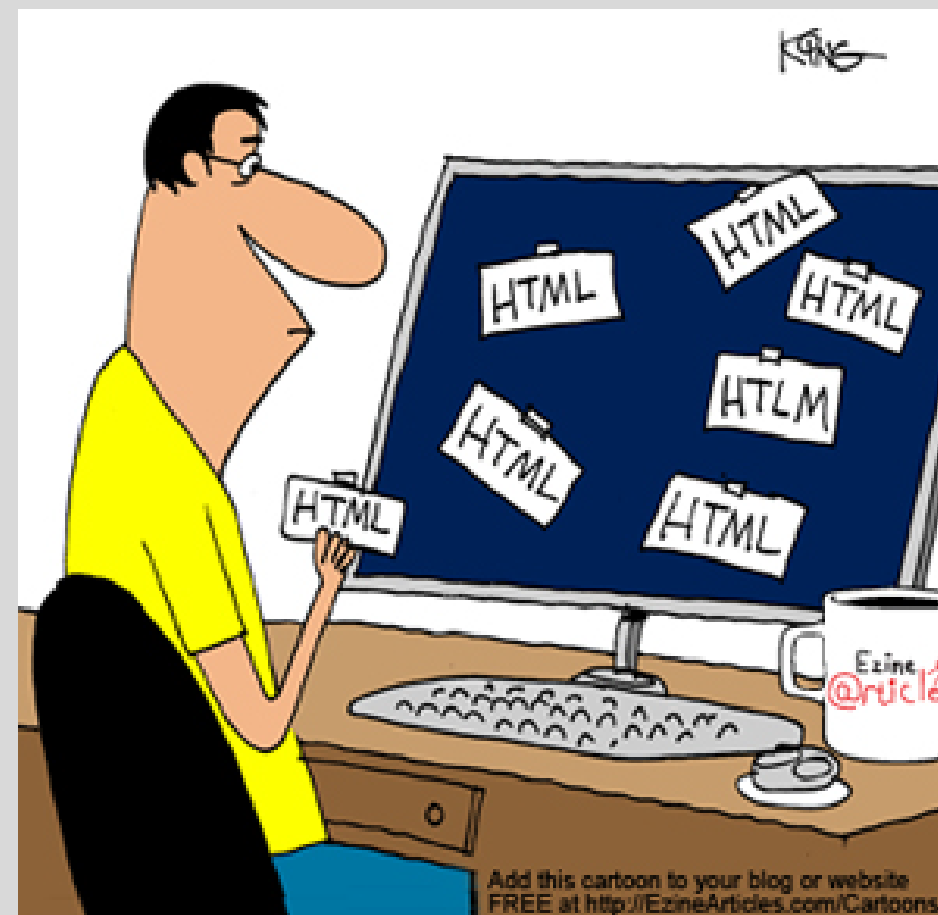




## <Tags für Text>

## Tags

1. Ab jetzt geht fast Alles was wir machen im Codeeditor.
2. Versuch und Fehler – jeweils bis zum Erfolg
3. Ich werde Niemanden zurücklassen – diese Grundlagen brauchen Sie unabhängig davon, was konkret Sie mal später machen.



*Felix always added HTML tags to his articles ... though he really questioned their effectiveness.*



## Tags für Text

### Absätze

**`<p> </p>`** bildet einen Absatz (engl. paragraph)

**`<br>`** Harter Umbruch – Text geht in nächster Zeile weiter  
(auch `</ br>` erlaubt, aber veraltet)

### Überschriften

**`<h1> </h1>`** Überschrift, die Zahl gibt den Rang an (H1 ist am wichtigsten)  
es gibt `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>` und `<h6>`



# Tags für Text

### Hervorhebungen im Text

`<strong> </strong>` fett hervorgehoben (logisch) (HTML5: dieser Text ist wichtig)

`<em> </em>` betonen von Text (oft kursiv dargestellt)



## Tags für Text

### Hervorhebungen im Text

`<sup> </sup>`      hochgestellter Text

`<sub> </sub>`      tiefgestellter Text

### Trennlinie

`<hr>`              Trennlinie ziehen



## Tags für Text

### HTML-Elemente verschachteln – zuerst geöffnet, zuletzt geschlossen

Beim Verschachteln von HTML-Elementen müssen Sie eine wichtige Grundregel beachten:

**Das zuerst geöffnete Element wird zuletzt geschlossen.**

Hier ein Beispiel:

`<p>Normal, <strong>fett, <em>fett und kursiv</em></strong>. Wieder normal.</p>`

Die Elemente p, strong und em werden geöffnet und in *umgekehrter* Reihenfolge geschlossen.

Die von den Elementen erzeugten Kästchen werden also wie russische Matroschka-Puppen ineinander verschachtelt: em steckt in strong, das wiederum in der p-Kiste sitzt.

Normal, **fett, *fett und kursiv***. Wieder normal.

## Tags für Text

HTML-Elemente verschachteln – zuerst geöffnet, zuletzt geschlossen

```
<p>Normal, <strong>fett, <em>fett und kursiv</em></strong>. Wieder normal.</p>
```



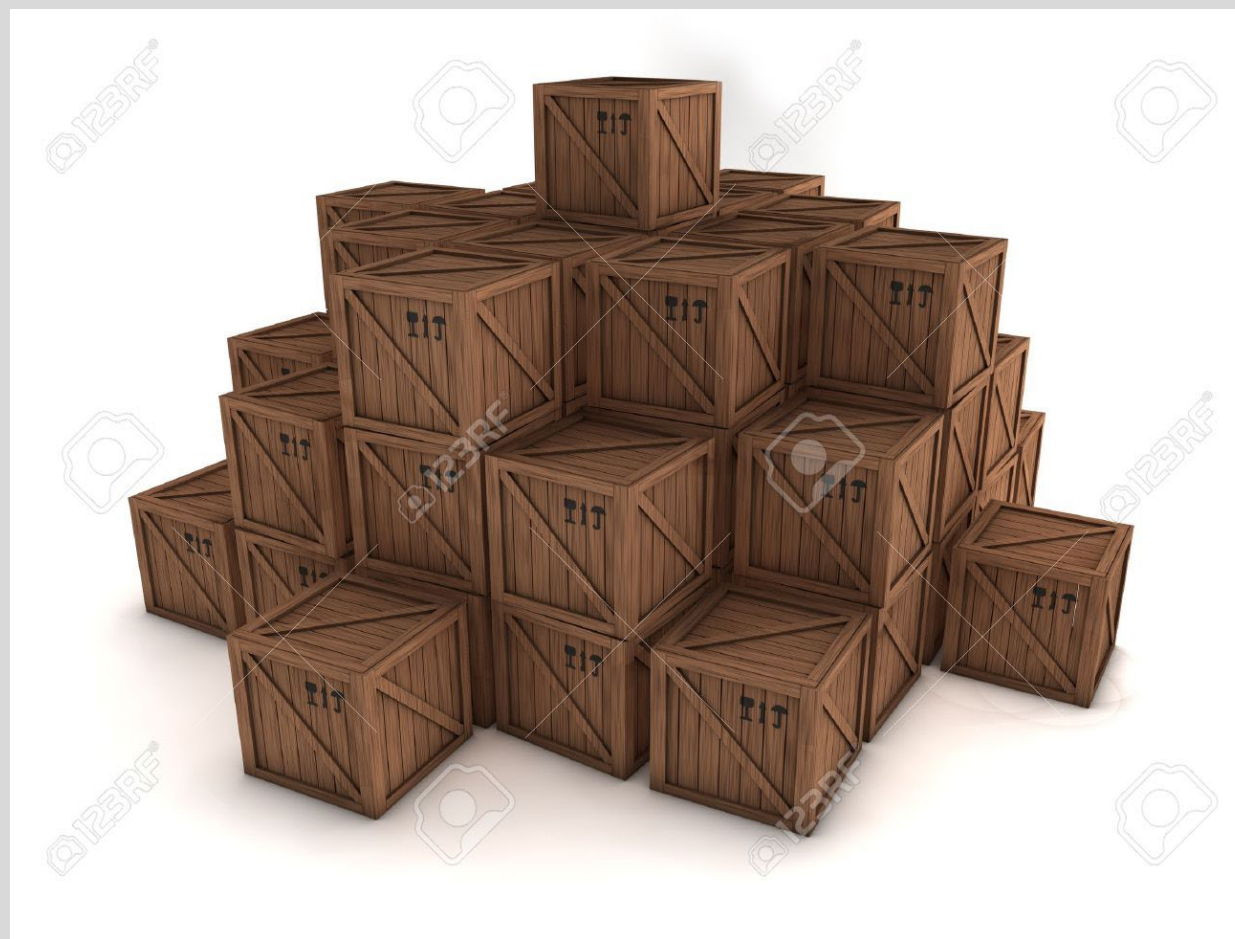


## Block- und Inline Elemente



## Tags | Block- & Inline Elemente

Mit HTML5 erzeugen wir vor allem Kästen / **Boxen** / Kisten / Rechtecke ... oder wie auch immer wir das bezeichnen wollen.





# Tags | Block- & Inline Elemente

## Über Block- und Inline-Elemente

Wie Sie bereits gesehen haben, gibt es verschiedene Arten von HTML-Elementen, die sich unterschiedlich verhalten.

Die beiden wichtigsten sind Block- und Inline-Elemente.

## Blockelemente werden so breit, wie es geht!

Blockelemente ähneln Absatzformaten in Word:

- Der Kasten (die Box) eines Blockelements wird automatisch so breit, wie es geht.
- Nachfolgende Elemente stehen unterhalb des Kastens in der nächsten Zeile.

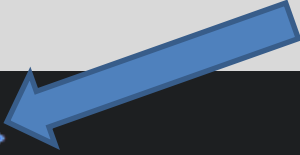
**Blockelemente enthalten normalen Text, Inline-Elemente und manchmal auch andere Blockelemente.**

Beispiele für Blockelemente sind die **Überschriften h1 bis h6**, **Fließtextabsätze** mit **p** sowie die **Listenelemente ul, ol und li** (die Sie gleich kennenlernen werden).

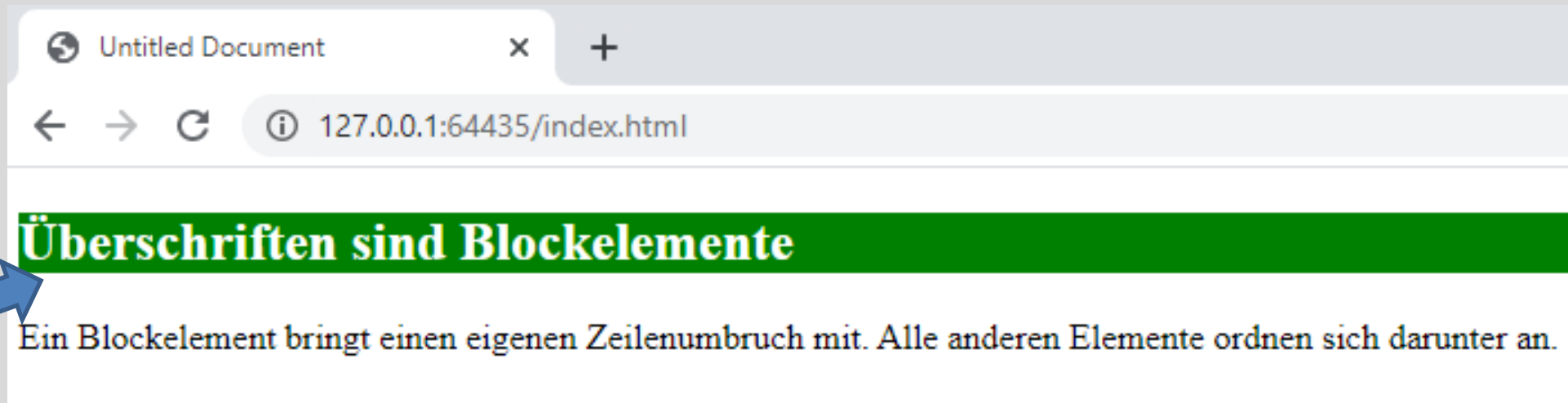


## Tags | Block- & Inline Elemente

```
<h2>Überschriften sind Blockelemente</h2>
```



```
<p>Ein Blockelement bringt einen eigenen Zeilenumbruch mit.  
Alle anderen Elemente ordnen sich darunter an.</p>
```





# Tags | Block- & Inline Elemente

**Inline-Elemente werden nur so breit wie ihr Inhalt**

**Inline-Elemente ähneln den Zeichenformaten aus Word:**

- Der Kasten eines Inline-Elements wird nur so breit wie sein Inhalt.
- Nachfolgender Text fließt direkt nach dem Element weiter.

**Inline-Elemente erzeugen keine neue Zeile und sind den Blockelementen untergeordnet. Inline Elemente dürfen normalen Text und andere Inline-Elemente enthalten, aber keine Blockelemente.**

Beispiele für Inline-Elemente sind **strong** und **em** sowie **Hyperlinks (a)** und **Bilder (img)**.

Hier ein link zu einer Liste aller Block und Inline Elemente:

[https://developer.mozilla.org/de/docs/Web/HTML/Block-level\\_elemente](https://developer.mozilla.org/de/docs/Web/HTML/Block-level_elemente)

[https://developer.mozilla.org/de/docs/Web/HTML/Inline\\_elemente](https://developer.mozilla.org/de/docs/Web/HTML/Inline_elemente)

## Tags | Block- & Inline Elemente

Inline-Elemente werden nur so breit wie ihr Inhalt

```
<h2>Überschriften sind Blockelemente</h2>
```

```
<p>Ein <em>Blockelement</em> bringt einen <strong>eigenen  
Zeilenumbruch</strong> mit.
```

```
<br>Alle anderen Elemente ordnen sich darunter an.</p>
```

← → ↻ ⓘ 127.0.0.1:64435/index.html

### Überschriften sind Blockelemente

Ein **Blockelement** bringt einen **eigenen Zeilenumbruch** mit.  
Alle anderen Elemente ordnen sich darunter an.



## Listen

## Tags für Text | Listen | ungeordnete Liste

Eine **ungeordnete Liste** (`<ul...ul>`) ist im Grunde nichts anderes als eine nicht nummerierte Aufzählungsliste, in der alle Listeneinträge mit einem Aufzählungszeichen versehen werden. **Die Webbrowser stellen dieses Aufzählungszeichen gewöhnlich mit einem Gliederungspunkt (auch Bullet Point genannt) dar.**

Eine solche Liste leiten Sie mit einem öffnenden `<ul>` (`ul` = unordered list) ein, gefolgt von den eigentlichen Aufzählungspunkten, die Sie zwischen `<li>` und `</li>` schreiben. Jedes `li`-Element (`li` = list item, deutsch: Listeneintrag) ist dabei ein Aufzählungspunkt.

Am Ende müssen Sie die ungeordnete Aufzählungsliste **mit dem schließenden `</ul>` beenden**. Zwischen `<ul>` und `</ul>` dürfen nur **li-Elemente enthalten sein**. Innerhalb der `li`-Elemente können Sie auch andere HTML-Elemente verwenden (mit Ausnahme von Sektionselementen).

### ungeordnete Liste

- Listenpunkt 1
- Listenpunkt 2
- Listenpunkt 3
- Listenpunkt 4
- Listenpunkt 5



## Tags für Text | Listen | ungeordnete Liste

### Aufzählungen und Listen | Auflistung mit Spiegelstrichen

<code>&lt;ul&gt;</code>	Start einer Aufzählung (durch ul mit Spiegelstrichen)
<code>&lt;li&gt; &lt;/li&gt;</code>	Aufzählungspunkt
<code>&lt;li&gt; &lt;/li&gt;</code>	nächster Aufzählungspunkt
<code>&lt;/ul&gt;</code>	Ende einer Aufzählung

```
<ul>
  <li>Listenpunkt 1</li>
  <li>Listenpunkt 2</li>
  <li>Listenpunkt 3</li>
  <li>Listenpunkt 4</li>
  <li>Listenpunkt 5</li>
</ul>
```



## Tags für Text | Listen | geordnete Liste

Das, was Sie gerade zum `ul`-Element gelesen haben, gilt auch für das `ol`-Element (`ol` = ordered list).  
**Nur ist das `ol`-Element eine geordnete Liste – genauer, eine nummerierte Liste, in der die einzelnen `li`-Elemente automatisch durchnummeriert werden.**

### Nummerierte Auflistung

<code>&lt;ol&gt;</code>	im Gegensatz zu <code>&lt;ul&gt;</code> Aufzählungen mit Nummerierung
<code>&lt;li&gt; &lt;/li&gt;</code>	Aufzählungspunkt
<code>&lt;li&gt; &lt;/li&gt;</code>	nächster Aufzählungspunkt
<code>&lt;/ol&gt;</code>	Ende einer Aufzählung

```
<h2>geordnete Liste</h2>
<ol>
  <li>Listenpunkt 1</li>
  <li>Listenpunkt 2</li>
  <li>Listenpunkt 2</li>
  <li>Listenpunkt 4</li>
  <li>Listenpunkt 5</li>
</ol>
```

### geordnete Liste

1. Listenpunkt 1
2. Listenpunkt 2
3. Listenpunkt 2
4. Listenpunkt 4
5. Listenpunkt 5



## Tags für Text | Listen

### ungeordnete Liste

- Listenpunkt 1
- Listenpunkt 2
- Listenpunkt 3
- Listenpunkt 4
- Listenpunkt 5

### geordnete Liste

1. Listenpunkt 1
2. Listenpunkt 2
3. Listenpunkt 3
4. Listenpunkt 4
5. Listenpunkt 5

```
<h2>ungeordnete Liste</h2>
```

```
<ul>
```

```
<li>Listenpunkt 1</li>
```

```
<li>Listenpunkt 2</li>
```

```
<li>Listenpunkt 3</li>
```

```
<li>Listenpunkt 4</li>
```

```
<li>Listenpunkt 5</li>
```

```
</ul>
```

```
<br>
```

```
<h2>geordnete Liste</h2>
```

```
<ol>
```

```
<li>Listenpunkt 1</li>
```

```
<li>Listenpunkt 2</li>
```

```
<li>Listenpunkt 3</li>
```

```
<li>Listenpunkt 4</li>
```

```
<li>Listenpunkt 5</li>
```

```
</ol>
```

## Tags für Text | Listen | geordnete Liste

### Nummerierung einer geordneten Liste umkehren

Mit HTML5 wurde die Möglichkeit eingeführt, die Reihenfolge der **Nummerierung** mit dem **HTML-Attribut reversed** im sich öffnenden `<ol>`-Tag **umzukehren**, sodass die Nummerierung in absteigender Reihenfolge dargestellt wird.

```
<ol reversed>
  <li>Listenpunkt 1</li>
  <li>Listenpunkt 2</li>
  <li>Listenpunkt 2</li>
  <li>Listenpunkt 4</li>
  <li>Listenpunkt 5</li>
</ol>
```

```
5. Listenpunkt 1
4. Listenpunkt 2
3. Listenpunkt 2
2. Listenpunkt 4
1. Listenpunkt 5
```

Neben `<ol reversed="reversed">` können Sie hier auch nur **`<ol reversed>`** verwenden, weil es sich um ein allein stehendes Attribut handelt.

## Tags für Text | Listen | geordnete Liste

### Die Nummerierung mit den Attributen »start« und »type« kontrollieren

Sie können dem Element `ol` mit dem Attribut `start` einen anderen Startwert geben:

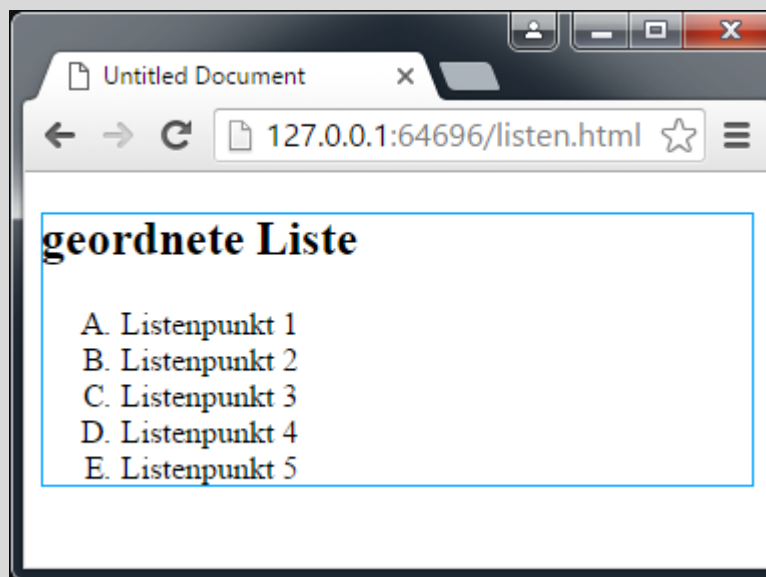
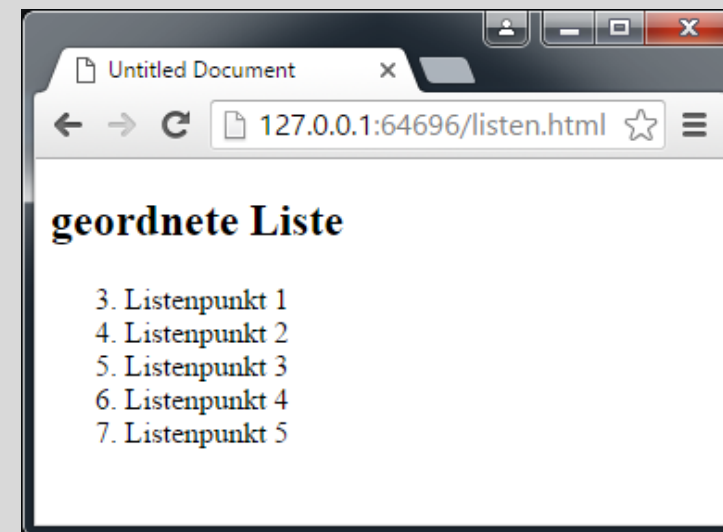
`<ol start=„3“> ... </ol>`

In diesem Fall beginnt die Liste nicht mit 1, sondern mit, ja, genau, 3.

Außerdem können Sie mit dem Attribut `type` die Art der Nummerierung beeinflussen:

`<ol type="A"> ... </ol>`

Das Attribut **reversed** verursacht eine umgekehrte Aufzählung.





# Tags für Text | Listen verschachteln

Ungeordnete und geordnete Listen können verschachtelt werden, um mehrstufige Aufzählungen zu erzeugen.

**Die ungeordnete Liste ist der populärste Struktur für die Navigation auf Webseiten:**

**Das HTML-Markup ist einfach, spiegelt die Hierarchie der Inhalte wieder und gibt Benutzern mit Readern einen direkten Überblick.**

Die geordnete Liste mit HTML ol ist das Gegenstück zur ungeordneten Liste.

HTML ol setzt einen Index vor jeden Listeneintrag.

Geordnete und ungeordnete Listen können Zusammenhänge im Schriftbild einer Seite besser herausstellen als einfache Textabsätze.

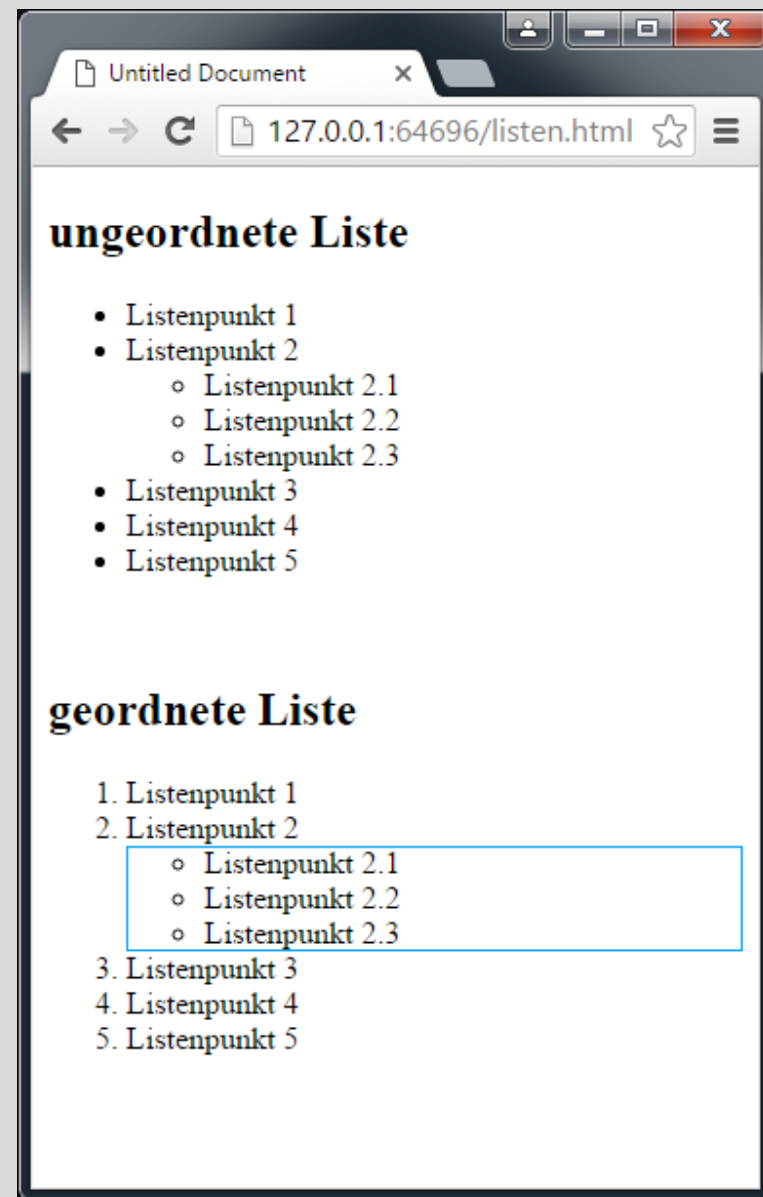
## Tags für Text | Listen verschachtelt

Ungeordnete und geordnete Listen können verschachtelt werden, um mehrstufige Aufzählungen zu erzeugen.

Natürlich können Sie die Listen noch tiefer verschachteln. Auch das Mischen von ungeordneten und geordneten Listen ist ohne Probleme möglich.

Leider ist es nicht möglich, bei den nummerierten Unterlisten eine automatische Nummerierung wie 1.2; 1.3; 1.4 usw. zu erzwingen.

```
<ul>
  <li>Home</li>
  <li>Wir über uns</li>
  <li>Produkte
    <ul>
      <li>Produkt 1</li>
      <li>Produkt 2</li>
    </ul>
  </li>
  <li>Dienstleistungen</li>
  <li>Kontakt</li>
</ul>
```



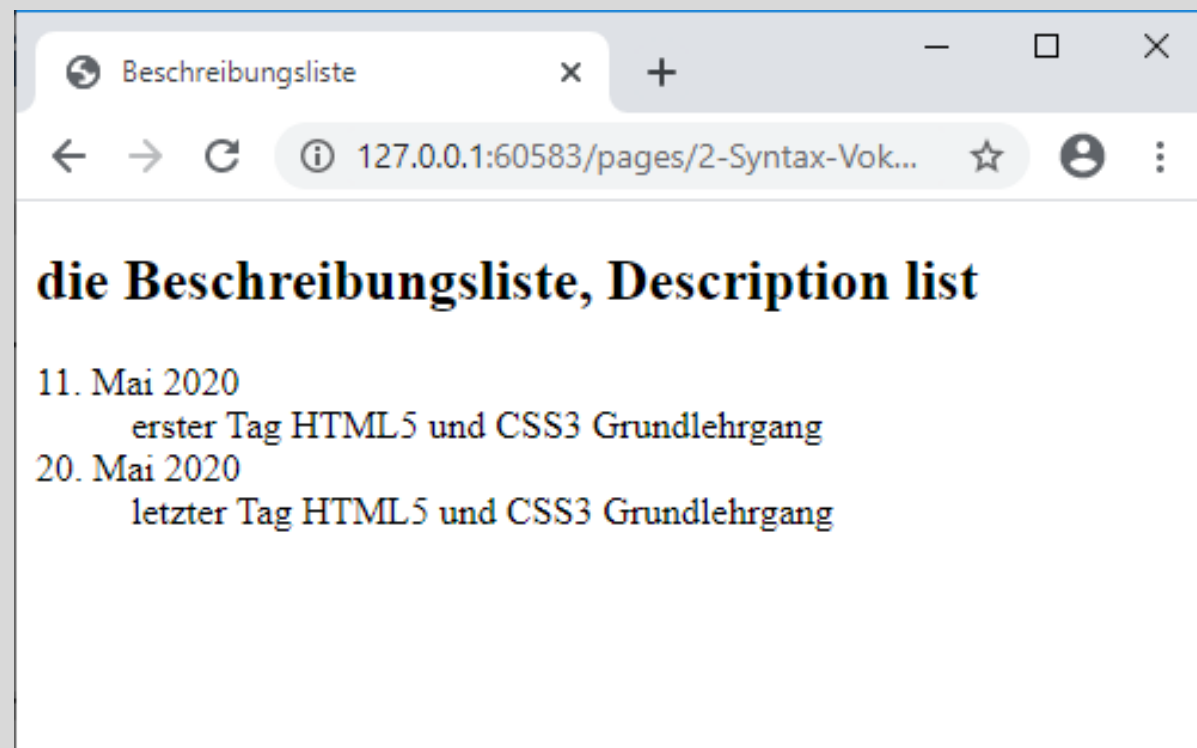
## Tags für Text | Listen | Eine Beschreibungsliste mit <dl>, <dt> und <dd> erstellen

In HTML existiert mit der Beschreibungsliste noch eine weitere Form von Liste, die Sie verwenden können. Während in HTML 4.01 diese Liste noch als Definitionsliste bezeichnet wurde, wurde sie in HTML5 semantisch als Beschreibungsliste umgedeutet.

Bei dieser Liste handelt es sich eher um eine Name-Wert-Zuordnungsliste.

Typische Anwendungsfälle für die Beschreibungsliste sind Glossare.

Auch die Auflistung spezieller Metadaten und Werte, eben einfach eine spezielle Liste mit bestimmten Daten, wo ein Wert oder eben eine Beschreibung zugeordnet wird.



## Tags für Text | Listen | Eine Beschreibungsliste mit <dl>, <dt> und <dd> erstellen

Eine Beschreibungsliste fassen Sie zwischen **<dl> und </dl>** zusammen (**dl = description list**, deutsch: Beschreibungsliste).

```
<dl> ... </dl>
```

Im dl-Element dürfen nur **dt- und dd-** Elemente enthalten sein.

- Der zu beschreibende Ausdruck, also der Name der Name-Wert-Zuordnungsliste, wird mit **<dt> und </dt>** gekennzeichnet (**dt = description term**, deutsch: Beschreibungsausdruck).

```
<dl>  
  <dt></dt><!--Name der Name-Wert-Zuordnungsliste-->
```

- Die dazugehörige Beschreibung wird hinter dem dt-Element zwischen **<dd> und </dd>** geschrieben (**dd = definition description**, deutsch: Beschreibung).

```
<dd></dd><!--dazugehörige Beschreibung-->
```

In dt und dd-Elementen wiederum dürfen auch andere HTML-Elemente verwendet werden – ausgenommen gruppierende Elemente (Grouping) und HTML-Elemente für neue Sektionen (Sectioning).



## Tags für Text | Listen | Eine Beschreibungsliste mit <dl>, <dt> und <dd> erstellen

```
<dl><!--Anfang der Beschreibungsliste-->  
  <dt>11. Mai 2020</dt><!--Begriff, Ausdruck der beschrieben werden soll-->  
  <dd>erster Tag HTML5 und CSS3 Grundlehrgang</dd><!--erklärender Text / Beschreibung-->  
  <dt>20. Mai 2020</dt>  
  <dd>letzter Tag HTML5 und CSS3 Grundlehrgang</dd>  
</dl><!--Ende der Beschreibungsliste-->
```



Eine solche Name/Wert-Paarliste kann ebenfalls innerhalb von anderen HTML-Elementen wie beispielsweise zwischen **<aside>** und **</aside>** oder dem neuen **details-Element** verwendet werden,



## Tags für Text | Listen | Eine Beschreibungsliste mit dl verschachteln

Definitionslisten mit HTML dl können ineinander verschachtelt werden.

```
<h2>verschachtelte Beschreibungsliste</h2>
<dl>
  <dt>11.Mai 2020</dt>
  <dd>erster Tag HTML5 / CSS3 Grundlehrgang
    <dl>
      <dt>Kennenlernen</dt><dd>Wer bin ich? / Wer seid Ihr?</dd>
      <dt>Absprachen und Regeln</dt><dd>Wie arbeiten wir die nächsten 8 Tage zusammen</dd>
    </dl>
  </dd>
</dl>
```

### verschachtelte Beschreibungsliste

11.Mai 2020  
erster Tag HTML5 / CSS3 Grundlehrgang

Kennenlernen  
Wer bin ich? / Wer seid Ihr?

Absprachen und Regeln  
Wie arbeiten wir die nächsten 8 Tage zusammen



**details | summary**

# Tags für Text | details | summary

Das **details-Element** ermöglicht es, Seiteninhalte zu verstecken und mit einem Klick auf **summary** aufzuklappen. Dieses Verhalten wird oft als "**Akkordeon**" bezeichnet, vor allem wenn mehrere solcher aufklappbarer Elemente vorhanden sind. Früher konnte ein solches Akkordeon nur mit vielen Zeilen JavaScript realisiert werden. Dabei wurde jedoch oft auf eine Bedienbarkeit mit der Tastatur keine Rücksicht genommen. Das **details-Element** bietet nun eine zugängliche und einfache Alternative.

```
<body>
  <h1>Akkordeon Verhalten mit details und summary</h1>
  <details>
    <summary>
      "Akkordeon" Verhalten ohne JavaScript
    </summary>
    <p> Das neue details-Element ermöglicht es, ganz ohne den Einsatz von JavaScript,
      ursprünglich verborgene Textinhalte aufzuklappen. </p>
  </details>
</body>
```

## Tags für Text | details / summary

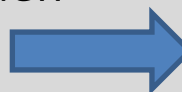
**details** bezeichnet einen semantischen Abschnitt, der mit Text, Bildern (oder einem figure-Element, einem Formular oder auch anderen details) gefüllt werden kann.

Ein Klick mit der Maus öffnet den versteckten Abschnitt. Alternativ bringt die Tab-Taste die Überschrift summary in den Fokus. Ein Klick auf Leertaste oder Enter öffnet dann den Text.

### Überschrift mit summary

Mit **summary** können Sie dem Akkordeon eine Überschrift geben. Links daneben wird ein Detailsmarker-Pseudoelement in Form eines schwarzen Dreiecks angezeigt.

Es verändert bei einer Zustandsänderung seine Richtung.



## Tags für Text | details / summary

### open

Mit dem Klick auf die summary wird details das Attribut **open** angehängt. Um die Details direkt anzuzeigen, können Sie es auch direkt im HTML-Code notieren. Es eignet sich auch als Zugriffspunkt, um das betreffende Element mit CSS zu gestalten.

```
<body>
<h1>Akkordeon Verhalten mit details und summary
<details open>
  <summary>
    "Akkordeon" Verhalten ohne JavaScript
  </summary>
  <p> Das neue details-Element ermöglicht es, ganz ohne den Einsatz von JavaScript,
    ursprünglich verborgene Textinhalte aufzuklappen.
  </p>
</details>
```





## </Tags für Text>