

# LU02c - Elemente um HTML zu strukturieren

## Lernziele:



- Den Unterschied zwischen Block- und Inline-Elementen verstehen
- Die grundlegenden Eigenschaften und das Verhalten von Block- und Inline-Elementen kennen
- Beispiele für Block- und Inline-Elemente identifizieren können
- Wissen, wie man Block- und Inline-Elemente verwendet, um eine Webseite zu gestalten

## Block- und Inline-Elemente

Sie werden hauptsächlich auf 2 Arten von HTML-Elementen stoßen:

**Block-Elemente** wie:

- Absätze `<p>`
- Listen: ungeordnete (mit Aufzählungspunkten) `<ul>` oder geordnete Listen (mit Zahlen) `<ol>`
- Überschriften: von der 1. Ebene `<h1>` bis zur 6. Ebene `<h6>`
- artikel `<artikel>`
- abschnitte `<section>`

**Inline-Elemente** wie:

- Links `<a>`
- hervorgehobene Wörter `<em>`
- wichtige Wörter `<strong>`
- kurze Zitate `<q>`

**Block-Elemente** dienen dazu, die wichtigsten Teile Ihrer Seite zu strukturieren, indem sie Ihren Inhalt in zusammenhängende Blöcke unterteilen.

**Inline-Elemente** dienen dazu, einen Teil eines Textes zu differenzieren, um ihm eine bestimmte Funktion oder Bedeutung zu geben. Inline-Elemente bestehen in der Regel aus einem einzigen oder wenigen Wörtern.

## Öffnende und schließende Tags

Alle Elemente auf Blockebene haben einen öffnenden und einen schließenden Tag.

Folglich sind selbstschließende Elemente Inline-Elemente, einfach weil ihre Syntax es nicht zulässt, dass sie ein anderes HTML-Element enthalten.

	Mit öffnenden und schließenden Tags	Selbstschließend
<b>Block-Elemente</b>	<code>&lt;p&gt; &lt;/p&gt;</code> <code>&lt;ul&gt; &lt;/ul&gt;</code> <code>&lt;ol&gt; &lt;/ol&gt;</code>	<b>Nicht möglich</b>
<b>Inline Elemente</b>	<code>&lt;a&gt; &lt;/a&gt;</code> <code>&lt;strong&gt; &lt;/strong&gt;</code> <code>&lt;em&gt; &lt;/em&gt;</code>	<code>&lt;input&gt;</code> <code>&lt;br&gt;</code> <code>&lt;img&gt;</code>

## Hierarchie von HTML Dokumenten

Ein HTML-Dokument ist wie ein großer Stammbaum, mit Eltern, Geschwistern, Kindern, Vorfahren und Nachkommen. Er entsteht durch die Möglichkeit, HTML-Elemente ineinander zu verschachteln.

## Verschachtelung

Schreiben wir einen einfachen Absatz und verbessern ihn, indem wir Teile des Textes durch zwei Inline-Elemente differenzieren:

```
<p>  
  Sir <strong>Alex Ferguson</strong> once said about Filippo Inzaghi:<q>"That  
  lad must have been born offside."</q>.  
</p>
```

Sir **Alex Ferguson** once said about Filippo Inzaghi: „That lad must have been born offside.“.

In dieser Anordnung:

- das `<strong>`-Element verleiht den Worten „Alex Ferguson“ mehr Bedeutung
- das `<q>` markiert sein Zitat über Inzaghi

Die Tatsache, dass `<strong>` in Fettdruck dargestellt wird, ist nur das Standardverhalten des Browsers. Denken Sie daran, dass Sie HTML-Elemente nach ihrer Bedeutung auswählen müssen, nicht danach, wie sie aussehen.

In diesem Fall:

- `<p>` ist das **übergeordnete** Element von `<strong>` und `<q>`
- `<strong>` und `<q>` sind **untergeordnete** Elemente von `<p>`
- `<strong>` und `<q>` sind **Geschwister**-Elemente

## Reihenfolge

Wie die **Verschachtelung** funktioniert, hängt von der Position der öffnenden und schließenden Tags

ab.

Da ein HTML-Element aus einem öffnenden Tag, einem schließenden Tag und allem, was dazwischen liegt, besteht, muss ein untergeordnetes Element **vor** dem Schließen des übergeordneten Elements geschlossen werden.

```
<!-- This is INVALID code! :( -->
<p>
  This HTML code won't work because I the "strong" tag is opened here
  <strong>but is only closed after the paragraph.
</p></strong>
```

Da das `<strong>`-Element nach dem `<p>`-Element geöffnet wurde (und somit als Kind von `<p>` gilt), muss das `<strong>`-Element vor dem übergeordneten `<p>`-Element geschlossen werden.

```
<!-- This is valid code. :) -->
<p>
  This HTML code will work because I the "strong" tag is opened <strong>and
  closed</strong> properly.
</p>
```

## Tiefe

Da untergeordnete Elemente selbst wiederum andere untergeordnete Elemente enthalten können, ist es möglich, eine tiefere Hierarchie innerhalb eines HTML-Dokuments zu erstellen.

Unser obiger Absatz könnte Teil eines Blog-Artikels sein:

```
<article>
  <h1>Famous football quotes</h1>
  <p>
    Sir <strong>Alex Ferguson</strong> once said about Filippo
    Inzaghi:<q>"That lad must have been born offside"</q>.
  </p>
  <p>
    When criticized by John Carew, <strong>Zlatan Ibrahimovic</strong>
    replied: <q>"What Carew does with a football, I can do with an orange"</q>.
  </p>
  <p>
    <strong>George Best</strong> said <q>"I spent a lot of money on booze,
    birds and fast cars. The rest I just squandered."</q> when asked about his
    lifestyle.
  </p>
</article>
```

In diesem Aufbau:

- `<article>` ist der Vorfahre jedes anderen Elements
- `<article>` ist der Elternteil von `<h1>` und den 3 `<p>`
- `<h1>` und die 3 `<p>` sind Geschwister

- jedes `<p>` ist das Elternteil des `<strong>` und des `<q>`, die sie enthalten
- jedes `<h1>`, `<p>`, `<strong>` und `<q>` sind alle Nachkommen von `<article>`

Die Analogie des Stammbaums gilt auch dann, wenn mehrere Ebenen der HTML-Schachtelung durchlaufen werden:

- Ein **Nachkomme** von Element X ist jedes Element, das in X enthalten ist.
- ein **Kind** ist nur ein direkter Nachkomme
- ein **Vorfahre** von Element Y ist jedes Element, das Y enthält
- der **Elternteil** ist nur der erste direkte Vorfahre
- ein **Geschwister** von Element X ist jedes Element, das denselben Elternteil hat

## Block- und Inline-Schachtelung

**Block**-Elemente können entweder Block- oder Inline-Elemente enthalten.

**Inline**-Elemente können jedoch nur andere Inline-Elemente enthalten. (`<a>` ist davon die einzige Ausnahme)

```
<!-- This is INVALID code! :( -->
<strong>
  <p>You can't put a paragraph inside a "strong" tag.
</strong>
```

Erinnern Sie sich einfach an Vorfahren/Nachkommen, Eltern/Kinder und Geschwister.

[m293-LU02](#), [m293-E1G](#), [m293-E1F](#)



Kevin Maurizi, Marcel Suter

From:  
<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:  
<https://wiki.bzz.ch/modul/m293/learningunits/lu02/strukturelemente?rev=1712592902>

Last update: **2024/04/08 18:15**

