

# HTML

---

## Generell

- Für **neue Projekte** wird der HTML5 Doctype gewählt.
- Bei der **Migration** bestehender Templates in neue Projekte wird auf HTML5 umgestellt.
- Bei der **Bearbeitung / Ergänzung bestehender Projekte** wird der verwendete Doctype beibehalten.

## Validierung

Das HTML-Konstrukt ist valide.

## Struktur

1. HTML hat die Aufgabe, die inhaltliche Struktur eines Dokumentes wiederzugeben und ist nicht für Präsentationszwecke zu verwenden. Alle visuellen Informationen sind in externe Stylesheets auszulagern. Jegliche Art von Inline-Script und Inline-Styles sind zu vermeiden.
2. Es sind jene HTML-Elemente zu verwenden, deren Bedeutung dem Inhalt am besten entsprechen.
3. Es ist auf eine sinnvolle Reihenfolge des Inhalts zu achten: das Wichtigste zuerst.
4. Es ist die Variante mit möglichst wenig Code zu wählen.
5. Dem HTML-Element ist auf jeder Seite die entsprechende generelle Dokumentsprache mitzugeben (beispielsweise `lang="de-CH"` ).
6. Auf neutrale Elemente wie `div` oder `span` wird verzichtet, solange bedeutsamere Elemente (beispielsweise `nav` , `ul` , `strong` , `blockquote` ) für einen Inhalt passender sind.
7. Alle Inhalte sind für sämtliche Devices, für Suchmaschinen und Screenreader vorhanden und zugänglich.

Beispiel einer grundlegenden HTML-Seitenstruktur:

```
<!DOCTYPE html>
<html lang="de-CH" class="no-js">
  <head>
    <meta charset="utf-8">
    <title>Seitentitel</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="/css/screen.css" rel="stylesheet" type="text/css" media="screen">
    <link href="/css/print.css" rel="stylesheet" type="text/css" media="print">
    <script src="/global.js"></script>
  </head>
  <body>

  </body>
</html>
```

## Klassen und ID's

Die Verwendung von Klassen sind grundsätzlich immer der von ID's vorzuziehen. Dies aus Gründen der Wiederverwendbarkeit und der Spezifität. CSS wird ausschliesslich auf Klassen angewendet.

Falsch:

```
<header id="header">
  <h2 id="header__logo"></h2>
</header>
```

Richtig:

```
<header class="header">
  <h2 class="header__logo">
</header>
```

Klassen oder ID's, die mit JavaScript angesprochen, modifiziert oder in den DOM geschrieben werden, sind mit einem entsprechenden Präfix `js-` versehen, damit ihr Zweck klar ist:

```
<div class="accordion__panel js-panel js-panel--open"></div>
```

Solche js-Klassen sind nicht für die Gestaltung der Elemente mit CSS zu verwenden, ausser bewusst im Sinne von «Progressive enhancement».

## Namenskonventionen

Sprechende ID- und Klassen-Namen machen den Quelltext lesbarer und werten ihn semantisch auf. Es gilt der Grundsatz: Namen sind so kurz wie möglich ( `.nav` ), so lang wie nötig ( `.footer__list--active` ).

1. Es ist eine BEM oder BEM-verwandte Notation zu verwenden.
2. Klassen sind ID's immer vorzuziehen (Ziel: Mehrfachnutzung).
3. Klassen- und ID-Namen sind englisch zu benennen.
4. Klassen- und ID-Namen sind ausschliesslich klein (lowercase) geschrieben. CamelCase wird nicht verwendet.
5. Klassen- und ID-Namen beginnen immer mit einem Kleinbuchstaben, nie mit Zahlen.
6. Wo möglich sind in der Namensgebung vorhandene Best Practices zu verwenden, z.B. `.wrapper`, `.row`, `.col`, `.subnav`, `.container`, `.content` ...
7. Klassen- und ID-Namen sind immer generisch oder funktional benannt und beschreiben nicht deren Aussehen oder Verhalten.

Falsch:

```
<div class="box box--redbox"></div>
```

Richtig:

```
<div class="box box--error"></div>
```

8. Wenn sinnvoll sind Klassen-Kombinationen neuen Klassen oder ID's vorzuziehen. Bei der Reihenfolge gilt die umgekehrte Spezifität.

```
<div class="box box--warning required"></div>
```

9. Komplexere Klassennamen können, wenn nötig mit einem Bindestrich getrennt werden. Logisch oder funktionell zugehörige Blöcke verwenden denselben Namespace.  
Beispiel:

```
<h2 class="sidebar__title"></h2>
```

10. Modifier sind mit einem Doppelbindestrich getrennt.

Beispiel Modifier:

```
<div class="box box--warning"></div>
```

## Musterblock

Ein Modulblock könnte nach diesem Muster aufgebaut sein:

```
<div class="tile">
  <h2 class="tile__title">Titel</h2>
  <p class="tile__desc">Beschreibung</p>
  <ul class="tile__lst">
    <li class="tile__lst-item"><a href="#">Editieren</a></li>
    <li class="tile__lst-item"><a href="#">Löschen</a></li>
    <li class="tile__lst-item tile__lst-item--logout"><a href="#">Logout</a></li>
  </ul>
</div>
<!-- end .tile -->
```

## externe Quellen

### HTML 5

- [MDN web docs](#)
- [W3C HTML 5.2](#)
- [WHATWG HTML](#)

### Dokumentationen

- [Übersicht aller HTML-Elemente](#)
- [SELFHTML](#)

### Zeichen & Symbole

- [HTML Sonderzeichen und Symbole aus der Typografie](#)
- [A delightful reference for HTML Symbols, Entities and ASCII Character Codes](#)
- [Unicode code converter](#)
- [CopyChar](#)
- [EntityCode](#)