

HTML Cheatsheet

Intro HTML

- Der lila Dino tanzt:
 - Dino = HTML (Nomen)
 - Lila = CSS (Adjektive)
 - Tanzt = JavaScript (Verben)

HTML

- HTML = Hypertext Markup Language
- Entwickelt 1989/90 zur Beschreibung akademischer Forschungsarbeiten

HTML Elemente

- Browser erkennt folgende Elemente:
 - <p>: Paragraph
 - <h1>: Hauptüberschrift
 - : Bild einbinden
 - <form>: Formular

Überschriftenelement

- Tags: <h1> und </h1>
- Beispiel: <h1>Hello World</h1>

Tags und Elemente

- **Tag:** Text in spitzen Klammern, z.B. <h1> oder </h1>
- **Element:** Tag-Paar und Inhalt dazwischen, z.B. <h1>Hello World</h1>

Hierarchie

- Überschriften definieren Dokumenthierarchie:
 - <h1>: wichtigste/größte Überschrift
 - <h6>: unwichtigste/kleinste Überschrift

Hierarchie-Beispiel:

Buch

- Kapitel 1
 - Abschnitt 1
 - Abschnitt 2
- Kapitel 2
 - Abschnitt 1
 - Schaubild 1
- Kapitel 3
 - Abschnitt 1
 - Abschnitt 2

wird zu:

```
<h1>Buch</h1>
<h2>Kapitel 1</h2>
  <h3>Abschnitt 1</h3>
  <h3>Abschnitt 2</h3>
<h2>Kapitel 2</h2>
  <h3>Abschnitt 1</h3>
    <h4>Schaubild 1</h4>
<h2>Kapitel 3</h2>
  <h3>Abschnitt 1</h3>
  <h3>Abschnitt 2</h3>
```

Erklärung:

- `<h1>` wird für die Hauptüberschrift des Dokuments (Buch) verwendet.
- `<h2>` wird für die Kapitelüberschriften verwendet.
- `<h3>` wird für die Abschnitte innerhalb der Kapitel verwendet.
- `<h4>` wird für Unterabschnitte oder spezielle Inhalte wie Schaubilder verwendet.

Kommentare

- HTML-Kommentar:

```
<!-- Dies ist ein Kommentar -->
```

Paragrafen

- **Absatz-Element:** Zum Formatieren und Einfügen von Textabsätzen.
- **Syntax:** `<p>Text des Absatzes</p>`

```
<p>Dies ist ein Absatz.</p>
```

- **Übung:** Öffne die Übungsdatei und füge `<p>`-Tags um jeden Absatz, um die Absätze klar voneinander zu trennen.

Selbstschießende Tags

- **Nichtige Elemente:** HTML-Elemente ohne Inhalt.
- **Beispiele:**
 - **Horizontale Linie:** `<hr />`
 - **Zeilenumbruch:** `
`

```
<p>Zeile 1<br />Zeile 2</p>
<hr />
<p>Absatz 2</p>
```

- **Übung:** Verwende `<hr />`, um Bereiche zu trennen, und `
`, um Zeilenumbrüche innerhalb von Texten zu erzwingen.

Boilerplate

HTML Boilerplate

Eine HTML Boilerplate ist eine Standardvorlage für HTML-Dokumente, die als Ausgangspunkt für die Erstellung von Webseiten dient. Sie enthält die grundlegende Struktur und notwendige Elemente, die in fast jedem HTML-Dokument vorhanden sein sollten. Die Verwendung einer Boilerplate hilft Entwicklern, schnell mit einem neuen Projekt zu starten, indem sie eine konsistente und bewährte Basis bereitstellt.

Bestandteile einer HTML Boilerplate

1. DOCTYPE-Deklaration

- **Tag:** `<!DOCTYPE html>`
- **Beschreibung:** Gibt die Version von HTML an, die verwendet wird. Die Angabe `<!DOCTYPE html>` zeigt an, dass es sich um HTML5 handelt.

2. HTML-Tag

- **Tag:** `<html lang="de">`
- **Beschreibung:** Der Wurzel-Tag des Dokuments. Das `lang`-Attribut gibt die Sprache des Inhalts an (z.B. de für Deutsch).

3. Kopfbereich (head)

- **Tag:** `<head>`
- **Beschreibung:** Enthält Metadaten über das Dokument, wie z.B. den Zeichensatz, den Titel und Links zu CSS-Dateien oder Skripten.

4. Zeichensatz (charset)

- **Tag:** `<meta charset="UTF-8">`
- **Beschreibung:** Gibt den Zeichensatz der Webseite an, hier UTF-8.

5. Viewport

- **Tag:** `<meta name="viewport" content="width=device-width, initial-scale=1.0">`
- **Beschreibung:** Steuert, wie die Webseite auf verschiedenen Geräten angezeigt wird.

6. Titel

- **Tag:** `<title>`
- **Beschreibung:** Der Titel der Webseite, der im Browser-Tab angezeigt wird.

7. Körper (body)

- **Tag:** `<body>`
- **Beschreibung:** Enthält den sichtbaren Inhalt der Webseite.

Beispiel einer HTML Boilerplate

```
<!DOCTYPE html>
<html lang="de">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Titel der Seite</title>
</head>
<body>
  <!-- Hier kommt der sichtbare Inhalt der Webseite -->
</body>
</html>
```

Vorteile der Verwendung einer Boilerplate

1. **Schneller Start:** Sie bietet eine sofort einsatzbereite Struktur für neue HTML-Dokumente.
2. **Best Practices:** Eine Boilerplate enthält empfohlene Einstellungen und Strukturen, die für eine korrekte und effiziente Darstellung der Webseite sorgen.
3. **Konsistenz:** Sie sorgt für eine konsistente Struktur und Metadaten in allen Projekten.
4. **Wartbarkeit:** Durch die Verwendung einer Standardvorlage wird der Code leichter wartbar und verständlich für andere Entwickler.

Zusammenfassung

Eine HTML Boilerplate ist eine grundlegende Vorlage, die die Basisstruktur eines HTML-Dokuments umfasst. Sie enthält wichtige Elemente wie die DOCTYPE-Deklaration, HTML-Tag, Kopfbereich mit Metadaten und den Körper für den sichtbaren Inhalt. Die Verwendung einer Boilerplate spart Zeit, fördert Best Practices und sorgt für eine konsistente und wartbare Codebasis.

Tipp

Tippe ! in VS Code, um die Boilerplate zu erstellen.

Emmet

Emmet ist ein Plugin, das Entwicklern hilft, HTML und CSS schneller zu schreiben, indem es Abkürzungen in vollständigen Code umwandelt.

Beispiel-Abkürzungen:

- `ul>li*5` generiert:

```
<ul>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
</ul>
```

Für mehr Informationen und Abkürzungen besuche die [Emmet Dokumentation](#).

Automatische Formatierung in VS Code

1. **Befehlspalette öffnen:**

- **Mac:** Cmd + Shift + P
- **Windows:** Ctrl + Shift + P

2. Dokument formatieren:

- **Mac:** Shift + Option + F
- **Windows:** Shift + Alt + F

3. Automatische Formatierung beim Speichern:

- **Mac:** Cmd + ,
- **Windows:** Ctrl + ,
- Suche und aktiviere "Format on Save".

HTML-Listen, Anker-Tags und Bilder

Listen

Im Internet gibt es viele Listen, wie auf BuzzFeed. Wie erstellen wir diese Listen selbst?

Ungeordnete Listen ()

- Verwende für ungeordnete Listen mit Aufzählungspunkten.
- Jedes Listenelement wird mit erstellt.

Beispiel:

```
<ul>
  <li>Element 1</li>
  <li>Element 2</li>
  <li>Element 3</li>
</ul>
```

Geordnete Listen ()

- Verwende für geordnete Listen mit nummerierten Elementen.
- Jedes Listenelement wird mit erstellt.

Beispiel:

```
<ol>
  <li>Schritt 1</li>
  <li>Schritt 2</li>
  <li>Schritt 3</li>
</ol>
```

Verschachtelte Listen (Nesting)

- Verschachtel Listen, indem du oder innerhalb von -Tags platzierst.

Beispiel:

```
<ul>
  <li>A</li>
  <li>
    B
    <ol>
      <li>B1</li>
      <li>B2
        <ul>
          <li>B2a
            <ul>
              <li>B2aa</li>
              <li>B2ab</li>
            </ul>
          </li>
          <li>B2b</li>
          <li>B2c</li>
        </ul>
      </li>
      <li>B3

```

```
<ol>
  <li>B31</li>
  <li>B32</li>
</ol>
</li>
</ol>
</li>
<li>C</li>
</ul>
```

Anker-Tag (<a>)

Mit dem A-Tag (<a>) können wir Hyperlinks erstellen, die auf andere Webseiten, E-Mail-Adressen, Dateien oder Orte auf derselben Seite verlinken.

Externe Links

Beispiel:

```
<a href="http://www.google.com">Ich bin ein Link</a>
```

Interne Links

- Erstelle einen Link zu einer anderen HTML-Datei:

```
<a href="about.html">Über mich</a>
```

Interne Sprungmarken

- Erstelle Links zu einem bestimmten Abschnitt auf derselben Seite:

```
<a href="#section-id">Zum Abschnitt</a>
```

- Der Zielabschnitt:

```
<h2 id="section-id">Abschnitt</h2>
```

Bilder ()

Füge Bilder mit dem Bild-Tag () ein. Das src-Attribut gibt die Quelle des Bildes an und das alt-Attribut bietet eine Beschreibung für Personen mit Einschränkungen beim Sehen.

Einfaches Bild

Beispiel:

```

```

Beispiel mit Beschreibung

```

```

Beispiel für eine vollständige HTML-Struktur mit Listen, Anker-Tags und Bildern

```
<!DOCTYPE html>
<html lang="de">
<head>
  <meta charset="UTF-8">
  <title>HTML Cheatsheet</title>
</head>
<body>
  <h1>Listen-Beispiele</h1>

  <h2>Ungeordnete Liste</h2>
  <ul>
    <li>Element 1</li>
```

```

    <li>Element 2</li>
    <li>Element 3</li>
</ul>

<h2>Geordnete Liste</h2>
<ol>
    <li>Schritt 1</li>
    <li>Schritt 2</li>
    <li>Schritt 3</li>
</ol>

<h2>Verschachtelte Liste</h2>
<ul>
    <li>A</li>
    <li>
        B
        <ol>
            <li>B1</li>
            <li>B2
                <ul>
                    <li>B2a
                        <ul>
                            <li>B2aa</li>
                            <li>B2ab</li>
                        </ul>
                    </li>
                    <li>B2b</li>
                    <li>B2c</li>
                </ul>
            </li>
            <li>B3
                <ol>
                    <li>B31</li>
                    <li>B32</li>
                </ol>
            </li>
        </ol>
    </li>
    <li>C</li>
</ul>

<h1>Links und Bilder</h1>

<h2>Externer Link</h2>
<a href="http://www.google.com">Google</a>

<h2>Interner Link</h2>
<a href="about.html">Über mich</a>

<h2>Bild-Beispiel</h2>
<h1>Ich bin ein Katzenmensch</h1>

</body>
</html>

```

Semantic HTML

Semantische HTML-Tags sind HTML5-Elemente, die den Inhalt, den sie umschließen, klar und verständlich machen. Diese Tags vermitteln dem Browser und den Entwicklern die Bedeutung des Inhalts und tragen zur Verbesserung der Barrierefreiheit und der Suchmaschinenoptimierung (SEO) bei.

1. <header>

Das <header>-Tag repräsentiert einen Container für Einführungs- und Navigationsinhalte. Es enthält oft Logo, Navigationsleiste, Suchformular, etc.

Beispiel:

```

<header>
    <h1>Meine Website</h1>
    <nav>
        <ul>
            <li><a href="#home">Startseite</a></li>

```

```

        <li><a href="#about">Über uns</a></li>
        <li><a href="#contact">Kontakt</a></li>
    </ul>
</nav>
</header>

```

2. <nav>

Das <nav>-Tag wird für den Hauptblock der Navigationslinks verwendet.

Beispiel:

```

<nav>
  <ul>
    <li><a href="#home">Startseite</a></li>
    <li><a href="#services">Dienstleistungen</a></li>
    <li><a href="#contact">Kontakt</a></li>
  </ul>
</nav>

```

3. <section>

Das <section>-Tag definiert Abschnitte in einem Dokument, wie Kapitel, Themen oder einen Bereich einer Webseite.

Beispiel:

```

<section>
  <h2>Unsere Dienstleistungen</h2>
  <p>Wir bieten eine Vielzahl von Dienstleistungen an, um Ihre Bedürfnisse zu erfüllen.</p>
</section>

```

4. <article>

Das <article>-Tag repräsentiert einen eigenständigen Inhalt, der in sich abgeschlossen ist und wiederverwendet werden kann, wie Blog-Einträge oder Nachrichtenartikel.

Beispiel:

```

<article>
  <h2>Neuste Nachrichten</h2>
  <p>Hier sind die neuesten Nachrichten aus der Tech-Welt.</p>
</article>

```

5. <aside>

Das <aside>-Tag enthält Inhalte, die nicht direkt zum Hauptinhalt gehören, wie Seitenleisten, zusätzliche Informationen oder Werbefbanner.

Beispiel:

```

<aside>
  <h3>Zusätzliche Informationen</h3>
  <p>Hier finden Sie weitere Ressourcen und Links.</p>
</aside>

```

6. <footer>

Das <footer>-Tag enthält Fußzeileninformationen einer Sektion oder eines Dokuments, wie Autorinformationen, Copyright, Links zu Nutzungsbedingungen, etc.

Beispiel:

```

<footer>
  <p>&copy; 2024 Meine Website. Alle Rechte vorbehalten.</p>
  <nav>
    <ul>
      <li><a href="#privacy">Datenschutz</a></li>
      <li><a href="#terms">Nutzungsbedingungen</a></li>
    </ul>
  </nav>

```

```
</nav>
</footer>
```

7. <main>

Das <main>-Tag definiert den Hauptinhalt des <body>-Elements. Es sollte nur einen <main>-Tag pro Dokument geben.

Beispiel:

```
<main>
  <h1>Willkommen auf unserer Website</h1>
  <p>Hier finden Sie die wichtigsten Informationen und Inhalte.</p>
</main>
```

8. <figure> und <figcaption>

Das <figure>-Tag wird verwendet, um selbständige Inhalte, wie Bilder, Diagramme oder Illustrationen, einzuschließen. Das <figcaption>-Tag liefert eine Beschreibung.

Beispiel:

```
<figure>
  
  <figcaption>Bildbeschreibung</figcaption>
</figure>
```

9. <time>

Das <time>-Tag repräsentiert eine bestimmte Zeit oder ein Datum.

Beispiel:

```
<p>Das Event findet am <time datetime="2024-07-20">20. Juli 2024</time> statt.</p>
```

Vorteile von semantischen HTML-Tags:

- **Bessere SEO:** Suchmaschinen können den Inhalt besser verstehen und indexieren.
- **Barrierefreiheit:** Screenreader und andere Hilfstechnologien können den Inhalt besser interpretieren.
- **Code-Lesbarkeit:** Der Code wird für Entwickler klarer und verständlicher.

Durch die Verwendung semantischer HTML-Tags wird die Struktur einer Webseite logisch und zugänglich, was sowohl für Benutzer als auch für Suchmaschinen von Vorteil ist.

HTML Formulare

Grundlegende Elemente eines Formulars

<form>-Element:

Das <form>-Element definiert ein HTML-Formular, das zur Erfassung von Benutzereingaben verwendet wird. Es kann verschiedene Eingabeelemente wie Textfelder, Passwortfelder, Auswahllisten usw. enthalten. Hier sind einige wichtige Eigenschaften und Verwendungszwecke des <form>-Elements:

- **Aktion (action):** Das action-Attribut bestimmt die URL, an die die Formulardaten gesendet werden sollen, wenn das Formular abgesendet wird. Wenn kein action-Attribut angegeben ist, wird das Formular an die aktuelle Seite gesendet.
- **Autofokus (autofocus):** Das autofocus-Attribut bestimmt, ob das erste Eingabefeld im Formular automatisch den Fokus erhält, wenn die Seite geladen wird.
- **Formularsteuerelemente:** Innerhalb des <form>-Tags können verschiedene Formularsteuerelemente wie Textfelder (<input type="text">), Passwortfelder (<input type="password">), Auswahllisten (<select> und <option>) usw. platziert werden.

<label>-Element:

Das <label>-Element definiert ein Label für ein Formularelement, normalerweise ein <input>, <select> oder <textarea>-Element. Es wird häufig verwendet, um das Formularelement zu beschreiben oder den Benutzer dazu

aufzufordern, was er in das Eingabefeld eingeben soll. Hier sind einige wichtige Eigenschaften und Verwendungszwecke des `<label>`-Elements:

- **For-Attribut (for):** Das `for`-Attribut legt fest, welches Formularelement mit diesem Label verknüpft ist. Es sollte den Wert der `id` des verknüpften Formularelements enthalten.
- **Barrierefreiheit:** Die Verwendung von `<label>`-Elementen verbessert die Zugänglichkeit von Formularen, da Benutzer nicht nur auf den Text klicken können, um das zugehörige Formularelement auszuwählen, sondern auch auf das Label klicken können.

`<input>`-Element:

Das `<input>`-Element definiert ein Eingabefeld, in das Benutzer Daten eingeben können. Es ist ein leeres Element und hat verschiedene Typen, je nachdem, welche Art von Daten erwartet wird.

- **Typ (type):** Das `type`-Attribut bestimmt den Typ des Eingabefeldes. Es kann verschiedene Werte haben, wie z.B. "text" für Textfelder, "password" für Passwortfelder, "checkbox" für Kontrollkästchen, "radio" für Optionsfelder usw.
- **Name (name):** Das `name`-Attribut definiert den Namen des Eingabefelds, der verwendet wird, um die Formulardaten zu identifizieren, wenn sie an den Server gesendet werden.
- **ID (id):** Das `id`-Attribut definiert eine eindeutige ID für das Eingabefeld. Sie wird normalerweise mit dem `for`-Attribut des zugehörigen `<label>`-Elements verwendet, um eine Verbindung zwischen dem Label und dem Eingabefeld herzustellen.
- **Erforderlichkeit (required):** Das `required`-Attribut bestimmt, ob das Eingabefeld obligatorisch ist und nicht leer sein darf, bevor das Formular abgesendet werden kann.

Diese Elemente werden kombiniert, um ein Formular mit Eingabefeldern und zugehörigen Labels zu erstellen, das es dem Benutzer ermöglicht, Daten einzugeben und an den Server zu senden. In der Anmeldeseite werden `<form>`, `<label>` und `<input>` verwendet, um Benutzernamen, Passwort und den Anmelde-Button bereitzustellen und zu steuern.

`<input>`-Element

Typen und Beispiele

- **Texteingabe:**

```
<input type="text" name="username">
```

- **Passworteingabe:**

```
<input type="password" name="password">
```

- **Farbauswahl:**

```
<input type="color" name="favcolor">
```

- **Nummerneingabe:**

```
<input type="number" name="quantity">
```

- **Zeiteingabe:**

```
<input type="time" name="appt-time">
```

Es gibt beim Input-Typ im HTML-Formular so viele verschiedene Möglichkeiten, weil unterschiedliche Eingabetypen verschiedene Arten von Daten erfassen und benutzerfreundlichere Interaktionen ermöglichen.

1. **Spezifität und Datenvalidierung:** Bestimmte Input-Typen stellen sicher, dass nur spezifische Arten von Daten eingegeben werden können, was die Validierung und Verarbeitung der Daten vereinfacht. Zum Beispiel sorgt `type="email"` dafür, dass die Eingabe ein gültiges E-Mail-Format hat.
2. **Benutzerfreundlichkeit:** Verschiedene Eingabetypen verbessern die Benutzerfreundlichkeit, indem sie den Nutzern spezifische Eingabemethoden anbieten. Zum Beispiel können Nutzer bei `type="date"` ein Datum aus einem Kalender auswählen, was einfacher und weniger fehleranfällig ist als eine manuelle Eingabe.
3. **Responsives Design und mobile Geräte:** Auf mobilen Geräten passen sich verschiedene Input-Typen an und bieten spezielle Tastaturen oder Eingabemethoden an, die die Eingabe erleichtern. Zum Beispiel zeigt

`type="tel"` eine numerische Tastatur auf mobilen Geräten an.

4. **Erleichterung der Barrierefreiheit:** Unterschiedliche Eingabetypen können dazu beitragen, Formulare für Nutzer mit Behinderungen zugänglicher zu machen. Zum Beispiel können Screenreader spezifische Eingabetypen besser interpretieren und entsprechend an den Nutzer kommunizieren.
5. **Ästhetik und Benutzererfahrung:** Verschiedene Input-Typen ermöglichen es Entwicklern, benutzerfreundlichere und optisch ansprechendere Formulare zu erstellen, die die Benutzererfahrung verbessern. Zum Beispiel kann `type="range"` für eine visuelle und intuitive Steuerung verwendet werden.

Einiger gängige Input-Typen und deren Verwendungszwecke:

- **text:** Einfache Texteingabe
- **email:** Eingabe einer E-Mail-Adresse
- **number:** Eingabe einer numerischen Wert
- **tel:** Eingabe einer Telefonnummer
- **url:** Eingabe einer URL
- **password:** Eingabe eines Passworts (mit verdeckter Anzeige)
- **date:** Eingabe eines Datums
- **time:** Eingabe einer Uhrzeit
- **datetime-local:** Eingabe eines Datums und einer Uhrzeit (ohne Zeitzone)
- **month:** Eingabe eines Monats und Jahres
- **week:** Eingabe einer Kalenderwoche und Jahres
- **color:** Auswahl einer Farbe
- **file:** Hochladen einer Datei
- **checkbox:** Ankreuzfeld für Ja/Nein-Optionen
- **radio:** Optionsfeld für eine von mehreren Auswahlmöglichkeiten
- **range:** Schieberegler zur Auswahl eines Wertebereichs

Attribute

- **Platzhalter:**

```
<input type="text" name="username" placeholder="Enter your username">
```

Labels in Formularen

In HTML werden "for"-Attribute in Formularen verwendet, um ein Etikett mit einem entsprechenden Formularelement zu verknüpfen. Das "for"-Attribut wird normalerweise in einem `<label>`-Tag verwendet und mit dem `id`-Attribut des zugehörigen Formularelements verknüpft. Dies ermöglicht es Benutzern, auf das Etikett zu klicken, um das zugehörige Formularelement zu aktivieren.

Zum Beispiel:

```
<label for="username">Benutzername:</label>
<input type="text" id="username" name="username">
```

In diesem Beispiel wird das Etikett "Benutzername:" mit dem Eingabefeld für den Benutzernamen verknüpft. Wenn ein Benutzer auf das Etikett klickt, wird das zugehörige Eingabefeld aktiviert. Das `for`-Attribut im `<label>`-Tag entspricht dem `id`-Attribut des zugehörigen Formularelements (`<input>`).

Beispielformular

```
<form>
  <label for="username">Username:</label>
  <input type="text" id="username" name="username" placeholder="Gib deinen Username ein">

  <label for="password">Passwort:</label>
  <input type="password" id="password" name="password" placeholder="Passworteingabe">

  <label for="favcolor">Lieblingsfarbe:</label>
  <input type="color" id="favcolor" name="favcolor">

  <label for="quantity">Anzahl:</label>
  <input type="number" id="quantity" name="quantity">

  <label for="termin">Termin:</label>
  <input type="time" id="termin" name="termin">
```

```
<button type="submit">Absenden</button>
</form>
```

Buttons

Typen

1. **Submit-Button:**
html <button type="submit">Absenden</button>
2. **Normaler Button:**
html <button type="button">Klicken</button>

Kontrollkästchen und Optionsfelder

Kontrollkästchen

```
<input type="checkbox" id="terms" name="terms" checked>
<label for="terms">Ich stimme den AGB zu</label>
```

Optionsfelder

```
<input type="radio" id="size-small" name="size" value="small">
<label for="size-small">Klein</label>

<input type="radio" id="size-medium" name="size" value="medium">
<label for="size-medium">Medium</label>

<input type="radio" id="size-large" name="size" value="large">
<label for="size-large">Groß</label>
```

Dropdown-Menüs

```
<select id="meal" name="meal">
  <option value="fish">Fisch</option>
  <option value="vegetarian">Vegetarisch</option>
  <option value="meat" selected>Fleisch</option>
</select>
```

Spezielle Eingabetypen

Bereichseingabe

```
<input type="range" id="cheese" name="cheese" min="1" max="100" step="1" value="75">
<label for="cheese">Menge Käse</label>
```

Nummerneingabe

```
<input type="number" id="men" name="persons" min="1" max="1000" step="1" value="10">
<label for="men">Anzahl der Personen</label>
```

Textbereich

```
<label for="requests">Spezielle Anfrage</label>
<textarea id="requests" name="requests" rows="10" cols="20" placeholder="Geben Sie hier Ihre An">
```

Form Validierung

Eingebaute Validierung

- **Pflichtfeld:**

```
<input type="text" id="firstName" name="firstName" required>
<label for="firstName">Vorname:</label>
```

- Minimale und maximale Länge:

```
<input type="text" id="username" name="username" minlength="5" maxlength="20" required>  
<label for="username">Username:</label>
```