

# Das `<form>`-Element

---

Das `<form>`-Element ist in HTML unerlässlich, wenn es darum geht, Daten von Benutzern zu sammeln und an einen Server zu senden. Es fungiert als Container für Eingabefelder, Schaltflächen und andere interaktive Steuerelemente.

[mdn web docs - Form-Element](#)

## Grundlegende Attribute von `<form>`:

- **action:** Gibt die URL an, an die die Formulardaten gesendet werden sollen.

```
action="/submit-form.php"
```

- **method:** Bestimmt die HTTP-Methode, die verwendet wird, um die Formulardaten zu senden. Übliche Werte sind:
  - **GET:** Sendet die Daten in der URL. Eignet sich für Suchanfragen.
  - **POST:** Sendet die Daten im Anfragekörper. Wird häufig für sensible Daten verwendet.

Beispiel:

```
method="POST"
```

- **enctype:** Gibt den MIME-Typ an, der verwendet wird, um die Formulardaten zu senden. Gängige Werte sind:
  - **application/x-www-form-urlencoded:** Standardwert, verwendet für einfache Textdaten.
  - **multipart/form-data:** Wird verwendet, wenn Dateien hochgeladen werden sollen.
  - **text/plain:** Sendet Daten ohne Codierung.

Beispiel:

```
enctype="multipart/form-data"
```

- **target:** Bestimmt, wo das Ergebnis der Formulardaten gesendet werden soll (z.B. im selben Fenster oder in einem neuen Tab).

Beispiel:

```
target="_blank"
```

- **novalidate:** Deaktiviert die integrierte Formularvalidierung des Browsers.

Beispiel:

```
novalidate
```

- **autocomplete:** Aktiviert oder deaktiviert die automatische Vervollständigung von Formularfeldern durch den Browser.

Beispiel:

```
autocomplete="on"
```

## Das `<label>`-Element

Das `<label>`-Element verknüpft ein Label mit einem Eingabefeld und verbessert die Benutzerfreundlichkeit sowie die Barrierefreiheit von Formularen. Durch Anklicken des Labels wird das verknüpfte Eingabefeld aktiviert.

- **for:** Verknüpft das Label mit einem Eingabefeld durch die ID des entsprechenden Feldes.

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

- **Nutzung ohne for:** Man kann das `<input>`-Element auch innerhalb des `<label>`-Tags platzieren, wodurch die Verknüpfung automatisch geschieht.

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

Das `<label>`-Element verbessert die Benutzererfahrung und hilft auch bei der Zugänglichkeit, da Screenreader Labels den Benutzern vorlesen.

## Das `<input>`-Element: Typen und Attribute

Das `<input>`-Element ist eines der vielseitigsten Elemente in HTML. Es kann verschiedene Dateneingabetypen erfassen, abhängig vom `type`-Attribut.

Wichtige Attribute für `<input>`:

- **name:** Der Name des Eingabefeldes. Dieser Wert wird beim Senden der Daten verwendet.

Beispiel:

```
name="username"
```

- **value:** Der Standardwert des Eingabefeldes. Bei der Übermittlung des Formulars wird dieser Wert gesendet.

Beispiel:

```
value="John Doe"
```

- **required:** Macht das Feld erforderlich. Der Benutzer muss das Feld ausfüllen, bevor das Formular gesendet werden kann.

Beispiel:

```
required
```

- **placeholder:** Ein Beispieltext, der angezeigt wird, wenn das Feld leer ist.

Beispiel:

```
placeholder="Geben Sie Ihren Namen ein"
```

- **readonly:** Macht das Feld schreibgeschützt.

Beispiel:

readonly

- **disabled:** Macht das Feld deaktiviert, sodass keine Eingabe möglich ist und es nicht gesendet wird.

Beispiel:

disabled

## Input-Typen:

1. **text:** Ein einzeliges Textfeld für normale Eingaben.

```
<input type="text" name="username" placeholder="Ihr Name" required>
```

2. **password:** Ein Textfeld für Passworteingaben. Der Text wird nicht angezeigt.

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

3. **email:** Ein Feld für E-Mail-Adressen. Der Browser prüft die Eingabe auf das korrekte Format.

```
<input type="email" name="email" placeholder="name@example.com" required>
```

4. **number:** Ein Eingabefeld für Zahlen. Optional können `min`, `max`, und `step` verwendet werden, um den gültigen Bereich festzulegen.

```
<input type="number" name="age" min="1" max="100" step="1" required>
```

5. **date:** Ermöglicht die Eingabe eines Datums.

```
<input type="date" name="birthday" required>
```

6. **file:** Ein Feld zum Hochladen von Dateien. Das Attribut `multiple` erlaubt das Hochladen mehrerer Dateien.

```
<input type="file" name="resume" accept=".pdf,.doc" required>
```

7. **checkbox:** Ein Kontrollkästchen, das entweder markiert oder nicht markiert sein kann.

```
<input type="checkbox" name="newsletter" value="yes"> Newsletter abonnieren
```

8. **radio:** Ein Auswahlfeld, bei dem nur eine Option aus einer Gruppe ausgewählt werden kann.

```
<input type="radio" name="gender" value="male"> Männlich  
<input type="radio" name="gender" value="female"> Weiblich
```

9. **submit**: Eine Schaltfläche, die das Formular absendet.

```
<input type="submit" value="Absenden">
```

10. **reset**: Eine Schaltfläche, die alle Eingabefelder auf ihre Standardwerte zurücksetzt.

```
<input type="reset" value="Zurücksetzen">
```

## Weitere wichtige Formularelemente:

### `<textarea>`:

Ermöglicht die Eingabe von mehrzeiligem Text.

```
<textarea name="comments" rows="5" cols="40" placeholder="Ihr Kommentar"></textarea>
```

### `<select>` und `option`:

Ein Dropdown-Menü mit mehreren Auswahlmöglichkeiten.

```
<select name="country">
  <option value="de">Deutschland</option>
  <option value="us">USA</option>
  <option value="fr">Frankreich</option>
</select>
```

### `<button>`:

Eine Schaltfläche, die als Submit-Button oder eine normale Schaltfläche fungieren kann.

```
<button type="submit">Formular absenden</button>
```

## Unterschied zwischen `<button>` und `<input type="button">`

Der Unterschied zwischen `<button>` und `<input type="button">` besteht hauptsächlich in ihrer Flexibilität und Anwendbarkeit.

Das `<button>`-Element:

Das `<button>`-Element ist ein vielseitigeres Element als das `<input type="button">`, da es mehr Möglichkeiten zur Anpassung und Funktionalität bietet.

### Eigenschaften von `<button>`:

- **Inhalt**: Das `<button>`-Element kann nicht nur Text enthalten, sondern auch andere HTML-Elemente wie Bilder, Icons oder sogar Formularelemente.

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

- **Typen**:

- `type="submit"`: Schickt das Formular ab (Standardwert).
- `type="reset"`: Setzt alle Formularfelder auf ihren Ausgangszustand zurück.

- `type="button"`: Funktioniert wie ein generischer Button, der keine Aktion von sich aus durchführt, sondern über JavaScript eine Funktion aufgerufen werden kann.

### Wann verwendet man `<button>`?

- Verwende `<button>`, wenn du komplexeren Inhalt (Text + Icons/Bilder) innerhalb der Schaltfläche benötigst oder erweiterte Funktionen durch verschiedene Typen (`submit`, `reset`, `button`) verwenden möchtest.
- Ideal für die Verwendung in Formularen oder interaktiven Elementen, bei denen die Flexibilität von Schaltflächeninhalten erforderlich ist.

### Das `<input type="button">`-Element:

Das `<input type="button">`-Element ist ein rein statisches Button-Element, das nur Text als Wert enthalten kann und in erster Linie in Verbindung mit JavaScript verwendet wird.

### Eigenschaften von `<input type="button">`:

- **Inhalt:** Der Textinhalt wird ausschließlich durch das `value`-Attribut festgelegt und kann keine anderen HTML-Elemente enthalten.

```
<input type="button" value="Klick mich" onclick="myFunction()">
```

- **Typen:**
  - `type="button"`: Führt keine eigene Aktion aus, sondern muss durch JavaScript gesteuert werden.
  - `type="submit"`: Sendet das Formular ab.
  - `type="reset"`: Setzt das Formular zurück.

### Wann verwendet man `<input type="button">`?

- Verwende `<input type="button">`, wenn du einen einfachen statischen Button benötigst, der nur durch JavaScript gesteuert wird und keinen weiteren HTML-Inhalt benötigt.
- Es wird häufig für einfache Schaltflächen verwendet, die per JavaScript eine Funktion aufrufen, z.B. zum Aktualisieren einer Seite oder Auslösen einer spezifischen Aktion.

### Beispiele:

```
<!-- Mit <button> kann mehr Flexibilität erreicht werden -->
<button type="submit">Absenden</button>
<button type="button" onclick="myFunction()">Klick mich</button>
<button type="reset">Zurücksetzen</button>

<!-- <input type="button"> ist auf das value-Attribut beschränkt -->
<input type="button" value="Klick mich" onclick="myFunction()">
<input type="submit" value="Absenden">
<input type="reset" value="Zurücksetzen">
```

## Das `<fieldset>`-Element

Das `<fieldset>`-Element gruppiert zusammengehörige Formularfelder und sorgt für eine visuelle Trennung. Es kann für komplexere Formulare verwendet werden, bei denen mehrere zusammenhängende Eingabefelder eine Einheit bilden.

```
<fieldset>
  <legend>Persönliche Informationen</legend>
  <label for="name">Name:</label>
  <input type="text" id="name" name="name" required>

  <label for="email">E-Mail:</label>
  <input type="email" id="email" name="email" required>
</fieldset>
```

## Das <legend>-Element

Das <legend>-Element wird innerhalb eines <fieldset> verwendet und stellt eine Überschrift dar, die den Inhalt des Feldsatzes beschreibt. Es sollte immer verwendet werden, um die Gruppierung von Formularfeldern zu erläutern.

```
<fieldset>
  <legend>Kontaktinformationen</legend>
  <label for="phone">Telefonnummer:</label>
  <input type="tel" id="phone" name="phone">
</fieldset>
```

## Das <summary>-Element

Das <summary>-Element wird zusammen mit dem <details>-Element verwendet, um zusätzliche Informationen anzuzeigen, die der Benutzer ein- und ausklappen kann. Es eignet sich gut, um dem Benutzer optionale oder erklärende Informationen anzubieten.

```
<details>
  <summary>Mehr Informationen zur Datenschutzerklärung</summary>
  <p>Hier können Sie weitere Details zu unserer Datenschutzerklärung nachlesen.</p>
</details>
```

Das <details>- und <summary>-Element sind nützlich, um lange Informationen zu verbergen, bis der Benutzer explizit danach sucht.

## Beispiel für ein umfassendes Formular

```
<form action="/submit-form.php" method="POST">
  <fieldset>
    <legend>Benutzerinformationen</legend>

    <label for="username">Benutzername:</label>
    <input type="text" id="username" name="username" required placeholder="Ihr Name">

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

    <label for="email">E-Mail:</label>
    <input type="email" id="email" name="email" placeholder="name@example.com" required>

    <label for="gender">Geschlecht:</label>
    <input type="radio" id="male" name="gender" value="male">
    <label for="male">Männlich</label>
    <input type="radio" id="female" name="gender" value="female">
    <label for="female">Weiblich</label>
  </fieldset>

  <fieldset>
    <legend>Kontaktdaten</legend>

    <label for="phone">Telefonnummer:</label>
    <input type="tel" id="phone" name="phone" placeholder="0123 456789">

    <label for="address">Adresse:</label>
    <input type="text" id="address" name="address" placeholder="Ihre Adresse">
  </fieldset>

  <details>
    <summary>Mehr Informationen</summary>
    <p>Wir sammeln Ihre Daten nur zur Verarbeitung dieses Formulars und zu keinem anderen Zweck.</p>
  </details>
</form>
```

```
</details>

<input type="submit" value="Absenden">
<input type="reset" value="Zurücksetzen">
</form>
```