



HTML(5) Formulare



<https://wiki.selfhtml.org/wiki/HTML/Formulare>



```
<form action="..." method="...">  
  ...  
  <input type="reset" value="Reset">  
  <input type="submit" value="Submit">  
</form>
```

- action

- Das Attribut action erhält als Wert eine URL, an welche beim Klicken auf den submit-Button Daten gesendet werden.
- Dadurch wird ein HTTP-Request ausgelöst.
- In den meisten Fällen wird das Attribut die URL eines Skriptes enthalten, mit welchem auf dem Server die Formulareingaben ausgewertet werden.

- method

- Legt mit get als Standardwert oder mit post die HTTP-Methode fest, mit der die Daten gesendet werden.



- Mit dem Input-Element erzeugen Sie Elemente innerhalb des Formulars, beispielsweise Buttons, Textfelder, Auswahlfelder usw.
- Es gibt drei verschiedene Arten von Buttons, gekennzeichnet durch das type-Attribut:
 - type = "button"
zum Auslösen clientseitiger Aktionen, üblicherweise zum Anstoßen einer JavaScript-Funktion.
 - type = "submit"
zum Absenden eines Formulars.
 - type = "reset"
zum Zurücksetzen eines Formulars.



Typischer Anwendungsfall von Formularen



```
<form action="auswertungsscript.php">  
  Benutzername: <br/>  
  <input type="text" name="benutzername"><br/>  
  Passwort: <br/>  
  <input type="text" name="password"><br/>  
  <br/>  
  <input type="reset" value="Zurücksetzen">  
  <input type="submit" value="Absenden">  
</form>
```

Benutzername:

Passwort:

Zurücksetzen

Absenden

Benutzername:

Passwort:

Zurücksetzen

Absenden



Probleme bei diesem beispielhaften Formular?



- Dieses vereinfachte Formular für ein Login-System enthält zwei neue Input-Elemente:
 - `<input type="text" name="benutzername">`
 - `<input type="text" name="password">`
- Das Input-Element vom Typ „text“ nimmt einfache Texteingaben vom Benutzer entgegen und schickt diese beim Absenden des Formulars mit.
- Es gibt aber zwei Probleme mit diesem Formular:
 - Passwörter sollten nicht als Text angezeigt werden, wir benötigen also einen neuen Input-Typen.
 - Das Formular wird standardmäßig mit der HTTP-Methode GET versendet, dies bedeutet, dass sowohl der Benutzername als auch das Passwort in der Adressleiste des Browsers angezeigt werden.
 - Mit der Angabe `method="post"` verhindern wir dies.



- Während der Standardwert für `input type="text"` ist und eine Texteingabe erlaubt, gibt es weitere einzeilige Eingabefelder, deren Inhalt jeweils eine andere Bedeutung hat:
 - `type = "text"`
 - `type = "password"`
 - `type = "search"`
 - `type = "tel"`
 - `type = "url"`
 - `type = "email"`
- Zudem gibt es seit HTML5 weitere Input-Typen für z.B. Zahlen, Datum, Uhrzeit, von denen hier aber nur einige vorgestellt werden.



```
<form action="auswertungsscript.php">  
  Passwort: <br/>  
  <input type="password" name="beispiel">  
</form>
```

Passwort:

Passwort:

- In einem Eingabefeld können Sie mit `type="password"` Geheimnummern, Passwörter usw. eingeben.
- Die eingegebenen Zeichen werden dabei durch Platzhalter (meistens Sternchen) dargestellt, sodass Personen im Raum des Anwenders nicht zufällig das eingegebene Passwort mitlesen können.
- Passwörter werden beim normalen HTTP-Protokoll trotz der verdeckten Eingabe im Klartext über das Internet übertragen!



```
<form action="auswertungsscript.php">
  <input type="search" list="Vögel" />
  <datalist id="Vögel">
    <option value="Amsel" />
    <option value="Buntspecht" />
    <option value="Drossel" />
    <option value="Eisvogel" />
    <option value="Fink" />
    <option value="Graugans" />
  </datalist>
  <button>finden!</button>
</form>
```

The top screenshot shows a search input field with a dropdown menu open, listing bird species: Amsel, Buntspecht, Drossel, Eisvogel, Fink, and Graugans. The bottom screenshot shows the search field containing the text 'Drossel' and a 'finden!' button next to it.

- Ein Eingabefeld vom Typ search ist für Eingabe von Suchbegriffen vorgesehen.
- Einige Browser ergänzen möglicherweise ein Lupensymbol, fügen ein Bedienfeld hinzu, welches das Eingabefeld leert oder passen ggf. automatisch den Wert eines input-Elements vom Typ submit an.



```
<form action="auswertungsscript.php">
```

```
  Telefonnummer: <br/>
```

```
  <input type="tel" name="beispiel">
```

```
</form>
```

The screenshot shows a mobile browser interface. At the top, there is a dark header with the text "(pure JS) ▾". Below this, the label "Telefonnummer:" is followed by an input field containing the number "01602468334". Below the input field, there is a light gray bar with a blue arrow icon, the text "Kontakt autom. ausfüllen", and a blue button labeled "Fertig". Below this bar is a numeric keypad with buttons for digits 1-9, 0, and symbols for +, *, #, and a backspace icon.

- Mit `input type="tel"` können Sie ein Eingabefeld für eine Telefonnummer vorbereiten.
- Bei mobilen Geräten wird eine angepasste Tastatur eingeblendet.



```
<form action="auswertungsscript.php">  
  Webadresse: <br/>  
  <input type="url" name="beispiel" value="https://">  
</form>
```

Webadresse:

Webadresse:

- Mit `input type="url"` können Sie Web-Adressen eingeben. Schon während der Eingabe wird überprüft, ob diese eine gültige Web-Adresse ist.
- Es wird überprüft, ob die Internet-Adresse valide ist.
- Im Internet Explorer 10 und Chrome erscheint sonst eine Fehlermeldung; Firefox und Opera verhindern ein Senden.



```
<form action="auswertungsscript.php">
  E-Mail: <br/>
  <input type="email" name="beispiel">
</form>
```

E-Mail:

E-Mail-Adresse eingeben

E-Mail:

test|

E-Mail:

test@gmail.com

- Mit `input type="email"` können Sie Email-Adressen eingeben.
- Bei mobilen Geräten (iOS, Android) erscheint bei der Eingabe das `@`-Symbol automatisch im Tastenfeld.
- Die Browser verwenden zur Prüfung einen regulären Ausdruck.



- Sie können Felder in einem Formular definieren, die dem Anwender nicht angezeigt werden.
- Beim Absenden des Formulars werden die Daten versteckter Felder mit übertragen.
- Auf diese Weise können Sie beispielsweise zusätzliche Informationen an das verarbeitende Script übergeben.
- Auch für JavaScript ist diese Möglichkeit interessant.
 - Da ein JavaScript Formularfelder problemlos auslesen und deren Werte auch ändern kann, ist es auf diese Weise bequem möglich, Daten zu speichern, die nicht am Bildschirm angezeigt werden.
 - Die Daten werden dann, wenn der Anwender das Formular abschickt, mit übertragen.



```
<form action="auswertungsscript.php" method="post">
  Benutzername: <br/>
  <input type="text" name="benutzername"><br/>
  Passwort: <br/>
  <input type="password" name="benutzername"><br/>
  <!-- nicht sichtbar -->
  <input type="hidden" value="Information...">
  <!-- nicht sichtbar -->
  <input type="submit" value="Absenden">
</form>
```

Benutzername:

Passwort:



- HTML5 bietet eine Vielzahl von neuen Eingabe-Typen für die Eingabe von Zahlen.
- Bei mobilen Geräten wird dann ein Nummernfeld eingeblendet, das die Eingabe erleichtert.
- Sie können ein Eingabefeld mit `<input type="number">` nur auf Zahleneingaben beschränken. Folgende Attribute können verwendet werden:
 - `max`: legt Maximalwert fest.
 - `min`: legt Mindestwert fest.
 - `step`: legt fest, in welchen Stufen die Werte durch eine Scrollbar eingegeben werden dürfen.
 - `value`: Vorbelegung mit einem bestimmten Wert



Einzeilige Felder: Zahlen

input type number



```
<form action="auswertungsscript.php">
```

```
  Nummer: <br/>
```

```
  <input type="number" name="beispiel" min="30" max="225" step="0.5" value="30">
</form>
```

Nummer:

Nummer:



```
<form action="auswertungsscript.php">
```

```
Wert auswählen: <br/>
```

```
<input type="range" name="beispiel" min="0" max="100" value="30">
```

```
</form>
```

Wert auswählen:



- Das Eingabefeld `<input type="range">` definiert einen Schieberegler, einen sogenannten Slider.
- Folgende Attribute können verwendet werden:
 - `max`: legt den Maximalwert fest, der Standardwert ist 100
 - `min`: legt den Mindestwert fest, der Standardwert ist 0
 - `step`: legt fest, in welchen Stufen die Werte eingegeben werden dürfen.
 - `orient: vertical` legt fest, dass der Slider vertikal angezeigt werden soll.
 - `value`: Vorbelegung mit einem bestimmten Wert, an dem der Schieberegler eingestellt ist.



Einzeilige Felder: Datum und Zeit

input type="date"



```
<form action="auswertungsscript.php">
  Datum: <br/>
  <input type="date" name="beispiel">
</form>
```

Datum:

Datum:

- HTML5 bietet eine Vielzahl von neuen Eingabe-Typen für Datums- und Zeitangaben.
- Bei mobilen Geräten wird dann ein Datumsfeld oder sogar der Kalender eingeblendet, was die Eingabe erleichtert.
- Mit `input type="date"` können Sie Datumseingaben abfragen.



Einzeilige Felder: Datum und Zeit

input type datetime-local



```
<form action="auswertungsscript.php">  
  Datum und Uhrzeit: <br/>  
  <input type="datetime-local" name="beispiel">  
</form>
```

Datum und Uhrzeit:

Datum und Uhrzeit:

- Mit `input type="datetime-local"` können Sie eine Datumseingabe mit einer Zeitangabe ohne Zeitzone kombinieren.



- Es gibt einige spezifische Input-Typen, wie z.B.:
 - tel
 - date
 - datetime-local
 - week
 - month
 - usw.
- Diese sind nicht in jedem Browser implementiert und können daher Probleme verursachen.
- In der Regel ersetzen Browser Input-Typen, die sie nicht kennen, mit dem Input-Typ „text“, die Funktionalität bleibt so erhalten.
- Wird dennoch ein spezifisches Eingabeformat benötigt, muss dies z.B. mit JavaScript erneut geprüft werden.



```
<form action="auswertungsscript.php">
```

```
  Ihr Text: <br/>
```

```
  <textarea cols="40" rows="5">
```

```
  Darts ist ein Sport, der in den letzten Jahren...
```

```
  </textarea>
```

```
</form>
```

Ihr Text:

Darts ist ein Sport, der in den letzten Jahren...

- Ein textarea-Element ist ein mehrzeiliges Eingabefeld.
- Es dient z.B. der Eingabe von mehrzeiligen Texten, etwa Kommentaren, Forumsbeiträgen oder Benachrichtigungen.
- Das textarea-Element kann über die rechte untere Ecke in seiner Größe verändert werden.
- Sie können seine Größe aber auch über Attribute oder über CSS festlegen.



- Folgende Attribute sind möglich:
 - name: Für die weitere serverseitige Verarbeitung benötigen auch die Textareas ein name-Attribut.
 - rows bestimmt die Anzahl der angezeigten Zeilen und damit die Höhe.
 - cols bestimmt die Anzahl der angezeigten Spalten.
 - „Spalten“ bedeutet dabei die Anzahl der Zeichen pro Zeile.
 - readonly: Lässt keine Eingaben zu.



- Radio-Buttons sind eine Gruppe von beschrifteten meist runden Knöpfen, von denen der Anwender einen auswählen kann.
- Es kann immer nur einer der Radio-Buttons ausgewählt sein.
- Folgende Attribute sind möglich:
 - name: Jeder Radio-Button erhält einen internen Bezeichnernamen. Alle Radio-Buttons, die den gleichen Namen haben, gehören zu einer Gruppe, d.h. von diesen Buttons kann der Anwender genau einen markieren.
 - value: bestimmt einen internen Bezeichnerwert für jeden Radio-Button. Wenn man das Formular abschickt, wird der Bezeichnerwert des markierten Buttons übertragen.
 - checked: Wenn Sie eine der Auswahlmöglichkeiten vorselektieren wollen, dann notieren Sie in dem <input>-Tag des entsprechenden Radio-Buttons das alleinstehende Attribut checked.



Treffen einer Auswahl

input type radio



```
<form action="auswertungsscript.php">
```

```
Ihre Lieblingsband: <br/>
```

```
<input type="radio" name="band" value="chainsmokers" checked> The Chainsmokers<br/>
```

```
<input type="radio" name="band" value="coldplay"> Coldplay<br/>
```

```
<input type="radio" name="band" value="dragons"> Imagine Dragons<br/>
```

```
</form>
```

Ihre Lieblingsband:

- The Chainsmokers
- Coldplay
- Imagine Dragons



- Eine Checkbox ist ein ankreuzbares, meist quadratisches Kontrollfeld.
- Der Benutzer kann es auswählen oder die Auswahl entfernen.
- Die Werte von ausgewählten Checkboxen werden beim Absenden des Formulars mit übertragen.
- Es lassen sich auch Gruppen von Checkboxen mit gleichen Namen, was technisch gesehen nur für die Übertragung wichtig ist, da sich Checkboxen auch in einer Gruppe unabhängig voneinander auswählen lassen.
- Checkboxen in HTML kennen nur die Zustände ausgewählt und nicht ausgewählt.
- Die möglichen Attribute sind identisch zu den Radio-Buttons.



Treffen einer Auswahl input type checkbox



```
<form action="auswertungsscript.php">
  Lieblingsgerichte: <br/>
  <input type="checkbox" name="gericht" value="pizza" /> Pizza<br/>
  <input type="checkbox" name="gericht" value="paella" /> Paella<br/>
  <input type="checkbox" name="gericht" value="bratkartoffeln" /> Bratkartoffeln<br/>
</form>
```

Lieblingsgerichte:

- Pizza
- Paella
- Bratkartoffeln



- Um die barrierefreie Nutzung von Webseiten zu gewährleisten, sollten alle Formulare sowohl gruppiert als auch ausreichend beschriftet werden.
- Zudem wird die anklickbare Fläche vergrößert, wenn Input-Elemente gruppiert werden.
- Der aktuelle Stand des letzten Formulars war wie folgt:

```
<form action="auswertungsscript.php">
  Lieblingsgerichte: <br/>
  <input type="checkbox" name="gericht" value="pizza"/> Pizza<br/>
  <input type="checkbox" name="gericht" value="paella"/> Paella<br/>
  <input type="checkbox" name="gericht" value="bratkartoffeln"/> Bratkartoffeln<br/>
</form>
```



- Angepasstes Beispiel, nachdem alle notwendigen Label hinzugefügt wurden:

```
<form action="auswertungsscript.php">
  Lieblingsgerichte: <br/>
  <input id="g1" type="checkbox" name="gericht" value="pizza"/>
  <label for="g1">Pizza</label>
  <br/>
  <input id="g2" type="checkbox" name="gericht" value="paella"/>
  <label for="g2">Paella</label>
  <br/>
  <input id="g3" type="checkbox" name="gericht" value="bratkartoffeln"/>
  <label for="g3">Bratkartoffeln</label>
</form>
```



Gruppieren von Formularelementen fieldset und legend



- Mit dem fieldset-Element können Elemente innerhalb von Formularen gruppiert werden.
- Mit dem legend-Element kann zusätzlich eine Überschrift für das fieldset-Element definiert werden.
- Standardmäßig zeichnen die Browser einen Rahmen um das fieldset.
- Das legend-Element liegt genau auf diesem Rahmen.



- Das Element fieldset kann unter HTML5 folgende Attribute haben:
 - disabled:
Mit dem Wert disabled kann das fieldset ausgegraut angezeigt, aber nicht auswählbar gemacht werden.
 - form:
Mit dem Wert der id eines Formulars kann es diesem zugeordnet werden.
 - name:
Durch den Namen kann es in einem verarbeitenden Script ausgewertet werden.



- Angepasstes Beispiel, nachdem die Formularelemente gruppiert wurden:

```
<form action="auswertungsscript.php">
  <fieldset>
    <legend>Lieblingsgerichte</legend>
    <input id="g1" type="checkbox" name="gericht" value="pizza"/>
    <label for="g1">Pizza</label>
    <br/>
    <input id="g2" type="checkbox" name="gericht" value="paella"/>
    <label for="g2">Paella</label>
    <br/>
    <input id="g3" type="checkbox" name="gericht" value="bratkartoffeln"/>
    <label for="g3">Bratkartoffeln</label>
  </fieldset>
</form>
```

<p>Lieblingsgerichte</p> <p><input type="checkbox"/> Pizza</p> <p><input type="checkbox"/> Paella</p> <p><input type="checkbox"/> Bratkartoffeln</p>
