

Formulare-Spickzettel

Webformulare erfassen Eingaben, bieten Auswahlmöglichkeiten und senden Daten zur Verarbeitung. Gute Formulare sind beschriftet, gegliedert und für Tastatur, Screenreader und Mobilgeräte nutzbar.



Grundgerüst

```
<form action="/kontakt" method="post">
  <fieldset>
    <legend>Kontakt</legend>
    <label for="mail">E-Mail</label>
    <input id="mail" name="mail" type="email"
required>
    <button>Nachricht senden</button>
  </fieldset>
</form>
```

form Umschließt alle Formular-Steuer-elemente.
action Zieladresse für die Verarbeitung der Daten.
method get für Abfragen, post für größere oder verändernde Daten.

Jedes übermittelte Feld braucht ein `name`-Attribut; ohne `name` wird der Wert nicht gesendet.

Beschriften und gliedern

```
<fieldset>
  <legend>Lieferadresse</legend>
  <label for="street">Straße</label>
  <input id="street" name="street"
autocomplete="street-address">
</fieldset>
```

label Verknüpft Text mit einem Feld. Klick auf den Text fokussiert das Feld.
fieldset Gruppiert zusammengehörige Eingaben.
legend Benennung einer Feldgruppe.

Platzhalter sind Beispiele, keine Beschriftungen. Ein sichtbares `label` bleibt auch nach der Eingabe verständlich.

Wichtige Eingabetypen

Text und Kontakt

text Einzeiliger Text.
email E-Mail-Tastatur und Formatprüfung.
tel Telefon-Tastatur, kein festes Format.
url Webadresse.

Zahlen und Zeit

number Zahl mit min, max, step.
range Schieberegler.
date Datumsauswahl.
time Uhrzeit.

Weitere Typen

password Maskierte Eingabe.
file Dateiauswahl, oft mit accept.
color Farbwähler.
hidden Versteckter Wert.

Auswählen: Radio

```
<label>
  <input type="radio" name="kontakt" value="mail"> E-
Mail
</label>
<label>
  <input type="radio" name="kontakt" value="tel">
Telefon
</label>
```

Radio-Buttons wählen genau eine Option aus einer Gruppe. Alle Optionen teilen denselben `name`.

Kontaktweg

- ☒ E-Mail
☐ Telefon

Wenige Optionen sind als Radio-Buttons oft schneller erfassbar als ein Dropdown.

Auswählen: Checkbox

```
<label>
  <input type="checkbox" name="themen" value="html">
HTML
</label>
<label>
  <input type="checkbox" name="themen" value="css">
CSS
</label>
```

Checkboxen stehen für mehrere unabhängige Ja/Nein-Entscheidungen.

Themen

- ☒ HTML
☐ CSS

Auswählen: Select

```
<select name="land">
  <option value="de">Deutschland</option>
  <option value="at">Österreich</option>
  <option value="ch">Schweiz</option>
</select>
```

`select` ist bei vielen Optionen platzsparend. Jede Option sollte einen sinnvollen `value` haben.

Land

Deutschland ▼

Vorschläge: Datalist

```
<input list="browser" name="browser">
<datalist id="browser">
  <option value="Firefox">
  <option value="Chrome">
</datalist>
```

`datalist` bietet Vorschläge für ein Eingabefeld; freie Eingabe bleibt möglich.

Browser

Attribute und Validierung

Pflicht und Hilfe

required Pflichtfeld.
placeholder Beispielttext, kein Label.
autocomplete Hilft beim Ausfüllen; bei Logins vorsichtig.
inputmode Steuert die Bildschirmrastatur, etwa `numeric`.

Grenzen

min, max Wertebereich.
minlength Textlänge.
maxlength
pattern Regulärer Ausdruck als Zusatzprüfung.

Status

disabled Nicht bedienbar, wird nicht gesendet.
readonly Nicht änderbar, wird gesendet.
checked Vorauswahl bei Checkbox und Radio.
accept Dateitypen begrenzen,
multiple mehrere Dateien erlauben.

Browserspezifische Validierung ist hilfreich, ersetzt aber keine serverseitige Prüfung.

Mit CSS reagieren

```
input:invalid {
  border-color: #dd6200;
}
input:required {
  border-inline-start: thick solid;
}
input:disabled {
  opacity: .6;
}
```

:required Pflichtfelder und freiwillige Felder unterscheiden.
:optional
:valid Auf gültige oder ungültige Eingaben reagieren.
:invalid
:checked Ausgewählte oder deaktivierte Steuerelemente
:disabled gestalten.

Buttons und Daten senden

```
<button type="submit">Bestellung abschicken</button>
<button type="button">Vorschau zeigen</button>
<button type="reset">Zurücksetzen</button>
```

submit Sendet das Formular.
button Neutraler Button für JavaScript-Aktionen.
reset Setzt Eingaben zurück; selten sinnvoll.

Buttontexte sollten die Aktion benennen: „Konto erstellen“, „Nachricht senden“, „Datei hochladen“.

Kurze Prüfliste

- Jedes Feld sichtbar mit `label` beschriften.
- Zusammengehöriges mit `fieldset` und `legend` gliedern.
- Passenden `type`, `autocomplete` und `inputmode` wählen.
- Labels oben, Fehler verständlich und nahe am Feld.

Häufige Stolperstellen

- `placeholder` nicht als Label verwenden.
- Radio-Gruppen brauchen denselben `name`.
- `disabled`-Felder werden nicht gesendet.
- Clientseitige Validierung nicht als einzige Prüfung nutzen.

