

# HTML Grundlagen



# Übersicht

- Was ist HTML?
- HTML Elemente
- HTML Attribute
- HTML Boilerplate
- HTML Tags



# Was ist HTML?

HTML



HTML the Skeleton



CSS



CSS the Skin



JS



Javascript the Brain



# Was ist HTML?

- HTML steht für **Hypertext Markup Language**
- Hypertext = Text verlinkt zu weiteren Websites
- Markup Language = kennzeichne Teile des Inhalts, wie "das ist eine Überschrift", "das ist ein Absatz", "das ist ein Link", usw.
- HTML beschreibt die Struktur einer Webseite.
- HTML besteht aus einer Reihe von Elementen.
- HTML-Elemente sagen dem Browser, wie der Inhalt angezeigt werden soll.

## World Wide Web

The WorldWideWeb (W3) is a wide-area [hypermedia](#) information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an [executive summary](#) of the project, [Mailing lists](#) , [Policy](#) , November's [W3 news](#) , [Frequently Asked Questions](#) .

### [What's out there?](#)

Pointers to the world's online information, [subjects](#) , [W3 servers](#), etc.

### [Help](#)

on the browser you are using

### [Software Products](#)

A list of W3 project components and their current state. (e.g. [Line Mode](#) ,X11 [Viola](#) , [NeXTStep](#) , [Servers](#) , [Tools](#) , [Mail robot](#) , [Library](#) )

### [Technical](#)

Details of protocols, formats, program internals etc

### [Bibliography](#)

Paper documentation on W3 and references.

### [People](#)

A list of some people involved in the project.

### [History](#)

A summary of the history of the project.

### [How can I help ?](#)

If you would like to support the web..

### [Getting code](#)

Getting the code by [anonymous FTP](#) , etc.

Erste Website: <https://info.cern.ch/hypertext/WWW/TheProject.html>

# HTML Elemente

- HTML Elemente beinhalten Start und End-Tag
- Inhalt steht zwischen Start und End-Tag
- HTML-Element können verschachtelt werden

```
<tagname>Content goes here...</tagname>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1>My First Heading</h1>
```

```
<p>My first paragraph.</p>
```

```
</body>
```

```
</html>
```

# HTML Attribute

- Alle HTML-Elemente können Attribute haben
- Attribute liefern zusätzliche Informationen über Elemente
- Attribute werden immer im Start-Tag angegeben
- Attribute kommen in Namen/Wert-Paaren vor, wie zum Beispiel: name="Wert", oder ohne Werte

```
<a href="https://www.w3schools.com">Visit W3Schools</a>
```

```

```

```
<input width="500" height="600" disabled></input>
```

# Die wichtigsten HTML Attribute

- id: Eindeutige Kennzeichnung für HTML-Elemente.
- class: Hinzufügen von CSS-Klassen zur Stilgestaltung.
- src: Angabe der Quell-URL für Bilder und Multimedia.
- href: Ziel-URL für Hyperlinks.
- alt: Alternative Textbeschreibung für Bilder.
- title: Anzeige eines Mouseover-Titels.
- width und height: Bildabmessungen in Pixeln.
- target: Festlegung, wie Hyperlinks geöffnet werden.
- type: Angabe des Typs von Formularelementen.
- disabled: Deaktivierung von Formularelementen.

```
<select id="mySelect" class="myClass"></input>
```

```

```

```
<a href="https://www.w3schools.com">Visit W3Schools</a>
```

```
<input title="myTitle" type="checkbox" disabled></input>
```

# HTML Boilerplate

- **Boilerplate:** ein vordefiniertes und wiederverwendbares Grundgerüst oder eine Vorlage, die in verschiedenen Kontexten verwendet werden kann
- **<!DOCTYPE html>:** Deklaration des HTML-Dokumenttyps für Browser.
- **<html lang="de">:** Start des HTML-Dokuments mit Angabe der Sprache (deutsch).
- **<head>:** Kopfbereich des Dokuments für Meta-Informationen und Verknüpfungen zu externen Ressourcen.
- **<meta charset="UTF-8">:** Einstellung der Zeichencodierung auf UTF-8.
- **<meta name="viewport" content="width=device-width, initial-scale=1.0">:** Einstellungen für die responsive Darstellung auf verschiedenen Geräten.
- **<title>:** Titel der Webseite im Browser-Tab.
- **<body>:** Hauptinhalt der Webseite.
- **<header>:** Kopfbereich der Webseite.

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



# Übung zu HTML Boilerplate

- Nehmt euch 5 Minuten Zeit, um das HTML Boilerplate genau zu studieren.
- Danach versucht ihr es, ohne auf die Vorlage zu blicken nachzuschreiben
- Wenn ihr nicht mehr weiterkommt, werft erneut einen kurzen Blick auf die Vorlage

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
  </head>
  <body>
    <!--Hier steht der Inhalt der Seite-->
  </body>
</html>
```

# HTML5 Tags

**New** [tags added in HTML5]

<b>&lt;article&gt;</b>	self-contained compilation that is independently distributable	<b>&lt;details&gt;</b>	details of an element	<b>&lt;output&gt;</b>	represents results of calculation
<b>&lt;aside&gt;</b>	section of page that consists of content tangentially related to content around it	<b>&lt;embed&gt;</b>	embedded content	<b>&lt;progress&gt;</b>	progress of any kind of task
<b>&lt;audio&gt;</b>	sound content	<b>&lt;figcaption&gt;</b>	caption of figure element	<b>&lt;rp&gt;</b>	parenthesized ruby text
<b>&lt;bdi&gt;</b>	span of text to be isolated from surroundings for bidirectional formatting purposes	<b>&lt;figure&gt;</b>	group of media content	<b>&lt;rt&gt;</b>	ruby text
<b>&lt;canvas&gt;</b>	area that can be used to draw graphics via JavaScript	<b>&lt;footer&gt;</b>	footer for section or page	<b>&lt;ruby&gt;</b>	ruby annotations
<b>&lt;command&gt;</b>	user invokable command	<b>&lt;header&gt;</b>	header for section or page	<b>&lt;section&gt;</b>	section in a document
<b>&lt;datalist&gt;</b>	dropdown list	<b>&lt;hgroup&gt;</b>	group of headings for section	<b>&lt;source&gt;</b>	media resources
<b>&lt;datatemplate&gt;</b>	data template	<b>&lt;keygen&gt;</b>	generated key in a form	<b>&lt;summary&gt;</b>	header of a detail element
		<b>&lt;mark&gt;</b>	marked text	<b>&lt;time&gt;</b>	date/time
		<b>&lt;meter&gt;</b>	measurement in defined range	<b>&lt;video&gt;</b>	video
		<b>&lt;nav&gt;</b>	navigation links	<b>&lt;br&gt;</b>	possible line break

**Old** [unsupported tags]

<code>&lt;acronym&gt;</code>	<i>acronym</i>	<code>&lt;isindex&gt;</code>	<i>provides searchable index related to current document</i>
<code>&lt;applet&gt;</code>	<i>applet</i>	<code>&lt;dir&gt;</code>	<i>directory list</i>
<code>&lt;basefont&gt;</code>	<i>base font</i>	<code>&lt;noembed&gt;</code>	<i>no embed section</i>
<code>&lt;bgsound&gt;</code>	<i>background sound</i>	<code>&lt;noframes&gt;</code>	<i>no frame section</i>
<code>&lt;big&gt;</code>	<i>big text</i>	<code>&lt;s&gt;</code>	<i>strikethrough text</i>
<code>&lt;center&gt;</code>	<i>centered text</i>	<code>&lt;strike&gt;</code>	<i>strikethrough text</i>
<code>&lt;fn&gt;</code>	<i>footnotes</i>	<code>&lt;tt&gt;</code>	<i>teletype text</i>
<code>&lt;font&gt;</code>	<i>text font, size, and color</i>	<code>&lt;u&gt;</code>	<i>underlined text</i>
<code>&lt;frame&gt;</code>	<i>sub window</i>	<code>&lt;xmp&gt;</code>	<i>preformatted text</i>
<code>&lt;frameset&gt;</code>	<i>set of frames</i>		

**Existing** [tags in HTML4 & 5]

<code>&lt;!--&gt;</code>	comment	<code>&lt;code&gt;</code>	code text
<code>&lt;!doctype&gt;</code>	document type	<code>&lt;col&gt;</code>	attributes for columns
<code>&lt;a&gt;</code>	hyperlink	<code>&lt;colgroup&gt;</code>	groups of columns
<code>&lt;abbr&gt;</code>	abbreviation	<code>&lt;dd&gt;</code>	definition description
<code>&lt;address&gt;</code>	address element	<code>&lt;del&gt;</code>	deleted text
<code>&lt;area&gt;</code>	image map area	<code>&lt;div&gt;</code>	generic block-level element
<code>&lt;b&gt;</code>	bold text	<code>&lt;dfn&gt;</code>	defining instance of a term
<code>&lt;base&gt;</code>	base URL for all links in page relative to document root	<code>&lt;dl&gt;</code>	definition list
<code>&lt;bdo&gt;</code>	text direction	<code>&lt;dt&gt;</code>	definition term
<code>&lt;blockquote&gt;</code>	long quotation	<code>&lt;em&gt;</code>	emphasized text
<code>&lt;body&gt;</code>	body element	<code>&lt;fieldset&gt;</code>	logically group items in a form
<code>&lt;br&gt;</code>	single line break	<code>&lt;form&gt;</code>	defines a form
<code>&lt;button&gt;</code>	push button	<code>&lt;h1&gt; to &lt;h6&gt;</code>	header 1 to header 6
<code>&lt;caption&gt;</code>	table caption	<code>&lt;head&gt;</code>	document information
<code>&lt;cite&gt;</code>	citation	<code>&lt;hr&gt;</code>	horizontal rule

<html>	html document
<i>	italic text
<iframe>	inline sub window
<img>	image
<input>	input field
<ins>	inserted text
<kbd>	keyboard text
<label>	label for a form control
<legend>	title in a fieldset
<li>	list item
<link>	resource reference
<map>	image map
<menu>	menu list
<meta>	meta information
<noscript>	no script section

<code>&lt;object&gt;</code>	<i>embedded object</i>	<code>&lt;sub&gt;</code>	<i>subscripted text</i>
<code>&lt;ol&gt;</code>	<i>ordered list</i>	<code>&lt;sup&gt;</code>	<i>superscripted text</i>
<code>&lt;optgroup&gt;</code>	<i>option group</i>	<code>&lt;table&gt;</code>	<i>table</i>
<code>&lt;option&gt;</code>	<i>option in a drop-down list</i>	<code>&lt;tbody&gt;</code>	<i>table body</i>
<code>&lt;p&gt;</code>	<i>paragraph</i>	<code>&lt;td&gt;</code>	<i>table cell</i>
<code>&lt;param&gt;</code>	<i>parameter for an object</i>	<code>&lt;textarea&gt;</code>	<i>text area</i>
<code>&lt;pre&gt;</code>	<i>preformatted object</i>	<code>&lt;tfoot&gt;</code>	<i>table footer</i>
<code>&lt;q&gt;</code>	<i>short quotation</i>	<code>&lt;th&gt;</code>	<i>table header</i>
<code>&lt;samp&gt;</code>	<i>sample computer code</i>	<code>&lt;thead&gt;</code>	<i>wraps row containing table headers</i>
<code>&lt;script&gt;</code>	<i>script</i>	<code>&lt;title&gt;</code>	<i>document title</i>
<code>&lt;select&gt;</code>	<i>selectable list</i>	<code>&lt;tr&gt;</code>	<i>table row</i>
<code>&lt;small&gt;</code>	<i>small text</i>	<code>&lt;ul&gt;</code>	<i>unordered list</i>
<code>&lt;span&gt;</code>	<i>inline generic container</i>	<code>&lt;var&gt;</code>	<i>variable</i>
<code>&lt;strong&gt;</code>	<i>strong text</i>		
<code>&lt;style&gt;</code>	<i>style definition</i>		

*Brought to you by*



Brought to you by:  
**inmotion**  
hosting

<https://html.com/blog/html-5-cheat-sheets/>

# Die wichtigsten HTML5 Tags

## Formattierung

- Division `<div>`
- Heading `<h1>...<h6>`
- Paragraph `<p>`
- List `<ul>`, `<ol>`
- Break `<br>`

## Inhalt

- Table `<table>`
- Image `<img>`
- Textarea `<textarea>`

## Funktional

- Anchor `<a>`
- Select `<select>`
- Input `<input>`
- Button `<button>`

Diese List ist nur ein Auszug aus häufig verwendeten Tags.  
Im Laufe des Kurses, werden wir durch Übungen noch viele  
weitere Tags kennen lernen und auch anwenden

# Übung zu HTML Tags „Kochrezept“

- Befüllt euer zuvor erstelltes Boilerplate mit Inhalt
- Welches Kochrezept würdet ihr gerne mit uns teilen? Jetzt habt ihr die Chance dazu
- Verwendet dafür mindestens 5 der bekannten HTML-Tags

## Rezepte

### Kartoffelpuffer

[Kartoffelpuffer](#) sind schnell selbst gemacht und können süß oder herzhaft genossen werden. Hier das Grundrezept für vier Personen.

#### Zutaten

- 1 kg Kartoffeln
- 2 Eier
- Salz
- Öl zum Braten

#### Zubereitung

Die rohen Kartoffeln werden geschält und gerieben. Die Kartoffelmasse wird dann ausgedrückt, mit den Eiern verrührt und leicht gesalzen.

Anschließend etwas Öl in einer Pfanne erhitzen, kleine Fladen aus der Kartoffelmasse mit einem Löffel in die Pfanne geben und braten, bis sie goldbraun sind.

# HTML Elemente

- Weitere HTML Elemente im Detail
- Live Preview im online Editor
- Styling (Vorschau wird mit CSS vertieft)
- <https://www.w3schools.com/html/default.asp>



# Iframe Element

- Einbetten von externen oder internen Webseiten und Inhalten im aktuellen HTML File

```
<iframe src="demo_iframe.htm" height="200" width="300" title="Iframe Example"></iframe>
```



🐱 Funniest Cats and Dogs Videos 🐱🐶 || 🐱🐶 Hilarious Animal Compilation №141

# HTML Tabelle

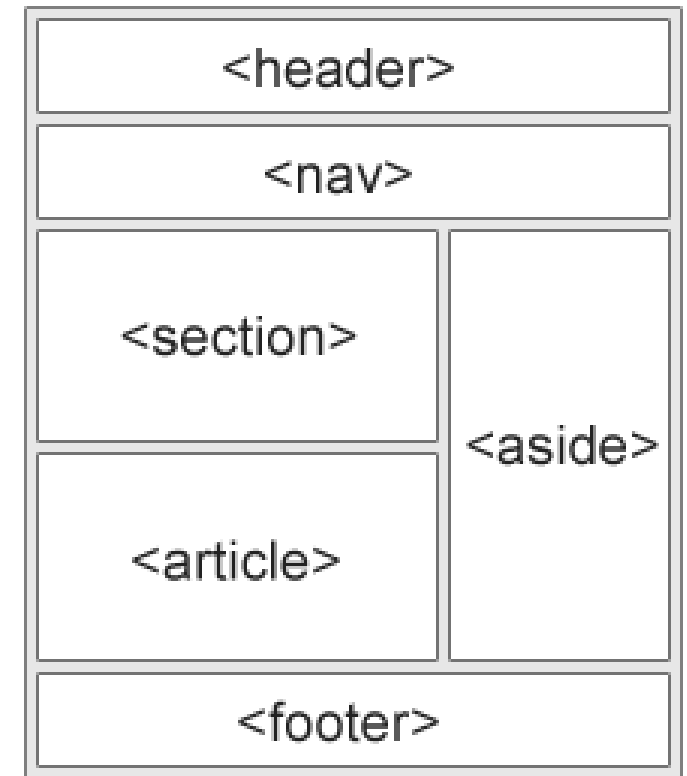
- HTML-Tabellen ermöglichen es, Daten in Zeilen und Spalten anzuordnen.
- `<table>`: Tabellen-Tag
- `<tr>`: table row – Reihen
- `<td>`: table data – Zellen
- `<th>`: table header – Kopfzeile
- `<td colspan = "2">` umfasst 2 Spalten
- `<td rowspan = "2">` umfasst 2 Reihen

```
<table>
  <tr>
    <th>Company</th>
    <th>Contact</th>
    <th>Country</th>
  </tr>
  <tr>
    <td>Alfreds Futterkiste</td>
    <td>Maria Anders</td>
    <td>Germany</td>
  </tr>
  <tr>
    <td>Centro comercial Moctezuma</td>
    <td>Francisco Chang</td>
    <td>Mexico</td>
  </tr>
  <tr>
    <td>Codersbay</td>
    <td>Dominic Koch</td>
    <td>Austria</td>
  </tr>
</table>
```

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Centro comercial Moctezuma	Francisco Chang	Mexico
Codersbay	Dominic Koch	Austria

# Semantische HTML Tags

- **Klarere Struktur und Bedeutung:**  
Sie beschreiben den Inhalt genauer und machen deutlich, welche Rolle verschiedene Teile des Dokuments spielen. Dies erleichtert es Entwicklern und Suchmaschinen, den Inhalt zu verstehen und richtig zu interpretieren.
- **Barrierefreiheit und Zugänglichkeit:**  
Bildschirmlesegeräte und andere assistive Technologien können semantische Tags verwenden, um den Inhalt sinnvoll vorzulesen. Dies ist besonders wichtig für Menschen mit Behinderungen, die auf solche Technologien angewiesen sind.
- **SEO-Vorteile:**  
Suchmaschinen wie Google bewerten Webseiten besser, wenn semantische Tags verwendet werden. Dies kann dazu beitragen, dass Ihre Webseite in den Suchergebnissen höher gerankt wird, da Suchmaschinen den Inhalt besser verstehen und interpretieren können.





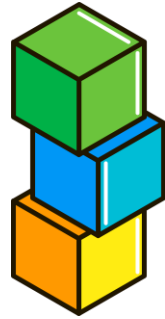
# HTML Übung „My Personal IMDb“

- Erstelle eine Tabelle mit den TOP 5 der besten Filme aller Zeiten
- Die Tabelle sollte folgende Spalten beinhalten
  - Filmtitel (ersten 3 Plätze mit Link zu Detailseite)
  - Genre
  - Regisseur
  - Erscheinungsjahr
- Der Filmtitel verlinkt zu einer Detailseite des Films, in der ein Trailer des Films und eine kurze Zusammenfassung des Films angezeigt wird. Hier bieten sich zusätzlich semantische Tags an.
- [https://www.imdb.com/search/title/?groups=top\\_100&sort=user\\_rating,desc](https://www.imdb.com/search/title/?groups=top_100&sort=user_rating,desc)





vs.



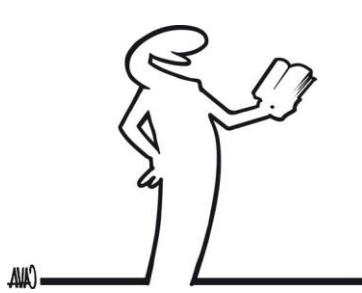
# Elemente

## INLINE ELEMENTE

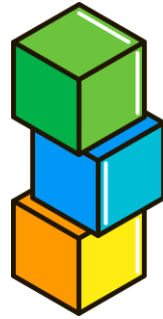
- **Inline-Elemente sind innerhalb von Textblöcken eingebettet:** Inline-Elemente sind HTML-Elemente, die inmitten von Text oder anderen Inline-Elementen platziert werden können, ohne einen Zeilenumbruch zu erzwingen.
- **Sie nehmen nur so viel Platz wie notwendig ein:** Sie erstrecken sich horizontal und reihen sich nebeneinander, solange genügend Platz vorhanden ist, bevor sie zur nächsten Zeile wechseln.
- **Beispiele für Inline-Elemente sind:** `<a>`, `<strong>`, `<em>`, `<span>`, `<img>`, `<br>`

## BLOCK ELEMENTE

- **Block-Elemente erzwingen Zeilenumbrüche:** Block-Elemente nehmen die gesamte verfügbare Breite ihres Elternelements ein und erzwingen einen Zeilenumbruch danach.
- **Sie werden oft für strukturelle Elemente verwendet:** Block-Elemente werden häufig verwendet, um strukturelle Teile einer HTML-Seite zu definieren
- **Beispiele für Block-Elemente sind:** `<div>`, `<p>`, `<h1>`, `<ul>`, `<li>`, `<table>`.



vs.



# Elemente

## INLINE ELEMENTE

- **Inline-Elemente sind innerhalb von Textblöcken eingebettet:** Inline-Elemente sind HTML-Elemente, die inmitten von Text oder anderen Inline-Elementen platziert werden können, ohne einen Zeilenumbruch zu erzwingen.

- **Sie nehmen nur Platz ein, wenn Text vorhanden ist, bei dem ein Zeilenumbruch erforderlich ist.** Sie erstrecken sich von Text verwendet zu werden, ohne Zeilenumbrüche zu erzwingen, während Block-Elemente dazu dienen, den Textfluss zu unterbrechen und strukturelle Elemente auf der Seite zu definieren.

- **Beispiele für Inline-Elemente sind:** `<a>`, `<strong>`, `<em>`, `<span>`, `<img>`, `<br>`

## BLOCK ELEMENTE

- **Block-Elemente erzwingen einen Zeilenumbruch:** Block-Elemente erzwingen einen Zeilenumbruch, indem sie die gesamte verfügbare Breite des Containers einnehmen und erzwingen, dass der Text auf einer neuen Zeile beginnt.
- **Sie werden häufig für strukturelle Elemente verwendet:** Block-Elemente werden häufig verwendet, um strukturelle Teile einer HTML-Seite zu definieren.

- **Beispiele für Block-Elemente sind:** `<div>`, `<p>`, `<h1>`, `<ul>`, `<li>`, `<table>`.

# HTML „Kontaktformular“ Übung

- Erstelle eine Formular `<form></form>` mit Eingabefeldern um eine Kontaktanfrage zu versenden
- Folgende Eingabefelder sollen vorhanden sein:
  - Name (Input)
  - Email (Input)
  - Nachricht (Textarea)
  - Geschlecht (Radio)
  - Interessen (Checkbox)
  - Submit Button `<input type=„submit“>`
- Verwende dafür sowohl Inline als auch Block Elemente.
- Über styling können inline und block Elemente definiert werden Beispiel: `p{display:inline}`

## Contact Us

Name:

Email:

Message:

Gender: ☐ Male ☐ Female ☐ Other

Interests: ☐ Programming ☐ Design ☐ Gaming

# HTML Formulare

- Ein HTML-Formular wird verwendet, um Benutzereingaben zu sammeln. Die Benutzereingabe wird in der Regel an einen Server zur Verarbeitung gesendet
- Ein HTML-Formular ist ein Container für verschiedene Arten von Eingabeelementen, wie zum Beispiel: Textfelder, Kontrollkästchen, Optionsfelder (Radio-Buttons), Absendebutons usw.
- Felder werden über ein serverseitiges Skript (PHP, Nodejs,...) verarbeitet
- Über einen Button `<input type="submit">` wird das Formular abgeschickt

## Contact Us

Name:

Email:

Message:

Gender: ☐ Male ☐ Female ☐ Other

Interests: ☐ Programming ☐ Design ☐ Gaming

```
<form action="./process.php" method="post">
  <section class="personal">
    <!-- Text Input for Name -->
    <label for="name">Name:</label>
    <input type="text" id="name" name="name" required />
```

# HTML Formular Attribute

- Mit dem **action**-Attribut wird das Backend Skript angegeben, welches beim Absenden aufgerufen wird
- Das Attribut **method** gibt den Typ des HTTP Requests an (GET, POST, PUT oder DELETE)
- Das Attribut **autocomplete="on"** definiert ob ein Formular Autocomplete verwendet oder nicht
- Das Attribut „**novalidate**“ definiert ob das Formular validiert wird oder nicht

## Contact Us

Name:

Email:

Message:

Gender: ☐ Male ☐ Female ☐ Other

Interests: ☐ Programming ☐ Design ☐ Gaming

```
<form action="./process.php" method="post">
  <section class="personal">
    <!-- Text Input for Name -->
    <label for="name">Name:</label>
    <input type="text" id="name" name="name" required />
```

# HTML Formular Input Attribute

- Daten der Felder werden über das **name**-Attribut an das Backend übergeben
- Über das Attribut **type** kann die Art des Formularfelds angegeben werden. Beispiel: `<input type="email" />`
- Das **readonly** Attribut gibt an dass das Feld nicht bearbeitet werden kann. Feldwerte werden aber an das Backend übertragen
- Felder mit dem Attribut **disabled** können nicht bearbeitet werden und werden auch nicht an das Backend geschickt
- Pflichtfelder erhalten das Attribut **required** und werden beim Senden validiert
- Der Input-type **submit** schickt das Formular mit den eingegebenen Feldwerten an das Backend `<input type="submit" />`

## Contact Us

Name:

Email:

Message:

Gender: ☐ Male ☐ Female ☐ Other

Interests: ☐ Programming ☐ Design ☐ Gaming

```
<form action="./process.php" method="post">
  <section class="personal">
    <!-- Text Input for Name -->
    <label for="name">Name:</label>
    <input type="text" id="name" name="name" required />
```

# Exkurs: php als Backend

- Siehe Codebeispiel im Kursrepository „process.php“





# Fragen & Antworten

- Zeit, um eure Fragen zu klären!



# HTML Wiederholungsübung

## „Coffee-Time“

- Erstelle eine Webseite für eine fiktive Café-Eröffnung, die eine Überschrift, einen Absatz, ein eingebettetes Video, ein Formular zur Anmeldung für den Newsletter und eine Tabelle mit den Angeboten des Cafés enthält.
- Zusatzaufgabe:  
Die Anmeldung zum Newsletter soll auf eine extra Seite „newsletter.html“ und die Preise auf eine extra Seite „preise.html“ ausgelagert werden und über Links erreichbar sein.

### Willkommen zur großen Eröffnung unseres Cafés!

Wir freuen uns, Sie bald in unserem neuen Café begrüßen zu dürfen! Erleben Sie feinste Kaffeespezialitäten, frisches Gebäck und eine gemütliche Atmosphäre. Besuchen Sie uns am Eröffnungstag, um tolle Eröffnungsangebote zu genießen.

### Schauen Sie sich unser Café in Aktion an!



### Melden Sie sich für unseren Newsletter an!

Name:

E-Mail:

Newsletter auswählen:  
Wöchentliche Updates ▼

### Unsere Eröffnungsangebote

Produkt	Beschreibung	Preis
Cappuccino	Cremiger Cappuccino mit einem Hauch Zimt	2,50 €
Latte Macchiato	Sanfter Latte mit extra Milchschaum	3,00 €
Schokokuchen	Hausgemachter Schokoladenkuchen	3,50 €

Besuchen Sie uns bald: Musterstraße 123, 12345 Musterstadt | Telefon: 01234/567890

**Viel Erfolg beim Entwickeln!**

