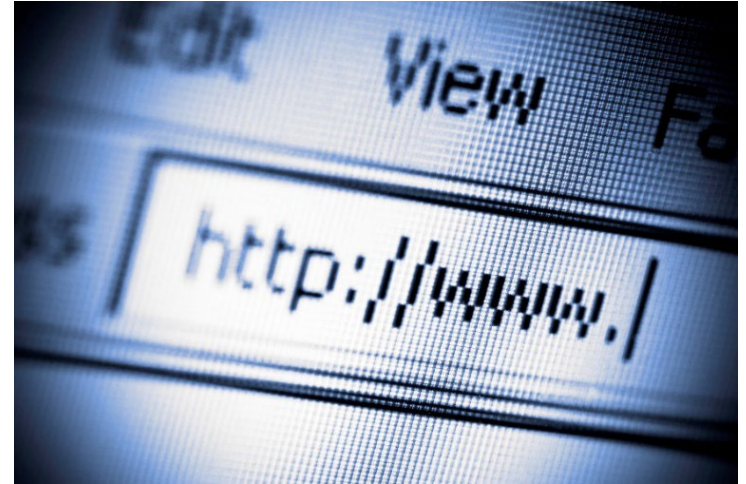


# Web-Technologien

## HTML-Grundlagen

- Syntax und Struktur von HTML-Dokumenten
- URLs und Hyperlinks
- Formulare
- Ausgewählte Elemente

**Letzte Aktualisierung: 24. September 2022**



# Lernziele

- Wissen wie der syntaktische Aufbau von HTML ist.
- Verstehen wie HTML Informationen strukturiert und wie der Aufbau von HTML Dokumenten ist.
- Wissen wie URLs in HTML-Dokumenten verwendet werden.
- Die vorgestellten HTML-Elemente kennen und anwenden können.

# Das erste HTML-Dokument

```
<html>
  <head>
    <title>Hello World</title>
  </head>
  <body>
    <p>Hallo WorldWideWeb!</p>
  </body>
</html>
```

- Man sieht den durch die Tags <...> **strukturierten** Aufbau.
- Zur besseren Lesbarkeit lohnt es sich den Code entsprechend "einzurücken".

# Wer schaut zu HTML?

- Das WHATWG-Konsortium (früher W3C) ist für die Standardisierung zuständig <https://whatwg.org>.
- HTML Living Standard: <https://html.spec.whatwg.org/multipage/>.
- Verschiedene Themen rund um HTML5 werden in eigenen Spezifikationen verabschiedet bzw. sind in der Entwicklung (AAM, HTML Extension Specifications, ARIA, ...). <https://www.w3.org/standards/techs/html#stds>

# Historisches: am Anfang war ..... SGML

- Entstanden Ende der 80er Jahre im CERN.
- HTML ist eine "Auszeichnungssprache" (**H**yper**T**ext **M**arkup **L**anguage) (deklaratives vs. imperatives Programmieren).
- Wurzeln liegen bei SGML (**S**tructure **G**eneralized **M**arkup **L**anguage) aus den 60er Jahren.
- Idee dahinter: "Inhalt/Information strukturieren".
- Beispiel dazu: mit Struktur ist z.B. Titel, Überschrift, Absatz, Block, etc. gemeint.
- Strukturierung erfolgt über besondere Auszeichnungen, den sogenannten Markups (oder Tags).

# Dokumentenstruktur

- Verschachtelung der Strukturen ineinander (Baum).
- Strukturanfang und Ende wird durch öffnenden `<tag>` und schliessenden `</tag>` Markup (oder Tag) gekennzeichnet.
- Äusserste Struktur ist immer `<html>...</html>` (damit man weiss, was drin ist 😊).
- Verschachtelung **muss** korrekt sein.

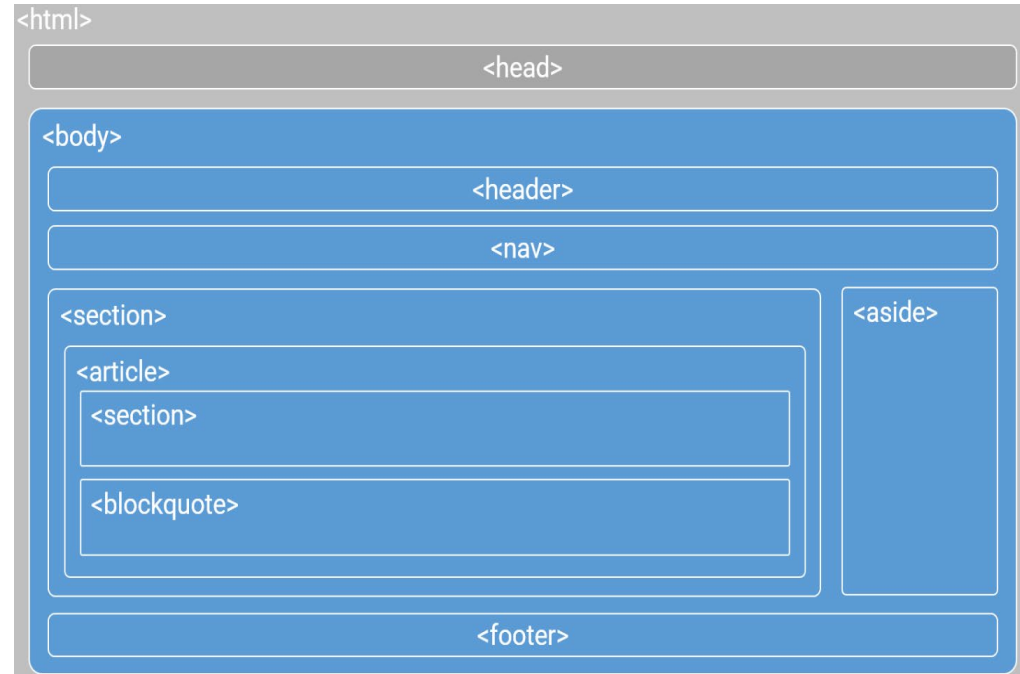


Bild: <https://www.seonative.de/seo-ueberschriften-und-html5-sections/>

# Markups / Tags (Strukturmarkierungen)

- Öffnende und zugehörige schliessende Markups.
- Alleinstehende Markups (die zweite Variante ist XHTML/XML kompatibel!).
- Können Attribute besitzen (immer in Gänsefüsschen!) .

```
<body> ... </body>
```

```
<br> oder <br/>
```

```
<body bgcolor="#AABBCC">
```

# Grundsätzliche Dokumentenstruktur in HTML

- Ein HTML-Dokument besitzt grundsätzlich (mit wenigen Ausnahmen) immer folgende Dokumentenstruktur:

```
<html>
  <head>
    <title>Hello World</title>
  </head>
  <body>
    <p>Hallo WorldWideWeb!</p>
  </body>
</html>
```

- Erweiterungen (**Attribute**), beispielsweise Stilangaben zur Formatierung etc., können zusätzlich angegeben werden.



# Das <head>-Element

- Enthält die Kopfdaten wie:
  - Metainformation (<meta>).
  - Titel (<title>).
  - Stil (<style>).
  - Script-Definitionen (<script>).
  - etc.
- Diese Informationen werden vom Webbrowser nicht im Dokumentenbereich angezeigt.

# Das <body>-Element

- HTML-Element <body> kennzeichnet den Anfang und das Ende des sichtbaren Inhalts des HTML-Dokuments.
- Browser zeigen nur den Inhalt zwischen dem öffnenden und schliessenden body-Tag im Browserfenster.
- Enthält weitere HTML-Elemente, welche die Information strukturieren aber **auch Scripts**, welche an der aufgeführten Stelle ausgeführt werden.
- Ein HTML-Dokument darf nur einen body-Tag haben.

# Text- bzw. Informationsstrukturierung

- Mit Strukturen sind *Absätze, Titel, Überschriften, Listen* etc. gemeint.

## Beispiele:

- Absatz: `<p>..</p>`
- Zeilenumbruch: `<br/>`
- Vorformatierung: `<pre>.. </pre>`
- Überschriften: `<h1>..</h1>` bis `<h6>..</h6>`
- Waagrechte Linien: `<hr/>`
- Container: `<div>...</div>` oder `<span>..</span>`

# URLs in HTML-Dokumenten

- An vielen Stellen benötigt: Hyperlinks, Bilder, Videos, etc.

**Absolute Angabe:** Referenz in anderer Domain als die des HTML-Dokuments

- Vollständige URL angeben, z.B.: <http://www.google/maps>.

**Relative Angabe:** Referenz in gleicher Domain des HTML-Dokuments

- Nicht vollständige URLs werden als relative Angaben interpretiert.
- Basis: URL des aktuellen Dokuments ohne das letzte Pfadelement.
- URL wird **relativ** zur Basis des aktuellen Dokuments angegeben.
- **Beispiel:** "2020/artikel123.html"
  - HTML-Dokument: <http://www.example.com/news/overview.html>

**Basis**

⇒ <http://www.example.com/news/2020/artikel123.html>

# Verknüpfungen (Hypertext-Referenzen)



- Link-Element `<a>`, Attribute `href` ist Ziel, Inhalt vom Benutzer gesehen:

```
<a href="ziel">Inhalt</a>
```

- Externe Verknüpfungen (ausserhalb des HTML-Dokuments): Angabe einer URL

**Beispiele:**

```
<a href="http://www.hslu.ch">HSLU</a>
```

...

```
<a href="artikel/webt.html">Web Technologien</a>
```

- Interne Verknüpfungen (innerhalb des HTML-Dokuments):

Verweis mittels `#` und Verweis auf `id`-Attribut:

**Beispiel:**

```
<a href="#Kapitel1">Kapitel 1</a>
```

...

```
<p id="Kapitel1">Kapitel 1</p>
```

- Mit dem `target`-Attribut kann zudem festgelegt werden wie sich der Link öffnet.

# Grafiken



- Grafikformate gif, jpg, png, svg, ... (viele weitere möglich)

- Einbinden mit:

Absolute oder relative URL

```

```

- alt-Attribut verwenden spezielle Browser z.B. für Blinde oder Suchmaschinen (z.B. Google Bildersuche).

# Imagemaps

Imagemap: Bild mit verlinkbaren Bereichen.

## 1. Definition des Bildes

```

```

## 2. Definitor der Map und der klickbaren Flächen

```
<map name="karte">  
  <area shape="circle" coords="50, 50, 45" href="Ziel.html"/>  
</map>
```

→ Wird zu Navigationszwecken verwendet.



# Metatags

- Werden nicht zur Gestaltung, sondern zur Beschreibung des Inhalts verwendet (daher der Name: Metainformation).
- Werden im head Bereich eingefügt.
- Prinzipieller Aufbau: `<meta name="Schlüsselwort" content="Inhalt">`

## Beispiele von Metatags:

keywords  
description  
language  
robots  
viewport  
copyright

- Kontrolle der Search-Engine-Indexierung
- benötigt für responsive Verhalten





# Sonderzeichen (Entities)

- Damit Sonderzeichen korrekt dargestellt werden, muss das charset Metatag korrekt gesetzt sein.

## Beispiele:

- `<meta charset="utf8">` falls der UTF8-Zeichensatz verwendet wird.
- `<meta charset="iso-8859-1">` falls der ISO-8859-1-Zeichensatz verwendet wird.

- Eine Alternative ist die Zeichen speziell zu kodieren:

**Beispiel:** aus ü wird `&uuml;`;

- Dies ist auch notwendig bei Zeichen, welche mit dem HTML-Markup kollidieren:  
& zu `&amp;`, < zu `&lt;`, > zu `&gt;`, " zu `&quot;`, ' zu `&#39;`.
- Liste mit Beispielen: <http://www.htmlhelp.com/reference/html40/entities/latin1.html>

# Masseinheiten

- Bestimmung der Dimensionen verschiedener Elemente wie Bilder, Schriften, Ränder, Abstände etc.
- Heutzutage werden eher relative Masseinheiten verwendet.

## Relative Einheiten:

- px: (Pixel) Abhängig von der **Pixeldichte** des Ausgabegeräts.
- em: relativ zur Schriftgrösse des Elements M.
- %: (Prozent) typischerweise relativ zum Elternelement (mit Ausnahmen)

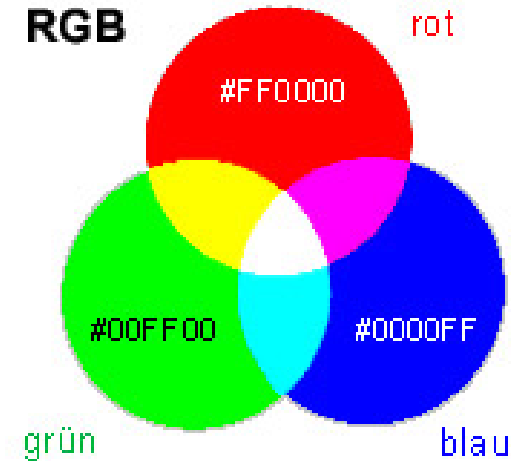
## Absolute Einheiten (eher selten verwendet):

- mm: Millimeter
- in: Inch, 1 Inch entspricht 2.54 Zentimetern
- pt: Punkt, 1 Punkt entspricht 1/72 Inches

# Farben

- Farben werden aus den RGB (rot-grün-blau) Wertangaben gebildet.

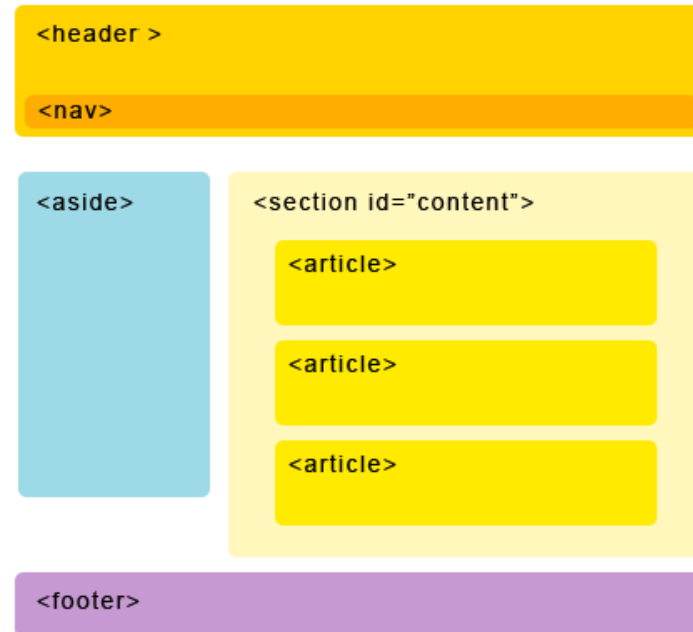
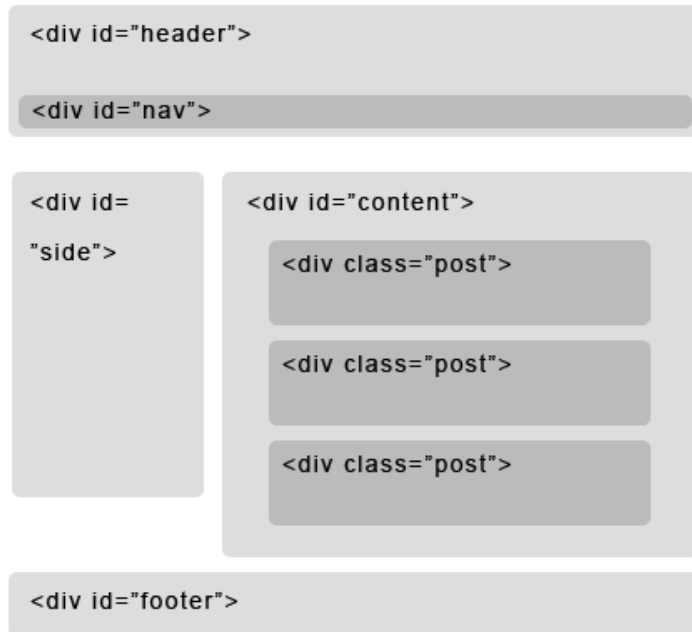
**Beispiel:** #00FF00 ist grün.



- Werte werden in hexadezimaler Form angegeben
- Pro Farbe sind Werte von #00 bis #FF möglich (nicht case-sensitive).
- Einige Farben sind per Namen vordefiniert (Auswahl):

|        |   |         |        |   |         |         |   |         |      |   |         |
|--------|---|---------|--------|---|---------|---------|---|---------|------|---|---------|
| Black  | = | #000000 | Green  | = | #008000 | Maroon  | = | #800000 | Navy | = | #000080 |
| Silver | = | #C0C0C0 | Lime   | = | #00FF00 | Red     | = | #FF0000 | Blue | = | #0000FF |
| Gray   | = | #808080 | Olive  | = | #808000 | Purple  | = | #800080 | Teal | = | #008080 |
| White  | = | #FFFFFF | Yellow | = | #FFFF00 | Fuchsia | = | #FF00FF | Aqua | = | #00FFFF |

# Struktur von Webseiten



[https://wiki.selfhtml.org/wiki/HTML/Tutorials/HTML5/Seitenstrukturierung#article\\_oder\\_section.3F](https://wiki.selfhtml.org/wiki/HTML/Tutorials/HTML5/Seitenstrukturierung#article_oder_section.3F)

# Struktur von Webseiten: Elemente (1)

## **<header>-Element:**

- Enthält sichtbaren Kopfbereich einer Webseite.
- Gruppierung einleitender Inhalte (Firmenlogos, ein Motto, Links zum Impressum oder zur Kontaktseite).

## **<footer>-Element:**

- enthält Informationen, die in Webseiten am Ende stehen: Autor, Hinweise zum Urheberrecht, ein Link zum Impressum.
- Position ist nicht notwendigerweise am unteren Rand
  - bei Blog-Einträgen steht der footer oft neben dem Text.

## Struktur von Webseiten: Elemente (2)

### **<article>-Element:**

- Stellt in sich geschlossene Abschnitte eines Dokuments dar
  - vergleichbar mit einem Zeitungsartikel.
  - innerhalb von article-Elementen weitere strukturierende Elemente wie header, section oder footer.

### **<section>-Element:**

- Enthält eine thematische Gruppierung von Inhalten typischerweise mit einer Überschrift.
- Dient dazu, den Inhalt oder auch einen article in semantische Abschnitte zu gliedern.

## Struktur von Webseiten: Elemente (3)

### **<nav>-Element:**

- Umschliesst insbesondere Navigationsleisten.
- Kann neben einer ungeordneten Liste mit den Verweisen auch eine Überschrift oder ähnliches enthalten.

### **<aside>-Element:**

- Umschliesst Abschnitte einer Seite, deren Inhalt nur in einem indirekten Zusammenhang mit dem umgebenden Inhalt stehen.
- Beispiele: Randbemerkungen, Fussnoten oder Links zu weitergehenden Webseiten.

# Struktur von Webseiten: Beispiele

```
<header>  
  <h1>HTML5 Seminar</h1>  
</header>
```

```
<footer>  
  <p>macro</p>  
  <p>CC 3.0 Lizenz</p>  
</footer>
```

```
<figure>  
  Diagramm, Abbildung  
</figure>
```

```
<aside>  
  <p>Weiterführende Infos</p>  
</aside>
```

```
<nav>  
  <ul>Hauptnavigation  
    <li>Startseite</li>  
  </ul>  
</nav>
```

```
<figcaption>  
  Bildunterschrift  
</figcaption>
```

Achtung: Unvollständige Liste!



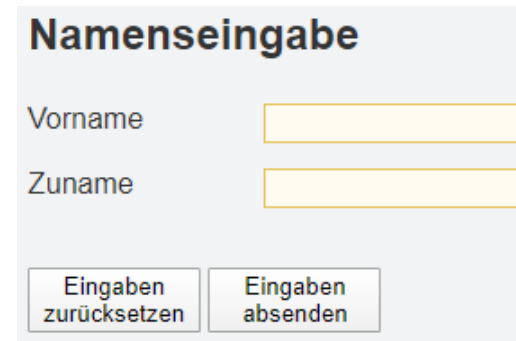
# Formulare

Formulare ermöglichen es Benutzern zwischen Optionen zu wählen oder Daten wie Adressen, Termine und Texte einzugeben und diese zur Weiterverarbeitung abzusenden.

## Formularelemente:

- werden mit dem `<form>`-Tag erzeugt.
- bestehen aus Formularelementen (z.B. Eingabefelder, Buttons, etc.).

## Beispiel:



The example shows a form titled "Namenseingabe" with two input fields: "Vorname" and "Zuname". Below the fields are two buttons: "Eingaben zurücksetzen" and "Eingaben absenden".

(HTML-Code nächste Seite)

# Formulare: Integration in die Webapplikation

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

Attribut "action": erhält als Wert eine URL (Adresse), an welche das Formular gesendet wird

```
<h2>Namenseingabe</h2>
```

```
<label for="vorname">Vorname</label>
```

Attribut "for": verknüpft das <label>-Element mit <input>-Feld

```
<input type="text" id="vorname" name="vorname">
```

Attribut "name": legt Namen fest (verwendet von auswertenden Programmen)

```
<label for="zuname">Zuname</label>
```

```
<input type="text" id="zuname" name="zuname">
```

Attribut "id" wird für Verknüpfung mit dem Label benötigt

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

```
</form>
```

Button: Formular an URL (action) senden

# Formularelemente: Radiobutton

```
<form action="..." >
  <h2>Namenseingabe</h2>
  <label for="male" >Mann</label>

  <input type="radio" name="gender" id="male" >

  <label for="female">Frau</label>
  <input type="radio" name="gender" id="female">

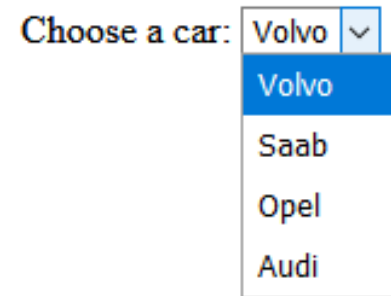
  <button type="submit">Eingaben absenden</button>
</form>
```

Attribut "name": wird u.A. verwendet  
um festzulegen welche Radiobuttons  
zur selben Gruppe gehören

# Formularelemente: Dropdownlisten und Platzhalter

- Dropdown-Listen (<select>): Verwendung analog zu Inputelementen (insbesondere das name-Attribut):

```
<label for="cars">Choose a car:</label>
<select name="cars">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="mercedes">Mercedes</option>
  <option value="audi">Audi</option>
</select>
```



Quelle:

[https://www.w3schools.com/tags/tag\\_select.asp](https://www.w3schools.com/tags/tag_select.asp)

- Placeholder: Platzhalter für Eingabevorschläge (kein Ersatz für das Label!):

```
<input type="text" id="i" name="i" placeholder="beautiful">
```

# Formularelemente: Validierung mittels HTML

Clientseitige Eingabevalidierungen können direkt mittels HTML-Attributen vorgenommen werden (Absenden nicht möglich, falls Validation fehlgeschlagen).

## Beispiele:

- required-Attribut: Input **muss** angegeben werden:

```
<input type="text" id="name" name="name" required>
```

- type="email": Erwartet eine korrekte **Emailadresse** (nur Formatprüfung):

```
<input type="email" id="mail" name="mail">
```

- min und max: Erwartet eine **Nummer X im Bereich** ( $\text{min} \leq X \leq \text{max}$ )

```
<input type="number" id="e" name="e" min="5" max="20">
```

# Videoelemente



- Einbindung von Videos in HTML-Dokumente:

```
<video src="movie.mp4" controls width="360" height="240"  
poster="startbild.jpg"> </video>
```

- Mehrere Formate möglich: Browser zeigt das erste Video, dessen Format er unterstützt:

```
<video controls width="360" height="240" poster="startbild.jpg">  
  <source src="movie.ogv" type="video/ogg">  
  <source src="movie.mp4" type="video/mp4">  
</video>
```

- Boolesche-Attribute (HTML-Attribute ohne Wert)
  - autoplay: Wiedergabe startet automatisch (sparsam einsetzen!).
  - controls: Videosteuerung wird eingeblendet (Start/Stop/Pause/etc.).

# Zusammenfassung

- HTML ist eine Sprache zur Strukturierung und Auszeichnung von Informationen eines Dokuments.
- Der HTML-Standard spezifiziert Tags und Attribute zur Auszeichnung dieser Informationen.
- Diese Standardisierung ermöglicht einem Programm (z.B. ein Webbrowser) die Inhalte eines HTML-Dokuments auszuwerten und zu verarbeiten (z.B. darstellen).
- Eines der wichtigsten HTML-Elemente ist der Hyperlink, also der Verweis zu weiterführenden Informationen.
- Formulare bieten eine erste Form der Interaktivität: Sie erlauben einem Benutzer eines HTML-Dokuments, Informationen an einen Webserver zu schicken.