



WESTFÄLISCHE
WILHELMS-UNIVERSITÄT
MÜNSTER

Erstellen dynamischer Webseiten mit PHP

WS 2016/2017



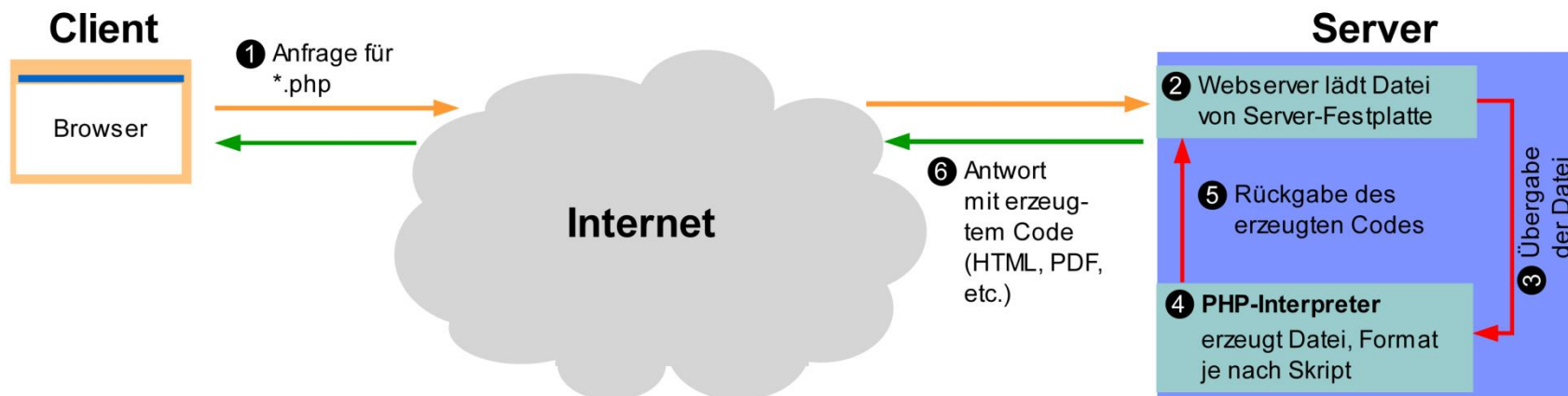
wissen.leben
WWU Münster

Dozenten: Patrick Förster, Michael Hasseler

Übersicht der heutigen Inhalte

- Formulare
 - Ablauf und Funktion
 - Übertragung: POST vs. GET
 - Felderarten
 - Validierung

Wiederholung: Ablauf beim Aufruf dynamischer PHP-Webseiten 1/2



vgl. <http://de.wikipedia.org/wiki/Php> ; Urheber: MovGP0; modifizierte Version

Wiederholung: Ablauf beim Aufruf dynamischer PHP-Webseiten 2/2

1. Im Browser wird die Adresse einer php-Datei eingegeben.
→ Browser fordert die Seite als **Request** beim Server an.
2. Webserver lädt die php-Datei von der Server-Festplatte.
3. Datei hat die Endung *.php
→ Der Webserver übergibt die php-Datei an den PHP-Interpreter.
4. PHP-Interpreter erkennt php-Befehle und erzeugt z.B. eine HTML-Seite.
5. Der Code wird vom PHP-Interpreter an den Webserver zurückgegeben.
6. Webserver schickt **Response** an den Browser

vgl. <http://de.wikipedia.org/wiki/Php> ; Urheber: MovGP0; modifizierte Version

Formulare

- „Formulare sind die Schnittstelle zwischen Benutzer und PHP.“ [s. PHP Grundlagen – Erstellung dynamischer Webseiten, RRZN, 9.Aufl., Seite 66]
 1. Daten werden in die Eingabefelder des Formulars eingegeben
 2. Eingabedaten werden mit http per POST oder GET-Methode an den Server geschickt
 3. PHP-Skript empfängt serverseitig die Eingabedaten und nimmt die Verarbeitung vor
- Formular beginnt mit dem html-Tag **<form>** und endet mit **</form>**
 - Attribut **method** legt die Verarbeitungsmethode fest: post und get
 - Attribut **action** legt Zieladresse fest, welches für die Verarbeitung zuständig ist

Formulardaten per POST oder GET-Methode übertragen

- **GET-Methode:**

- Eingabedaten hinter der Adresse mitgeschickt, und zwar hinter einem Fragezeichen in der Form **?name1=wert1&name2=wert2** usw. Dort können sie auch manipuliert und wieder abgeschickt werden.
- Datenmenge auf etwa 2 KB begrenzt
- input-Werte stehen im superglobalen Array **\$_GET** zur Verfügung

- **POST-Methode:**

- Adresszeile bleibt unverändert, Daten werden ähnlich wie eine Mail verpackt und abgeschickt. Auf diese Weise ist die Größe der Daten nicht beschränkt.
- input-Werte stehen im superglobalen Array **\$_POST** zur Verfügung.

Vgl. PHP Grundlagen – Erstellung dynamischer Webseiten, RRZN, 9.Aufl., Seite 66/67

Verwendungszweck von POST und GET

- GET-Methode wird bei Suchbegriffen oder der Navigation auf Webseiten verwendet.
- Bei der POST-Methode sind die Formulardaten nicht im Verlauf des Browsers oder in der Protokolldatei des Servers sichtbar.
- POST-Methode wird generell empfohlen, da es die sicherere Variante ist. Manipulation der Daten ist nicht so leicht möglich wie bei der GET-Methode. Für die Sicherheit ist hierbei auch die Verwendung von https anstatt von http wichtig.

Vgl. PHP Grundlagen – Erstellung dynamischer Webseiten, RRZN, 9.Aufl., Seite 66/67

Superglobale Variablen in PHP

- superglobale Variablen/Arrays sind von PHP selbst definiert
- von überall im Programm aufrufbar
- Syntax `$_GROSSBUCHSTABEN` z. B. `$_POST` oder `$_GET`
- Können nicht vom Nutzer selbst angelegt werden

Vgl. PHP Grundlagen – Erstellung dynamischer Webseiten, RRZN, 9.Aufl., Seite 86

Eingabefelder

- **input**-Elemente:
 - **text** (Eingabezeile)
 - **password** (Eingabezeile, nicht lesbar)
 - **number** (numerische Eingabe)
 - **date** (Eingabe einer Datumseingabe)
 - **checkbox** (Schalter)
 - **radio** (Umschalter)
 - **submit** (Knopf zum Start der Übertragung)
 - **reset** (Knopf, setzt „offline“ alle **input**-Elemente auf Anfangswert)
 - **file** (Dateiauswahl zwecks Hochladens)
 - **hidden** (zur Aufbewahrung von Werten)
 - **button** (für JavaScript etc.)
- **select**-Element (Listbox oder Klappliste)
- **textarea**-Element (einfacher Editor)

Textfelder

- Eingabe von Text in einem Formular
- Syntax: **<input** name="bezeichner">
 - Für bezeichner ist ein eindeutiger Name zu vergeben
 - Beschriftung im html-Dokument vor dem Tag input
 - Bei input erfolgt kein automatischer Umbruch
 - Attribut **size** legt die Größe des Textfeldes fest
 - Attribut **maxlength** legt die maximal eingebbaren Zeichen fest
 - Attribut **type** legt die Feldart fest: **text** ist default, **password** für Sternchenanzeige, **int** für numerische Werte und **date** für eine Datumsangabe



Vorname:

Nachname:

Vgl. HTML Das Einsteigerseminar, Thomas Kobert, 2012 Seite 149-154

Schaltflächen

- Absenden der Formulardaten erfolgt mit Schaltflächen
- Syntax: `<input type="submit" value="anzeigertext">`
 - Attribut **type** legt die Art der Schaltfläche fest:
 - **submit** schickt die Formulardaten an den Server mit der Zieladresse, die in action angegeben wird
 - **reset** löscht die Eingabedaten und setzt das Formular zurück in den Ursprungszustand
 - Attribut **value** dient zur Angabe der Beschriftung der Schaltfläche, wobei sich die Breite automatisch anpasst
- Seit HTML 4 existiert ein tag `<button>`, das von den Browsern aber noch nicht uneingeschränkt zur Verfügung gestellt wird



Vgl. HTML Das Einsteigerseminar, Thomas Kobert, 2012 Seite 154-158

Kontrollkästchen

- Kontrollkästchen (Checkbuttons) werden in der Oberfläche als kleine Kästchen dargestellt
- Kästchen können angeklickt werden und mit einem Häkchen versehen werden, dann ist das Auswahl aktiviert
- Syntax: `<input type="checkbox" name="bezeichnung" value="wert">`
 - Attribut **type** auf checkbox setzen
 - Attribut **name** ist die Bezeichnung des Feldes
 - Attribut **value** ist der Wert, den man serverseitig sieht, falls das Häkchen gesetzt ist
- Mehrfachauswahl ist möglich, dann Attribut **name** als Array angeben `name="bezeichnung[]"`

Extras:

☒ 24 Stunden Express ☐ Versicherung

Vgl. HTML Das Einsteigerseminar, Thomas Kobert, 2012 Seite 158-159

Optionsfelder

- Optionsfelder (Radiobuttons) werden in der Oberfläche als kleine Kreise dargestellt

- Kreise können angeklickt werden und

mit einem Punkt versehen werden, dann ist das Auswahl aktiviert

- Syntax: **<input type="radio" name="bezeichnung" value="wert">**

- Attribut **type** auf radio setzen

- Attribut **name** ist die Bezeichnung des Feldes

- Attribut **value** ist der Wert, den man serverseitig sieht, falls der Button gesetzt ist

- Mehrfachauswahl in einer Gruppe (gleicher Name) ist nicht möglich

- Beschriftung erfolgt im HTML-Dokument selbst



Ort:
☒ Inland ☐ EU ☐ weltweit

Vgl. HTML Das Einsteigerseminar, Thomas Kobert, 2012 Seite 158-159

Listenfelder 1/2

- Listenfelder beginnen mit dem tag **<select>** und enden dem tag **</select>**
 - Attribut **name** ist ein Pflichtangabe und dient serverseitig für den Zugriff auf die Formularliste
 - Attribut **size** legt fest, wie viele Einträge zu sehen sind, falls die Liste nicht aufgeklappt ist
 - Attribut **multiple** gibt einem die Möglichkeit mehrere Einträge auszuwählen
- Innerhalb des select-tags kann ein Listeneintrag mit dem tag **<option>** ... **</option>** zu der Auswahlliste hinzugefügt werden.
- Beim Anklicken eines Listenfeldes werde die auswählbaren Optionen aufgelistet.

Vgl. HTML Das Einsteigerseminar, Thomas Kobert, 2012 Seite 162-165

Listenfelder 2/2

- Zugeklappt:



Gewichtsklasse:
<500 g ▼

- Aufgeklappt:



Gewichtsklasse:
<500 g ▼
<500 g
0,5-2 kg
2-5 kg
5-10 kg
10-20 kg

Mehrzeiliges Texteingabefeld

- Eingabe von mehrzeiligen Text in einem Formular
- Syntax: `<textarea name="bezeichner"> </textarea>`
 - Für bezeichner ist ein eindeutiger Name zu vergeben
 - Beschriftung im html-Dokument vor dem Tag textarea
 - Attribut **cols** legt Anzahl der Zeichen für die Breite fest
 - Attribut **rows** legt Anzahl der sichtbaren Zeilen fest
- Falls der sichtbare Bereich der Texteingabe nicht ausreichen sollte, dann werden automatisch Scroll-Balken eingeblendet

Vgl. HTML Das Einsteigerseminar, Thomas Kobert, 2012 Seite 160-162

Validierung von Formularinhalten in PHP

- Funktion **isset(...)** prüft, ob ein Parameter übermittelt wurde
- Einsatz: Wenn mehrere Schaltflächen möglich sind, kann so geprüft werden, welche ausgewählt wurde
- Beispiel: `if (isset($_POST['addieren']))`
- Funktion **empty(...)** prüft, ob ein Formularfeld ein Wert enthalten hat
Beispiel: `if (empty($_POST['vorname']))`
- Funktion **is_numeric(...)** prüft, ob eine Variable numerisch ist und liefert ein Boolean zurück
Beispiel: `if (is_numeric($zahl)) { ... }`

Beispiel: Berechnung der Versandkosten eines Pakets (b03.html)

- Auf der Seite b03.html soll man die Gewichtsklasse eines Paketes, den Empfängerort und die Extras 24 Stunden Express und Versicherung auswählen können. Die Berechnung und Ausgabe der Gesamtkosten des Versands erfolgt mit dem Aufruf des PHP-Skripts b03.php.
- Gewichtsklassen:
 - < 500g: 4 Euro
 - 0,5-2kg: 6 Euro
 - 2-5kg: 8 Euro
 - 5-10kg: 12 Euro
 - 10-20kg: 18 Euro
- Der Versand innerhalb der EU hat einen Aufschlag von 20% und weltweit von 100%.
- Bei einem 24 Stunden Express entstehen Zusatzkosten von 5 Euro und bei der Versicherung auch von 5 Euro.

Zusammenfassung

- Wie funktioniert ein Formular?
- Wofür und wie werden Textfelder, Schaltflächen, Kontrollkästchen (checkboxes), Optionsfelder (radiobuttons) und Listenfelder in einem Formular verwendet?
- Wie funktioniert die Übergabe der Werte an das PHP-Skript?
- Wann sollte man POST und wann GET verwenden?
- Wie kann man Formularinhalte in PHP validieren?

Aufgaben

1. Erstellen Sie eine Datei a01.html, die ein Textfeld Benutzername und ein Textfeld mit type="password" für das Passwort hat. Die Verarbeitung der Eingabe soll von der Datei a01.php vorgenommen werden. Bei der Kombination admin und geheim soll der Text: "Sie sind angemeldet als Administrator." erscheinen, ansonsten ein Hinweis auf ungültige Anmeldedaten. Die Übertragung soll mit der GET-Variante erfolgen. Manipulieren Sie die Werte in der URL bei Ihren Tests.
2. Analog zu Aufgabe 1 soll nun die Übertragung mit der POST-Variante erfolgen. Speichern Sie die Dateinamen mit dem Präfix a02... . Vergleichen Sie die Unterschiede zu Aufgabe 1).
3. Bauen Sie die html-Seite a03.html mit folgenden Lieblingsrestaurants als Kontrollkästchen: chinesisch, deutsch, griechisch, italienisch, japanisch und mexikanisch. Die Verarbeitungsseite a03.php soll die Auswahl der Lieblingsrestaurants als Satz ausgeben. Eine Mehrfachauswahl soll möglich sein! Für den Fall, dass keine Auswahl erfolgt, geben sie hierzu eine Meldung aus.

Anhang: ZIVinteraktiv-Abstimmung

- QR-Code zur anonymen Abstimmung:



- <https://www.uni-muenster.de/ZIV/anw/zivinteraktiv/abstimmung.php?code=5804a66a>