

Kommunikationstechnik

12. Praktikum

Thema: Case sensitive Bereiche Maps

Florian Stocker

18.01.2023 KW3

Lehrer: Herrn Maulhardt

E-Mail: maulhardt@nta-isny.de

NTA Isny



Zusammenfassung der Aufgabe:

Gestalten Sie mehrere anklickbare Bereiche innerhalb einer Graphik

- Orange mittels Kreisform (z.B.5 Ecken)
- Weintauben mittels Polygon
- Ananas mittels Rechteck



Inhaltsverzeichnis

1. Aufgabe.....	Fehler! Textmarke nicht definiert.
2. Aufgabe.....	Fehler! Textmarke nicht definiert.
3. Aufgabe.....	Fehler! Textmarke nicht definiert.
4. Aufgabe.....	Fehler! Textmarke nicht definiert.
5. Aufgabe.....	Fehler! Textmarke nicht definiert.
6. Aufgabe.....	Fehler! Textmarke nicht definiert.
7. Quellenangabe.....	12



1. Definition

- Wird durch das Tag `<form>` eingeleitet
- Dann folgt die Beschreibung der einzelnen Steuerelemente, bevor die Definition mit `</form>` abgeschlossen wird
- Parameter für FORM = ACTION & METHOD
- METHOD bestimmt, was mit dem Inhalt des Formulars geschieht, wenn der Anwender es abgeschickt hat
- POST = als Email verschickt
- GET = an einen HTML-Server weiterreichen, wo sie in der Umgebungsvariable QUERY_STRING abgelegt wird
- ACTION = gibt an, wo genau die Eingaben des Gastes landen sollen

→ die Adresse muss die Dienste Kennung(HTTP://), den Namen des Servers(search.yahoo.de) und das Unterverzeichnis(/search/de) enthalten

2. Einzeilige Eingabefelder & Eingabebereiche innerhalb eines Formulars

- Wichtigste Element = das Eingabefeld
→ Wird mit `<input>` eingeleitet
- NAME legt eine Bezeichnung fest, die von den Webentwickler festgelegt wird(Bsp. Hobby)
- SIZE bestimmt die Breite des Eingabefeldes
- MAXLENGTH beschränkt die Maximallänge
- Das Format der Eingabe kann mit TYPE festgelegt werden



- Parameter für den Datentyp
- eines Eingabefeldes:

TYPE	BEDEUTUNG
PASSWORD	Die Eingabe des Benutzers erfolgt verdeckt, es werden nur Sternzeichen anstatt de eingegebenen Zeichen dargestellt
INT	Nur ganze Zahlen. Mit SIZE die Länge begrenzen. Durch MAX und MIN können die Grenzen für den Wertebereich vorgegeben werden.
FLOAT	Fließkommazahlen werden akzeptiert. Die Länge kann durch SIZE begrenzt werden.
DATE	Akzeptiert eine Kalenderdatum
URL	Zur Eingabe von Internet-Adressen
TEXT	Beliebige Kombinationen aus Buchstaben und/oder Ziffern bzw. Sonderzeichen

Ohne blinde Tabelle

Name

E-Mail

Notiz

Mit einer blinden Tabelle

Name

E-Mail

Notiz

3. Mehrzeilige Eingabefelder & Eingabebereich innerhalb eines Formulars

- Eingabebereiche werden mit dem Tag <textarea> definiert
- Parameter ROWS und COLS legen die Anzahl der Zeilen und Spalten fest
- Ein Name muss vergeben werden, bevor mit </textarea> die Definition abgeschlossen wird

Readonly:

Manchmal sollen Felder nicht verändert werden, wenn z.B. eine automatische Kundennummer angezeigt wird. Dafür können Felder den Zusatz <readonly> erhalten Dann können die vordefinierten Werte nicht verändert werden. Allerdings



zeigen die Browser diesen Nur-Lese-Status nicht gesondert an und die Felder sehen aus wie sonst auch. Das kann irritierend sein.

Bsp.:

```
<p>Kunden-Nummer: <input name="kdnr" type="text" size="9" maxlength="9" value="33015" readonly> </p>
```

Größe:

Die Größe kann man mit den Attributen COL und ROW angegeben werden.

Bsp.:

```
<textarea col="20" row="5"> </textarea>
```

4. Auswahllisten mit festen Einträgen

- Bieten die Möglichkeit zwischen verschiedenen vorgegebenen Antworten auszuwählen
- Wird mit dem Tag `<select>` erzeugt
- Mit dem MULTIPLE im SELECT-Tag kann auch eine Mehrfachauswahl zugelassen werden
- Vorschläge werden mit dem `<option>` Tag hinzugefügt

Beispiel:

```
58 <h2>Auswahlliste</h2>
59 <p><select name="klassen"></p>
60   <option>1AK</option>
61   <option>2AK</option>
62   <option>3AK</option>
63   <option>4AK</option>
64   <option>5AK</option>
65 </select>
66
```

→ Ergebnis





5. Weitere HTML-Themen

Absendwerte von Einträgen bestimmen(values):

Value weist einem input-Feld einen Anfangswert zu, z.B. einen Text oder eine Zahl. NAME und VALUE bilden die Name-/Wert-Paare, die das Formular an die verarbeitende Anwendung überträgt.

Bsp.:

```
<input type="text" name="mytext" value="">
```

Verschachtelte Auswahllisten(optgroup label):

Das HTML Tag <optgroup> enthält in einer Gruppe kombinierte Tags <option>. (Das Tag <option> bestimmt die Punkte in einer Pop-up-Liste, die ihrerseits durch das Tag <select> bestimmt wird).

Der Inhalt des Tages <optgroup> sieht wie eine fettgedruckte Überschrift aus. Die im Tag <option> eingeschlossene Elemente werden sich nach rechts verschoben. Das Tag <option> wird in der Regel mit dem Attribut value verwendet und bezeichnet, welcher Wert an den Server für weitere Verarbeitung gesendet wird.

Bsp.:

```
<select>
  <optgroup label="Bücher">
    <option value="html">HTML</option>
    <option value="css">CSS</option>
```



</optgroup>

Buttons(Radiobutton,Checkbox, Klick-Button):

Radiobutton dient für die Auswahl zwischen verschiedenen Optionen. Die Antwort-Möglichkeiten sind dabei vorgegeben, z.B. Schulnoten von 1-5 oder ja, Nein oder Weiß nicht.

Alle Attribute dieser Auswahlliste müssen denselben Namen haben. Nur so werden sie als zusammengehörige Auswahlliste erkannt. Weicht der Name bei einem Element ab, funktioniert die Auswahl nicht.

Dem <input>-Befehl muss der Typ <radio> zugewiesen werden.

Damit der Webserver und CGIs die Angaben richtig verarbeiten können, muss man noch das Attribut

<value> mit dem entsprechenden Wert anhängen.

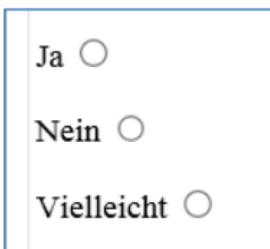
Bsp.:

<p> Ja <input type="radio" name="antwort" value="ja"></p>

<p> Nein <input type="radio" name="antwort" value="nein"></p>

<p> Vielleicht <input type="radio" name="antwort" value="vielleicht"></p>

Ergebnis:

A screenshot of a web browser displaying a form with three radio buttons. The first button is labeled 'Ja', the second 'Nein', and the third 'Vielleicht'. All three buttons are currently unselected.

Checkbox wird verwendet um eine mehrfache Auswahl zu ermöglichen. Dafür gibt man im Input-Feld den Type <checkbox> an. Wie beim Radio-Button ist es hier wichtig, dass alle Checkboxes im Attribut <name> denselben Namen besitzen. So werden sie als eine zusammengehörige Gruppe erkannt und verarbeitet. Der Besucher kann so viele Boxen anklicken wie er möchte.

Checkboxes, auch wenn sie den selben Gruppennamen haben, werden NICHT zu Gruppen zusammengefasst. Klickt man eine an, ändert das nichts am Zustand der andern, selbst wenn sie denselben Namen haben. Stattdessen schaltet man sie mit



einem Klick an, mit einem weiteren Klick wieder aus. In HTML ist der einzige Unterschied, dass der Typ nicht `<radio>` sondern `<checkbox>` lautet.

Bsp.:

```
<p>Checkbox: Wähle deine Hobbys: </p>
<p><input type="checkbox" name="hobbys" value="lesen">Lesen</p>
<p><input type="checkbox" name="hobbys" value="tennis">Tennis</p>
<p><input type="checkbox" name="hobbys" value="fuß">Fußball</p>
<p><input type="checkbox" name="hobbys" value="schwim">Schwimmen</p>
```

Ergebnis:

Checkbox: Wähle deine Hobbys:

- ☐ Lesen
- ☐ Tennis
- ☐ Fußball
- ☐ Schwimmen



Klick-Buttons leiten Sie mit `<button>` ein. Dieses Tag hat ein Abschluß-Tag `</button>`, mit dem Sie die Definition des Buttons am Ende abschließen. Alles, was Sie innerhalb von `<button>...</button>` notieren, wird als "Beschriftung" des Buttons angezeigt. Diese Inhalte werden innerhalb der Button-Fläche zentriert ausgerichtet.

Im einleitenden `<button>`-Tag notieren Sie verschiedene Angaben zum Button. Etwas komisch erscheint die Angabe `type=button`, wo doch das Tag schon so heißt. Notieren Sie diese Angabe jedoch bei allen Buttons, die Sie für Scriptsprachen verwenden. Denn mit Hilfe des `<button>`-Tags können Sie auch zwei andere Button-Typen definieren, nämlich Buttons zum Absenden und Abbrechen.

Mit dem Attribut `name=` können Sie einen Namen für den Button vergeben. Mit dem Attribut `value=` können Sie eine Beschriftung für den Button bestimmen, falls Sie keinen Inhalt innerhalb von `<button>...</button>` notieren.

Um anzugeben, was passieren soll, wenn der Button angeklickt wird, können Sie beispielsweise den JavaScript Event-Handler `onClick=` verwenden. Hinter dem Istgleichzeiten geben Sie einen JavaScript-Befehl ein, z.B. den Aufruf einer selbstgeschriebenen JavaScript-Funktion, oder - wie im Beispiel - einen einfachen JavaScript-Befehl.

Bsp.:

`<p>Wenn bei Ihnen JavaScript funktioniert, hat der folgende Button die gleiche Bedeutung wie der Back-Button im WWW-Browser:</p>`

`<input type=button value="Zurück" onClick="history.back()">`

Dateiupload-Felder(`input type="file"`):

Mithilfe des Formularelements **`input type="file"`** kann der Anwender eine Datei von seinem lokalen Rechner zusammen mit dem Formular übertragen (Upload). Durch das Attribut `type="file"` erstellt der Browser automatisch ein Eingabefeld für den Pfad zur Datei und einen Button mit dessen Hilfe eine lokale Datei ausgewählt werden kann. Die Größe des Eingabefeldes kann wie üblich mit dem `size`-Attribut festgelegt werden.

Dieses Element funktioniert nur mit der POST-Methode und nicht mit GET. Außerdem muss im Formular das `enctype="multipart/form-data"`-Attribut notiert werden, da sonst nur der Name und nicht die Datei selbst übertragen wird.



Bsp.:

```

<form>
  <label for="vname">Vorname:
    <input id="vname" name="vname">
  </label>
  <label for="zname">Zuname:
    <input id="zname" name="zname">
  </label>

  <label for="männl">männlich</label>
    <input type="radio" id="männl" name="geschlecht" value="0">
  <label for="weibl">weiblich</label>
    <input type="radio" id="weibl" name="geschlecht" value="1">
  <label for="queer">queer</label>
    <input type="radio" id="queer" name="geschlecht" value="2">

  <label for="alter">über 18:</label>
    <input type="checkbox" id="alter" name="alter">

  <input type="submit" value="senden"><nowiki>
</form>

```

Absenden/Abbrechen-Buttons:

Zwei Standard-

Buttons stellt HTML zur Verfügung, um Formulareingaben zu handhaben: einen Button zum "Absenden" und einen zum "Abbrechen". Mit dem **Absende-Button** kann der Anwender das ausgefüllte Formular losschicken. Mit den Formulardaten geschieht dann das, was im einleitenden <form>-Tag bei action= festgelegt wurde (siehe hierzu Formular definieren). Mit dem **Abbrechen-Button** kann der Anwender alle Eingaben verwerfen. Das Formular wird nicht abgeschickt.

Bsp.:

```

<input type=submit value="Absenden">
<input type=reset value="Abbrechen">

```

Mit <input type=submit ...> definieren Sie einen Absende-Button (input = Eingabe, submit = bestätigen). Beim Anklicken dieses Buttons werden die Formulardaten abgeschickt. Mit <input type=reset...> definieren Sie einen Abbrechen-Button (reset = zurücksetzen). Eingegebene Daten werden verworfen und nicht abgeschickt. Mit dem Attribut value= bestimmen Sie die Beschriftung der Buttons (value = Wert). Die Angabe muß in Anführungszeichen stehen.



6. Wichtige Begrifflichkeiten

CGI = Common Gateway Interface:

Das Common Gateway Interface (CGI) ist eine Schnittstelle von Webservern, über die standardisierter Datenaustausch zwischen externen Anwendungen und Servern möglich ist. Mit CGI müssen HTML-Seiten nicht vollständig auf dem Server bereitstehen, sondern werden dynamisch erzeugt, sobald ein Nutzer eine passende Anfrage über die Website stellt.

CSS-Stylesheets:

Cascading Style Sheets (CSS) ist eine Programmiersprache, die es Ihnen ermöglicht, das Design von elektronischen Dokumenten zu bestimmen. Anhand einfacher Anweisungen – dargestellt in übersichtlichen Quellcodes – lassen sich so Webseiten-Elemente wie Layout, Farbe und Typografie nach Belieben anpassen. Dank der gestuften Gestaltungsbögen bleiben die semantische Struktur und der Inhalt des Dokuments vollkommen unberührt.

XML:

XML (Extensible Markup Language) ist abgeleitet vom älteren Standard SGML, ein textbasiertes Format und für den Austausch strukturierter Informationen gedacht. Dazu gehören Konfigurationen, Dokumente, usw. Grundsätzlich kann XML zum Beschreiben, Speichern und Austauschen von Daten genutzt werden.

XHTML:

Die Abkürzung XHTML steht für Extensible Hyper Text Markup Language und ist sozusagen eine Parallelentwicklung zu HTML. XHTML bildet ein Zwischenstück zu XML und HTML, nutzt die typische, semantische Syntax von XML, ist aber weniger auf die Auszeichnung hierarchischer Strukturen ausgelegt, sondern mehr auf die Definition und Strukturierung von Dokumenten und Inhalten.

JavaScript und DOM:



Das DOM beschreibt HTML-, XML- und SVG-Dokumente und ist die Schnittstelle für den Zugriff auf HTML-Elemente, Attribute und Inhalte von HTML-Seiten, XML-Dokumenten und SVG-Grafiken.

Das DOM verbindet Javascript mit den HTML-Elementen und erzeugt eine Baumstruktur, in der jedes HTML-Element eindeutig erreicht wird.

Dynamisches HTML:

Dynamisches HTML (dynamic HTML, DHTML) ist ein Sammelbegriff für eine Kombination von HTML-Tags und -Optionen, mit denen Webseiten lebendiger und interaktiver gestaltet werden können als mit früheren Versionen von HTML.

Zu den einfachen Beispielen für dynamische HTML-Funktionen gehört, dass sich die Farbe einer Textüberschrift ändert, wenn ein Benutzer mit der Maus darüberfährt, oder dass ein Benutzer ein Bild per Drag and Drop an eine andere Stelle auf einer Webseite ziehen kann.

Perl:

Bei **PERL** handelt es sich um eine freie und interpretierte Programmiersprache bzw. Skriptsprache, die plattformunabhängig ist und mehrere Programmierparadigmen unterstützt.

Zu den Vorzügen von PERL zählen neben einer schnellen Problemlösung auch die Möglichkeit der Textbearbeitung mittels regulärer Ausdrücke sowie eine Vielzahl freier verfügbarer Module.

PHP:

Es handelt sich bei **PHP** um eine Skriptsprache, die in einigen Details mit Programmiersprachen wie C oder Perl vergleichbar ist. Der wichtigste Einsatzbereich von PHP ist das Erstellen dynamischer Webinhalte. Ursprünglich war PHP die Abkürzung für „Personal Home Page Tools“. Heute ist PHP ein rekursives Akronym für „Hypertext Preprocessor“.

Java:

Java ist eine besondere Programmiersprache, die sich an der Syntax der Programmiersprachen C und C++ anlehnt. Bei der Entwicklung von Java wurde von Beginn an genauestens darauf geachtet, dass bestimmte schwere Programmierfehler erst gar nicht auftreten können.



7. Quellenangabe

