

Tough validation



CSc 337 HTML Forms

Benjamin Dicken

Announcements

- PA 3 “required” elements
- PA 4

GETting vs POSTing

- Previously only wrote pages that display information, not much interaction other than `<a>` and resizing
- HTML forms allow us to get information from the user, receive it on our server

HTML Forms

- An HTML form consists of a `<form>` element with a number of `<label>` and `<input>` fields within
- The labels and inputs should be paired up 1 to 1

Full Name

Password

Age

Email Address

How tall are you? Tall About average Short

Favorite Color!

Form Input types

- [Many, Many types](#)
- A few covered in class, but might be asked to use others in assignments and such

The image shows a form with several input fields and a submit button, each with an arrow pointing to a label on the right. The labels are: text, password, number, email, radio, color, and submit.

Full Name ← **text**

Password ← **password**

Age ← **number**

Email Address ← **email**

How tall are you? Tall About average Short ← **radio**

Favorite Color! ← **color**

← **submit**

Form Element Attributes

- Every `<form>` element should have an **action** attribute
(can also specify `method=".."`)
- Every `<input>` element should have a **type**, **id**, and **name** attribute
- Every `<label>` element should have a **for** attribute, which should match the **id** of the corresponding input

Submitting via Get vs POST

POST URL:

`https://benjdd.com/sp`

POST request:

`POST / HTTP/1.1`

`Host: localhost`

`Content-Type: application/x-www-form-urlencoded`

`Content-Length: 13`

`name=ben&a=hi`

GET URL:

`https://benjdd.com/sp?name=ben&a=hi`

GET request:

`GET / HTTP/1.1`

`Host: localhost`

`Content-Length: 0`

Built It!

