

## DIG3716c: Internet Interaction

# Lab Assignment 4 Specifications (Validated Form Scripting)

---

*The assignment is due 12/3/2010 by 4:20pm.*

### Overview

Build a survey form that offers up hints, uses a variety of input types, and performs basic validation before delivering the results of the survey to the page.

### Specifications

#### Implementation

##### General

- Code should be indented to show tag parent/child relationships.
- The root directory for this assignment should be ~/public\_html/dig3716c/assignment04
- You should have a single XHTML page called survey.html and it should be located in the root directory for this assignment.
- The image directory for this assignment should be ~/public\_html/dig3716c/assignment04/img
- The <title> of the page should be "Survey – your\_first\_name your\_last\_name", for example "Survey – Dan Novatnak".
- Your page should contain proper semantic formatting as discussed in previous lectures/labs.
- All directories and filenames should contain no spaces or uppercase letters.
- Any and all links need to be functional.

##### XHTML

- Web page validates as XHTML Transitional (<http://validator.w3.org>)

##### CSS

- The CSS directory for this assignment should be ~/public\_html/dig3716c/assignment04/css
- You may use the 960gs Grid System for your layout of this assignment.
- All styles must be documented in an external CSS file called styles.css and linked to the document using the @import url rule.
  - All presentational HTML attributes should be replaced with CSS rules.
  - Use classes and/or id's where appropriate.

##### JavaScript

- The JavaScript directory for this assignment should be ~/public\_html/dig3716c/assignment04/js
- You may not use any JavaScript libraries or any piece of a JavaScript tutorial.
- You may not use the innerHTML property anywhere in this assignment.
- You may not use the XMLHttpRequest object or any variation of it.
- Your JavaScript must be documented in a single file called process\_form.js and be linked in the <head> of the page.
  - **Form Processing**
    - All user input must come from input fields in a single form tag.
    - All user submitted data must be stored and displayed to the page using DOM Core methods and properties after the quiz is submitted.
    - After the survey is submitted, the user must be shown their personal info, the answers they submitted, and a "badge" to place on their homepage. The caption of the badge will be an absolute URL of that badge image that the user can copy and paste to include in their webpage.

- **Anonymous (optional)**

- If the anonymous checkbox is checked, immediately after being checked the text fields should all be disabled. If the checkbox is then unchecked, the text fields should be enabled again.

- **Hints**

- When focused, text fields should offer a hint to the right of the text field that indicates how the user should input data.

The screenshot shows a 'Personal Info' form with a green header. The form contains five text input fields: 'First Name', 'Last Name', 'E-mail Address', 'Phone Number', and 'Sulley Address'. To the right of each field, a hint is displayed in a light blue box. The hints are: 'Please enter a valid e-mail address! (must contain @)' for E-mail Address, 'Please enter a valid phone number! (must contain -)' for Phone Number, and 'Please enter a valid Sulley address! (must contain ~)' for Sulley Address. There is also an 'Anonymous' checkbox at the bottom, which is checked. A 'HINTS' section is visible, stating 'Only shown when the field is in focus'.

- When unfocused, the validateData() function should be called.

- **Validation**

- First name check – something has been entered, the user has entered only letters
- Last name check – something has been entered, the user has entered only letters
- Phone check – contains numbers and in the pattern XXX-XXX-XXXX
- Email check – at least follows the pattern [blah@blah.com](mailto:blah@blah.com)
- Sulley address check – at least contains a tilde (~) (<http://www.dm.ucf.edu/~dnovatnak>) and begins with http://
- All validation should occur in a function called validateData()
  - If invalid – the hint area should update to tell the user to try again and show an iconic representation of the error.

The screenshot shows the same 'Personal Info' form, but now with validation errors. The 'First Name' and 'Last Name' fields have red error messages: 'You didn't enter a valid name!'. The 'Phone Number' field has a red error message: 'You didn't enter a valid phone number!'. The 'E-mail Address' and 'Sulley Address' fields are empty. The 'Anonymous' checkbox is checked. An 'ERRORS' section is visible, stating 'Only shown when validation fails'.

- If valid – the hint area should update to show an iconic representation of the success.

The image shows a web form titled 'Personal Info' with a green background. It contains several text input fields: 'First Name', 'Last Name', 'E-mail Address', 'Phone Number', and 'Sulley Address'. The 'E-mail Address' field has a green checkmark icon next to it. At the bottom, there is a checkbox labeled 'Anonymous' which is checked. A red box with the text 'SUCCESS' and 'Only shown when validation succeeds' is overlaid on the right side of the form.

## Presentation

### General - Now is your time to shine!

- Develop a new layout and website that meets the following criteria
  - Column, grid-based layout.
  - Contains at least header, sidebar, content, and footer sections using different layouts than assignments 2 and 3.
    - Header contains the name of the quiz.
    - Footer contains the text "Created by first\_name last\_name"..
  - Centered in the browser window.
  - Font-sizes and font-families should be consistent.
  - Your survey should be divided visually into at least 2 different styled fieldsets with legends.

The image shows the same 'Personal Info' form as above, but without the success message. It features a green background and several text input fields: 'First Name', 'Last Name', 'E-mail Address', 'Phone Number', and 'Sulley Address'. At the bottom, there is a checkbox labeled 'Anonymous' which is checked.

- Questions should have obvious visual dividers.

## Content

### General

- Develop a survey form that meets the following criteria:
  - **Text fields**
    - First name
    - Last name
    - Phone number
    - Email address
    - Sulley address

- **Survey questions**

- You will develop all of the questions for the survey, and while they have to make sense together, they do not have to be about any one topic.
- Use common sense in your selection of content. I always think "Can I show this to my grandmother?" as a yardstick.
- All survey questions must be radio buttons.
- Each survey question must have at least 2 options.
- Surveys must have at least 2 questions.

- **Badges**

- You must create at least 2 different "badge" images. You must build logic into your script so that the answers a user submits will determine the badge they are shown when they complete the quiz. (here's an example of a badge creation: [http://www.top-callingcards.com/blog\\_voice/music\\_type.php](http://www.top-callingcards.com/blog_voice/music_type.php))

## Grading Rubric:

### Assessment:

- /30 validateData() works according to spec
- /10 anonymous checkbox working according to spec
- /20 hints appear according to spec
- /40 Presentation and Content specs followed

### Deductions:

- 100 LATE SUBMISSIONS
- 40 SUBMISSION TO INCORRECT LOCATION
- 100 if any libraries(like jQuery, etc.), tutorial code, innerHTML and/or XMLHttpRequest are used
- 30 if the instructor feels that you didn't put significant effort into the assignment
- 30 if all requested content is not included
- 30 if all requested content is not included
- 10 fieldset and legend specs not followed
- 10 if your font-sizes and families are not consistent
- 30 if your survey questions are not included as requested
- 20 if badges are not displayed as requested
- 10 if footer content is not represented as requested
- 30 if any XHTML tables are used for formatting
- (-1 per error) Implementation General specs followed
- (-2 per error, -2 per warning) Implementation XHTML specs followed
- (-1 per error) Implementation CSS specs followed

/100 TOTAL