Creating HTML5 Content

Overview

In this lab you will start using HTML5 features in your Web pages, including semantic tags and new form input controls. You'll also get another chance to use the HTML5 selector API, which we mentioned earlier in the course.

Roadmap

There are 4 exercises in this lab, of which the last two exercises are "if time permits". Here is a brief summary of the tasks you will perform in each exercise; more detailed instructions follow later:

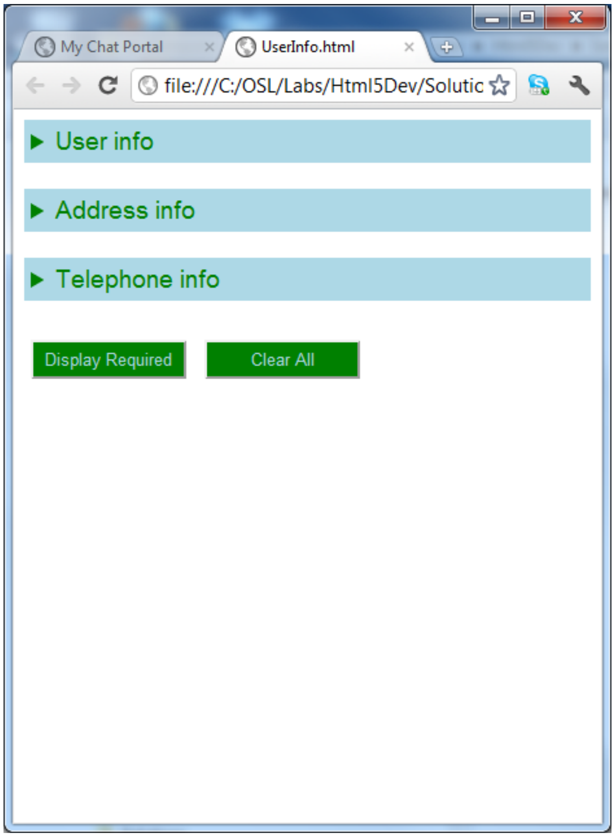
1. Using semantic tags
2. Using the selector API
3. Updating the DOM (if time permits)
4. Additional suggestions (if time permits)

Exercise 1 – Using semantic tags

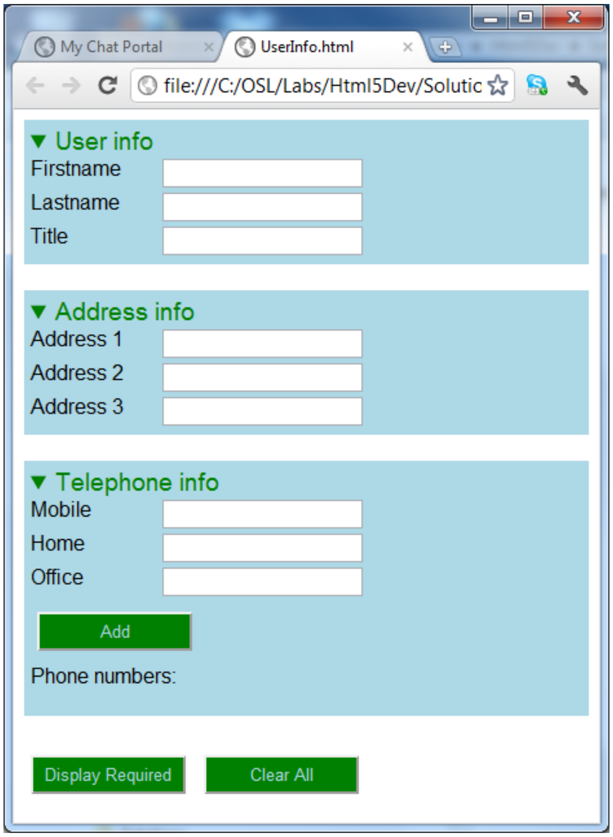
In the *Start* folder, open `UserInfo.html` in an editor.

Take a look at the existing markup, and then refactor it to use HTML5 semantic markup tags to define the structure of the document. When you're done, the page should look something like the screenshots show below (you can use `<details>` elements to create expander controls, and use `<summary>` to create the caption for each `<details>` element):

Unexpanded view (in Google Chrome)



Expanded view (in Google Chrome)

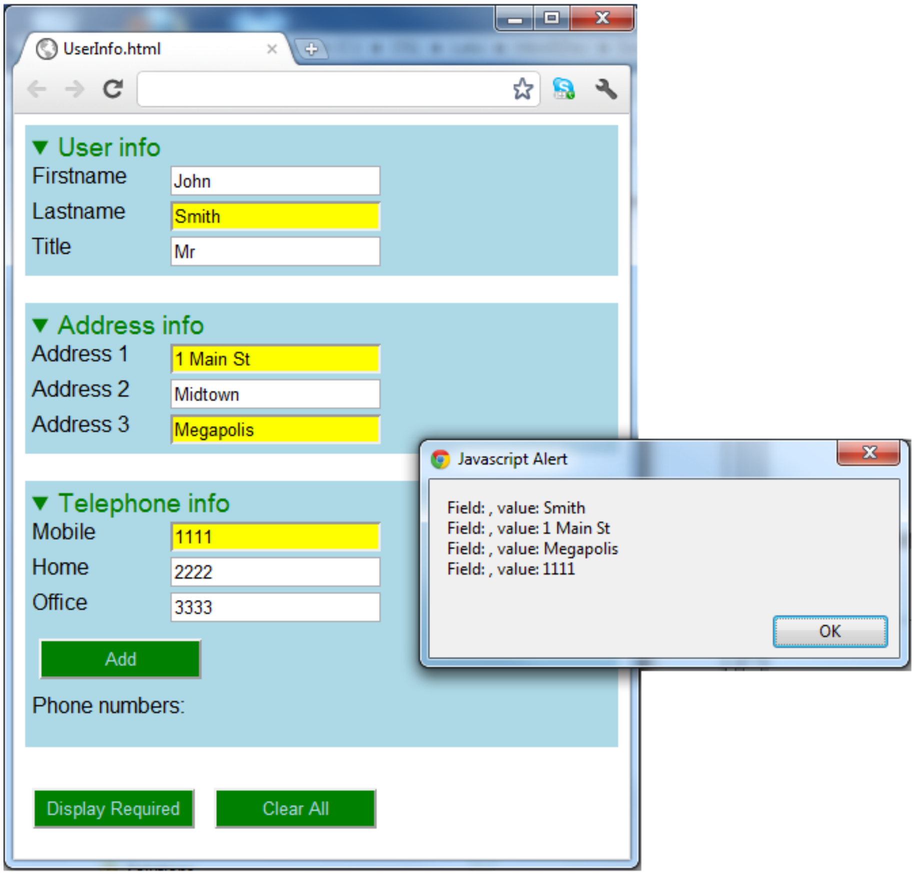


Exercise 2 – Using the selector API

You might have noticed that some of the text boxes are "required" – i.e. they have a `required="required"` attribute. Bearing this in mind, implement the *Display Required* button-click handler as follows:

- Get all the required text boxes by using the HTML5 selector API.
- Change the background colour of all the required text boxes to yellow.
- Display an alert popup showing the current values from all the required text boxes.
- After the user closes the alter popup, revert all the required text boxes to a white background colour.

This is the kind of thing you're aiming for:
Unfortunately the yellow background will show up after you close the popup in Chrome: https://stackoverflow.com/questions/38960101/why-is-element-not-being-shown-before-alert



Firstname	John
Lastname	Smith
Title	Mr
Address 1	1 Main St
Address 2	Midtown
Address 3	Megapolis
Mobile	1111
Home	2222
Office	3333

```
Field: , value: Smith
Field: , value: 1 Main St
Field: , value: Megapolis
Field: , value: 1111
```

Then implement the *Clear All* button-click handler, so that it clears the content of all the text boxes (use the HTML5 selector API to get all the text boxes).

Exercise 3 (if time permits) – Updating the DOM

Enhance the Web page so that it displays a bulleted list of all the telephone numbers entered by the user (the *Telephone info* part of the Web page has an empty `<ul>` element named `phoneNumbers` ready to receive list items).

Here are some hints:

- In the button-click handler for the *Add* button, get the text entered in the three text boxes and create a string in the format *mobileNumber homeNumber officeNumber*.
- Use DOM to create a new `<li>` element containing the string, and append the `<li>` to the `phoneNumbers` element.

You might also like to enhance the *Clear All* functionality from earlier, so that as well as clearing all the text boxes, it also clears the list items.

Exercise 4 (if time permits) – Additional suggestions

Think about how you can add the following features to the Web page:

- Placeholder text and regular expressions for input controls
- Data lists, via the `<datalist>` element
- Various different types of input control

You might also like to enhance the *Clear All* functionality from earlier, so that as well as clearing all the text boxes, it also clears the list items.