# Information Technology 3

# Working with forms

## Kristof Michiels

# Forms: good interaction design can make or break a modern web application

# Forms revisited

- We already learned about forms in IT1

- You may use these examples and slides if you wish

- Today we revisit forms with a few twists and a focus on the right structure

# Forms

- Important part of modern web applications: allow users to interact with systems

- Forms are where visitors of a web page actually do something else than consume content

- Forms enable visitors to do everything from filling out a contact form, write a contribution, enter payment information,...

- Despite being very important, forms are often poorly designed and coded

- Forms can be as simple as an e-mail address and a message field

- Or they can be hugely complex, spanning multiple pages

# Forms

- Forms have traditionally been hard to style => often black box, browser decides

- We can, however, style at least some parts of the appearance of form controls

- We will learn how to style fieldsets, labels, text inputs, and buttons

- In addition: how to style various form elements, including custom styling of checkboxes and select menus

# Forms

- Short and relatively simple forms are easiest to fill in when the form labels appear vertically above their associated form elements

- Users can simply move down the form step by step, reading each label and completing the following form element

- This method works well for short forms collecting relatively simple and predictable information such as contact details

- It is also a very good baseline for viewing forms on smaller viewports such as mobile browsers.

# Fieldsets and Legends help add structure and meaning to a form

# Fieldsets

- A fieldset is used for grouping related blocks of information

- In the example, 3 fieldsets are used: one for the contact details, one for the comments, and one for the "Remember me" preference

# Legends

- Use a legend element to identify the purpose of a fieldset

- Legends act a little like a fieldset's heading

- Usually appearing vertically centered over the top border of the fieldset and indented a little to the right

- By default fieldsets are often rendered as having a double border

- This slightly unusual appearance is implemented by different browsers in different ways

- It seems to be a special case in the browser's rendering engine, and throwing normal CSS properties at it to undo the odd positioning rarely has the effect you expect

**Label elements help add structure and increase the usability and accessibility**

# Labels

- Labels are used to add a meaningful and descriptive label to each form element

- Clicking the label element will - in many browsers - cause the associated form element to gain focus

- They increase form usability for people using assistive devices

- Screen readers will correctly associate a form element with its label

- Their users can also bring up a list of all the labels in a form, allowing them to audibly scan through the form

# Labels

Associating a label element with a form control can be done in one of two ways: either implicitly by nesting the form control inside the label element:

```
<label>Email <input name="comment-email" type="email"></label>
```

Or explicitly by setting the for attribute of the label equal to the id attribute value of the associated form element:

```
<label for="comment-email">Email<label>
<input name="comment-email" id="comment-email" type="email">
```
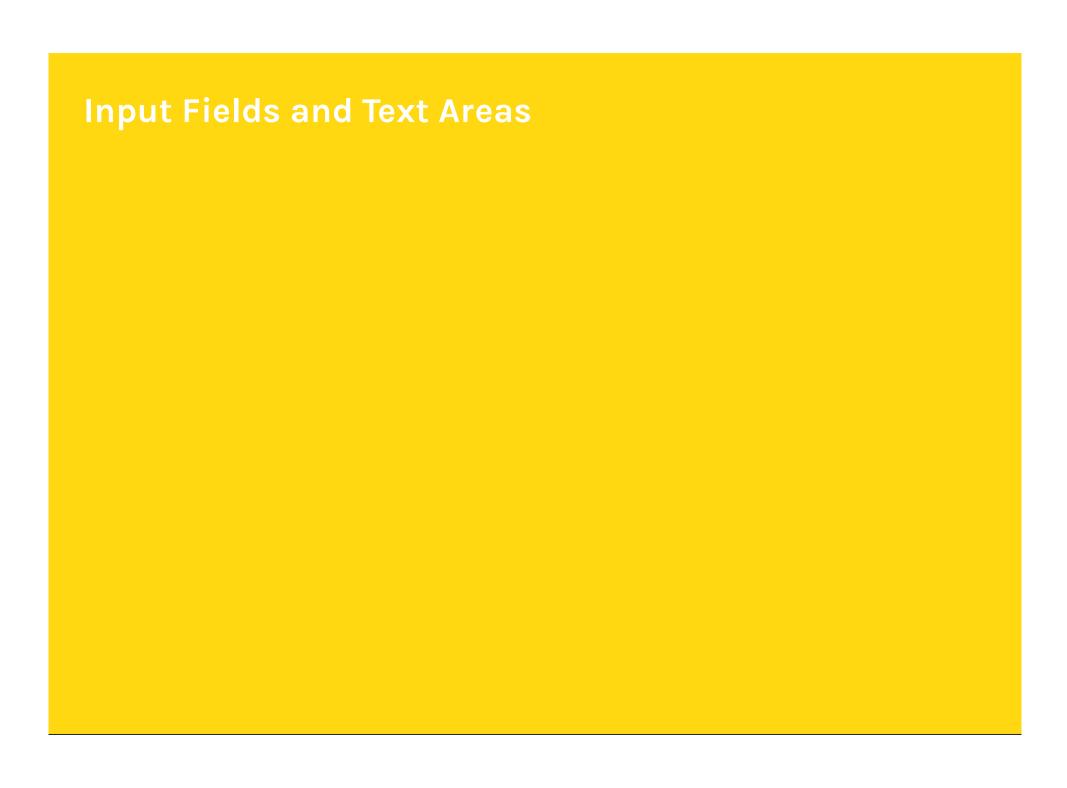
I prefer the explicit method.

# Name and id attributes

- Most form controls contain both a name attribute and an id attribute

- The id attribute is required to create the association between the form input and the label

- The name attribute is required so that the form data can be sent back to the server

- The id and name don't have to be the same, but it can be a handy convention to keep them identical when possible

# For-attribute

- Labels associated with form controls using the for attribute don't need to be near those controls in the source code

- They could be in a completely different part of the document

- From a structural point of view, separating form controls from their labels should be avoided wherever possible

# Input Fields and Text Areas

# Text Areas

- Text areas are for typing in multiple lines of text, as in the comment field

- The cols and rows attributes can be used to set a default size of the text area, mostly to indicate the approximate length of the expected content

- We will be able to further style the textarea with CSS

```
<textarea name="comment-text" id="comment-text" cols="20" rows="10"></textarea>
```

# Input Fields

- The input element is a versatile form control

- Default = single-line text input, but type attribute can change this to a variety of different form controls

- Setting type="password" obscures the value, and type="checkbox" creates a checkbox

- There are a lot of different values for the type attribute, many of them added in HTML5

- Some are mostly variations of the text input but with special behavior: e.g. email, url, and search

- Some types create different interface controls where supported: checkbox, radio, color, range, and file

- Apart from type, there is also a whole host of attributes to use on the inputs, to declare the expected format

# Input Fields

- The different types of form inputs and their attributes are useful for automatic validation of forms

- On devices with an onscreen keyboard, changing the type triggers the software keyboard to change its layout

- If we add an e-mail field and a URL field with the correct types, the keyboard on smartphones and tablets will automatically adjust to make it easier to type in the correct value when we focus each field

```
<input name="comment-author" id="comment-author" type="text">
```

# Let's look at an example

# CodePen

- Find the example on CodePen in the IT3 collection: https://codepen.io/admkrm/pen/OBReKX

# Your assignment for this week

# Assignment for this week

- You will make a web page for a fictive real estate agency, or you redesign a page for an existing brand

- The page you will build will be a landing page for a real house (you can reuse existing content e.g. from immoscoop) that is on sale.

- This page will contain all the necessary information and photos on that house

- The page will contain real estate branding information

- The page will contain a smart contact form for people interested in this particular house. Use the example we have seen today as an inspiration. Be creative!

- Your site needs to be typography (and mobile) first, and should be responsive

- Please take note that this IS an actual web site that you are making. It should look real-world and the design should be your own

# Immoscoop

Immoscoop

[https://www.immoscoop.be](https://www.immoscoop.be)

# Immoscoop

Immoscoop

[https://www.immoscoop.be](https://www.immoscoop.be)

# Working with forms

## Questions?

kristof.michiels01@ap.be