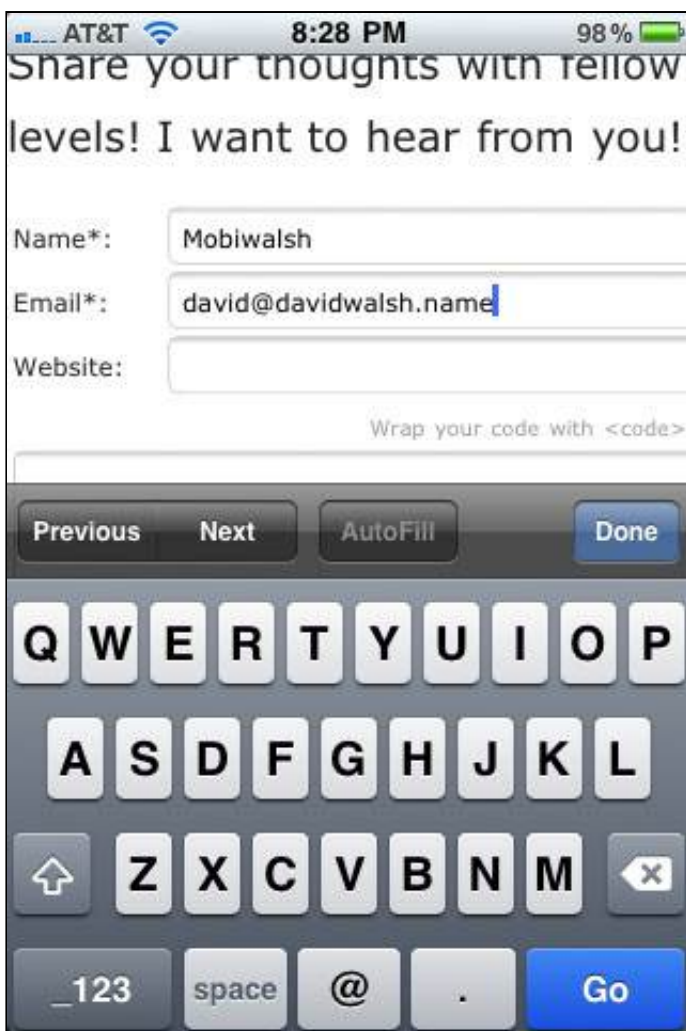


`<input type="email">`, `<input type="tel">`, `<input type="URL">` and `<input type="search">`

`<INPUT TYPE="EMAIL">`

This input type is relatively straightforward to use. In mobile applications, this new input type pops up a keyboard layout adapted to email input. Note the "@" key, the "." key, etc.



This input type is very interesting as it provides default validation behaviors:

- If the value entered looks like an email address (contains a "@"...), the field is *valid*,

and gets the pseudo CSS class `:valid`

- If the value entered does not contain an "@", and does not look like an email address, the field is *invalid* and gets the pseudo CSS class `:invalid`

See the next example to see this in action. More details will be presented in a later section dedicated to form validation.

Typical use

[Online example at JS Bin](#)

Try it on your browser:

Enter your email:

```
10. <!DOCTYPE html>
    <html>
    <head>
      <meta charset="utf-8">
      <title>Example of input type=email</title>
      <style>
        input:invalid {
          background-color:pink;
        }
      </style>
    </head>
    <body>
      <label for="email">Enter your email</label>
      <input type="email" id="email">
    </body>
    </html>
```

Note the CSS rule that turns the background color of the email input field to `pink` if a user enters an invalid address (lines 7-8). Also note that the validation is based only on matching a regular expression (the address should contain a "@", a ".", etc.). It does not

check if the address is an existing one.

Current support: all browsers, on mobile or desktop versions, since 2012.

`<INPUT TYPE="TEL">`

This input field is really useful on smartphones and tablets, as it makes the browser pop up a keyboard layout suitable for entering phone numbers:



This input type is often used with the new `placeholder` and `pattern` attributes that are detailed in another section of this course. It is supported by all recent major Web browsers, on mobile devices and desktops.

[Online example at JS Bin](#)

Try it in your browser (we used the same CSS for changing the background-color when the input value is invalid):

Enter a telephone number:

Source code:

```
<!DOCTYPE html>
```

```

<html>
<body>
<head>
  <title>Example of input type=tel</title>
<style>
  input:invalid {
    background-color: pink;
  }
10. </style>
</head>
<body>
  <label for="tel">Enter a telephone number:</label>
  <input type="tel" id="tel"
    placeholder="(555) 555-5555"
    pattern="^(?\\d{3})?[-\\s]\\d{3}[-\\s]\\d{4}.*?\\)$/"/>
</body>
</html>

```

<INPUT TYPE="URL">

This input field is really useful on smartphones and tablets, as it makes the browser pop up a keyboard layout suitable for entering URLs:



This field is also compatible with *the validation API* (more on this in another section).

Here is an online example that shows the use of

the `placeholder` and `pattern` attributes for entering only URLs that start with `ftp://` or `http://`. [Check it at JS Bin!](#)

Or try it here in your browser:

Enter a URL (default validation):

Enter a URL (custom validation, must start with http, https or ftp):

Source code:

```
<!DOCTYPE html>
<html>
<head>
  <title>Example of input type=url</title>
<style>
  input:invalid {
    background-color: lightPink;
  }
</style>
</head>
<body>
12. <label for="url1">Enter a URL (default validation):</label>
    <input type="url" id="url1"/>
    <p>
      <label for="url2">Enter a URL (custom validation, must start
with http, https or ftp):</label>

      <input id="url2" type="url" placeholder="http://www.domain.com"
        pattern="(http|https|ftp)\\:\\\\/\\[a-zA-Z0-9\\-
\\.\\/]*/></p>
    </body>
</html>
```

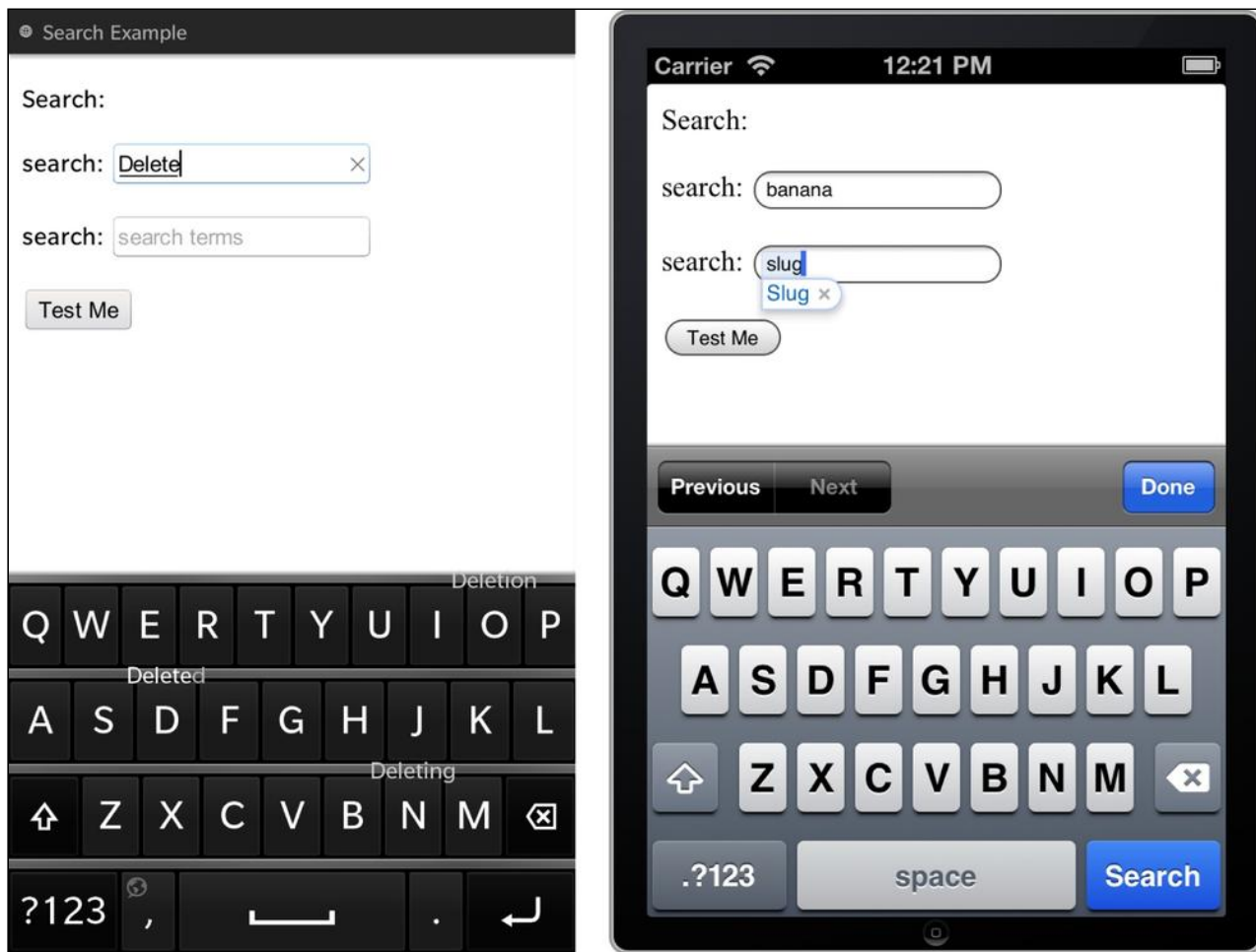
Lines 16-17 show the use of a `pattern` attribute with a JavaScript regexp that accepts only URLs starting with `http`, `https` or `ftp`. More details on the `pattern` attribute are given in the section that presents the new HTML5 form attributes.

<INPUT TYPE="SEARCH">

The search type is used for search fields (i.e., for a search engine). A search field behaves like a regular text field, except that it may provide some feedback GUI for stopping the current request and emptying the search field, or it may provide a drop-down list of recent search results.

The specification does not state what the GUI should look like, so current implementations show variations in the look and feel.

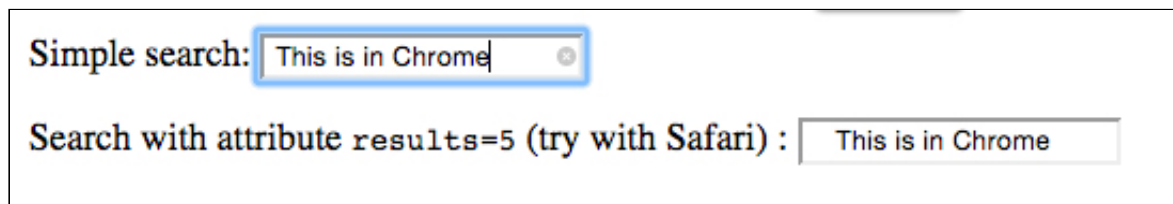
Mobile (left blackberry, right Safari IOS):



Typical use:

```
<label for="search1">Simple search:</label>
<input type="search" id="search1">
<p>
<label for="search2">Search with
attribute <code>results=5</code> (try with Safari): </label>
<input type="search" id="search2"results=5>
```

Results on Chrome and Opera desktop - notice the small cross on the right when one enters a value:



Simple search:

Search with attribute `results=5` (try with Safari) :

Same example with Safari desktop, this time the second line with an attribute `results=5` shows a small icon on the left:



Simple search:

Search with attribute `results=5` (try with Safari) :

Example that shows a drop down list of recent searches (Safari screenshot borrowed from [this excellent site about HTML5 forms that is worth reading](#)):

Q- Submit

Q- | Submit

No recent searches

Q- | Submit

Recent Searches

test

bananna

Clear Recent Searches

SOURCE CODE FOR THE KNOWLEDGE CHECK BELOW

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Example of input type=email</title>
  <style>
    input:? {
      background-color:pink;
    }
10.   input:?? {
      background-color:lightGreen;
    }
  </style>
</head>
<body>
  <label for="email">Enter your email:</label>
  <input type="email" id="email">
</body>
</html>
```


KNOWLEDGE CHECK 5.4.4 (NOT GRADED)

How could you fix the above code so that an invalid email address entered in the input field would turn its background color to pink?

☐ Instead of "?", I would put the string "invalid" and instead of "??", I would put the string "valid"

☐ Instead of "?", I would put the string "valid" and instead of "??", I would put the string "invalid"
