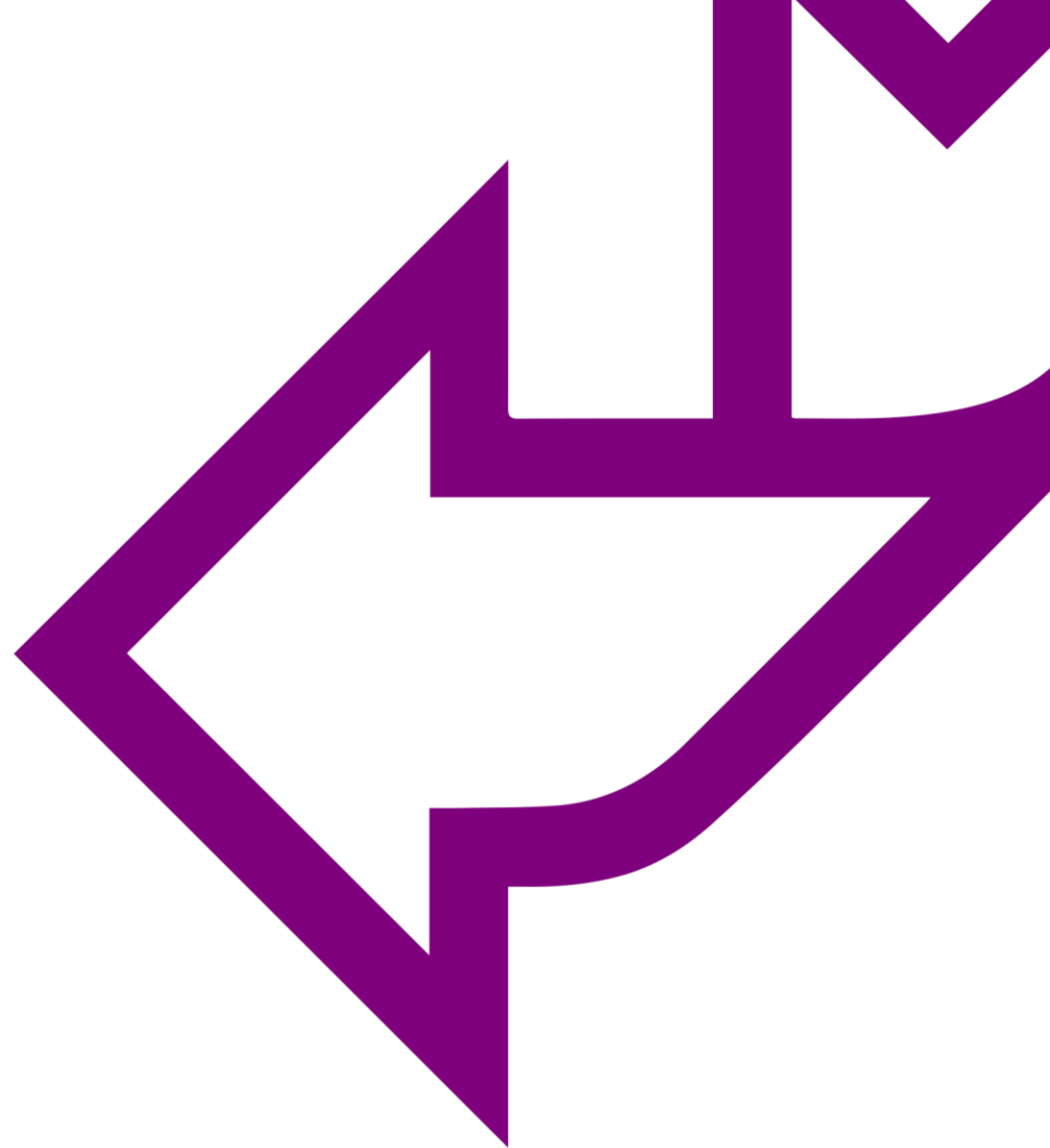




Forms

Basic HTML



QA The `<form>` tag

Defines a form in HTML

Attributes

- **action**: URL to execute when form is submitted
- **method**: How the information is passed to server (GET or POST)

```
<form method="post" action="cgi-bin/subscribe.pl">  
...  
</form>
```

QA Text and push buttons

Kinds of push button

- **submit**: Send the form information to the server for processing
- **reset**: Reset all form fields
- **button**: Generic push button

Single line text field

- **type="text"** attribute of **<input>** tag
- **type="password"** as text, but text is not displayed on screen – browser default hidden used

```
<form
  method="post"
  action="scripts/subscribe.pl"
>
  Please enter your name:
  <br/>
  <input type="text" size="20" name="UsrName"/>
  <br/>
  <input type="submit" name="OKBtn" value="OK" />
</form>
```

Please enter your name:

QA Multi-line text input

`<textarea> .. </textarea>`

- Initial text can also be supplied
- Browser will supply scroll bars if necessary

```
Address: <br/>
<textarea
  name="UstrAddr"
  rows="7"
  cols="24"
>
  Enter address here
</textarea>
```

Address:

Enter address here

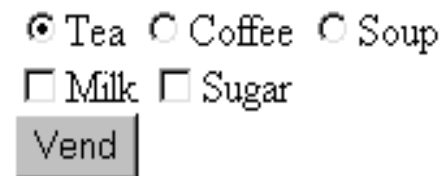
QA Check and radio buttons

Radio buttons: Select from one of a group

- **name** attribute groups buttons together

Check buttons: Independent yes/no value

```
<form ...>
  <input type="radio" checked name="RadioDrink" value="Tea"/>Tea
  <input type="radio" name="RadioDrink" value="Coffee"/>Coffee
  <input type="radio" name="RadioDrink" value="Soup"/>Soup<br/>
  <input type="checkbox" name="CheckMilk" value="Yes"/>Milk
  <input type="checkbox" name="CheckSugar" value="Yes"/>Sugar<br/>
  <input type="submit" name="OKButton" value="Vend"/>
</form>
```



QA <select> and <option>

Select one from a drop-down list

<select> tag defines the list

- **size** attribute is 1 for drop down list, > 1 for scrolled list
- **multiple** attribute specifies multiple selection list

<option> tags define the contents

```
<select name="Drop1" size="1">  
  <option>Cookies</option>  
  <option>Tortillas</option>  
  <option selected>  
    Poppadoms (my favourite)  
  </option>  
</select>
```



```
<select name="Drop2" size="3">  
  <option value="2">High</option>  
  <option value="1">Medium</option>  
  <option value="0">Low</option>  
</select>
```





HTML5 input elements



The HTML5 spec started with forms

- Opera and Safari are the driving force and most complete

13 type options to date

Mostly extend the `<input>` tags with additional type values

- If a browser does not understand the extension rendered as:

No requirement in the spec for how browsers present

- Different browsers show different UI and error messages
- Browsers that do not understand the new types treat them as text
- JavaScript defence is needed for legacy browsers

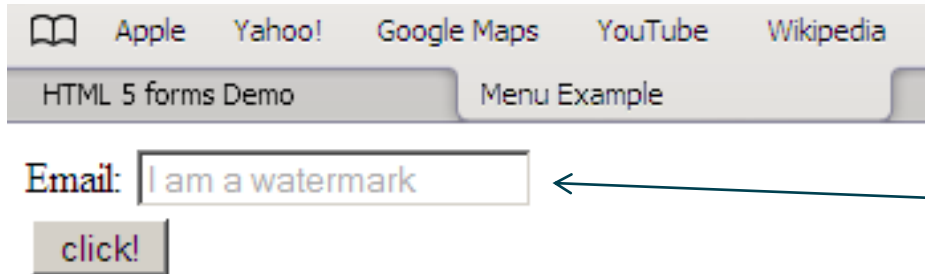
```
<input type="text" />
```

QA Placeholder attribute

The placeholder attribute offers default text

- Gives the user example or instruction for the field
- Sometimes called a watermark
- Can only be used for text values
- Is not a default value

```
<input type="text" placeholder="I am a watermark" />
```



A screenshot of a web browser interface. At the top, there is a navigation bar with links: Apple, Yahoo!, Google Maps, YouTube, and Wikipedia. Below this, there are two tabs: 'HTML 5 forms Demo' and 'Menu Example'. The main content area shows a form with a label 'Email:' followed by a text input field. The input field contains the placeholder text 'I am a watermark'. Below the input field is a button labeled 'click!'.

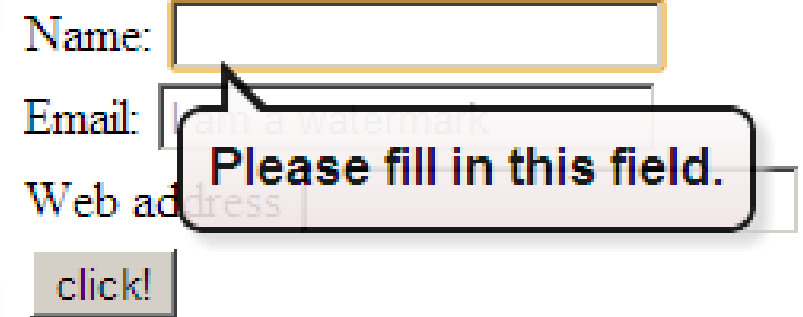
Text disappears as soon as you tab into the box or start typing

QA Required Fields

- You can force a field to be mandatory on the client

```
<input type="text" autofocus="true" required />
```

- On a submit action, an error message may appear:
 - Appears in:
 - Safari 6+
 - Firefox 4.0
 - Opera 9+
 - Chrome 9+
 - IE10+
 - Message will appear differently in each browser



A screenshot of a web form with three input fields: "Name:", "Email:", and "Web address:". The "Name:" field is empty and has a red border. A red speech bubble with the text "Please fill in this field." points to the "Name:" field. The "Email:" field contains the text "Email watermark". The "Web address" field is empty. Below the fields is a button labeled "click!".

QA Autofocus attribute

It is common to have the first field of a form to focus

- To have the cursor flashing ready to type

Previously achieved with JavaScript

- The markup representation is faster
 - Part of the page rendering rather than code execution

Supported in all browsers other than IE9 and less:

- Use JavaScript to support legacy browsers

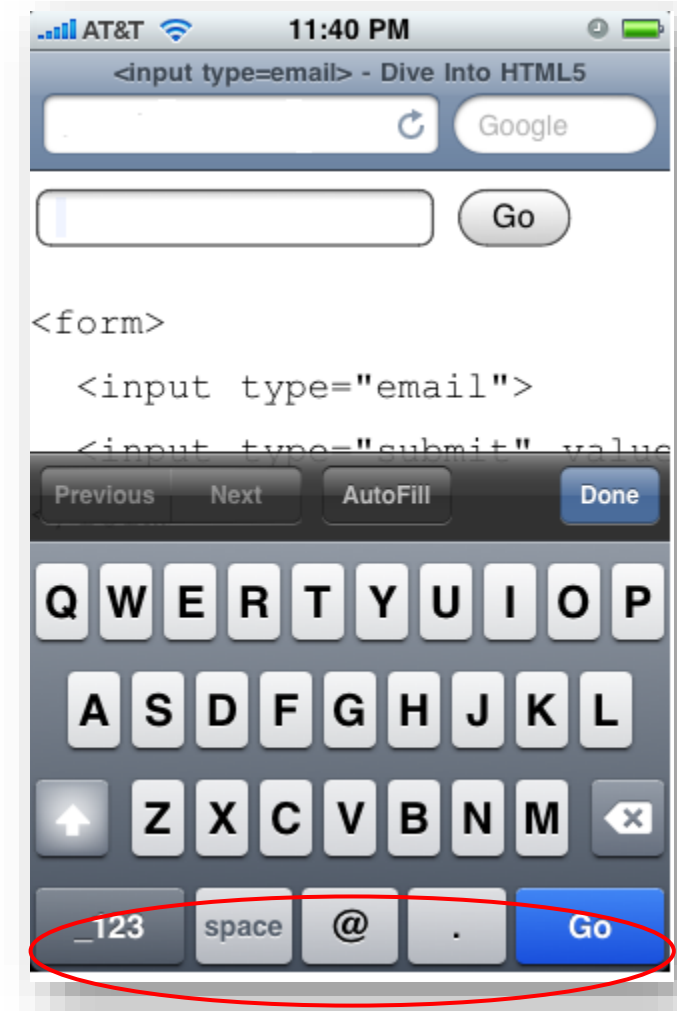
```
<form>
  <input name="q" autofocus="true">
  <input type="submit" value="Search">
</form>
```

QA Email input type

Add type value of email

What happens in the client is not consistent

- HTML5 spec does not demand it
 - Opera and safari provides submit validation
 - Firefox provides client validation on blur
 - Safari mobile changes the input keyboard
 - IE 9< does nothing
- Form will not submit until the error is solved
- Provides a simple input mask to check input
 - e.g. boffin@qa.com



QA Web address input type

Denotes the input must have schema prefixing an address

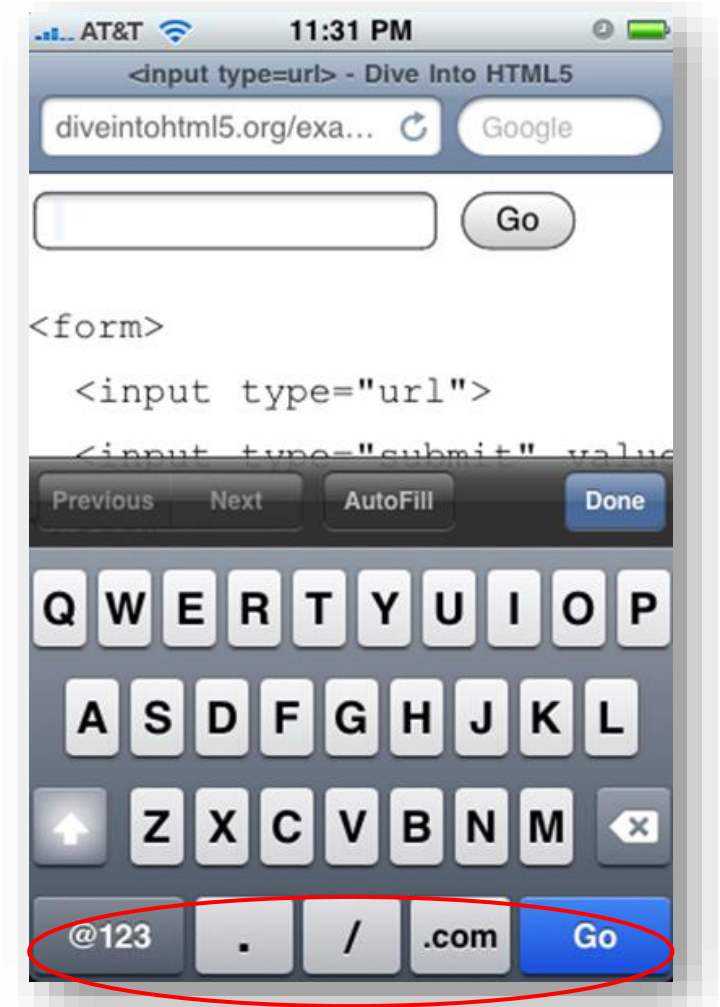
- e.g.: `http://www.qa.com` or `ftp://ftp.mysite.com`

Different browsers have different functionality

- More of an issue than previously due to different functionality
 - Chrome and Firefox 4 force user to add schema
 - Opera prefixes an address with `http://`
 - Safari mobile provides a different keyboard

To use effectively, combine with a placeholder

- Users don't really understand schemas



QA Number input type

Numbers often need to be constrained by range

New number type provides this functionality

Four attributes:

```
<input type="number" min="1" max="12" step="2" value="6" />
```

- **min** - lowest range
- **max** - upper range
- **step** - what value the control enumerates by
- **value** - default value

Browser support issues

- Firefox, Chrome, and Opera display these as 'spinboxes'
- No increment/decrement buttons in IE
- iOS UI widget doesn't consider step, min, or max values



QA Range input type

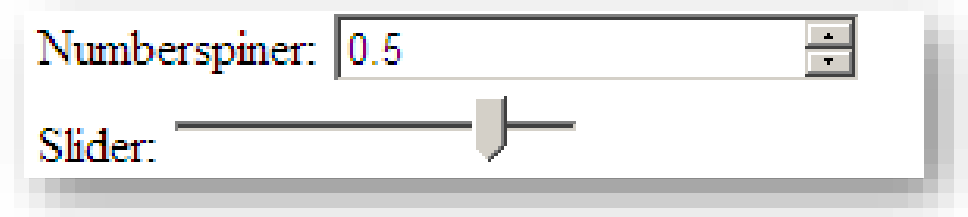
Creates a slider bar in Chrome, Firefox, IE10, & Opera

```
<input name="r" type="range" min="1" max="11" value="9">
```

Has the same attributes as the Number type

Only IE10 displays the current slider value

- In other browsers, you need JavaScript to display the value
 - Obtain a pointer to the slider
 - Subscribe to its change event
 - Add the output to a span



QA Date input type

A popup calendar is standard for date selection

- Normally requires a JavaScript framework
- Around 10% of web users do not support JavaScript

HTML5 defines six date time types

- Use UTC in the same way as time element

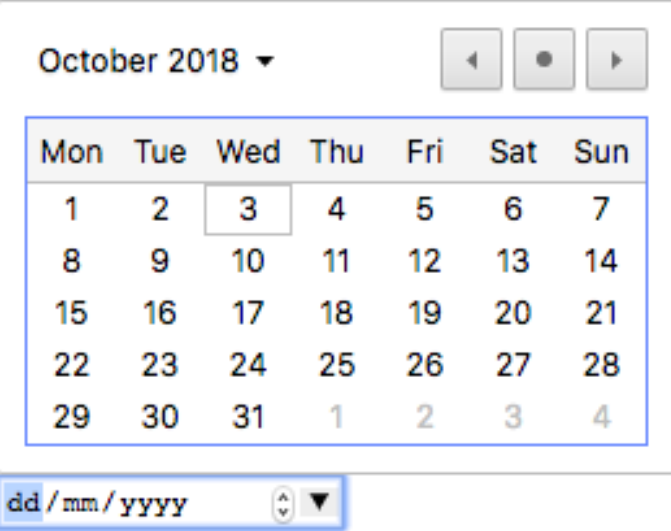
No support in IE or Safari as of Oct 2018

Will not require JavaScript enabled

- Native support is the ultimate aim

ECMAScript5 allows you to create dates from UTC

Formatting is dependent on browser defaults



A screenshot of a date picker interface. At the top, it shows 'October 2018' with a dropdown arrow and three navigation buttons (previous, current, next). Below this is a calendar grid with days of the week as headers. The date '3' is highlighted with a blue border. At the bottom, there is a text input field with the placeholder 'dd/mm/yyyy' and a small dropdown arrow.

Mon	Tue	Wed	Thu	Fri	Sat	Sun
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

dd/mm/yyyy



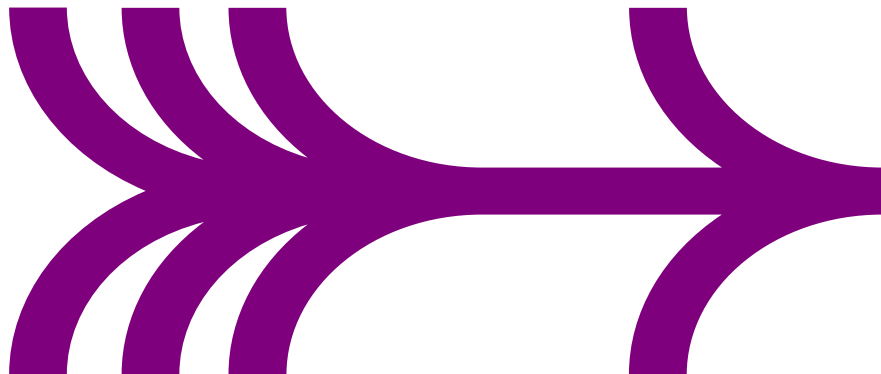
Search input type

Provides a semantic definition for search input

- Keep to one per page

Need to set a name for the search field otherwise nothing will be submitted

Most common name is q



Search Google:

```
<input type="search" q="googlesearch"/>
```


QA Colour input type

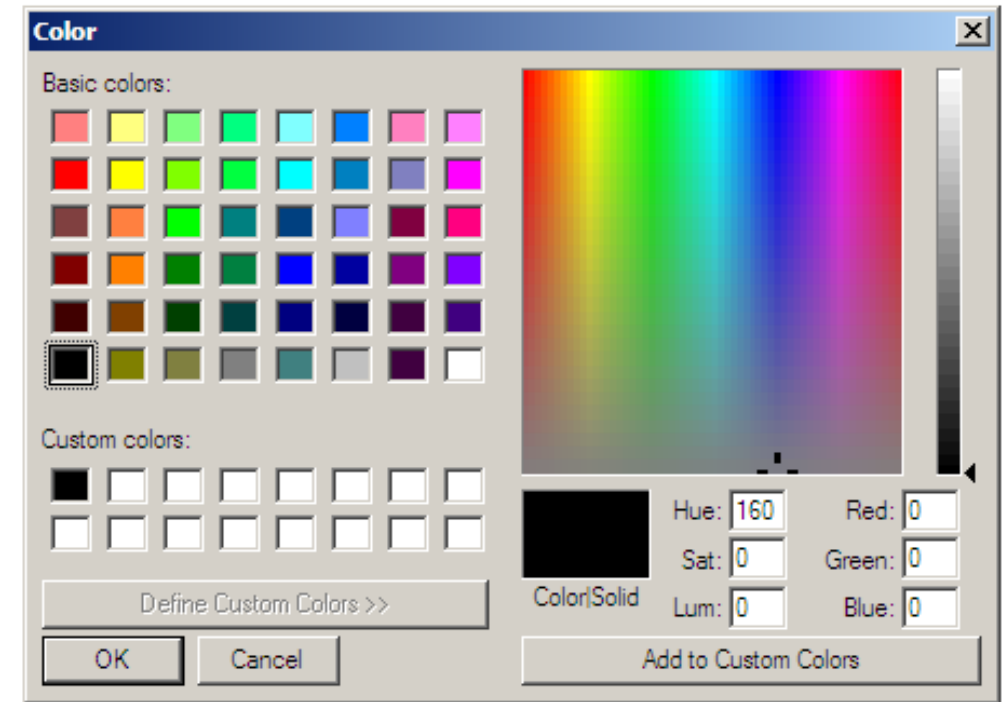
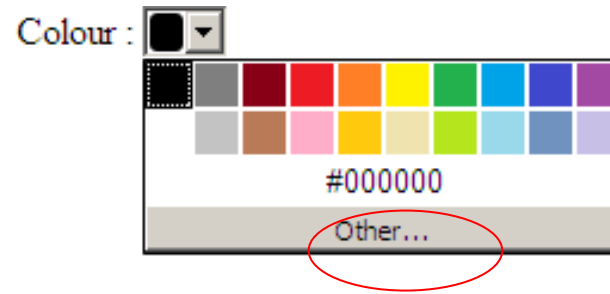
Currently implemented in Chrome, Firefox, and Opera

Returns a six-digit hex value

- In other browsers should be verified as a hex value

On MacOS and Windows returns a colour picker

- On Linux only the basic colour picker so far



QA Pattern

The pattern attribute allows use of regular expressions

Pattern works with the following input types:

- text
- search
- url
- tel
- email
- password

```
<input type="text" pattern="[0-9]{13,16}" name="CreditCardNumber" />
```

Ensure the user understands the regular expression

- Support with a placeholder

QA Datalist

The `<datalist>` tag provides a list of pre-defined options

- Use the `<input>` element's list attribute to bind it to the `datalist`

Provides an "autocomplete" feature on `<input>` elements

- Users see a drop-down list of options as they input data

```
<input list="browsers" />
<datalist id="browsers">
  <option value="Internet Explorer">
  <option value="Firefox">
  <option value="Google Chrome">
  <option value="Opera">
  <option value="Safari">
</datalist>
```

QA Form validation

As we have seen, some browsers ship with validation

- IE offers no UI implementation in any version
- Firefox and Opera often the most complete implementation
- Chrome is pretty good and Safari will get better
 - Some controls have silent errors, not enough UI feedback

These are JavaScript free client validation

Uneven support may be more trouble than benefit

- You can tell a browser to switch it off
- Still benefiting from the semantic types

```
<form novalidate>
  <input type="email" id="addr">
  <input type="submit" value="Subscribe">
</form>
```

QA <fieldset>

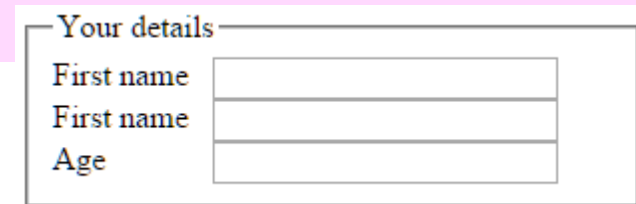
- Group related form elements together
- Meaningful legend provides accessibility
- Can disable all contained elements

```
<fieldset>
  <legend>Your details</legend>

  <label for="fname">First name</label>
  <input id="fname" type="text">

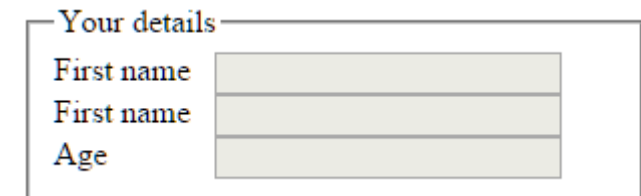
  <label for="sname">First name</label>
  <input id="sname" type="text">

  <label for="age">Age</label>
  <input id="age" type="number">
</fieldset>
```



A visual representation of a functional form fieldset. It features a legend titled "Your details" at the top. Below the legend, there are three input fields: two text inputs labeled "First name" and one number input labeled "Age". The inputs are arranged vertically and are all active and ready for use.

```
<fieldset disabled>
  ...
</fieldset>
```



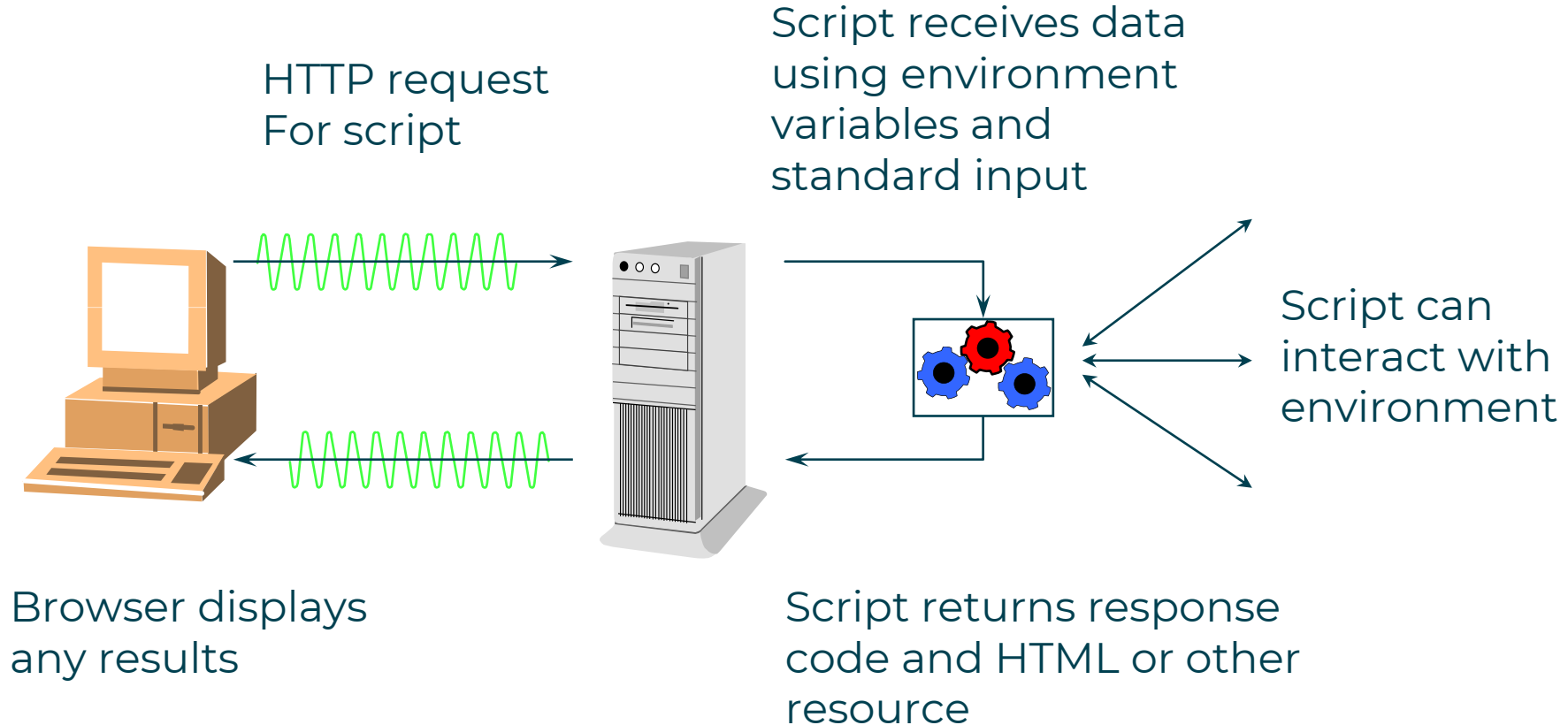
A visual representation of a disabled form fieldset. It features a legend titled "Your details" at the top. Below the legend, there are three input fields: two text inputs labeled "First name" and one number input labeled "Age". The inputs are arranged vertically and are all disabled, indicated by a light gray background and a border.

QA Submitting the form

- The “submit” button triggers the script
- The values of the form’s fields are sent to the server
- Two different mechanisms: GET and POST



QA Server scripting mechanism



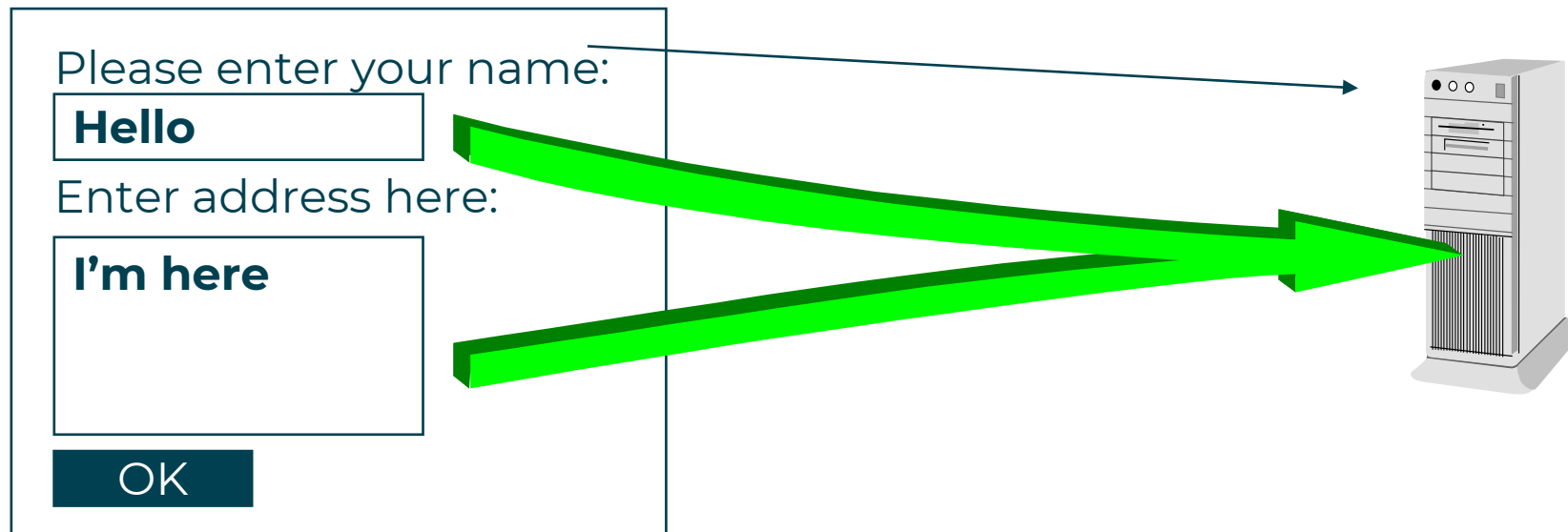
QA Form security issues

Scripts should check data received from a form

- Validate all data carefully!

Data sent from a form is not normally encrypted

- Unless a secure connection is made (e.g., SSL)





Quick Lab Chapter 6 – Forms

Create a form on a web page using different input types and in-built validation.