



# Forms

Javascript for Emulating Events

# Overview

---

- ▶ Revision of Forms
- ▶ Form Properties
- ▶ Form Elements
- ▶ Faked Events using JavaScript
- ▶ Examples



# A Quick Summary of Forms

---

- ▶ A form is a page which interacts with the user
- ▶ Forms are made up of elements which can be shown as text boxes, check boxes, radio buttons, drop down lists etc.
- ▶ JavaScript functions will be used to interpret the data obtained from the forms
- ▶ A form in HTML is defined by the tag `<FORM>`

```
<form>
```

```
    <input type="text" size="25"  
    value="Enter your name here!">
```

```
</form>
```

---



# A Quick Summary of Forms

---

- ▶ Form attributes can be:
  - ▶ Method = get or post
    - ▶ [how to submit form data (get is the default method)]
  - ▶ Action = URL
    - ▶ [where to submit the data]
  - ▶ Target = name of window
    - ▶ [which window to display responses, e.g. target=blank will display the responses in a new window]
  - ▶ onSubmit = “JavaScript code”
    - ▶ [the event handler for a submit event]



# Form Properties

---

- ▶ Each form in a document is an object and it is defined by:

```
document.forms[0]
```

- ▶ If there is a second form within the document then it will be defined as:

```
document.forms[1]
```

- ▶ For each form there will be elements such as:

```
<INPUT TYPE = "button"> ...element 1
```

```
<INPUT TYPE = "checkbox"> ...element 2
```



# Form Properties

---

- ▶ To refer to specific elements within a form one can specify:

```
document.forms[0].elements[0]
```

- ▶ This would be referring to the first element in the first form of the document. This would be:

...the button element specified as `<INPUT TYPE = "button">`

- ▶ Likewise : `document.forms[0].elements[1]` will refer to the second element which is the checkbox
- 



## Form Properties – the action property

---

- ▶ Imagine you are using a form to gather data about the number of objects (e.g. pens) which a user has bought online
- ▶ You want to submit the form data to one URL if the user has bought more than 10 pens and to another URL if they purchase less than 10 pens. To do this, we have to use the `if...else` statement
- ▶ Such a statement would need to be written within a function e.g. `buypens (pens)`



# Form Properties – the action property

---

```
▶ <SCRIPT LANGUAGE="JavaScript">
function buypens (pens)
{
    if (pens>=10)
        {documents.forms[0].action =
"http://...1.html"}
    else
        {document.forms[0].action =
"http://...2.html" }
}
```



# Form Properties – the action property

---

```
<FORM method=get
  onSubmit="buypens (this.quantity.value) ">
<INPUT TYPE="text" name="quantity" ROWS=1
  COLUMNS=3>Quantity of order
</FORM>
```

- ▶ The form has an element which is a text box which is 3 spaces wide; the name of this element is “quantity”
- ▶ The form definition, specifies an **onSubmit** event handler



# Form Properties – the action property

---

```
<FORM method=get  
  onSubmit="buypens(this.quantity.value)">  
<INPUT TYPE="text" name="quantity" ROWS=1  
  COLUMNS=3>Quantity of order  
</FORM>
```

- ▶ The event handler will call the function `buypens()` with the parameter `this.quantity.value`
- ▶ The value is inputted by the user



# Emulating events using JavaScript – the validation

---

- ▶ Event emulators are methods of the form element objects
- ▶ These will be used to help and direct the user
- ▶ Why?
- ▶ Sometimes it happens that the user clicks on one element first but then clicks on the second element by mistake, making his responses inconsistent and invalid



# Emulating events using JavaScript – the validation

---

- ▶ So what will such a code involve:

```
<FORM>
```

```
<INPUT NAME="spicy" TYPE="checkbox">Do you like  
spicy foods?<p>
```

Please select which spicy food is your favourite:

```
<INPUT NAME="spicetype" TYPE="radio" value="mexican"  
onClick="checkspicy(document.forms[0].spicy.checked)">Mexican
```

```
<INPUT NAME="spicetype" TYPE="radio" value="indian"  
onClick="checkspicy(document.forms[0].spicy.checked)">Indian
```

```
<INPUT NAME="spicetype" TYPE="radio" value="notype"  
onClick="checkspicy(document.forms[0].spicy.checked)">I said I don't like spicy food!
```

```
</FORM>
```

---



# Emulating events using JavaScript – the validation

---

## ▶ And how will the function validate the script?

```
<SCRIPT language="JavaScript">
```

```
Function checkspicy(likes)
```

```
{
```

```
  if (likes==true)
```

```
  { if
```

```
    (document.forms[0].spicetype[2].checked==true)
```

```
      {document.forms[0].spicetype[0].click()}}
```

```
  else
```

```
  { if
```

```
    (document.forms[0].spicetype[2].checked==false)
```

```
      {document.forms[0].spicetype[2].click()}}
```

```
}
```

```
</SCRIPT>
```

---



# Conclusion

---

- ▶ Forms are used to gather data from the user interactively
- ▶ Forms will have properties and elements
- ▶ Properties include: action, method and target
- ▶ Elements include: button, text box, checkbox, radio button, drop down list
- ▶ Forms will be validated using JavaScript functions

