# Tutorial

## Activity 1: HTML <body> Elements

HTML <body> child elements have a number of different properties, for example:

* some have **paired** tags, others have **single / empty** tags
* some are **block** elements, others are **inline** elements
* some can be placed anywhere in <body>, others must be **nested** inside other elements.

Classify the following HTML <body> elements and identify at least one possible parent tag:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Element / Tag | Paired | Empty | Inline | Block | Parent Tag(s) |
| <a> | ✓ |  | ✓ |  | <div>, <p>, etc. |
| <br> |  | ✓ | ✓ |  | <div>, <p>, etc. |
| <div> | ✓ |  |  | ✓ | <div>, <body>, etc. |
| <em> | ✓ |  | ✓ |  | <div>, <p>, etc. |
| <form> | ✓ |  |  | ✓ | <div>, <body>, etc. |
| <header> | ✓ |  |  | ✓ | <body> |
| <h5> | ✓ |  |  | ✓ | <div>, <p>, etc. |
| <hr> |  | ✓ |  | ✓ | <div>, <p>, etc. |
| <img> |  | ✓ | ✓ |  | <body>, <article>, etc. |
| <input> |  | ✓ | ✓ |  | <form> |
| <li> | ✓ |  |  | ✓ | <ul>, <ol> |
| <mark> | ✓ |  | ✓ |  | <div>, <p>, etc. |
| <ol> | ✓ |  |  | ✓ | <div>, <p>, etc. |
| <p> | ✓ |  |  | ✓ | <div>, <body>, etc. |
| <span> | ✓ |  | ✓ |  | <div>, <p>, etc. |
| <table> | ✓ |  |  | ✓ | <div>, <body>, etc. |
| <th> | ✓ |  |  | ✓ | <table> |
| <ul> | ✓ |  |  | ✓ | <div>, <p>, etc. |

How are <div> and <span> similar? How are they different?

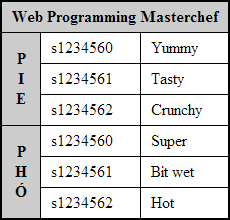
* Similarities:
  + Used as a container to group together other elements
  + Doesn’t provide any visual changes
  + Is used to style inline elements stored inside
* Differences:
  + <div> are usually used for dividing a part or section of an HTML document
  + <span> are usually used to style blocks of text

## Activity 2: Creating a Table.

A table element is a HTML structure made up of rows and cells that is used to lay out **tabular data** in a **grid** with neatly lined up rows and columns. Occasionally, you will want data or headings to expand to other cell spaces and you use rowspan and colspan attributes to make this happen.

The commonly used tags used to create a table are:

<table>, <tr>, <th> and <td>.



## 

## How many rows and how many columns does the table have before "cell spanning"?

* 7 rows and 3 columns

To give a table more structure and "semantic" meaning, the following tags are often included to contain rows and cells:

<thead>, <tbody> and <tfoot>.

These tags are placed inside the <table> tag in any order, but they will always be rendered in the order above.

**Please note:** When designing a layout, it is preferable to use the **CSS Grid layout model** which has many of the benefits of a table but without the fixed structure. Please avoid using tables for layout in your website as they do not adapt well when viewed in small screens such as mobile phones etc.

Tables can still be used but only for tabular data information, but even then always be mindful of the lack of layout adaption on smaller screens.

## Activity 3: Forms and Form Submission

Forms are used to submit data to a processing script. In this course we will write processing scripts in PHP, but there are many server side languages that can process form data.

A form has the following attributes:

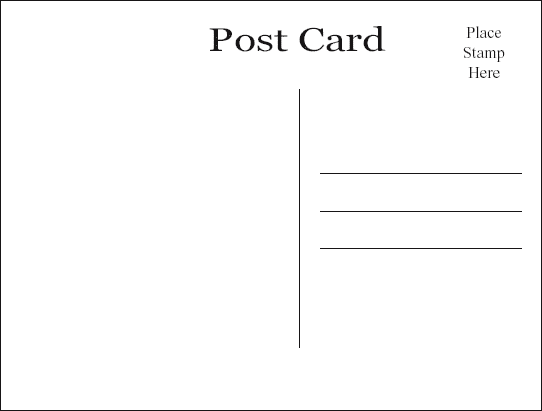
|  |
| --- |
| <form action="..." method="..." onsubmit="..." />  <!-- Input controls etc, will cover soon -->  </form> |

What goes in the action attribute?

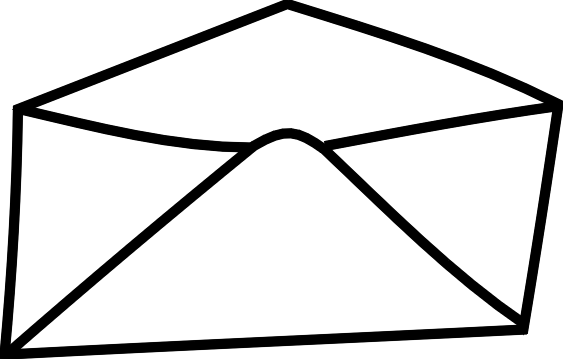
* The location where the data of the form will be sent to after submitted

The method attribute takes either post or get as a value. The post method puts the form data inside the body of the message, whereas get puts the form data inside the url. Discuss the benefits of each method option, think of examples where you would use one rather than the other.

* Benefit of post:
  + It is more secure
  + There are no limitations to the size of data being sent
  + The data are not saved or cached on client’s side
* Benefit of get:
  + The data is saved or cached on client’s side so they can be bookmarked by the user



GET



POST

Are there similarities with physical postcards and envelopes?

* There similarities with post and get form methods to physical postcards and envelopes
* Get method is similar with postcards because the information isn’t secured and can be read by people during the transportation from the sender to the receiver
* Post method is similar with envelopes because the information being sent is secured and covered by an outer layer, can only be read by the receiver

In the onsubmit attribute, if we write "return checkForm()" and this function returns false, what will happen when a user tries to submit the form?

* The data inside the form will not be submitted so the data won’t reach the server, saving time from the server

## Activity 4: Form Elements

### Input elements:

What do each of the following 3 input attributes do? Why are they important:

|  |
| --- |
| <input type="..." name="..." value="..." /> |

* Type: Specifying the input type of the element, e.g. button, checkbox, data, number, etc. should be the correct type for easier input parsing and easier for users to input information
* Name: Specifying the name of the input, this is for other functions to reference to
* Value: The initial value of the input, it is used differently for different types of input

What is the difference between text**,** password and hidden inputs? For example:

|  |
| --- |
| Name: <input type="text" name="name" value="" /><br/>  Password: <input type="password" name="password" value="" /><br/>  Rated: <input type="hidden" name="client-rating" value="2 stars" /> |

* Text: the input box will be a single-line text field
* Password: the input box will also be a single-line text field, but all the text entered will be blacked out
* Hidden: the input box will be hidden away and can’t be entered by the user

What is the difference between radio and checkbox inputs AND why are there square brackets in the checkboxes' name attributes?

|  |
| --- |
| <!-- Radios -->  <input type="radio" name="gender" value="male" /> Male<br/>  <input type="radio" name="gender" value="female" /> Female<br/>  <input type="radio" name="gender" value="other" /> Other  <!-- Checkboxes -->  <input type="checkbox" name="seeks[]" value="male" /> Male<br/>  <input type="checkbox" name="seeks[]" value="female"/> Female<br/>  <input type="checkbox" name="seeks[]" value="other"/> Other |

* Radio: the inputs are circle select buttons but only one option can be selected
* Checkbox: the inputs are square select boxes and many boxes can be selected at the same time
* The square brackets in checkbox’s name attributes help store the submitted values into an array or a list, which can then be use by other functions

What is the difference between submit, reset and button inputs? For example:

|  |
| --- |
| <input type="submit" value="Buy Now"/>  <input type="reset" value="Clear Form"/>  <input type="button" value="Calculate Something"/> |

* Submit: this will display a clickable button which the user can click and submit all the form data to another processing script
* Reset: this will display clickable button which the user can click to reset all the form fields to their initial values
* Button: this will display a clickable button which the user can click to do various functions or run scripts defined by the web developer

### 

### Textarea, Select & Option elements:

These controls have more features, have a different format and are paired tags.

The textarea control is a multi-line text field and has extra attributes to control height and width:

|  |
| --- |
| <textarea name="..." rows="..." cols="..."> ... </textarea> |

The select and option elements are combined to create **Drop down** boxes and **Combo lists**:

|  |
| --- |
| <!-- This will be rendered as a drop down list -->  <select name="...">  <option value="mr">Mr.</option>  <option value="mrs">Mrs.</option>  <option value="miss">Miss</option>  <option value="ms">Ms.</option>  <option value="dr">Dr.</option>  </select>  <!-- This will be a combo box -->  <select name="..." size='...'>  <option value="mr">Mr.</option>  <option value="mrs">Mrs.</option>  <option value="miss">Miss</option>  <option value="ms">Ms.</option>  <option value="dr">Dr.</option>  </select> |

The multiple attribute allows the user to select more than one option. The list loses "radio-like" behaviour and gains "checkbox-like" behavior.

|  |
| --- |
| <select name="..." multiple > ... </select> |

Code up these inputs and draw what each looks like.

* Textarea:

Simple   
multiline  
textarea

* Dropdown list:

Mr.

Mrs.  
Miss  
Ms.  
Dr.

Mr.  
Mrs.

* Combo box (size=2):
* Checkbox-like <select>:

Mr.  
Mrs.  
Miss  
Ms.

### 

### Labels:

For semantic reasons, it is useful to link a label with an input, for example:

|  |
| --- |
| <!-- Text, but no label element -->  <p>Name <input type='text' name='name' id= 'name' /></p>  <!-- Text inside a label element -->  <p><label>Name</label> <input type='text' name='name' id='name' /></p> |

In addition, we can make labels "clickable" by linking the label's for attribute to the id of an input:

|  |
| --- |
| <p><label for='name'>Name</label> <input id='name' type='text' name='name' /></p> |

Useful for text based inputs and very useful for small "hard to click" radio and checkbox inputs.

|  |
| --- |
| <!-- Gender is a "non-clickable" label -->  <p><label>Gender</label>  <!-- But labels below are clickable and increase the usability of the form -->  <input type='radio' name='gender' id='male' /> <label for='male' >Male</label>  <input type='radio' name='gender' id='female' /> <label for='female'>Female</label>  <input type='radio' name='gender' id='other' /> <label for='other' >Other</label>  </p> |

*If there is time, teaching staff will demonstrate some of the new HTML5 input elements and attributes.*