

Forms & Events

Tell Me About Forms...

The background of the slide features a series of horizontal, wavy lines in various shades of blue and white, creating a sense of depth and movement. The lines are more pronounced at the bottom and fade towards the top.

What's a Form?

- Forms are constructed with HTML
- They can be manipulated with Javascript
 - Validating emails, passwords, zip codes, etc
 - Adding/removing placeholder text
 - Exposing chunks of the form as it is completed

Forms Are Good For...

- Adding an interactive element to websites so that users can submit information to you
- Examples of Forms in our lives:
 - Sign up forms on Facebook, Groupon, Google, etc
 - Blog posts
 - Forum posts
 - Submitting your purchase to Amazon

Form Elements

- Input
- Checkboxes
- Radio buttons
- Text Areas
- Drop down menus
- Password
- Submit buttons

Exercise: Making a Form

- use `materials/forms/exercises/dolphin_ride.html`
- add some fields:
 - height (`type="text"`)
 - preferred dolphin type (`type="radio"` – the options should be: pink, grey, green)
 - password (`type="password"`)

Using jQuery To Access Form Elements

- jQuery gives us some methods that make manipulating forms easy:
 - `.val()` method returns the current value of a form element as well as allowing us to set the element's value.
 - `.click()` allows us to click checkboxes & radio buttons
 - `.attr()` allows us to access and set attributes of elements.

Exercise: Using jQuery To Access Form Elements

- Using the console, play around with the following:
 - `$('#name').val("Greg")`
 - `$('#name').val()`
 - `$('#lang-shark').is(':checked')`
 - `$('#lang-shark').attr('checked')`
 - `$('#lang-shark').attr('checked', true);`
 - `$('#lang-whale').click();`
 - `$('#no-swim').click();`



Events

What Is An Event?

The background of the slide features a series of horizontal, wavy lines in various shades of blue and white, creating a sense of depth and movement. The lines are more pronounced at the bottom and fade towards the top.

What Is An Event

- Actions in the web browser that JavaScript understands.

Examples of Events

- A mouse click
- A web page or an image loading
- Mousing over a hot spot on the web page
- Selecting an input field in an HTML form
- Submitting an HTML form
- A keystroke

Effect Of Events

- Events are normally attached to functions which cause something to happen when the event takes place. The function is not executed until the event happens in the browser.
- The function that is executed is called an *event handler*

How Events Get Triggered

- Every element on a webpage (body tag, paragraph tags, div tags, etc) can have events attached to them. When the event happens on the particular element, a function is triggered.



Attaching Events

The Old Way

...

```
<script>
```

```
    function sayHello() {
```

```
        alert("Hello");
```

```
    };
```

```
</script>
```

...

```
<body>
```

```
    <p onclick="sayHello()">Say Hello</p>
```

...

The Lazy Way

```
<script>
  function sayHello() {
    alert("Hello");
  };

  window.onload = function() {
    var element = document.getElementById('hello');
    element.onclick = sayHello;
  }
</script>
```

A Better Way

```
<script>
  function sayHello() {
    alert("Hello");
  };

  window.onload = function() {
    var element = document.getElementById('hello');
    element.addEventListener( 'click', sayHello,
    false);
  }
</script>
```

The IE Way

```
<script>
  function sayHello() {
    alert("Hello");
  };

  window.onload = function() {
    var element = document.getElementById('hello');
    element.attachEvent( 'onclick', sayHello);
  }
</script>
```

The jQuery Way

```
<script>
    function sayHello() {
        var element = $('#hello');
        element.bind( 'click', function() {
            alert("Hello");
        });
    };

    $(document).ready(sayHello);
</script>
```

Another jQuery Way

```
<script>
    var element = $('#hello');
    element.click(function() {
        alert("Hello");
    });
};

$(document).ready(sayHello);
</script>
```

Example

- [materials/events/examples/div_alert.html](#)
- .bind()
- .focus()
- .click()
- .mouseover()
- .hover()

Exercise

- `materials/events/exercises/paragraph_alert.html`
- write several different event handlers that alert the text within the paragraph when a paragraph is clicked on, along with the event that triggered it.
- Documentation for the jQuery event object: <http://api.jquery.com/category/events/event-object/>

Exercise

- use `materials/events/exercises/string_objects.html`
- Modify the code so that there is a form with four input boxes each with different ids: “split” should be the id for the first input, “substr” for the second, “charAt” for the third, “slice” for the forth.
- Instead of using alerts, use events that trigger when the different ids are “onfocus” or “click. When the events are triggered, use jQuery to change the value of the inputs to the values given by the string object operations.
- Hint: `$('#split').val('blah'.split(''))`

Exercise

- materials/events/exercises/table_stripes.html
- A little explanation: `tr:even` & `addClass`
- This html document has an html table already in it. Your task is to add & then remove a class when a user mouses over then mouses off of a row. The class to add is `“.highlight”`.

Exercise: Help Text

- `materials/events/exercises/input_help_text.html`
- Add a focus event that removes the help text.
- Add a blur event that puts the help text back in if the user hasn't typed anything