

# Developing Mobile Websites

**HTML5: Mobile Specific**

## Lesson 1, Activity 2: New HTML5 Features

### Forms

Our mobile sites will need to accept user input - registration forms, comment forms, and the like to ask users to send us, via form fields, some information. HTML5 makes available a set of new form input types and attributes that enhance the user experience. Keep in mind that adoption for some of these new features is spotty at best; we'll cover the better implemented options.

There are 13 new input types:

1. search
2. tel
3. url
4. email
5. datetime
6. date
7. month
8. week
9. time
10. datetime-local
11. number
12. range
13. color

We'll concentrate on a few of the more useful types for mobile.

### Telephone

The `tel` type input expects a phone number. Many phones will present

a numeric keyboard:



## Dates

The date type of input allows the user to enter a date with no time zone. Some smartphone browsers present a date picker when a field of type date receives focus. iPhones with iOS version 5 or later, for instance, offer a date picker - earlier versions do not.

AT&T 10:16 PM 61 %

**Date of Birth**

Apr 10, 2012 ▼

Submit

Previous Next AutoFill Clear Done

February	8	2010
March	9	2011
April	10	2012
May	11	2013
June	12	2014

## Email

Fields of type `email` expect a valid email address. Many phones will show a contextually appropriate set of characters for input; the iPhone, for instance, displays the "@" character:



## Web Address

Fields of type `url` are for entering full Web site address. This field type excludes invalid characters for a URL, such as a space. Many phones will show a contextually appropriate set of characters for input; the iPhone, for instance, displays the ".com" key:

A screenshot of an iPhone screen displaying a web form. The status bar at the top shows 'AT&T' with signal bars, a Wi-Fi icon, the time '10:16 PM', a Bluetooth icon, and a battery level of '61%'. The form has a light orange background. The first field is labeled 'Website URL' and has a white text input box with a blue cursor. The second field is labeled 'Date of Birth' and has a white date picker. Below the form is a navigation bar with four buttons: 'Previous', 'Next', 'AutoFill', and 'Done'. At the bottom is a QWERTY keyboard with a 'Go' button.

## New Field Attributes

The `placeholder` attribute is among the more useful new HTML5 field attributes: it allows us to specify an example value for the field, a value that disappears when the user begins to enter content in the field.

The `required` attribute is very useful - but, sadly, not well supported among mobile browsers: the automatic client-side HTML5 form validation that works on desktop browsers is lacking on their mobile counterparts.

## A Jazz Calendar Registration Form

Let's create a simple registration form for the *Jazz Calendar* site. Open <HTML5MobileSpecific/Demos/forms.html> in a mobile browser - this is page from which the above screen shots were taken.

### Code Sample:

---

#### <HTML5MobileSpecific/Demos/forms.html>

```
----- CODE OMITTED -----
```

Name

Phone

Email

Website URL

Date of Birth

```
---- CODE OMITTED ----
```

We use a placeholder value for the first ("name") field, and ask the user for information of type `tel`, `email`, `url`, and `date`.



## Lesson 1, Activity 4: **Local Storage**

Like cookies, HTML5 local storage offers a mechanism for storing key/value pairs locally, within the client browser, even after you navigate away from the page, close the browser, etc. Unlike cookies, these data are never sent back to the server.

In addition, HTML5 offers session storage - similar to local storage, session storage persists between page requests, but does not persist after the browser is closed.

We use HTML5 Web storage to store key/value pairs. Browsers typically limit the amount of client-side storage space allocated to a single domain to 5 megabytes and throw a `QUOTA_EXCEEDED_ERR` exception if you try to store more than that.

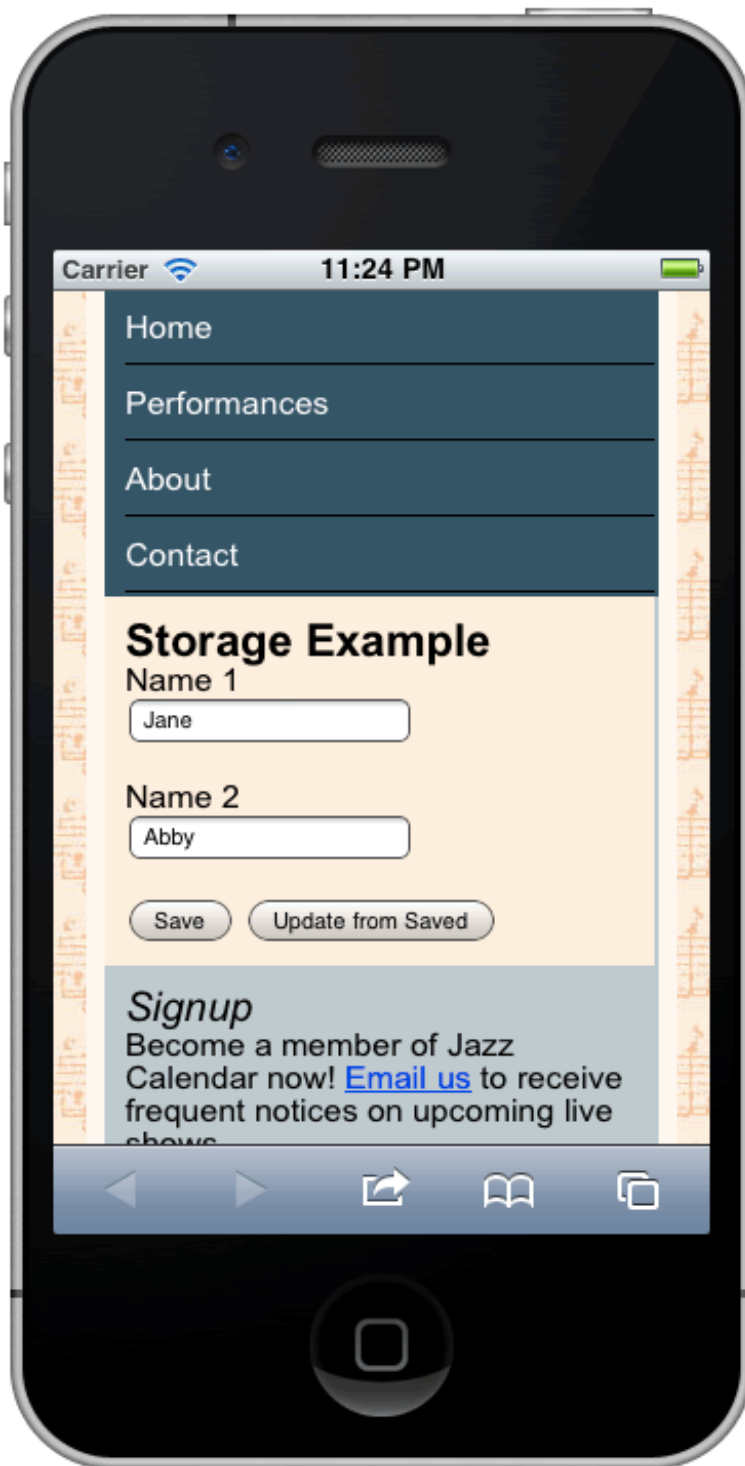
Here are the JavaScript methods and properties for working with Web storage:

**Web Storage Methods & Properties**

Method/Property	Description
length	Holds the number of key/value pairs
setItem(key, value)	Creates or updates a key/value pair
getItem(key)	Gets the value of the specified key
key(n)	Returns the nth key. Useful for iterating through key/value pairs
removeItem(key)	Removes the key/value pair for the given key
clear()	Removes all key/value pairs

## **Local Storage Example**

Let's look at a simple page on the *Jazz Calendar* site - open the file <HTML5MobileSpecific/Demos/storage.html> in your browser:



Clicking the **Save** button saves the currently entered names to the local browser; clicking **Update from Saved** restores the saved names to their respective fields.

### Code Sample:

## HTML5MobileSpecific/Demos/storage.html

----- C O D E    O M I T T E D -----

----- C O D E    O M I T T E D -----

Name 1

Name 2

----- C O D E    O M I T T E D -----

When the **Save** button is clicked, the `save ()` function is called: the current values from the `name1` and `name2` fields are saved (`localStorage.setItem`) to keys `name1` and `name2`, respectively.

Function `set ()` - called when the **Update from Saved** button is clicked - sets the values of the two fields from the last saved values.

Note that each function checks first, via function `html5_storage_support ()`, to see if the current browser supports Web storage.

## Lesson 1, Activity 6: Registration Form for *Pickup Soccer*

Duration: 20 to 30 minutes.

In this exercise, you will create a registration form for the *Pickup Soccer* site:

Carrier 12:16 AM

# Register Now

Name

Phone

Email

Website URL

Date of Birth

1. Open the file HTML5MobileSpecific/Exercises/soccer/index.html in your file editor and in your browser.
2. Add a form and input fields for name, phone, email, Web site URL, and date of birth - choose the appropriate input type for each.
3. Allow the user to save his or her values in the form using Web storage: **Save** saves the entered data, and **Restore from Saved** repopulates the fields with saved data.
4. Check your work with a mobile browser.

## Solution:

---

### HTML5MobileSpecific/Solutions/soccer/index.html

---- C O D E    O M I T T E D ----

```
<script type="text/javascript">
  function html5_storage_support() {
    try {
      return 'localStorage' in window && window['localStorage'] !== null;
    } catch (e) {
      return false;
    }
  }

  function saveit() {
    if (html5_storage_support()) {
      var name = document.getElementById("name").value;
      var phone = document.getElementById("phone").value;
      var email = document.getElementById("email").value;
      var website = document.getElementById("website").value;
      var DOB = document.getElementById("DOB").value;
      localStorage.setItem('name', name);
      localStorage.setItem('phone', phone);
      localStorage.setItem('email', email);
      localStorage.setItem('website', website);
      localStorage.setItem('DOB', DOB);
    }
  }

  function restoreit() {
    if (html5_storage_support()) {
      document.getElementById('name').value = localStorage.getItem('name');
      document.getElementById('phone').value = localStorage.getItem('phone');
```

```

        document.getElementById('email').value = localStorage.getItem('email');
        document.getElementById('website').value = localStorage.getItem('website');
        document.getElementById('DOB').value = localStorage.getItem('DOB');
    }
}
</script>
---- C O D E   O M I T T E D ----

<form action="index.html" method="post">
  <label for="name">Name</label><br>
  <input type="text" name="name" id="name" placeholder="Jane Doe"><br>
  <br>
  <label for="phone">Phone</label><br>
  <input type="tel" name="phone" id="phone"><br>
  <br>
  <label for="email">Email</label><br>
  <input type="email" name="email" id="email"><br>
  <br>
  <label for="website">Website URL</label><br>
  <input type="url" name="website" id="website"><br>
  <br>
  <label for="DOB">Date of Birth</label><br>
  <input type="date" name="DOB" id="DOB"><br>
  <br>
  <input type="submit">
</form>
<button onclick="saveit()">Save</button>
<button onclick="restoreit()">Restore from Saved</button>

---- C O D E   O M I T T E D ----

```

We add fields of type `text`, `tel`, `email`, `url`, and `date`. We use Web storage to save the current fields values when the user clicks the **Save** button, and restore them when the user click **Restore from Saved**.