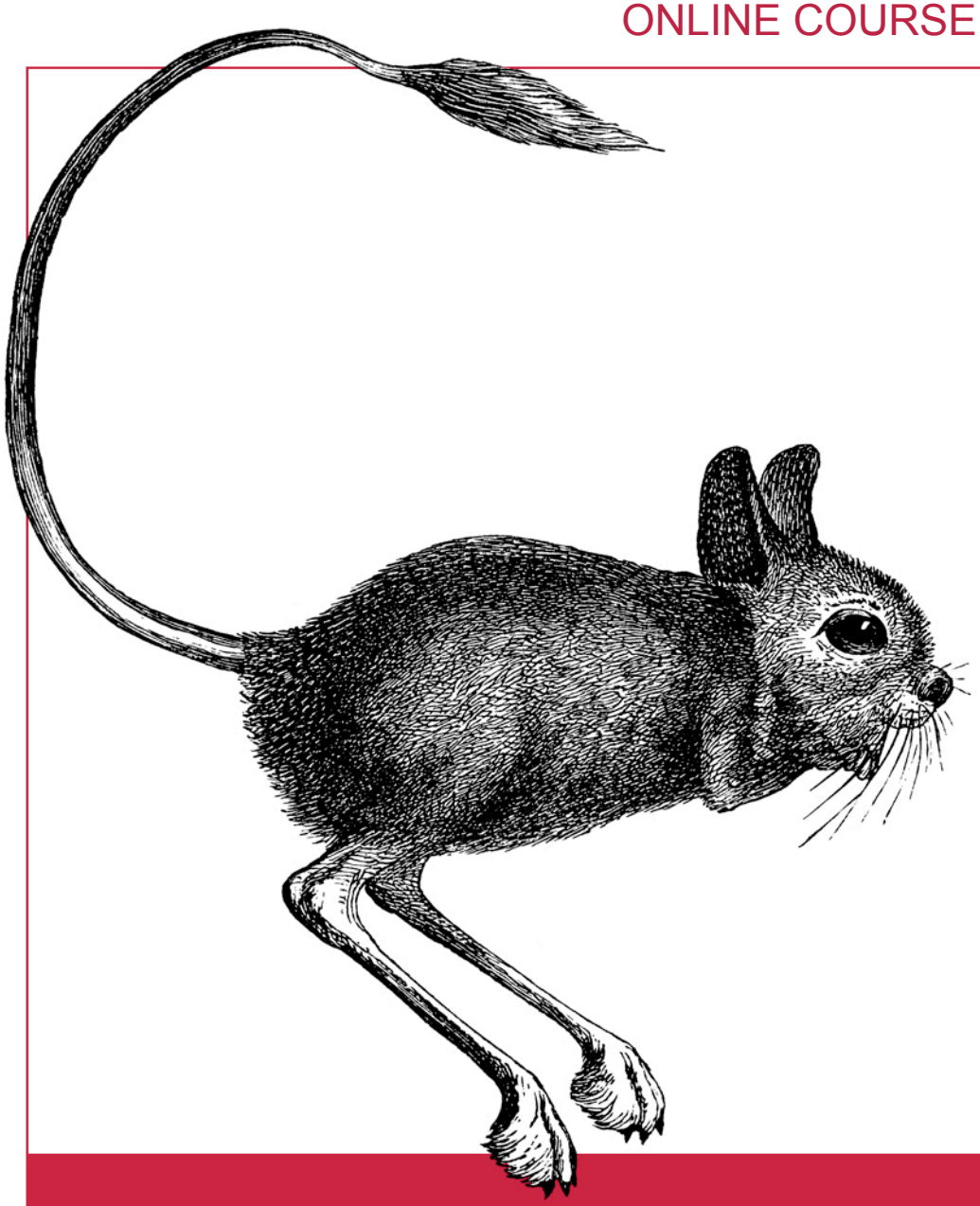
ONLINE COURSE TIP SHEET



HTML5 Mobile Web Development

Jake Carter

O'REILLY®

About the instructor

Jake Carter is a web and software developer at RogueSheep, an award-winning Seattle-based company dedicated to creating top-tier applications that focus on design and usability. While he currently develops for Apple's iOS devices, Jake has a personal and professional enthusiasm for web standards. Staying on the cutting edge of technology is more than just a job for Jake—it's a passion.

About O'Reilly Media, Inc.

O'Reilly Media spreads the knowledge of innovators through its books, online services, magazines, and conferences. Since 1978, O'Reilly has been a chronicler and catalyst of leading-edge development, homing in on the technology trends that really matter and spurring their adoption by amplifying "faint signals" from the alpha geeks who are creating the future. An active participant in the technology community, the company has a long history of advocacy, meme-making, and evangelism.

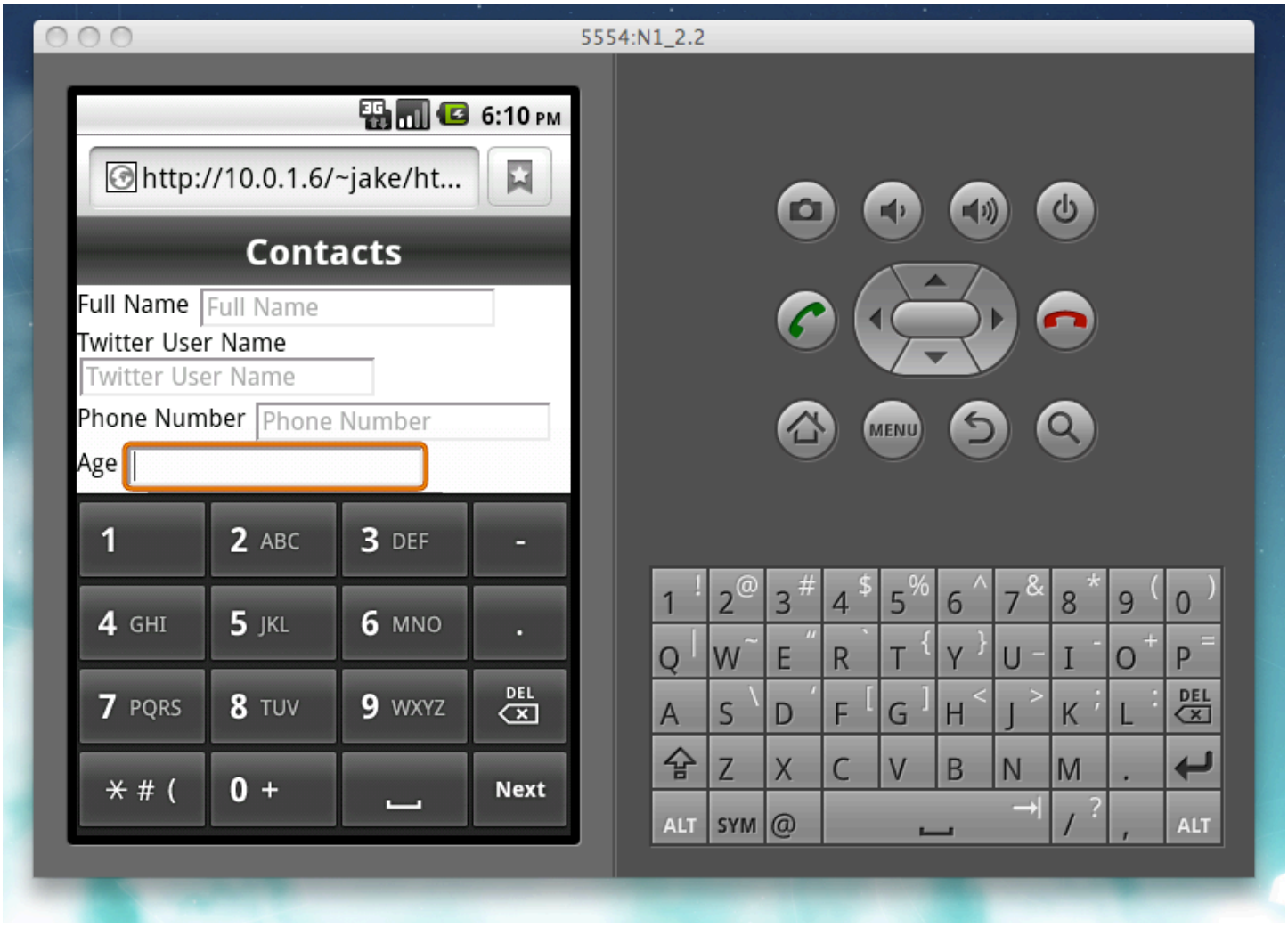
Week 6 Tip Sheet

HTML5 Input Types

The new HTML5 input types can be used to help the browser display an appropriate keyboard. Below is a list of a few of the ones that are currently implemented.

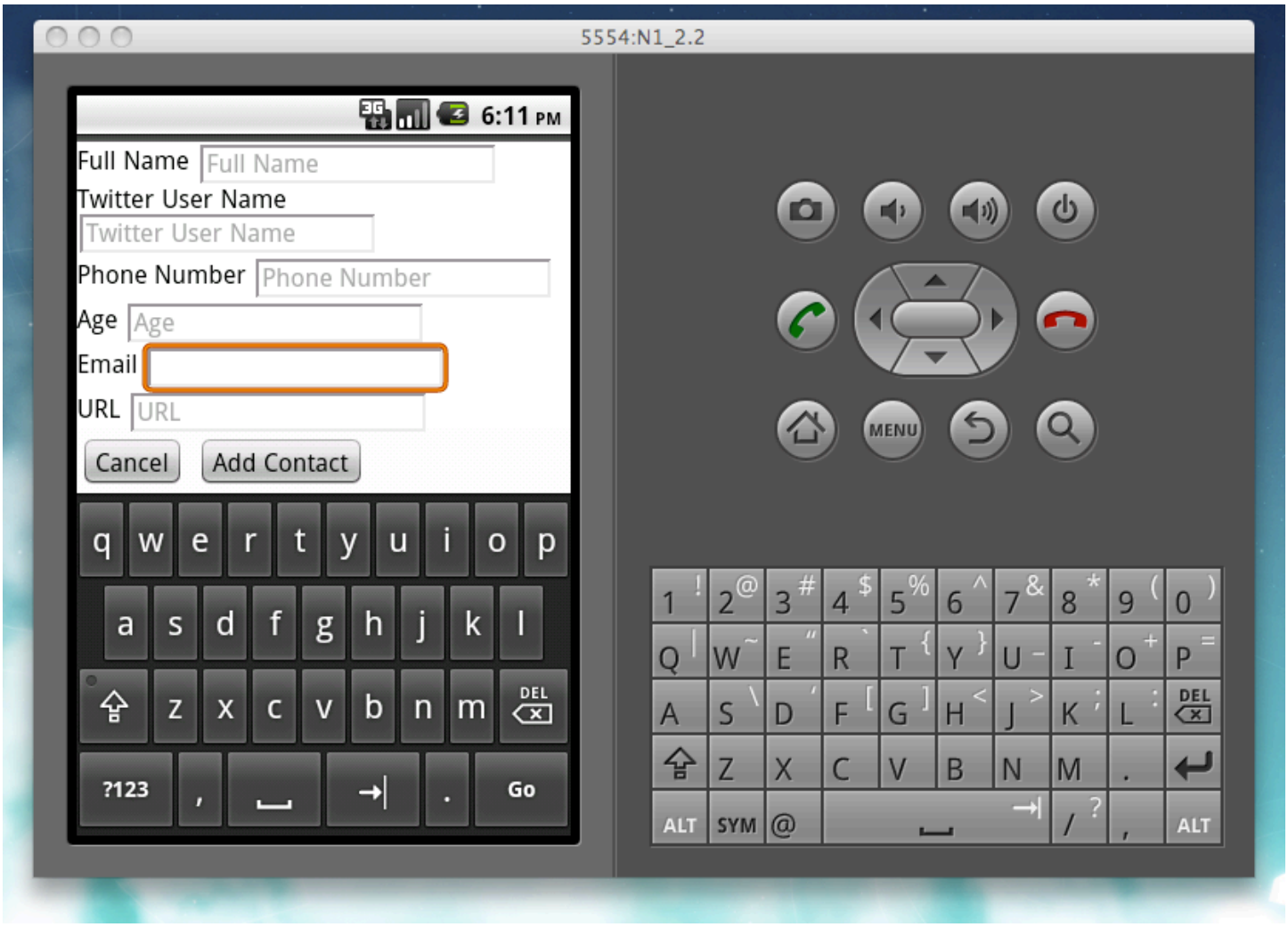
number


Carrier
6:09 PM
Full Name
Full Name
Twitter User Name
Twitter User Name
Phone Number
Phone Number
Age
Email
Email
URL
URL
Cancel
Add Contact
Previous
Next
AutoFill
Done
1 2 3 4 5 6 7 8 9 0
- / : ; ( ) $ & @ "
#+= . , ? ! '
ABC
space
Go


5554:N1_2.2
6:10 PM
http://10.0.1.6/~jake/ht...
Contacts
Full Name
Full Name
Twitter User Name
Twitter User Name
Phone Number
Phone Number
Age
1
2 ABC
3 DEF
-
4 GHI
5 JKL
6 MNO
.
7 PQRS
8 TUV
9 WXYZ
DEL
* # (
0 +
Next
MENU
ALT
SYM
@
/
,
ALT

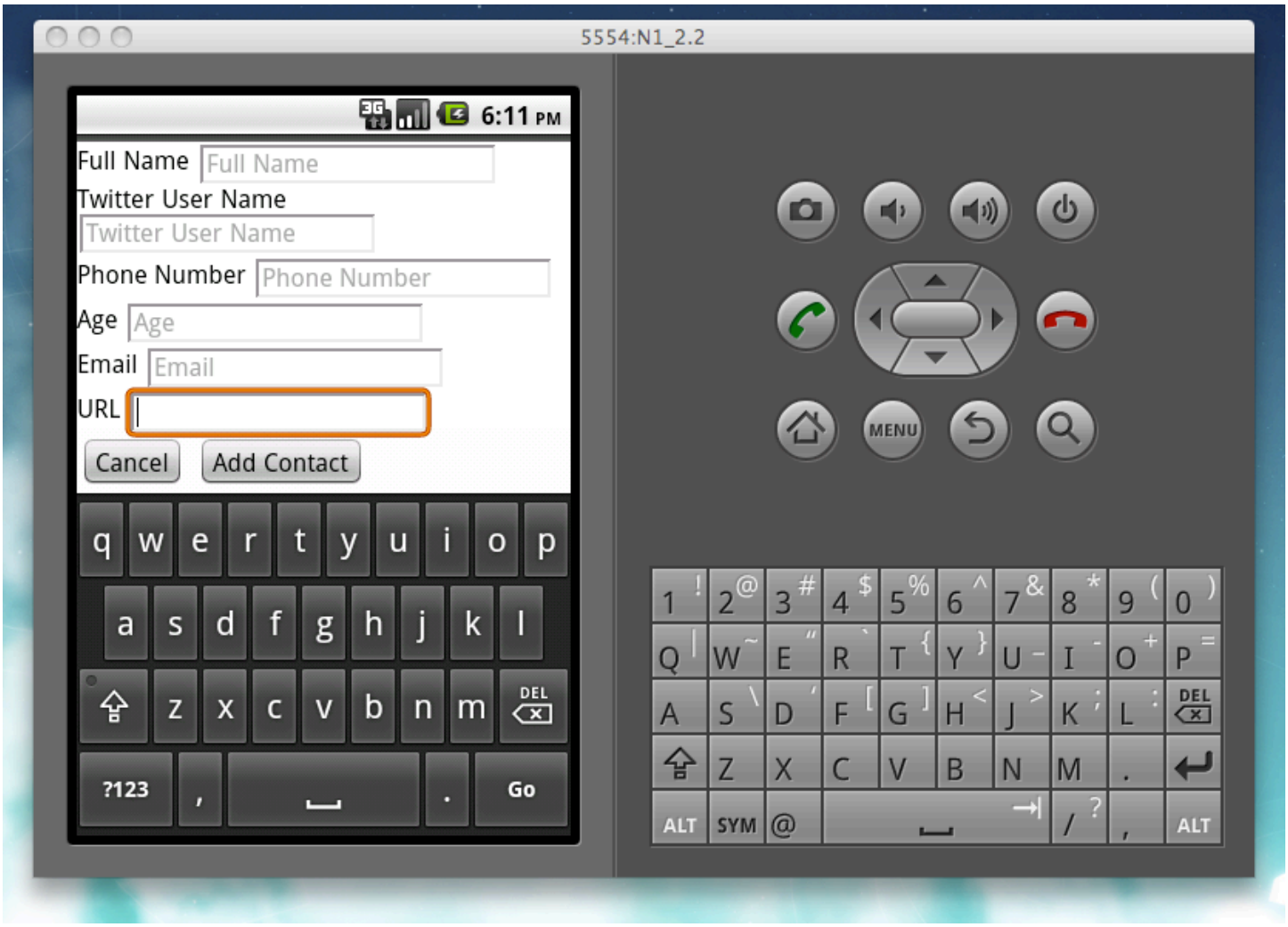
email


Carrier
6:09 PM
Full Name
Twitter User Name
Twitter User Name
Phone Number
Phone Number
Age
Age
Email
URL
URL
Cancel
Add Contact
Previous
Next
AutoFill
Done
Q W E R T Y U I O P
A S D F G H J K L
Z X C V B N M
_123
space
@
.
Go


5554:N1_2.2
6:11 PM
Full Name
Full Name
Twitter User Name
Twitter User Name
Phone Number
Phone Number
Age
Age
Email
URL
URL
Cancel
Add Contact
MENU

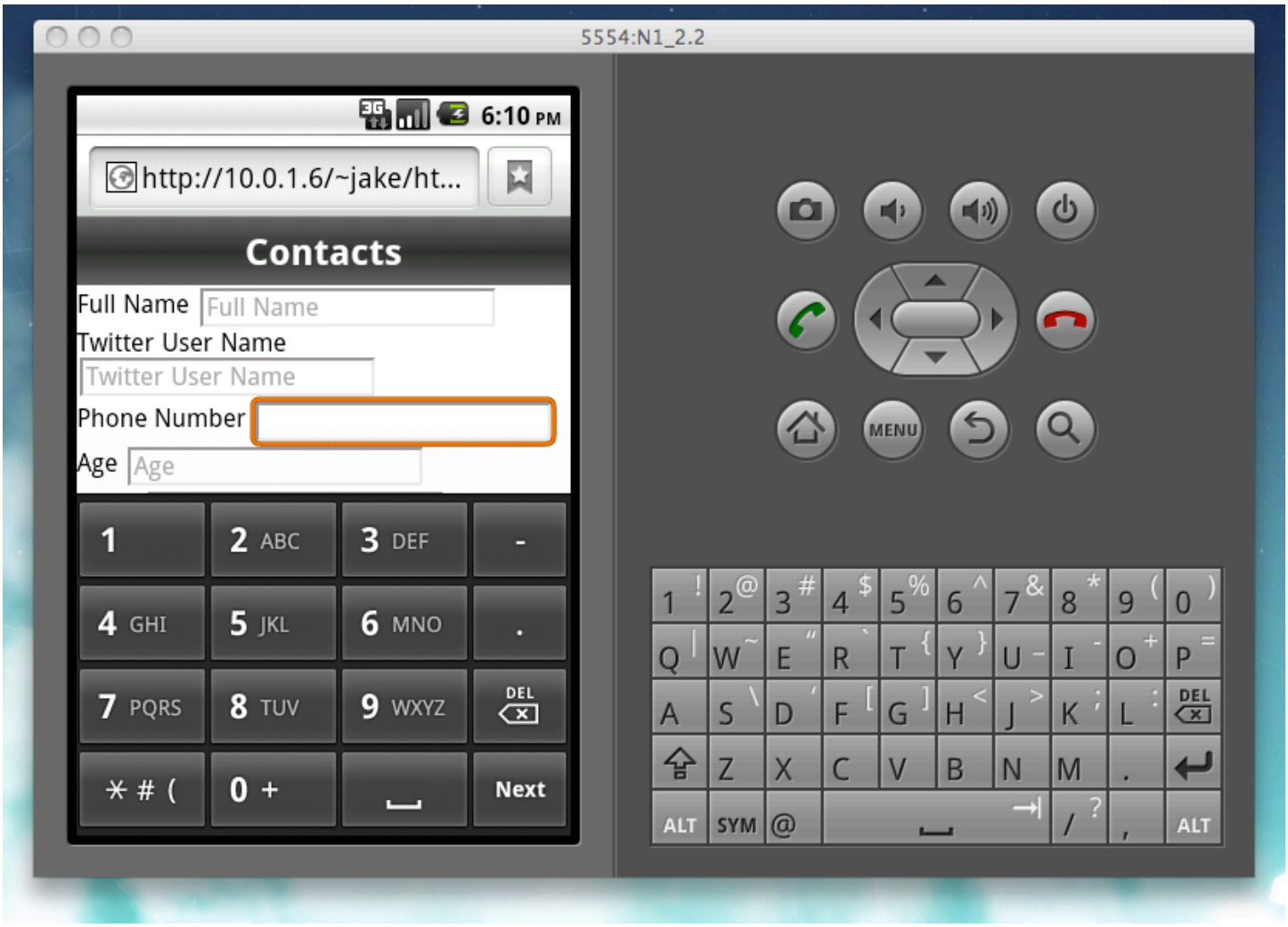
url


Carrier
6:10 PM
Twitter User Name
Phone Number
Phone Number
Age
Age
Email
Email
URL
Cancel
Add Contact
Previous
Next
AutoFill
Done
Q W E R T Y U I O P
A S D F G H J K L
Z X C V B N M
@123
.
/
.com
Go


5554:N1_2.2
6:11 PM
Full Name
Full Name
Twitter User Name
Twitter User Name
Phone Number
Phone Number
Age
Age
Email
Email
URL
Cancel
Add Contact
MENU

tel


Carrier
6:09 PM
Contacts
Full Name
Full Name
Twitter User Name
Twitter User Name
Phone Number
Age
Age
Email
Email
URL
URL
Cancel
Add Contact
Previous
Next
AutoFill
Done
1
2 ABC
3 DEF
4 GHI
5 JKL
6 MNO
7 PQRS
8 TUV
9 WXYZ
+ * #
0



Syntax

```
<input type='input-type' />
```

Related Links

http://dev.w3.org/html5/spec/Overview.html#attr-input-type
http://developer.apple.com/safari/library/codinghowtos/mobile/userexperience/index.html#GENERAL-CONTROL_WHICH_KEYBOARD_IS_DISPLAYED_WHEN_A_USER_TOUCHES_A_TEXT_FIELD
http://diveintohtml5.org/forms.html

Placeholder Text

Placeholder text is default text in an input element that gives the users some information about the input that is expected. When the input element gains focus, the text is removed and the user is allowed to type.

Syntax

```
<input placeholder='placeholder text' />
```

Related Links

http://dev.w3.org/html5/spec/Overview.html#attr-input-placeholder

Validation

The new form elements now also have a validity state that we can check. This is great, because now you don't have to write the code to check to see if the fields are valid. Here's how you use it:

```
if (document.form.checkValidity()) {
     // Form is valid.
}
else {
     // Form is NOT valid.
}
```

Validation can also be styled via CSS using the :valid or :invalid pseudo classes like so:

```
input:valid {
     /* Style valid inputs. */
     border: 1px solid #00ff00; /* Green Border */
}

input:invalid {
     /* Style invalid inputs. */
     border: 1px solid #ff0000; /* Red Border */
}
```

Related Links

http://www.the-art-of-web.com/html/html5-form-validation/

CSS3 Animations

CSS3 Transitions and Animations are very similar. Transitions let you specify one property to animate. Animations let you group multiple properties and give you even more control over how they are animated.

Keyframes

To create CSS3 Animations you need to define it's keyframes. These are the different states each of the properties will transition between.

```
@-webkit-keyframes 'animation name' {
     0% {
          left: 0px;
          background-color: #ff0000;
     }
     50% {
          left: 500px;
          background-color: #00ff00;
     }
     100% {
          left: 100px;
          background-color: #0000ff;
     }
}
```

Here we have defined our animation called 'animation name'. Apply it to a style use the following:

```
div.animate {
     /* Name of the animation */
     -webkit-animation-name: 'animation name';

     /* How many times you want the animation to loop  */
     -webkit-animation-iteration-count: 5;

     /* How long each iteration should last */
     -webkit-animation-duration: 1s;
}
```

Now once this css rule 'div.animate' matches an element, it will begin animating.

CSS3 Animation Events
There are several animation events that we can listen to via JavaScript.

animationstart
This gets fired when the animation starts.

animationend
This gets fired when the animation is completely done.

animationiteration
This gets fired each time an iteration ends. (Does not get fired for animations with an iteration count of one.)

You an attach one of these events just like we do with all other event listeners:

```
// Get a reference to the element that will be animated.
var box = document.getElementById('box');

// Add the event listener for the event you want to handle.
// Here we are handling the animationEnd event.
box.addEventListener('webkitAnimationEnd', function(event) { ... });
```

Keep in mind that until CSS3 is finalized, you will need to add the browser prefix to the event names. This is why mine in the example above has 'webkit' tacked onto the front.

Related Links
http://www.w3.org/TR/css3-animations

CSS3 Transition Events
We did not go over this in class, but you can also be notified when a CSS3 Transition has finished by listening for the 'webkitTransitionEnd' event.

To do so, you will need the DOM Element that will transition.

```
var box = document.getElementById('box');
```

Next you will add the event listener.

```
box.addEventListener('webkitTransitionEnd', function(event) { ... });
```

Related Links
http://developer.apple.com/safari/library/documentation/AudioVideo/Reference/WebKitTransitionEventClassReference/WebKitTransitionEvent/WebKitTransitionEvent.html#//apple_ref/javascript/cl/WebKitTransitionEvent