Your First Android Web Application

by

Alfred Samman, PhD

Your First Android Web Application

Alfred Samman

This book is for sale at http://leanpub.com/androidjavascriptapp

This version was published on 2014-08-10

This is a Leanpub book. Leanpub empowers authors and publishers with the Lean Publishing process. Lean Publishing is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

Contents

Preface

With the recent proliferation of apps I felt that there was a need for a book to teach the average Jane how to develop their own apps. The web has been with us for a while now and many of the technologies and terminologies are familiar to the average user. We are familiar with what a webpage is, what a link is, what a browser is etc. Building on this familiarity, my aim is to show that with relative ease anyone can start developing their own app.

How this book is organized

This book is divided into self contained chapters of increasing difficulty. You can jump between chapters and pick topics that interest you. Each chapter has accompanying code that can be downloaded and run, or read from within the text. Another way to read this book is to read it systematically from the first chapter to the last.

Who should read this book

This book can be read by a complete novice. It can also be read by anyone with basic web development knowledge.

The book begins with a gentle introduction to the main web technologies, html, css and Javascript. We then move on to learn how to set up a development environment for developing our apps. You will then create your first simple app. Subsequent chapters touch on the DOM, touch events, saving data, swipe action and finally how to publish your app.

Source Code

The code for each chapter can be downloaded from Github

https://github.com/budu3/the-book/tree/master/code

Introduction

The Apple App Store made its debut on July 2008 (http://9to5mac.com/2011/10/21/jobs-original-vision-for-the-iphone-no-third-party-native-apps/) on the iPhone 2 and has not looked back since. To say the App Store has been a success is an understatement. As of the date of writing this book the Apple App S tore has a record 800 000 apps available for download. The concept of an "App" and an "App Store" has now been adopted by a plethora of platforms including, Google (Google Play), Android (Amazon Appstore for Android), Microsoft (Windows Store), Firefox (Firefox Marketplace), Chrome (Chrome Web Store) and a host of other platforms. It seems like every device today has an App Store and for every task out there, there's an App for that.

In this book we will learn how to create Web apps. The aim is to show a novice how to develop simple Web Apps for mobile devices. Before one can develop Web Apps one needs to know some basics web technologies. Your App will consist of 3 main components, *HTML5*, *CSS* and *JavaScript*. *HTML5* is used for presentation, *CSS* is used for styling and *JavaScript* is used to code the actual functionality of your app.

HTML5

HTML5 (HyperText Markup Language 5) will be used to create all the visual elements of your app.

DOM

The DOM (Document Object Model) is an object representation of the elements that make up your app. It is organized as a tree. Every web app consists of a DOM and the functionality of your your app usually involves manipulating the DOM.

We will discuss the aforementioned technologies later on in this book.

Native apps versus Web apps

Native apps are apps written specifically for the platform on which they run on. They take advantage of the platform's native user interface elements. Web apps on the other hand are developed using web technologies, can be viewed in a browser and can usually work on any platform with a compliant browser.

Detecting Swipe Actions

In this chapter we will add a swipe effect to our app so that when you swipe an item it disappears from the screen.

We are going to learn how to perform swipe actions. Performing a swipe action involves three major tasks. Detecting the point at with the user begins swiping, detecting the point at which the user stops swiping and then transitioning or moving the html element between these two positions.

Detecting the start position of your touch

To detect the start position of a touch, we attach a function to the touchstart event. We then use `changeTouches[0].pageX` to get the starting point of the touch as shown below.

```
element.addEventListener('touchstart',function(e){
    startx = parseInt(e.changedTouches[0].pageX);
});
```

Note that `TouchEvent.changedTouches` will return a list of touch points. `Touch.pageX` will return the x coordinate of a particular touch point

Transitioning a swiped element

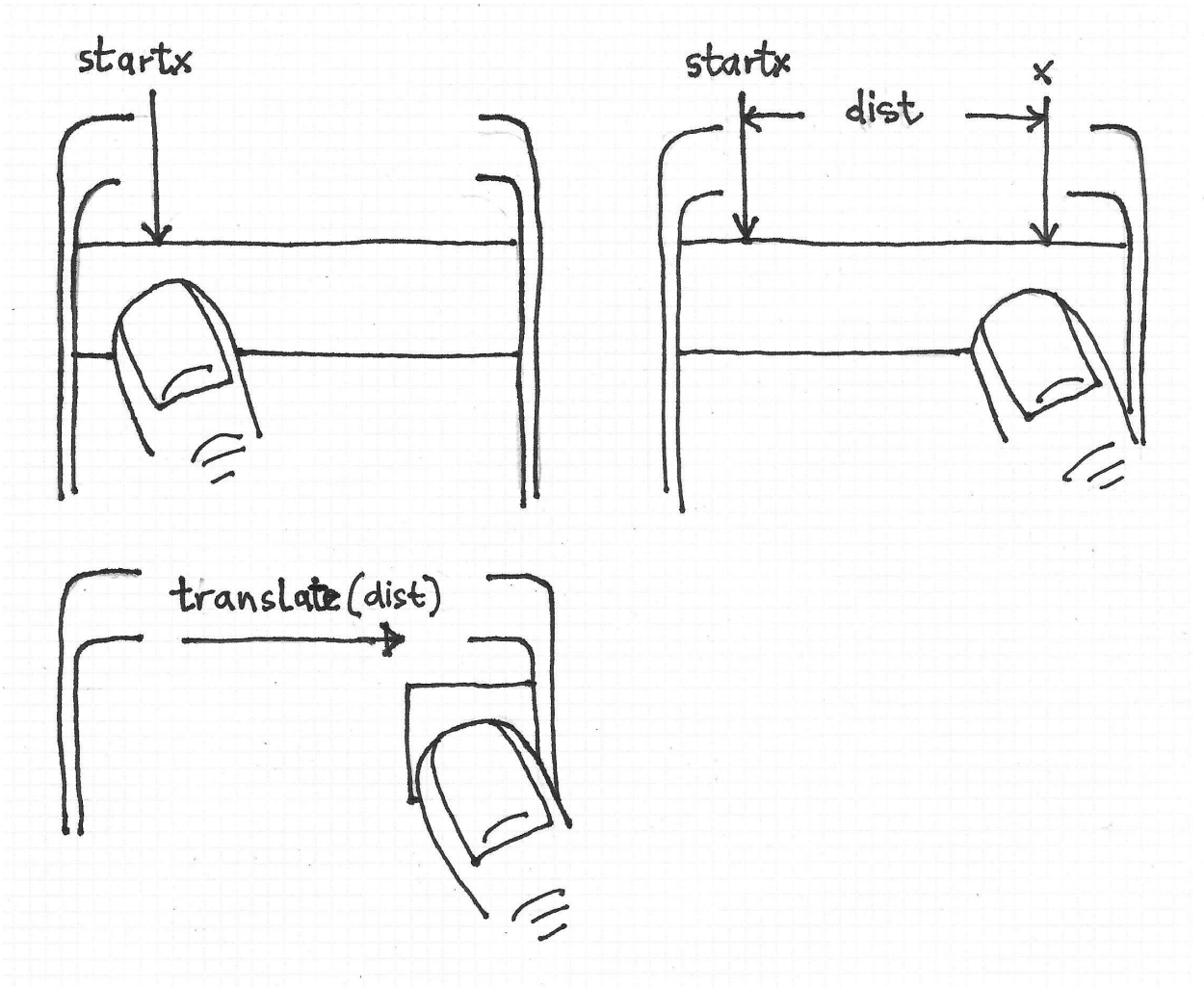
Now, to produce the swiping effect we translate the touched list element the distance that the finger has moved along the x axis. This is done by manipulating the list element's `webkitTransform` style attribute as shown below.

```
element.style.webkitTransform = 'translate(' + dist + 'px, 0)';
```

Detecting when the swipe ends

Finally, we need to detect when the swipe action is over. We do this by simply listening to the touchend event. Whatever action that needs to be performed when the touch ends can be attached to the event as shown.

```
element.addEventListener('touchend', function(e){
    ...
})
```



image

The code listing below adds swipe functionality to your app.

```
<!DOCTYPE html>
<html>
<head>
<title>todo list</title>
<link href="css/main.css" rel="stylesheet" type="text/css">
</head>
<body>

<div class="container">
        <ul>

        </ul>
                <input type="text" id="add-item"></input><input type="submit" id="submit-btn">\
</input>
</div>
<script>

        //retrieve items from local storage
        retrieveFromStorage();

        //dynamically add an <li> element when the submit button is clicked

        var submit_btn = document.getElementById('submit-btn'); //get the submit button
        //attach a function to the submit button
        submit_btn.onclick = function additem(){
                var count = new Date().getTime();
                var textInput = document.getElementById('add-item');
                var item = textInput.value;
                var ul = document.getElementsByTagName('ul');
                var li;
                var img;

                createListElement(count,item);
                localStorage.setItem(count, item);
        }

        function retrieveFromStorage(){
                var ul_item = document.getElementsByTagName('ul');

                for(index in localStorage){

                        var itemValue = localStorage[index];
```

```
43                          var li;
44
45                          li = createListElement(index, itemValue);
46                  }
47          }
48
49
50          function createListElement(id, value){
51                  var textInput = document.getElementById('add-item');
52                  var ul = document.getElementsByTagName('ul');
53                  var li;
54                  var img;
55
56                  ul[0].insertAdjacentHTML('beforeend','<li id="' + id +'" class="item">' + valu\
57  e + '</li>');
58
59                  li = document.getElementById(id);
60
61                  li.addEventListener("touchstart", handleStart);
62                  li.addEventListener("touchmove", handleMove);
63                  li.addEventListener("touchend", handleEnd);
64
65                  swipe(li);
66
67                  li.onclick = strikeThrough;
68                  li.addEventListener("touchstart", strikeThrough);
69          }
70
71
72          function handleStart(e){
73                  e.preventDefault();
74                  this.style.backgroundColor = "#FFFFCC";
75          }
76
77          function handleMove(e){
78                  e.preventDefault();
79                  this.style.backgroundColor = "#FFCC66";
80          }
81
82          function handleEnd(e){
83                  e.preventDefault();
84                  this.style.backgroundColor = "#FFFFFF";
```

```
85          }
86
87          function strikeThrough(e){
88                  e.preventDefault();
89
90                  if (this.style.textDecoration == "none"){
91                          this.style.textDecoration = "line-through";
92                  }else{
93                          this.style.textDecoration = "none";
94                  }
95          }
96
97          function swipe(element){
98                  var startx;
99                  var dist;
100                 var delta;
101
102                 element.addEventListener('touchstart',function(e){
103                         e.preventDefault();
104                         startx = parseInt(e.changedTouches[0].pageX);
105                 });
106
107                 element.addEventListener('touchmove', function(e){
108                         e.preventDefault();
109                         var x = parseInt(e.changedTouches[0].pageX);
110                         dist = x - startx;
111
112                         move(element,dist,0);
113                 });
114
115                 element.addEventListener('touchend', function(e){
116                         if (dist > 100){
117                                 localStorage.removeItem(this.id);
118                                 animate(element, 4);
119                         }else{
120                                 delta = (dist * -1)/2;
121                                 move(element,dist * -1,50);
122                                 move(element,0,0);
123                         }
124                 })
125         }
126
```

```
127         function move(element, dist, speed){
128             element.style.transitionDuration = speed + 'ms';
129             element.style.webkitTransform = 'translate(' + dist + 'px,0)';
130         }
131 
132         function slide(e) {
133 
134                 var speed = 10;
135                 var dist = 1000;
136 
137                 this.style.transform = 'translateX(10px)';
138             this.style.transitionDuration = speed + 'ms';
139             this.style.webkitTransform = 'translate(' + dist + 'px,0)' + 'translateZ(0)\
140 ';
141         }
142 
143 
144         function animate(element, speed){
145                 var children = element.childNodes;
146 
147                 element.style.setProperty('-webkit-transition', 'all 0.5s');
148                 element.style.setProperty('height', '0px');
149                 element.style.setProperty('padding', '0px');
150                 element.style.setProperty('border', '0px');
151                 element.removeChild(children[0]); //remove text node
152         }
153 
154 </script>
155 </body>
156 </html>
```

Conclusion

In this chapter we learnt how to detect that a user swipe action has taken place and how to retrieve touch points.