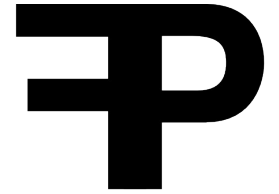


Salvatore Iaconesi  
salvatore.iaconesi@artisopensource.net



FakePress

# FROM DESIGN TO APPLICATION

Crash course per designer visuali: dal concept all'applicazione smartphone/tablet

# **CRASH COURSE**

# COSA USEREMO

**PHONEGAP**  
**IOS SDK BY APPLE**  
**XCODE DEVELOPMENT ENVIRONMENT**

per scaricare i software:

<http://www.phonegap.com/>

<http://developer.apple.com/>

LE APPLICAZIONI SVILUPPATE IN QUESTO MODO POTRANNO  
ESSERE IMMEDIATAMENTE PORTATE ALLA PIATTAFORMA  
ANDROID (*INSTALLARE AMBIENTE DI SVILUPPO **ECLIPSE** ED  
ESEGUIRE IL TOOL DI INSTALLAZIONE APPOSITO FORNITO  
INSIEME A PHONEGAP*)

## **STEP PRELIMINARI:**

- 1) INSTALLARE XCODE
- 2) INSTALLARE PHONEGAP
- 3) COMPILARE L'APPLICAZIONE DI DEFAULT

**TUTTO OK?**

PERFETTO ALLORA:

**IMPARIAMO COME SI COSTRUISCE  
UNA APPLICAZIONE IPHONE/IPAD  
(MA ANCHE ANDROID, BLACKBERRY, WEBOS,  
SYMBIAN...)**

**USANDO HTML5 E CSS**

IMPORTANTISSIMO!

# **LEGGERE QUESTE GUIDELINES**

IN CUI APPLE SPIEGA LE BEST PRACTICES DA  
SEGUIRE IN QUESTO GENERE DI ATTIVITA'

ORA NE VEDREMO ALCUNI PASSI FONDAMENTALI

# USE STANDARDS

HTML 4.01

XHTML 1.0

CSS 2.1 and partial CSS3

ECMAScript 3 (JavaScript)

DOM Level 2

AJAX technologies, including XMLHttpRequest

# GOOD WEB DESIGN PRACTICES

USE DOCTYPE  
SEPARATE HTML E CSS FILES  
WELL STRUCTURED HTML  
USE BROWSER INDEPENDENT CODE

# USE SECURITY FEATURES

SSL  
HTTPS

**NO FRAMESETS!**

**USE BLOCKS! (DIV, SPAN..)**

**CHECK SIZE OF WEB PAGE**

**USE THE `<SELECT>` ELEMENT**

**USE ALERT, CONFIRM, PROMPT**

**USE HTML5 AUDIO AND VIDEO**

**REMEMBER DIFFERENCE BETWEEN  
CLICK AND TOUCH**

## USE CONDITIONAL STYLESHEET

EXAMPLE:

SPECIFY STYLESHEET FOR IPHONE

```
<link media="only screen and (max-device-width: 480px)" href="small-device.css" type="text/css" rel="stylesheet">
```

OR, INSIDE THE CSS FILE:

```
@media screen and (min-device-width: 481px) { ... }
```

# USE VIEWPORT

```
<meta name = "viewport" content = "width = device-width">
```

```
<meta name = "viewport" content = "initial-scale = 1.0">
```

```
<meta name = "viewport" content = "initial-scale = 2.3, user-scalable = no">
```

```
<meta name = "viewport" content = "width = 590">
```

```
<meta name = "viewport" content = "user-scalable=no, width=device-width">
```

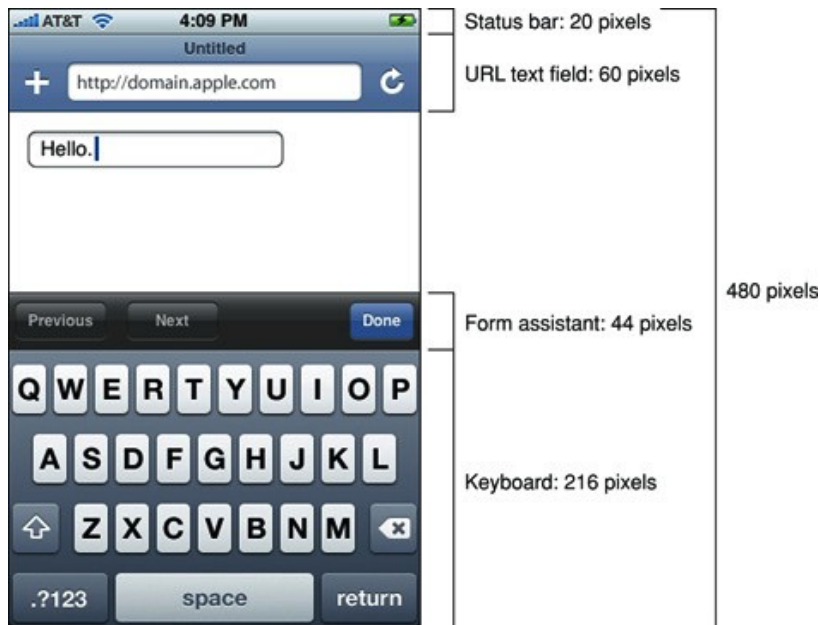
## USE WEBKIT

VEDI [THIS LINK](#) PER CAPIRE LE ESTENSIONI DEL BROWSER MOBILE DISPONIBILE SU IPHONE/IPAD

ESTENSIONI COME:

- webkit-border-image
- webkit-border-radius
- webkit-text-size-adjust
- webkit-tap-highlight-color
- webkit-animation
- webkit-mask
- webkit-transition

# FORMS



Object	Metrics in pixels
Status bar	Height = 20
URL text field	Height = 60
Form assistant	Height = 44
Keyboard	Portrait height = 216 Landscape height = 162
Button bar	Portrait height = 44 Landscape height = 32

## EVENTS

RENDERE GLI OGGETTI **CLICKABLE** PER  
CONTROLLARLI DA JAVASCRIPT

```
<span onmouseover = "..."  
      onmouseout  = "..."  
      onclick = "void(0)">
```

WHERE TO BUY

```
</span>
```

## EVENTS

### PER GESTIRE GLI EVENTI MULTITOUCH:

```
<div
  ontouchstart="touchStart(event);"
  ontouchmove="touchMove(event);"
  ontouchend="touchEnd(event);"
  ontouchcancel="touchCancel(event);"
></div>
```

Alternatively, register handlers in JavaScript as follows:

```
element.addEventListener("touchstart", touchStart, false);
element.addEventListener("touchmove", touchMove, false);
element.addEventListener("touchend", touchEnd, false);
element.addEventListener("touchcancel", touchCancel, false);
```

```
function touchStart(event) {
  // Insert your code here
}
```

Optionally, get all touches on a page using the `touches` property as follows:

```
var allTouches = event.touches;
```

Optionally, get all touches for the target element using the `targetTouches` property:

```
var targetTouches = event.targetTouches;
```

Optionally, get all changed touches for this event using the `changedTouches` property:

```
var changedTouches = event.changedTouches;
```

get the number of touches:

```
event.touches.length
```

Get a specific touch object at index `i`:

```
var touch = event.touches[i];
```

get the location in page coordinates for a single-finger event:

```
var x = event.touches[0].pageX;
var y = event.touches[0].pageY;
```

## EVENTS

### MULTITOUCH + JAVASCRIPT

```
function touchMove(event) {  
    event.preventDefault();  
    curX = event.targetTouches[0].pageX - startX;  
    curY = event.targetTouches[0].pageY - startY;  
    event.targetTouches[0].target.style.webkitTransform =  
        'translate(' + curX + 'px, ' + curY + 'px)';  
}
```

## GESTURES

```
<div
  ongesturestart="gestureStart(event);"
  ongesturechange="gestureChange(event);"
  ongestureend="gestureEnd(event);"
></div>
```

```
var angle = event.rotation;
```

```
var scale = event.scale;
```

```
function gestureChange(event) {
  // Insert your code here
}
```

```
onGestureChange: function(e) {
  e.preventDefault();
  e.target.style.webkitTransform =
    'scale(' + e.scale + startScale + ') rotate(' + e.rotation + startRotation + 'deg)';
}
```

**INIZIAMO**

# PRENDIAMO UN DOCUMENTO HTML

```
<html>
<head>
  <link rel="stylesheet" href="screen.css" type="text/css" />
  <title>Salvatore laconesi</title>
</head>
<body>
<div id="container">
  <div id="header">
    <h1><a href=".">Salvatore laconesi</a></h1>
    <div id="utility">
      <ul>
        <li><a href="about.html">About</a></li>
        <li><a href="blog.html">Blog</a></li>
      </ul>
    </div>
    <div id="nav">
      <ul>
        <li><a href="design.html">Design</a></li>
        <li><a href="progetti.html">Progetti</a></li>
        <li><a href="ricerca.html">Ricerca</a></li>
      </ul>
    </div>
  </div>
  <div id="content">
    <h2>About</h2>
    <p>Salvatore laconesi è ingegnere robotico, interaction designer
    artista e insegna cross media design alla Facoltà di
    Architettura dell'Università di Roma "La Sapienza".
    Adora gli animali, la musica elettronica e il circuit bending.
    ...
  </p>
  </div>
  <div id="sidebar">
    Salvatore laconesi è ingegnere robotico, artista, interaction
    designer e professore di design cross-mediale.
  </p>
  </div>
  <div id="footer">
    <ul>
      <li><a href="design.html">Design</a></li>
      <li><a href="about.html">About</a></li>
      <li><a href="blog.html">Blog</a></li>
    </ul>
    <p class="subtle">Salvatore laconesi, Inc.</p>
  </div>
</div>
</body>
</html>
```

# Mettiamo uno stylesheet dedicato

```
<link rel="stylesheet" type="text/css"  
      href="iphone.css" media="only screen and (max-width: 480px)" />  
<link rel="stylesheet" type="text/css"  
      href="desktop.css" media="screen and (min-width: 481px)" />
```

# Controlliamo viewport e scala

```
<meta name="viewport" content="user-scalable=no, width=device-width" />
```

# Creiamo look'n feel iOS in “iphone.css”

```
body {
  background-color: #ddd; /* Background color */
  color: #222; /* Foreground color used for text */
  font-family: Helvetica;
  font-size: 14px;
  margin: 0; /* Amount of negative space around the outside of the body */
  padding: 0; /* Amount of negative space around the inside of the body */
}

#header h1 {
  margin: 0;
  padding: 0;
}
#header h1 a {
  background-color: #ccc;
  border-bottom: 1px solid #666;
  color: #222;
  display: block;
  font-size: 20px;
  font-weight: bold;
  padding: 10px 0;
  text-align: center;
  text-decoration: none;
}

#header ul {
  list-style: none;
  margin: 10px;
  padding: 0;
}
#header ul li a {
  background-color: #FFFFFF;
  border: 1px solid #999999;
  color: #222222;
  display: block;
  font-size: 17px;
  font-weight: bold;
  margin-bottom: -1px;
  padding: 12px 10px;
  text-decoration: none;
}

#content, #sidebar {
  padding: 10px;
}

#footer {
  display: none;
}
```

# ANDIAMO OLTRE “iphone.css”

```
#header h1 a {  
  text-shadow: 0px 1px 0px #fff;  
  background-image: -webkit-gradient(linear, left top, left bottom,  
    from(#ccc), to(#999));  
}
```

```
#header ul li:first-child a {  
  -webkit-border-top-left-radius: 8px;  
  -webkit-border-top-right-radius: 8px;  
}  
#header ul li:last-child a {  
  -webkit-border-bottom-left-radius: 8px;  
  -webkit-border-bottom-right-radius: 8px;  
}
```

# CREIAMO IL BEHAVIOUR

Nascondiamo gli <UL> nell'#header

```
#header ul.hide {  
  display: none;  
}
```

DEFINIAMO LO STILE PER UN PULSANTE CHE AGGIUNGEREMO TRA UN ATTIMO:

```
#header div.leftButton {  
  position: absolute;  
  top: 7px;  
  left: 6px;  
  height: 30px;  
  font-weight: bold;  
  text-align: center;  
  color: white;  
  text-shadow: rgba(0,0,0,0.6) 0px -1px 0px;  
  line-height: 28px;  
  border-width: 0 8px 0 8px;  
  -webkit-border-image: url(images/button.png) 0 8 0 8;  
}
```

```
#header div.pressed {  
  -webkit-border-image: url(images/buttonActive.png) 0 8 0 8;  
}
```

# CREIAMO IL BEHAVIOUR

Aggiungiamo **jQuery** e predisponiamo un file per i nostri script:

```
<script type="text/javascript" src="jquery.js"></script>  
<script type="text/javascript" src="iphone.js"></script>
```

Dentro `iphone.js` aggiungiamo:

```
function jQueryStart(){  
    $('#header ul').addClass('hide');  
    $('#header').append('<div class="leftButton"  
        onclick="toggleMenu()">Menu</div>');  
}  
function toggleMenu() {  
    $('#header ul').toggleClass('hide');  
    $('#header .leftButton').toggleClass('pressed');  
}
```

E nella funzione **onDeviceReady()** aggiungiamo;

```
jQueryStart();
```

**MA ANDIAMO OLTRE**

# UNA SEMPLICE APP GEO-REFERENZIATA

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta name="viewport" content="width=device-width; height=device-height; user-scalable=no" />
<meta http-equiv="Content-type" content="text/html; charset=utf-8">
<title>Beer Me</title>
<link rel="stylesheet" href="/master.css" type="text/css" media="screen" />
<script type="text/javascript" src="scripts/phonegap.js"></script>
<script type="text/javascript">
function loader() {
var state = document.readyState;
if (state == 'loaded' || state == 'complete') {
run();
} else {
if (navigator.userAgent.indexOf('Browzr') > -1) {
setTimeout(run, 250);
} else {
document.addEventListener('deviceready',run,false);
}
}
}
function run() {
var win = function(position) { // Grab coordinates object from the Position object passed into success callback.
var coords = position.coords;
// Call for static google maps data - make sure you use your own Google Maps API key!
var url = "http://maps.google.com/maps/api/staticmap?center=" + coords.latitude + "," + coords.longitude + "&zoom=13&size=320x480&maptype=roadmap&key=MyGoogleMapsAPIKey&sensor=true";
document.getElementById('map').setAttribute('src',url);
};
var fail = function(e) {
alert('Can\'t retrieve position.\nError: ' + e);
};
navigator.geolocation.getCurrentPosition(win, fail);
}
</script>
</head>
<body onload="loader();">
<img id="map" />
</body>
</html>
```

**PER FINIRE:**

**Un po' di link sulle cose mostrate  
durante la giornata**

## **EMOZIONI:**

We feel fine:

<http://www.wefeelfine.org/>

E le API per interagire con il database di emozioni:

<http://www.wefeelfine.org/api.html>

## **PROCESSING:**

[Http://www.processing.org](http://www.processing.org)

È un ambiente di sviluppo molto semplice e pensato per le arti visive.

E' molto semplice da utilizzare, ci sono moltissime librerie ed esempi pronti.

Può essere utilizzato per produrre applicazioni per Android (bisogna però prima installare Android SDK)

Su questo altro sito:

<http://www.openprocessing.org/>

ci sono moltissimi esempi liberamente utilizzabili.

## SINTESI VOCALE:

<http://www.cstr.ed.ac.uk/projects/festival/>

È un motore Open Source per la sintesi vocale. Ci sono delle “voci” gratuite anche in italiano, ma bisogna cercare un po' sulla rete per trovarne di buone.

In particolare c'è un plugin per Processing che integra questo software direttamente, e permette di generare il suono del parlato con 2 righe di codice:

<http://www.local-guru.net/blog/pages/ttslib>

## **COCOS2d:**

<http://www.cocos2d-iphone.org/>

È una libreria potentissima per sviluppare app per iPhone/iPad ad alto contenuto di grafica animata e realtime.

Occorre però imparare il linguaggio c per usarla.

E' consigliato per sviluppare app come giochi, simulatori o eye-candy, quelle cioè che necessitano di molta grafica e multimedia spinto.

Qui, all'interno della documentazione, un how-to che spiega come integrare le animazioni fatte in flash all'interno di una app iPhone/iPad

[http://www.cocos2d-iphone.org/wiki/doku.php/prog\\_guide:animation](http://www.cocos2d-iphone.org/wiki/doku.php/prog_guide:animation)

## TIMELINE:

Questo è un tool fatto in JavaScript che permette di creare delle timeline interattive in maniera semplice e molto personalizzabile:

<http://www.simile-widgets.org/timeline/>

Le timeline create con SMILE sono completamente in HTML+CSS

Sono quindi completamente modificabili e personalizzabili.

Esiste un plugin per Wordpress:

<http://wordpress.org/extend/plugins/wp-simile-timeline/>

Usandolo la creazione delle timeline diventa immediata, creando contenuti direttamente con Wordpress

## Information visualization:

<http://thejit.org/>

Questa è una libreria javascript potentissima per creare visualizzazioni di informazioni interattive in maniera molto semplice.

Il modo più semplice di iniziare ad impararne l'uso è quello di prendere un esempio tra quelli forniti nella documentazione sul sito, impostarlo seguendo le indicazioni del sito e poi prendere i file di dati che usano i vari esempi, e modificarli, inserendo i dati che servono a voi, per la vostra applicazione.

Dal sito trovate anche i link al Google Group che esiste per la community internazionale che usa questa libreria: potete iscrivervi gratuitamente e fare domande a centinaia di programmatori che di solito sono molto ben disposti a fornire aiuti e indicazioni sull'uso della libreria e sulla risoluzione di problemi.

Il formato dei file di dati si chiama JSON: <http://json.org/>

## Information visualization:

<http://www.fakepress.it/knowners/>

Questo plugin per wordpress permette di realizzare visualizzazioni interattive molto utili per rappresentare informazioni e basi di conoscenza.

I grafi vengono generati automaticamente a partire dalle tag usate nei contenuti di Wordpress.

E' completamente personalizzabile usando CSS.

E' prodotto da FakePress e se volete potete chiedere direttamente a me le info sul suo utilizzo e sui trucchetti che ci si possono fare.

## Realtà Aumentata:

<https://github.com/xdxdVSxdxd/NeoReality>

Questa è una applicazione iPhone pronta per la personalizzazione ed una estensione di Wordpress che permette di realizzare un browser in realtà aumentata.

Installate il plugin per wordpress, inserite i contenuti dentro il sistema, ne Indicate la posizione geografica e, automaticamente, li vedrete dentro l'applicazione iPhone, in realtà aumentata, proprio come su software tipo Layar o Junaio.

Anche questo è un software di FakePress e quindi posso aiutarvi direttamente per il suo uso e personalizzazione.

Salvatore Iaconesi  
[salvatore.iaconesi@artisopensource.net](mailto:salvatore.iaconesi@artisopensource.net)



# FROM DESIGN TO APPLICATION

Crash course per designer visuali: dal concept all'applicazione smartphone/tablet

# **CRASH COURSE**

## **COSA USEREMO**

# **PHONEGAP IOS SDK BY APPLE XCODE DEVELOPMENT ENVIRONMENT**

per scaricare i software:

<http://www.phonegap.com/>

<http://developer.apple.com/>

LE APPLICAZIONI SVILUPPATE IN QUESTO MODO POTRANNO  
ESSERE IMMEDIATAMENTE PORTATE ALLA PIATTAFORMA  
ANDROID (INSTALLARE AMBIENTE DI SVILUPPO **ECLIPSE** ED  
ESEGUIRE IL TOOL DI INSTALLAZIONE APPOSITO FORNITO  
INSIEME A PHONEGAP)

**STEP PRELIMINARI:**

- 1) INSTALLARE XCODE
- 2) INSTALLARE PHONEGAP
- 3) COMPILARE L'APPLICAZIONE DI DEFAULT

**TUTTO OK?**

PERFETTO ALLORA:

**IMPARIAMO COME SI COSTRUISCE  
UNA APPLICAZIONE IPHONE/IPAD  
(MA ANCHE ANDROID, BLACKBERRY, WEBOS,  
SYMBIAN...)**

**USANDO HTML5 E CSS**

IMPORTANTISSIMO!

## **LEGGERE QUESTE GUIDELINES**

IN CUI APPLE SPIEGA LE BEST PRACTICES DA  
SEGUIRE IN QUESTO GENERE DI ATTIVITA'

ORA NE VEDREMO ALCUNI PASSI FONDAMENTALI

# USE STANDARDS

HTML 4.01  
XHTML 1.0  
CSS 2.1 and partial CSS3  
ECMAScript 3 (JavaScript)  
DOM Level 2  
AJAX technologies, including XMLHttpRequest

# GOOD WEB DESIGN PRACTICES

USE DOCTYPE  
SEPARATE HTML E CSS FILES  
WELL STRUCTURED HTML  
USE BROWSER INDEPENDENT CODE

# USE SECURITY FEATURES

SSL  
HTTPS

**NO FRAMESETS!**  
**USE BLOCKS! (DIV, SPAN..)**  
**CHECK SIZE OF WEB PAGE**

**USE THE <SELECT> ELEMENT  
USE ALERT, CONFIRM, PROMPT  
USE HTML5 AUDIO AND VIDEO**

**REMEMBER DIFFERENCE BETWEEN  
CLICK AND TOUCH**

# USE CONDITIONAL STYLESHEET

EXAMPLE:

SPECIFY STYLESHEET FOR IPHONE

```
<link media="only screen and (max-device-width: 480px)" href="small-device.css" type="text/css" rel="stylesheet">
```

OR, INSIDE THE CSS FILE:

```
@media screen and (min-device-width: 481px) { ... }
```

## USE VIEWPORT

```
<meta name = "viewport" content = "width = device-width">
```

```
<meta name = "viewport" content = "initial-scale = 1.0">
```

```
<meta name = "viewport" content = "initial-scale = 2.3, user-scalable = no">
```

```
<meta name = "viewport" content = "width = 590">
```

```
<meta name = "viewport" content = "user-scalable=no, width=device-width">
```

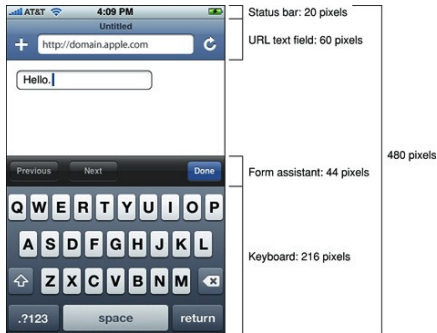
## USE WEBKIT

VEDI [THIS LINK](#) PER CAPIRE LE ESTENSIONI DEL BROWSER  
MOBILE DISPONIBILE SU IPHONE/IPAD

ESTENSIONI COME:

- webkit-border-image
- webkit-border-radius
- webkit-text-size-adjust
- webkit-tap-highlight-color
- webkit-animation
- webkit-mask
- webkit-transition

# FORMS



Object	Metrics in pixels
Status bar	Height = 20
URL text field	Height = 60
Form assistant	Height = 44
Keyboard	Portrait height = 216 Landscape height = 162
Button bar	Portrait height = 44 Landscape height = 32

# EVENTS

RENDERE GLI OGGETTI **CLICKABLE** PER  
CONTROLLARLI DA JAVASCRIPT

```
<span onmouseover = "..."  
      onmouseout = "..."  
      onclick = "void(0)">
```

WHERE TO BUY

```
</span>
```

# EVENTS

### PER GESTIRE GLI EVENTI MULTITOUCH:

```
<div
  ontouchstart="touchStart(event);"
  ontouchmove="touchMove(event);"
  ontouchend="touchEnd(event);"
  ontouchcancel="touchCancel(event);"
></div>
```

Alternatively, register handlers in JavaScript as follows:

```
element.addEventListener("touchstart", touchStart, false);
element.addEventListener("touchmove", touchMove, false);
element.addEventListener("touchend", touchEnd, false);
element.addEventListener("touchcancel", touchCancel, false);
```

```
function touchStart(event) {
  // Insert your code here
}
```

Optionally, get all touches on a page using the `touches` property as follows:

```
var allTouches = event.touches;
```

get the number of touches:

```
event.touches.length
```

Optionally, get all touches for the target element using the `targetTouches` property:

```
var targetTouches = event.targetTouches;
```

Get a specific touch object at index `i`:

```
var touch = event.touches[i];
```

Optionally, get all changed touches for this event using the `changedTouches` property:

```
var changedTouches = event.changedTouches;
```

get the location in page coordinates for a single-finger event:

```
var x = event.touches[0].pageX;
var y = event.touches[0].pageY;
```

# EVENTS

## MULTITOUCH + JAVASCRIPT

```
function touchMove(event) {  
  event.preventDefault();  
  curX = event.targetTouches[0].pageX - startX;  
  curY = event.targetTouches[0].pageY - startY;  
  event.targetTouches[0].target.style.webkitTransform =  
    'translate(' + curX + 'px, ' + curY + 'px)';  
}
```

# GESTURES

```
<div                                var angle = event.rotation;
  ongesturestart="gestureStart(event);"
  ongesturechange="gestureChange(event);"
  ongestureend="gestureEnd(event);"
></div>                                var scale = event.scale;

function gestureChange(event) {
  // Insert your code here
}

onGestureChange: function(e) {
  e.preventDefault();
  e.target.style.webkitTransform =
    'scale(' + e.scale + startScale + ') rotate(' + e.rotation + startRotation + 'deg)';
}
```

**INIZIAMO**

# PRENDIAMO UN DOCUMENTO HTML

```
<html>
<head>
<link rel="stylesheet" href="screen.css" type="text/css" />
<title>Salvatore Iaconesi</title>
</head>
<body>
<div id="container">
<div id="header">
<h1><a href="/">Salvatore Iaconesi</a></h1>
<div id="utility">
<ul>
<li><a href="about.html">About</a></li>
<li><a href="blog.html">Blog</a></li>
</ul>
</div>
<div id="nav">
<ul>
<li><a href="design.html">Design</a></li>
<li><a href="progetti.html">Progetti</a></li>
<li><a href="ricerca.html">Ricerca</a></li>
</ul>
</div>
<div id="content">
<h2>About</h2>
<p>Salvatore Iaconesi è ingegnere robotico, interaction designer
artista e insegna cross media design alla Facoltà di
Architettura dell'Università di Roma "La Sapienza".
Adora gli animali, la musica elettronica e il circuit bending.
...
</p>
</div>
<div id="sidebar">
Salvatore Iaconesi è ingegnere robotico, artista, interaction
designer e professore di design cross-mediale.
</p>
</div>
<div id="footer">
<ul>
<li><a href="design.html">Design</a></li>
<li><a href="about.html">About</a></li>
<li><a href="blog.html">Blog</a></li>
</ul>
<p class="subtle">Salvatore Iaconesi, Inc.</p>
</div>
</body>
</html>
```

# Mettiamo uno stylesheet dedicato

```
<link rel="stylesheet" type="text/css"  
      href="iphone.css" media="only screen and (max-width: 480px)" />  
<link rel="stylesheet" type="text/css"  
      href="desktop.css" media="screen and (min-width: 481px)" />
```

# Controlliamo viewport e scala

```
<meta name="viewport" content="user-scalable=no, width=device-width" />
```

# Creiamo look'n feel iOS in “iphone.css”

```
body {
  background-color: #ddd; /* Background color */
  color: #222; /* Foreground color used for text */
  font-family: Helvetica;
  font-size: 14px;
  margin: 0; /* Amount of negative space around the outside of the body */
  padding: 0; /* Amount of negative space around the inside of the body */
}

#header h1 {
  margin: 0;
  padding: 0;
}

#header h1 a {
  background-color: #ccc;
  border-bottom: 1px solid #666;
  color: #222;
  display: block;
  font-size: 20px;
  font-weight: bold;
  padding: 10px 0;
  text-align: center;
  text-decoration: none;
}

#header ul {
  list-style: none;
  margin: 10px;
  padding: 0;
}

#header ul li a {
  background-color: #FFFFFF;
  border: 1px solid #999999;
  color: #222222;
  display: block;
  font-size: 17px;
  font-weight: bold;
  margin-bottom: -1px;
  padding: 12px 10px;
  text-decoration: none;
}

#content, #sidebar {
  padding: 10px;
}

#footer {
  display: none;
}
```

# ANDIAMO OLTRE “iphone.css”

```
#header h1 a {  
  text-shadow: 0px 1px 0px #fff;  
  background-image: -webkit-gradient(linear, left top, left bottom,  
    from(#ccc), to(#999));  
}
```

```
#header ul li:first-child a {  
  -webkit-border-top-left-radius: 8px;  
  -webkit-border-top-right-radius: 8px;  
}  
#header ul li:last-child a {  
  -webkit-border-bottom-left-radius: 8px;  
  -webkit-border-bottom-right-radius: 8px;  
}
```

# CREIAMO IL BEHAVIOUR

Nascondiamo gli <UL> nell'#header

```
#header ul.hide {  
  display: none;  
}
```

DEFINIAMO LO STILE PER UN PULSANTE CHE AGGIUNGEREMO  
TRA UN ATTIMO:

```
#header div.leftButton {  
  position: absolute;  
  top: 7px;  
  left: 6px;  
  height: 30px;  
  font-weight: bold;  
  text-align: center;  
  color: white;  
  text-shadow: rgba(0,0,0,0.6) 0px -1px 0px;  
  line-height: 28px;  
  border-width: 0 8px 0 8px;  
  -webkit-border-image: url(images/button.png) 0 8 0 8;  
}
```

```
#header div.pressed {  
  -webkit-border-image: url(images/buttonActive.png) 0 8 0 8;  
}
```

Le immagini vengono dalla directory "images" di **jqtouch**

# CREIAMO IL BEHAVIOUR

Aggiungiamo **jQuery** e predisponiamo un file per i nostri script:

```
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript" src="iphone.js"></script>
```

Dentro `iphone.js` aggiungiamo:

```
function jQueryStart(){
    $('#header ul').addClass('hide');
    $('#header').append('<div class="leftButton"
        onclick="toggleMenu()">Menu</div>');
}
function toggleMenu() {
    $('#header ul').toggleClass('hide');
    $('#header .leftButton').toggleClass('pressed');
}
```

E nella funzione `onDeviceReady()` aggiungiamo:

```
jQueryStart();
```

Le immagini vengono dalla directory "images" di **jqtouch**

**MA ANDIAMO OLTRE**

# UNA SEMPLICE APP GEO-REFERENZIATA

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <meta name="viewport" content="width=device-width; height=device-height; user-scalable=no" />
    <meta http-equiv="Content-type" content="text/html; charset=utf-8">
    <title>Beer Me</title>
    <link rel="stylesheet" href="master.css" type="text/css" media="screen" />
    <script type="text/javascript" src="scripts/phonegap.js"></script>
    <script type="text/javascript">
      function loader() {
        var state = document.readyState;
        if (state == 'loaded' || state == 'complete') {
          run();
        } else {
          if (navigator.userAgent.indexOf('Browzr') > -1) {
            setTimeout(run, 250);
          } else {
            document.addEventListener('deviceready', run, false);
          }
        }
      }
      function run() {
        var win = function(position) { // Grab coordinates object from the Position object passed into success callback.
          var coords = position.coords;
          // Call for static google maps data - make sure you use your own Google Maps API key!
          var url = "http://maps.google.com/maps/api/staticmap?center=" + coords.latitude + "," + coords.longitude + "&zoom=13&size=320x480&maptype=roadmap&key=MyGoogleMapsAPIKey&sensor=true";
          document.getElementById('map').setAttribute('src',url);
        };
        var fail = function(e) {
          alert('Can't retrieve position. InError: ' + e);
        };
        navigator.geolocation.getCurrentPosition(win, fail);
      }
    </script>
  </head>
  <body onload="loader();" >
    <img id="map" />
  </body>
</html>
```

**PER FINIRE:**  
**Un po' di link sulle cose mostrate  
durante la giornata**

**EMOZIONI:**

We feel fine:

<http://www.wefeelfine.org/>

E le API per interagire con il database di emozioni:

<http://www.wefeelfine.org/api.html>

**PROCESSING:**

[Http://www.processing.org](http://www.processing.org)

È un ambiente di sviluppo molto semplice e pensato per le arti visive.

E' molto semplice da utilizzare, ci sono moltissime librerie ed esempi pronti.

Può essere utilizzato per produrre applicazioni per Android (bisogna però prima installare Android SDK)

Su questo altro sito:

<http://www.openprocessing.org/>

ci sono moltissimi esempi liberamente utilizzabili.

## **SINTESI VOCALE:**

<http://www.cstr.ed.ac.uk/projects/festival/>

È un motore Open Source per la sintesi vocale. Ci sono delle “voci” gratuite anche in italiano, ma bisogna cercare un po' sulla rete per trovarne di buone.

In particolare c'è un plugin per Processing che integra questo software direttamente, e permette di generare il suono del parlato con 2 righe di codice:

<http://www.local-guru.net/blog/pages/ttslib>

## **COCOS2d:**

<http://www.cocos2d-iphone.org/>

È una libreria potentissima per sviluppare app per iPhone/iPad ad alto contenuto di grafica animata e realtime.

Occorre però imparare il linguaggio c per usarla.

E' consigliato per sviluppare app come giochi, simulatori o eye-candy, quelle cioè che necessitano di molta grafica e multimedia spinto.

Qui, all'interno della documentazione, un how-to che spiega come integrare le animazioni fatte in flash all'interno di una app iPhone/iPad

[http://www.cocos2d-iphone.org/wiki/doku.php/prog\\_guide:animation](http://www.cocos2d-iphone.org/wiki/doku.php/prog_guide:animation)

## **TIMELINE:**

Questo è un tool fatto in JavaScript che permette di creare delle timeline interattive in maniera semplice e molto personalizzabile:

<http://www.simile-widgets.org/timeline/>

Le timeline create con SMILE sono completamente in HTML+CSS

Sono quindi completamente modificabili e personalizzabili.

Esiste un plugin per Wordpress:

<http://wordpress.org/extend/plugins/wp-simile-timeline/>

Usandolo la creazione delle timeline diventa immediata, creando contenuti direttamente con Wordpress

### **Information visualization:**

<http://thejit.org/>

Questa è una libreria javascript potentissima per creare visualizzazioni di informazioni interattive in maniera molto semplice.

Il modo più semplice di iniziare ad impararne l'uso è quello di prendere un esempio tra quelli forniti nella documentazione sul sito, impostarlo seguendo le indicazioni del sito e poi prendere i file di dati che usano i vari esempi, e modificarli, inserendo i dati che servono a voi, per la vostra applicazione.

Dal sito trovate anche i link al Google Group che esiste per la community internazionale che usa questa libreria: potete iscrivervi gratuitamente e fare domande a centinaia di programmatori che di solito sono molto ben disposti a fornire aiuti e indicazioni sull'uso della libreria e sulla risoluzione di problemi.

Il formato dei file di dati si chiama JSON: <http://json.org/>

**Information visualization:**

<http://www.fakepress.it/knowners/>

Questo plugin per wordpress permette di realizzare visualizzazioni interattive molto utili per rappresentare informazioni e basi di conoscenza.

I grafi vengono generati automaticamente a partire dalle tag usate nei contenuti di Wordpress.

E' completamente personalizzabile usando CSS.

E' prodotto da FakePress e se volete potete chiedere direttamente a me le info sul suo utilizzo e sui trucchetti che ci si possono fare.

### **Realtà Aumentata:**

<https://github.com/xdxdVSxdxd/NeoReality>

Questa è una applicazione iPhone pronta per la personalizzazione ed una estensione di Wordpress che permette di realizzare un browser in realtà aumentata.

Installate il plugin per wordpress, inserite i contenuti dentro il sistema, ne Indicate la posizione geografica e, automaticamente, li vedrete dentro l'applicazione iPhone, in realtà aumentata, proprio come su software tipo Layar o Junaio.

Anche questo è un software di FakePress e quindi posso aiutarvi direttamente per il suo uso e personalizzazione.