6 Trucos HTML5 Para Móviles

Actualmente existen una docena de navegadores, versiones, tamaños de pantalla, características indocumentadas, bugs y problemas nuevos. Es por ello que, en algunas situaciones, se tiene que romper algunos límites para lograr el objetivo del desarrollo. Existen algunos hacks que pueden ser usados para sacar provecho, te invitamos a conocerlos y así puedas implementar algunos trucos HTML5 para móviles.

1. Experiencia de pantalla completa

El navegador de Android es diferente a Chrome, y la única solución es tener un ***document*** con una altura al menos igual a la altura del dispositivo. Para esto ocultaremos la barra de la URL con el siguiente código:

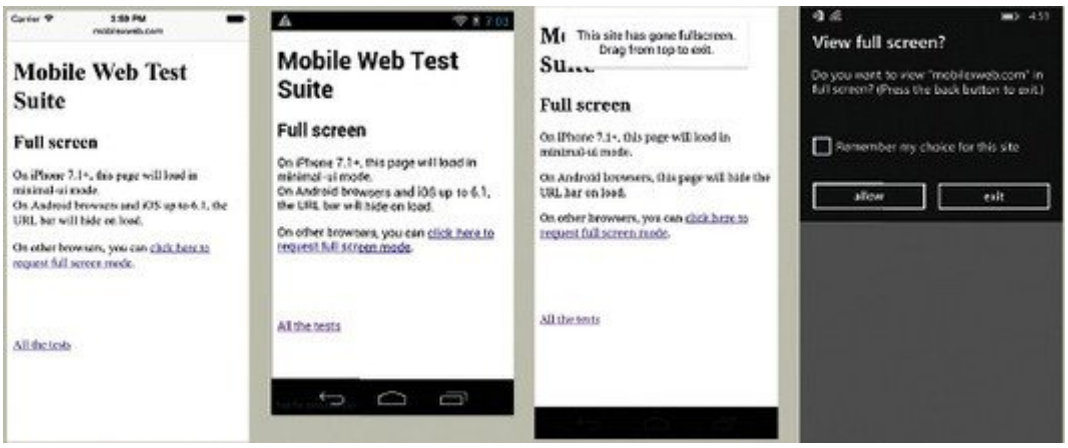
[js]window.addEventListener("load", function() { window. scrollTo(0, 0); });[/js]

Google Maps usa este truco para emular una experiencia de pantalla completa en este navegador. Cuando se usa con cuidado, también puede reducir la posibilidad de una reaparición de la barra de direcciones del navegador mediante la prevención de la gestión del desplazamiento táctil:

[js]document.addEventListener("touchmove", function(e) { e.preventDefault() });[/js]

En Google Chrome, Firefox OS, Firefox para Android, BlackBerry

OS 10 y Amazon Silk (el navegador desarrollado para Kindle Fire) podemos utilizar la API de pantalla completa estándar de la W3C.



Como ya debes saber, algunos navegadores siguen trabajando con prefijos, por lo que para ello necesitamos un código de múltiples proveedores:

[js]

```
var body = document.documentElement;
if (body.requestFullscreen) {
body.requestFullscreen();
} else if (body.webkitrequestFullscreen) {
body.webkitrequestFullscreen();
} else if (body.mozrequestFullscreen) {
body.mozrequestFullscreen();
} else if (body.msrequestFullscreen) {
body.msrequestFullscreen();
}
```

[/js]

La única restricción importante para recordar al solicitar el modo de pantalla completa es que debemos hacerlo sólo después de la interacción de un usuario, por ejemplo, activado por un evento táctil. Apple ofrece una ***solución para iPhone*** desde iOS 7.1, utilizando un atributo especial de la etiqueta meta.

Para activar el modo de interfaz de usuario mínima, sólo tienes que utilizar:

[html]<meta name="viewport" content="width=devicewidth, minimal-ui">[/html]

Cuando este modo está activado, necesitas tomar algún tipo de atención especial alrededor de las áreas que se pueden tocar cerca de los bordes.

Para detectar si el modo mínima de interfaz de usuario está habilitada:

[css]

```
@media (device-height: 568px) and (height: 529px),
(device-height: 480px) and (height: 441px) {
/* minimal-ui is active */
}
```

[/css]

2. Crear una aplicación web en la pantalla principal

En iOS y Chrome en Android, se puede tener una experiencia de pantalla completa después de que el usuario ha instalado un icono en la pantalla principal. La aplicación web se ejecuta fuera del navegador. Para habilitar esto, tenemos que añadir algunas etiquetas meta:

[html]

```
<!– for ios 7 style, multi-resolution icon of 152×152 –>
<meta name="apple-mobile-web-app-capable" content="yes">
```

```
<meta      name="apple-mobile-web-app-status-barstyle"
content="black-translucent">
<link rel="apple-touch-icon" href="icon-152.png">
<!– for Chrome on Android, multi-resolution icon of 196×196 –>
<meta name="mobile-web-app-capable" content="yes">
<link rel="shortcut icon" sizes="196×196" href="icon-196.png">

[/html]
```

En Firefox OS y Firefox para Android también podemos crear aplicaciones web en la pantalla principal, por medio de la creación de un JSON en el manifest y utilizando una API de JavaScript. Consulte la documentación oficial para los ejemplos.

3. Alta resolución en Canvas

El ***API Canvas*** es una interfaz de dibujo basado en mapa de bits que funciona como una imagen cargada desde un archivo. Por lo tanto, si crea un Canvas con width= 200, se crea una imagen de píxeles reales en 200, independientemente de la resolución.

Eso significa que en un iPhone 5S, la imagen de 200 píxeles será redimensionada a 400 píxeles de dispositivo, y en un dispositivo Nexus 5 será redimensionado a 600 píxeles de dispositivo. En ambos casos pierden parte de la resolución de la imagen. Si desea crear una imagen de alta resolución, digamos, el doble del dispositivo de resolución media original, es posible utilizar el siguiente truco:

[html]

```
<canvas width="400" height="400" style="width: 200px; height: 200px"></canvas>
<script>
document.queryselector("canvas").getContext("2d").scale(2, 2);
</script>
```

[/html]

Tenga en cuenta que el aumento del tamaño de Canvas, también aumentará la memoria utilizada y la CPU necesaria para crear los dibujos, por lo que deberíamos hacerlo sólo cuando estamos seguros de que se trata de un dispositivo de alta resolución.

4. Campo de texto verdaderamente numérico

Los diferentes teclados que puedes encontrar en los dispositivos móviles, serán un verdadero dolor de cabeza al momento usarlo en los campos de entrada. Este pequeño problema se puede ver en los campos de entrada numérico, para esto podemos usar el atributo **pattern=»[0-9]*»** por lo que recibirá un **teclado numérico en iOS** como en otros sistemas operativos:

[html]<input type="number" pattern="[0-9]*">[/html]

También puede utilizar este **truco con type=»password»** para

conseguir un teclado numérico para un campo de texto de la contraseña.

[html]<input type="password" pattern="[0-9]*">[/html]

5. Diseño Web Responsive y Windows 8

Si está trabajando con sitios web responsive, se podría pensar que la definición de una meta tag viewport móvil y diferentes puntos de ruptura con MediaQueries. Sin embargo, hay una situación particular que necesita un poco más de trabajo: Windows 8.x con Internet Explorer que funciona en modo de pantalla completa.

En Windows 8 y 8.1, un sitio web abierto en este navegador puede compartir la pantalla con otras aplicaciones en modo de pantalla completa, incluyendo el escritorio clásico. En ese caso, y cuando la anchura disponible es inferior a 1024 píxeles, IE aplicará un comportamiento móvil que aleja la página web y no se aplica el código responsive

Para resolver este problema, podemos utilizar la declaración Viewport CSS en una consulta MediaQuerie, tal como la siguiente:

[css]

```
@media only screen and (max-width: 400px) {
@-ms-viewport { width: 320px; }
}
```

[/css]

6. ¿Dónde está el selector de fecha y hora?

Se te dijo que mediante el uso de **<input type=datetime>** se obtiene un selector de fecha y hora, en la mayoría de los navegadores. Esto era cierto, hasta que se vio ser un problema en algunas versiones. Sin embargo, es posible que no tenga en cuenta que el selector de fecha y de tiempo todavía disponible en los navegadores que utilizan ***type=datetime-local***, ya que este cambio no fue debidamente anunciado o documentado por los proveedores de tu navegador.

[html]<input type="datetime-local">[/html]