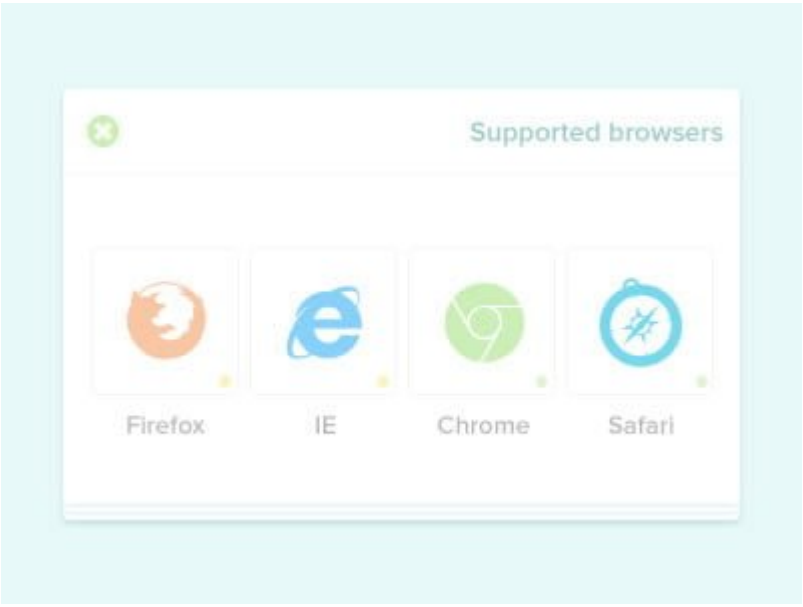
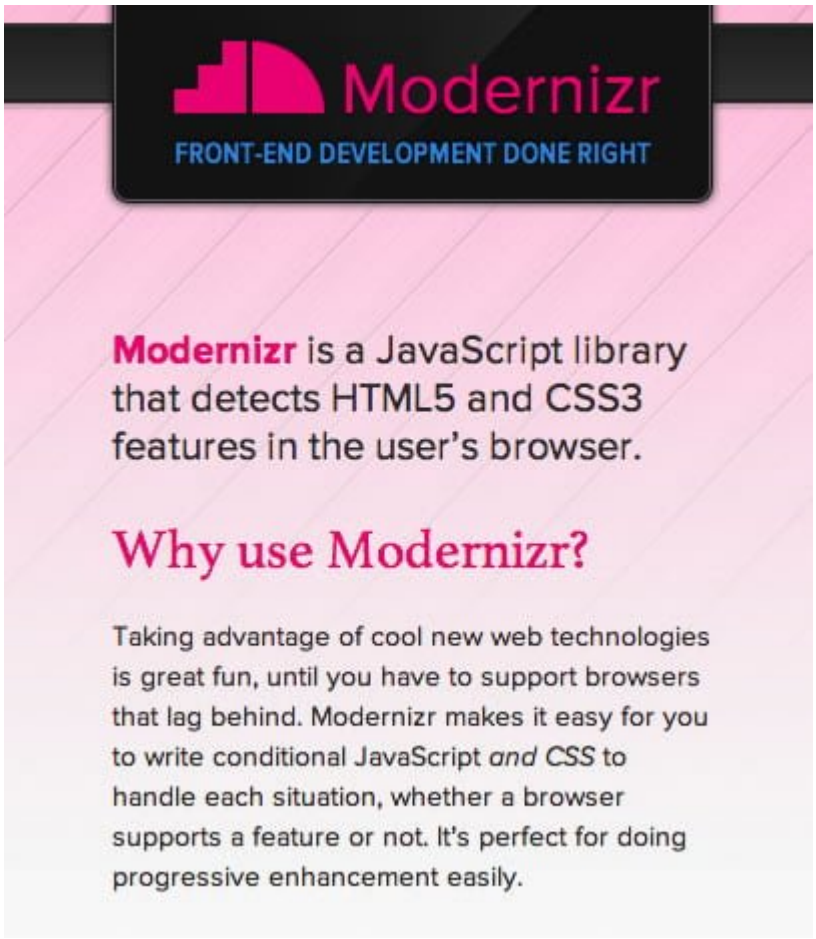
Consejos Modernizr y YepNope

Cada día contamos con nuevas y mejores versiones de los navegadores. Navegadores reales (como Firefox y Chrome) han llegado a su vigésima versión desde hace un tiempon y con una lista impresionante de características crecientes. Por otro lado, los desarrolladores deben asegurar que los productos funcionan bien en los navegadores antiguos (pero no en el obsoleto IE6), ya que muchos clientes de pago pueden estar usando IE8 (o 7?) O máquinas incluso más lentas que no permiten JavaScript o HTML5.

Así que, ¿cómo podemos evitar que nuestros sitios funcionen sin errores? Muchas personas simplemente se tira a las últimas tecnologías, pero un verdadero creyente debe beneficiar a los usuarios más actualizados, dándoles una experiencia impresionante. Si quieres hacer algo como esto, debe hacer un buen uso de las bibliotecas como Modernizr y YepNope.



Modernizr, ¿quién?



Es una impresionante biblioteca JS que le permite detectar características del navegador, por lo que condicionalmente puede cargar mejoras. Puede hacerlo ya sea a través de CSS o JS. El primer paso es la descarga (http://www.modernizr.com/download/) del scipt e instalarlo en su sitio como scripts JS habituales.

```
[code lang=»html»]
<!DOCTYPE HTML>
<HTML lang=”en”>
<HEAD>
<SCRIPT src=”modernizr.js”></SCRIPT>
</HEAD>
<BODY>
</BODY>
</HTML>
[/code]
```

Cómo usar Modernizr en su sitio

Por lo tanto, su uso es muy sencillo. Usted podría probar a través de JS si hay una cierta característica disponible como audio HTML, entonces si no se carga el contenido deseado, si no, hay que cargar las segundas opciones (y por lo general peor).

He aquí un ejemplo simple JS

[code lang=»JavaScript»]
if(Modernizr.audio){
alert(“HTML5 Audio enabled”);
}else{
alert(“HTML5 Audio NOT enabled”);
}
[/code]

También puede probar las propiedades CSS para que evitar la fractura de su sitio. Por ejemplo, si una parte de su sitio depende de un borde de imagen , podría condicionalmente cargarlo, y si no lo tiene puede cargar un CSS alternativo o efecto JS:

[code lang=»JavaScript»]
if(Modernizr.borderimage){
alert(“Border-image support”);
}else{
alert(“Border-image not supported”);
}
[/code]

También puede utilizar la detección de CSS para las propiedades CSS. Esto le ahorrará hacks y el código no confiable que con toda seguridad, rompera en el futuro (en cuanto IE lanza una nueva versión, al menos). Esta versión se basa en el hecho de que Modernizr crea una gran cantidad de clases en el código HTML, generando algo como esto:

[code lang=»html»]
<!DOCTYPE HTML>
<HTML lang=”en”>
<HEAD>
<SCRIPT src=”modernizr.js”></SCRIPT>
</HEAD>
<BODY>
</BODY>
</HTML>
[/code]

Así que usted puede utilizar las clases bonitas en el archivo CSS, así:

[code lang=»css»]
.no-borderimage{
border: 0px;
}
[/code]

Puedes consultar aquí la lista completa de las características detectadas por Modernizr (http://modernizr.com/docs/ # s2).

Sin embargo, es posible que se esté preguntando «Ok, se detecta la falta de una función, pero no se debe proporcionar una alternativa?». Bueno, puedo sentir su inquietud. Es por eso que podemos utilizar otras bibliotecas, los polyfills Permiten «crear» funcionalidad con HTML5 (y otros) en los navegadores que no lo soportan. Sin embargo, si se carga todo lo que necesita en su página sería demasiado lento, por eso podemos hacer uso de otra herramienta: YepNope

YepNope la versión modernizada de Modernizr

YepNope es un gestor condicional, que le permite cargar sólo las secuencias de comandos que se necesitan para un determinado navegador. Es personalizable y se puede descargar en la página del proyecto (http://yepnopejs.com/).

Tiene una sintaxis simple y es importante decir que está integrado con el Modernizr. El siguiente código muestra la sintaxis básica:

[code lang=»js»]
Yepnope({
test: /* condition to be tested, with a boolean result (true / false) */,
yep: /* what will be loaded if condition is true */,
nope: /*what will be loaded if condition is false */,
both: /* what Will be loaded either way */,
load: /* what will be loaded*/,
callback:/* what happens after loading */,
complete: /*what will be loaded when everything else is OK*/,
});
[/code]

Aquí está un ejemplo en acción para la detección de características HTML5

[code lang=»js»]
yepnope({
test: Modernizr.audio,
yep: ‘audio.js’
nope: ‘audio-polyfill.js’
});
[/code]

Por lo tanto va probar automáticamente la característica de el navegador y cargar la versión alternativa, que le ahorra un montón de pruebas if / else. Veamos un ejemplo más cool:

[code lang=»js»]
yepnope({
test: Modernizr.video,
yep: ‘video.css’,
nope: [‘video-html5.css’,’video-polyfill.js’],
callback: function(url, result, key){
if(url == ‘video- html5.css’){
alert(“HTML5 Video Ready”);
}
)};
[/code]

Aquí se pone a prueba la funcionalidad de vídeo HTML, si el navegador lo soporta va cargar el archivo video.css, si no lo hace los archivos vídeo-polyfill.js y Video-html5.css serán cargados, y después de que esto se hace va alertar con un un mensaje diciendo que la función de vídeo está listo ahora.