

Efecto De Transición De Números Con Javascript

Una de las maneras más eficientes de hacer una presentación de la información numérica, se hace mediante la transición de los números. La creación de una animación con [Javascript](#) nos puede llevar una larga cantidad de líneas de código. En el presente ejemplo usaremos un [plugin de Javascript](#), el cual nos facilitara la vida en el momento de implementar un efecto de este tipo.



[Odometer](#) es un **plugin de javascript** que puede ayudarle a mostrar su información numérica de una forma más atractiva con suaves transiciones. Es fácil de usar y es compatible con varios navegadores actuales.

Implementación

[Odometer](#) es un **plugin de javascript** independiente. Sólo tiene que incluir el archivo **js** y su **CSS** en su página con el siguiente código:

```
[html]
```

```
<link rel="stylesheet" href="odometer-theme-car.css" />
<script type="text/javascript" src="odometer.js"></script>
```

```
[/html]
```

¡Listo! Ahora, cualquier elemento que se envuelve con la **clase odometer** se transformará en un [odómetro](#).

En este ejemplo, se usara el estilo de un **odómetro** de tipo coche. **Odometer** viene con otros **seis temas diferentes**, los cuales son: **theme** (por defecto), **digital**, **minimal**, **plaza**,

slot machine y train station. Puedes ir a la [página de demostración](#) para verlos en acción.

Para actualizar el valor, puede utilizar **javascript nativo** o un [código de jQuery](#). En primer lugar, se llama la función **setTimeout**, a continuación, se define el valor actualizado, como en el siguiente fragmento:

```
[js]
```

```
<script>
setTimeout(function(){
odometer.innerHTML = 5555;
}, 1000);
</script>
```

```
[/js]
```

O puede usar un código jQuery de la siguiente forma:

```
[js]
```

```
setTimeout(function(){
$('.odometer').html(5555);
}, 1000);
```

```
[/js]
```

El valor de 1000 en el código significa que el proceso de actualización se ejecutará 1 segundo después de haber cargado completamente la página.

A continuación, agregue la **clase odometer** a cualquier elemento que desee, por ejemplo:

```
[html]
```

```
<p class="odometer">3252</p>
```

```
[/html]
```

Y el valor de 3252 habrá cambiado a 5555 (tal como se definió anteriormente en el **código javascript ó jquery**) con una transición bastante fluida.



Como vemos la aplicación de este efecto es bastante simple, pero si queremos aplicar ciertos cambios?

Opciones De Odometer

Para funciones más avanzadas, **Odometer** ofrece algunas opciones para personalizar. Esto es útil cuando la configuración por defecto no es lo que queremos. Para poder configurar las opciones, primero crear un objeto **odometerOptions** de la siguiente manera:

```
[js]
```

```
<script>
window.odometerOptions = {
  format: '(ddd).dd'
};
</script>
```

```
[/js]
```

La opción **format** afecta a la regla de formato de número, como mostrar un punto decimal antes de ciertos dígitos. (ddd) significa que no hay punto decimal en el número. Y para otras opciones, echa un vistazo a la siguiente lista:

```
[js]
```

```
window.odometerOptions = {
  auto: false, // No te permite inicializar automáticamente
              todo con clase "odómetro"
  selector: '.my-numbers', // Cambiar el selector utiliza para
                           encontrar automáticamente las cosas para ser animados
```

```
format: '(,ddd).dd', // Cambiar la forma de indicar los grupos
de dígitos, y cuántos dígitos se muestran tras el punto
decimal
duration: 3000, // Cambiar el tiempo que el javascript espera
la animación CSS para tomar
theme: 'car', // Especifique el tema (si tiene más de un
archivo de temas css en la página)
animation: 'count' //
El recuento es un método de animación más simple que sólo
incrementa el valor,
/ / Utilizar cuando usted está buscando algo más sutil.
};

[/js]
```

Para aquellos que suelen presentar la información numérica y le gustaría hacerlo más llamativo, **Odometer** es una buena opción.