# 5\_1 EJEMPLOS DE CONTENIDO INTERACTIVO

### ICOMOON

Fuente de iconos: <https://icomoon.io/>

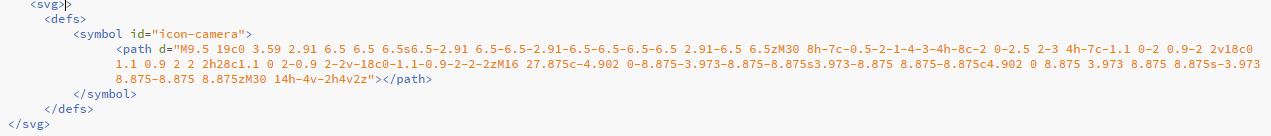
1. Ir a IconPacks🡪 IcoMoon App🡪
2. Seleccionar el icono
3. Crear enlace online (pero solo dur 24 horas) y copiar el link
4. Generar svg y descargarlo

2.b)

2.b.a) Linkar la imagen descargada (svg)



2.b.b) Generar el icono en el html copiando el texto que viene en demo.html



<svg>: Para indicar que se va a trabajar con gráficos vectoriales

<defs>: Para definir un elemento que se utilizará más tarde

<symbol> con id. Se utilizará para definir/dibujar dentro el elemento en base al parámetro Path (<https://www.w3schools.com/graphics/svg_path.asp> )

El id se utilizará después con la etiqueta use xlink:href



### AWESONE

Descargar: <https://fontawesome.com/start>

Linkar la fuente: <link href="/your-path-to-fontawesome/css/all.css" rel="stylesheet"> <!--load all styles -->

### ANIMATE

1. Linkar librería
2. Descargar y linkar después animate.min.css
3. Linkar la librería: <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/animate.css/4.1.1/animate.min.css"/>
4. Elegir el efecto y aplicar la clase a la etiqueta. Siempre poner previamente la clase animate\_\_animated



### FANCY BOX

1. Html válido
2. Descargar si se quiere desde: [http://fancyapps.com](http://fancyapps.com/fancybox/3/)
3. Incluir los fichero js

<script src="https://cdn.jsdelivr.net/npm/jquery@3.5.1/dist/jquery.min.js"></script>

<script src="https://cdn.jsdelivr.net/gh/fancyapps/fancybox@3.5.7/dist/jquery.fancybox.min.js"></script>

1. Linkar librería fancy

<link rel="stylesheet" href="https://cdn.jsdelivr.net/gh/fancyapps/fancybox@3.5.7/dist/jquery.fancybox.min.css" />

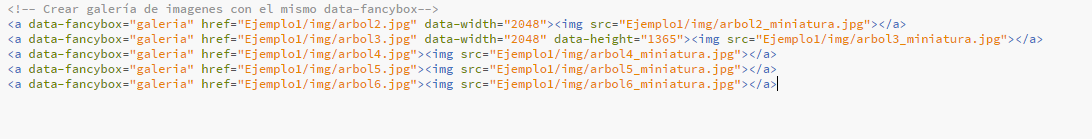
IMÁGENES Y VIDEOS

1. Crear un elemento enlace <a>
2. Elemento único. Abre un cuadro con la imagen



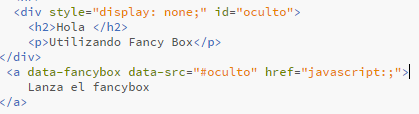
1. Galería de imágenes. Se enlazan en una galería con flechas que pasan las imágenes.

* Deben tener el mismo atributo data-fancybox
* La imagen en miniatura en la etiqueta img. La imagen grande en el href del a
* Se puede controlar la altura y anchura con los atributors data-width y data-height. De no ser así sacará el tamaño original.



ELEMENTO INLINE

1. Crear un elemento inline (un cuadro que salta)
   1. Crear un elemento oculto (con display:none) con un id
   2. Crear el enlace que lo lanzará referenciando a su id en data-src



### OWLCAROUSEL

1. Incluir css:

<link rel="stylesheet" href="owlcarousel/owl.carousel.min.css">

<link rel="stylesheet" href="owlcarousel/owl.theme.default.min.css">

1. Incluir JS en el footer

<script src="jquery.min.js"></script>

<script src="owlcarousel/owl.carousel.min.js"></script>

1. Crear en el html los divs/img que utilizarás para el carrosuel

<div class="owl-carousel">

<div> IMAGEN1 </div>

<div> IMAGEN2</div>

<div> IMAGEN3</div>

<div> IMAGEN4</div>

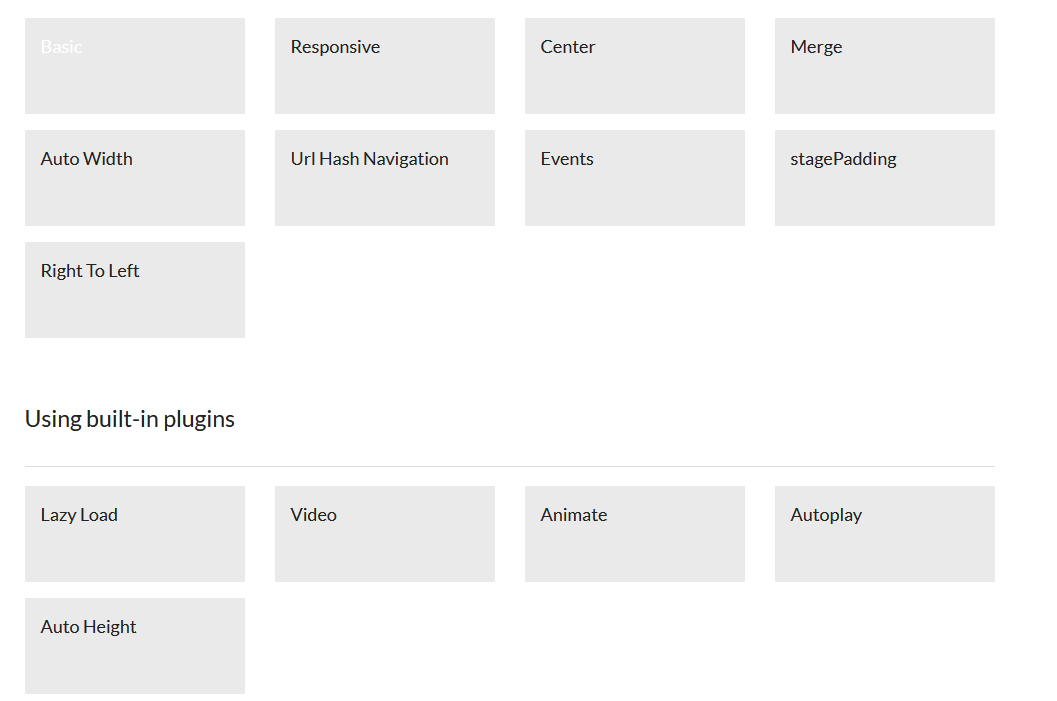
<div> IMAGEN5</div>

<div> IMAGEN6</div>

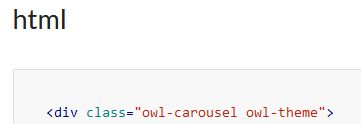
</div>

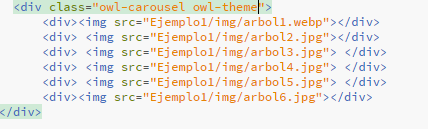
1. Elegir de la web de Demos el carrosuel que se quiere e incluir el código javascript









* 



### NORMALIZE

Fichero que normaliza el estilo en todos los navegadores (para evitar fuentes, tamaños, paddings, etc. diferentes). Armoniza la visualización inicial de los navegadores y corrige algunos errores.

1. Descargar el fichero desde su [página oficial](http://necolas.github.io/normalize.css/)  y colocarlo en la carpeta de nuestra web.
2. Linkarlo antes de cualquier otro estilo: <link rel="stylesheet" href="normalize.css" />

### SWIPER

1. Ir a la página [Swiper Demos (swiperjs.com)](https://swiperjs.com/demos/) y elegir el carrousel que queremos.
2. Descargar/Copiar el código
3. Linkar el estilo de swipper: <link rel="stylesheet" href="../package/swiper-bundle.min.css">

### MODERNIZR

Es una librería JavaScript que nos permite conocer la compatibilidad del navegador con tecnologías **HTML5** y **CSS3**, lo que nos permitirá desarrollar sitios web que se adapten a las capacidades cada navegador.

Modernizr se ejecuta automáticamente, solo con incluir la librería dentro de la página HTML.

1. Desde la página de descargas de Modernizr, una vez que sepamos qué funcionalidades vamos a usar y por lo tanto necesitamos detectar, nos permite construir **una versión personalizada que contenga tan solo las funciones deseadas**, con lo que minimizaremos el tamaño del archivo que necesitarán descargar los visitantes de nuestra web
2. Linkar la librería:
3. Modo local (descargado)

<script type="text/javascript" src="modernizr-latest.js"></script>

1. Online (pueden estar más desactualizados):

<script type="text/javascript" src="//cdnjs.cloudflare.com/ajax/libs/modernizr/2.8.3/modernizr.min.js"></script>

1. Añadir al html la etiqueta no-js.

<html class="no-js">

Al ejecutarse Modernizr, si el navegador soporta JavaScript, se reemplazará "no-js" por "js"

1. Modernizr no sólo cambia no-js por js sino que además incluye en la lista de clases los nombres de cada una de las características que ha conseguido detectar. De esta manera, la etiqueta <html> en tiempo de ejecución presentará un aspecto similar a este:

<html class="js canvas canvastext no-geolocation rgba hsla multiplebgs borderimage borderradius boxshadow opacity cssanimations csscolumns cssgradients cssreflections csstransforms csstransforms3d csstransitions video audio localstorage sessionstorage webworkers applicationcache fontface">

Las características soportadas simplemente se indican por su nombre, mientras que las no soportadas llevan el prefijo no-

* El cuerpo de la página contiene un <div> al que queremos aplicarle un reborde con las esquinas redondeadas. Podríamos simplemente añadir en el estilo la propiedad border-radius, y se despreciaría en caso de que el navegador no la soportase.

Pero lo que no conseguiríamos de esa manera es cambiar el color o el grosor del borde en aquellos casos en los que no se pueda redondear.

* En cambio, con Modernizr podemos añadir un segundo estilo basado en la clase borderradius que Modernizr añadió al <html> y por tanto fue heredado por el <div>. En ese segundo estilo, cambiamos las propiedades del borde del <div>

Ejemplo:

<head>

[...]

<style type="text/css">

#content {

background: yellow;

border: 3px outset red;

padding: 20px;

}

.borderradius #content {

border: 1px solid green;

-webkit-border-radius: 10px;

-moz-border-radius: 10px;

border-radius: 10px;

}

</style>

</head>

<body class="**no-js**">

[...]

<div id="content">

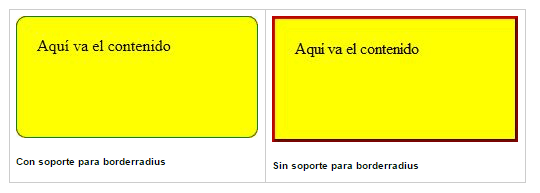
[...]

</div>

[...]

</body>

Resultado:



### PARALLAX

Parallax: efecto de desplazamiento que podemos dar a los elementos de un sitio web al ir bajando en la página.

Ejemplos:

* <http://falconmasters.com/demos/parallax_sencillo/>
* <http://matthew.wagerfield.com/parallax/>
* <https://www.madwell.com/>

Por ejemplo al ir bajando en una página y que el fondo se desplace a un tiempo diferente al resto del contenido.

1. Cosas que necesitaremos:

* Una imagen muy grande o textura
* 3 archivos:
  + index.html
  + estilos.css
  + parallax.js
* Jquery

**Recursos:**

Jquery: <http://code.jquery.com/jquery-latest.js>

Texturas: [6 Sitios web donde conseguir texturas para tu sitio web](http://www.falconmasters.com/recursos-herramientas/6-sitios-web-texturas-fondos/)

1. Enlazar los archivos en la web

<link rel="stylesheet" href="estilos.css">

<script src="<http://code.jquery.com/jquery-latest.js>"></script>

<script src="main.js"></script>

1. Crear el html (debe tener suficiente contenido para poder hacer scroll)
2. En el css. La parte importante son las propiedades background dentro de nuestro body porque con ella cargamos la imagen y le damos las propiedades necesarias.:

body {

background:url(bg.jpg); /\* Nuestra imagen de fondo \*/

background-repeat:no-repeat; /\* Indicamos que no se repetira \*/

background-size:cover; /\* Encajamos la imagen al 100% del ancho \*/

background-attachment: fixed; /\* Establecemos una posicion fija para la imagen \*/

1. Si trabajas con texturas tienes que cambiar las propiedades de background y dejar el body de la siguiente manera:

body {

background:url(bg.jpg); /\* Nuestra textura \*/

background-repeat:repeat; /\* Indicamos que la textura se repetira \*/

background-attachment: fixed; /\* Establecemos una posicion fija para la textura \*/

/\* Eliminamos la propiedad de background-size \*/

}

1. Mover el fondo con JS y background.

Lo siguiente por hacer es lograr que el fondo de nuestro sitio baje mas lento a comparacion del contenido. Primero como recordaras, establecimos un position:fixed; a nuestra imagen, por lo que se queda fija a pantalla. Aunque esta fija, nosotros vamos a desplazarla con Jquery, esto mediante la propiedad background-position:;

background-position nos permite mover el fondo de nuestro sitio mediante 2 valores, en el primer valor indicamos cuanto desplazaremos de izquierda a derecha y en el segundo cuanto desplazaremos de arriba a abajo. Tocaremos solo el segundo que nos permite desplazarlo de arriba a abajo.

Para desplazarlo lo haremos con un calculo y básicamente lo que haremos es tomar la posición en px del scroll y multiplicarla por un valor inferior al 1.

De esta forma el fondo se moverá mas despacio que el contenido.

$(document).ready(function(){

$(window).scroll(function(){

var barra = $(window).scrollTop();

var posicion =  (barra \* 0.10);

$('body').css({

'background-position': '0 -' + posicion + 'px'

});

});

});

Para cambiar la velocidad de movimiento de la pantalla, cambia el valor 0.1 por un valor más grande (cuidado, si tienes mucho texto, te puedes quedar sin imagen si el desplazamiento es muy grande). Valor recomendado: 0.1-0.4

1. Desactivar en caso de pantallas pequeñas el background-size:cover

$(document).ready(function(){

var ancho = $(window).width();

// Si trabajamos con una imagen desactivamos el background-size:cover;

if (ancho <= 1350){

$('body').css({

'background-size': 'initial'

});

}

// Resto del código parallax

});

1. Mover el fondo utilizando texturas

$(document).ready(function(){

$(window).scroll(function(){

var barra = $(window).scrollTop();

var posicion = barra \* 0.40;

$('body').css({

'background-position': '0 ' + posicion + 'px'

});

});

});