



clase n°6

CSS

<animate a/> revolución*
programar digital_

floats

Originalmente pensada para flotar imágenes dentro de bloques de texto, la propiedad float se convirtió en una de las herramientas más usadas para crear diseños multicolumna en las páginas web. Con la llegada de **Flexbox** y **Grid** ha vuelto ahora a su propósito original.

Un ejemplo de float simple

```
<h1>Ejemplo simple de caja flotante</h1>

<div class="box">Caja flotante</div>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luctus aliquam dolor, eu lacinia lorem placerat vulputate. Duis felis orci, pulvinar id metus ut, rutrum luctus orci. Cras porttitor imperdiet nunc, at ultricies tellus laoreet sit amet. </p>

<p>Sed auctor cursus massa at porta. Integer ligula ipsum, tristique sit amet orci vel, viverra egestas ligula. Curabitur vehicula tellus neque, ac ornare ex malesuada et.</p>

<p>Nam vulputate diam nec tempor bibendum. Donec luctus augue eget malesuada ultrices. Phasellus turpis est, posuere sit amet dapibus ut, facilisis sed est. Nam id risus quis ante semper consectetur eget aliquam lorem. Vivamus tristique elitdolor, sed pretium metus suscipit vel.</p>
```

Generar texto **Lorem Ipsum**: Escribir '**Lorem**' y pulsar **TAB**. Se puede hacer dentro de etiquetas **<p>** o donde te convenga.



Si guardas y recargas la página, verás algo parecido a lo que esperarías: la caja se encuentra por encima del texto, en un flujo normal. Para flotar el texto alrededor, añade las propiedades *float* y *margin-right* a la regla *.box*:

Ahora, si guardas y recargas, podrás ver algo parecido a lo siguiente:

Ejemplo simple de caja flotante

Caja flotante

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luctus aliquam dolor, eu lacinia lorem placerat vulputate.

Duis felis orci, pulvinar id metus ut, rutrum luctus orci. Cras porttitor imperdiet nunc, at ultricies tellus laoreet sit amet. Sed auctor cursus massa at porta. Integer ligula ipsum, tristique sit amet orci vel, viverra egestas ligula. Curabitur vehicula tellus neque, ac ornare ex malesuada et. In vitae convallis lacus. Aliquam erat volutpat. Suspendisse ac imperdiet turpis.

Aenean finibus sollicitudin eros pharetra congue. Duis ornare egestas augue ut luctus. Proin blandit quam nec lacus varius

commodo et a urna. Ut id ornare felis, eget fermentum sapien.

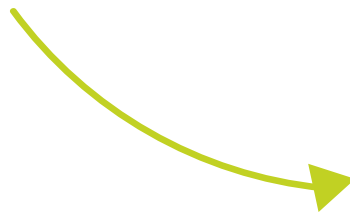
```
body {  
  width: 90%;  
  max-width: 900px; margin: 0 auto;  
  font: .9em/1.2 Arial, Helvetica, sans-serif  
}  
.box {  
  width: 150px; height: 100px; border-radius: 5px;  
  background-color: rgb(207,232,220); padding: 1em;  
}
```

```
float: left;  
  
margin-right: 15px;
```

Añade una clase especial al primer párrafo de texto, el que sucede inmediatamente a la caja flotante, y luego añade en tu CSS las siguientes reglas. Esto le dará al párrafo siguiente un color de fondo.

```
.special {  
  background-color: rgb (79, 185, 227);  
  padding: 10px;  
  color: #fff;  
}
```

Para que el efecto sea más fácil de ver, cambia el margin-right de tu elemento flotante a margin, para obtener espacio alrededor del elemento flotante. Verás que el fondo del párrafo pasa justo por debajo de la caja flotante.



Ejemplo simple de caja flotante

Caja flotante


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luctus aliquam dolor, eu lacinia lorem placerat vulputate.

Duis felis orci, pulvinar id metus ut, rutrum luctus orci. Cras porttitor imperdiet nunc, at ultricies tellus laoreet sit amet. Sed auctor cursus massa at porta. Integer ligula ipsum, tristique sit amet orci vel, viverra egestas ligula. Curabitur vehicula tellus neque, ac ornare ex malesuada et. In vitae convallis lacus. Aliquam erat volutpat. Suspendisse ac imperdiet turpis. Aenean finibus sollicitudin eros pharetra congue. Duis ornare egestas augue ut luctus. Proin blandit quam nec lacus varius commodo et a urna. Ut id ornare felis, eget fermentum sapien.

limpiando floats

Hemos visto que el flotador se elimina del flujo normal y que otros elementos se mostrarán a su lado, por lo tanto, si queremos evitar que un elemento siguiente se mueva hacia arriba, debemos limpiar el float. Esto se logra con la propiedad clear.

En el HTML añade una clase cleared al segundo párrafo debajo del elemento flotante. Luego añade lo siguiente a tu CSS



```
.cleared {  
  clear: left;  
}
```

flexbox

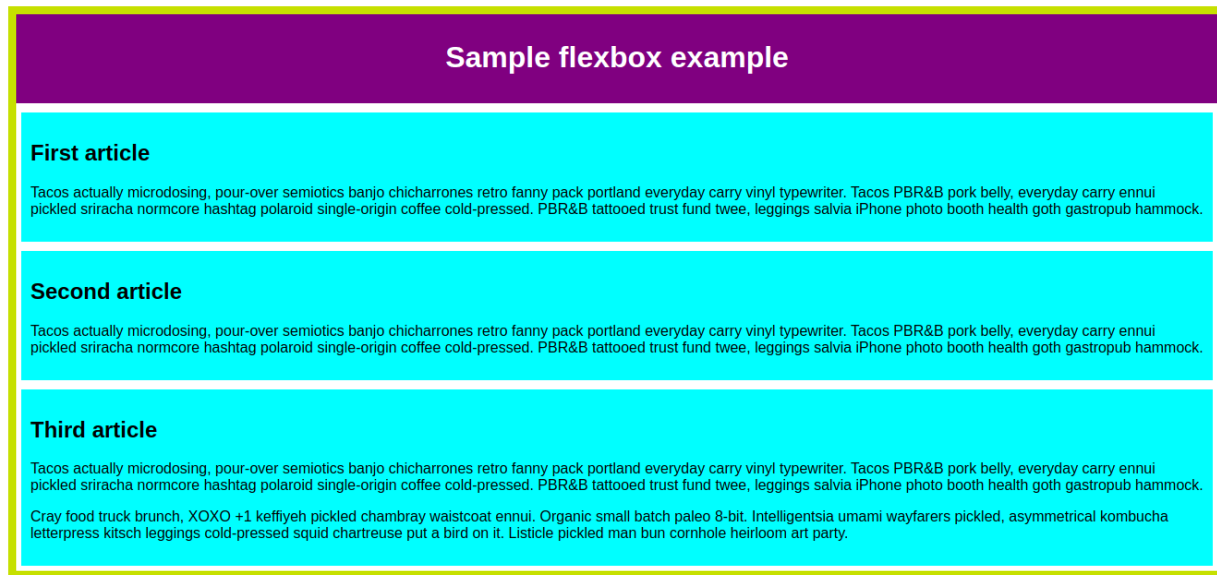
Es un método de diseño unidimensional para organizar elementos en filas o columnas. Los elementos se flexionan (expanden) para llenar espacio adicional o se encogen para que quepan en espacios más pequeños.

¿por qué flexbox?

Durante mucho tiempo, las únicas herramientas confiables compatibles con varios navegadores disponibles para crear diseños CSS fueron características como flotadores y posicionamiento. Estos funcionan, pero de alguna manera también son limitantes y frustrantes. Los siguientes diseños de diseño simples son difíciles o imposibles de lograr con tales herramientas de cualquier manera conveniente y flexible:

- Centrar verticalmente un bloque de contenido dentro de su padre.
- Hacer que todos los hijos de un contenedor ocupen una cantidad igual del ancho / alto disponible, independientemente de la cantidad de ancho / alto disponible.
- Hacer que todas las columnas en un diseño de varias columnas adopten la misma altura, incluso si contienen una cantidad diferente de contenido.

presentamos un ejemplo sencillo



Diseñamos una pagina con un elemento **<header>** con un encabezado de nivel superior en su interior y un elemento **<sección>** que contiene tres **<artículos>**. Vamos a usarlos para crear un diseño de tres columnas bastante estándar. Especificación de los elementos que se deben diseñar como cuadros flexibles. Para empezar, debemos seleccionar qué elementos se presentarán como cajas flexibles. Para ello, establecemos un valor especial de visualización en el elemento padre de los elementos a los que desea afectar. En este caso queremos exponer los elementos **<artículo>**, por lo que establecemos esto en la **<sección>**:

```
section {  
  display : Flex;  
}
```

Esto hace que el elemento **<sección>** se convierta en un recipiente flexible y sus hijos se conviertan en elementos flexibles. *El resultado de esto debería ser algo así:*

Sample flexbox example

First article

Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth health goth gastropub hammock.

Second article

Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth health goth gastropub hammock.

Third article

Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth health goth gastropub hammock.

Cray food truck brunch, XOXO +1 keffiyeh pickled chambray waistcoat ennui. Organic small batch paleo 8-bit. Intelligentsia umami wayfarers pickled, asymmetrical kombucha letterpress kitsch leggings cold-pressed squid chartreuse put a bird on it. Listicle pickled man bun cornhole heirloom art party.

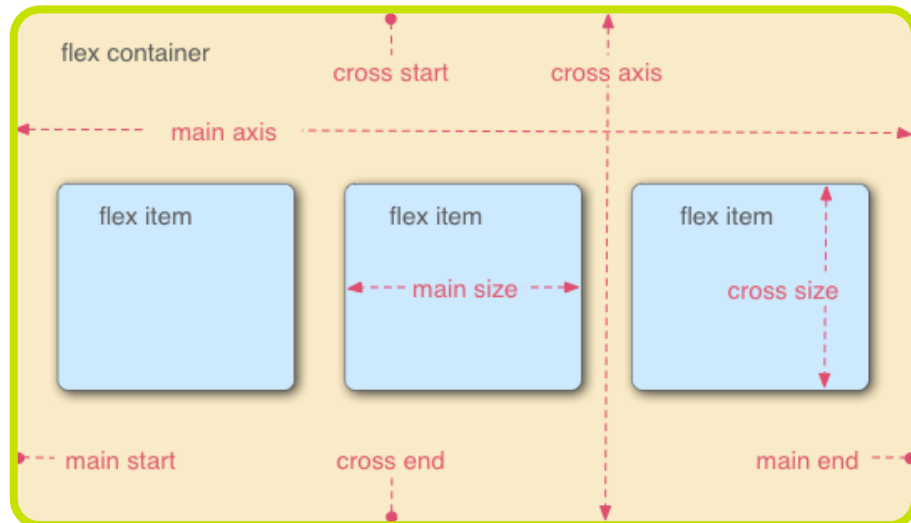
Tenemos nuestro diseño de columnas múltiples con columnas de igual tamaño, y las columnas tienen la misma altura. Esto se debe a que los valores predeterminados dados a los elementos flexibles (*los hijos del contenedor flexible*) están configurados para resolver problemas comunes como este.

Cuando los elementos se presentan como elementos flexibles, se colocan **a lo largo de dos ejes**:

El **eje principal** es el eje que se ejecuta en la dirección en la que se presentan los elementos flexibles (por ejemplo, como filas a través de la página o columnas en la página). El inicio y el final de este eje se denominan inicio principal y extremo principal.

El **eje transversal** es el eje que corre perpendicular a la dirección en la que se colocan los elementos flexibles. El inicio y el final de este eje se denominan inicio cruzado y extremo cruzado. El elemento padre que se ha establecido en él (*la <sección> en nuestro ejemplo*) se denomina **contenedor flexible**. `display: flex`

Los artículos dispuestos como cajas flexibles dentro del contenedor flexible se denominan **elementos flexibles** (*los elementos <artículo> en nuestro ejemplo*).



¿columnas o filas?

Flexbox proporciona una propiedad llamada `flex-direction` que especifica en qué dirección corre el eje principal (en qué dirección se colocan los hijos de flexbox). De forma predeterminada, esto se establece en `row`, lo que hace que se presenten en una fila en la dirección en la que funciona el idioma predeterminado de su navegador (de izquierda a derecha, en el caso de un navegador en inglés).

Probemos agregando la siguiente línea de código: `flex-direction: column;`

Para poder probar todas las posibilidades que ofrece Flexbox CSS, vamos a crear una capa `class="contenedor"` que hará de padre y siete capas `class="elemento"` numeradas. ¿Y por que siete? Para los ejemplos nos viene bien que sea un número impar para ver tanto el ordenamiento, como la adaptación y su resultado, y siete también para ver un listado de elementos no muy corto, aunque puedes probar a colocar cualquier número de elementos.

Realiza el código para ver lo que se muestra en el siguiente ejemplo:

EJEMPLO 1:

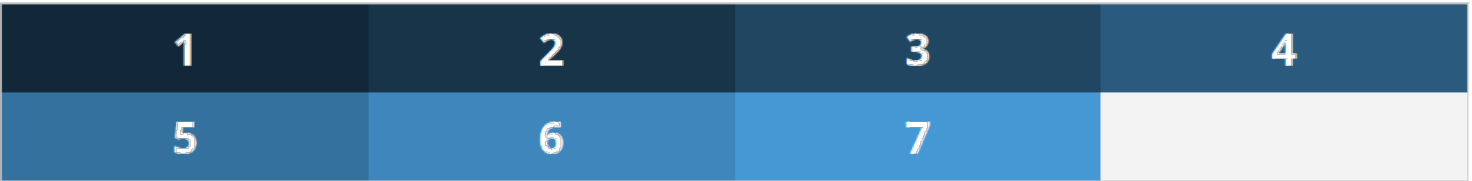


EJEMPLO 2:



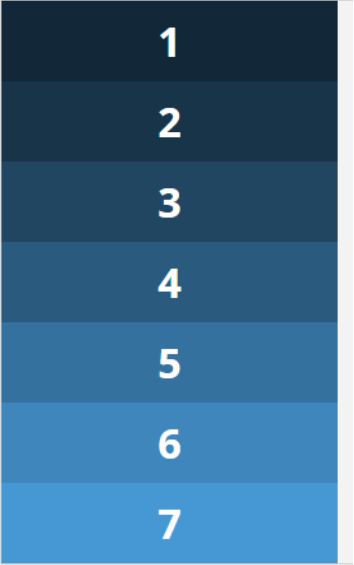
`flex-direction: row-reverse;`

EJEMPLO 5:



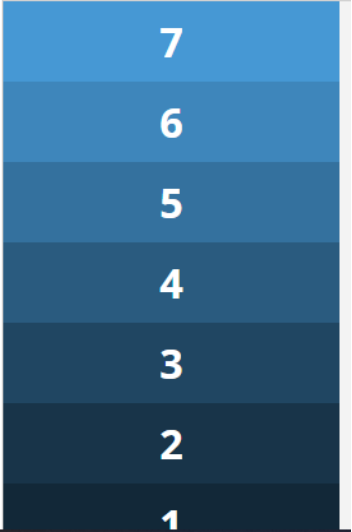
`flex-direction: column;`

EJEMPLO 3:



`flex-direction: column;`

EJEMPLO 4:



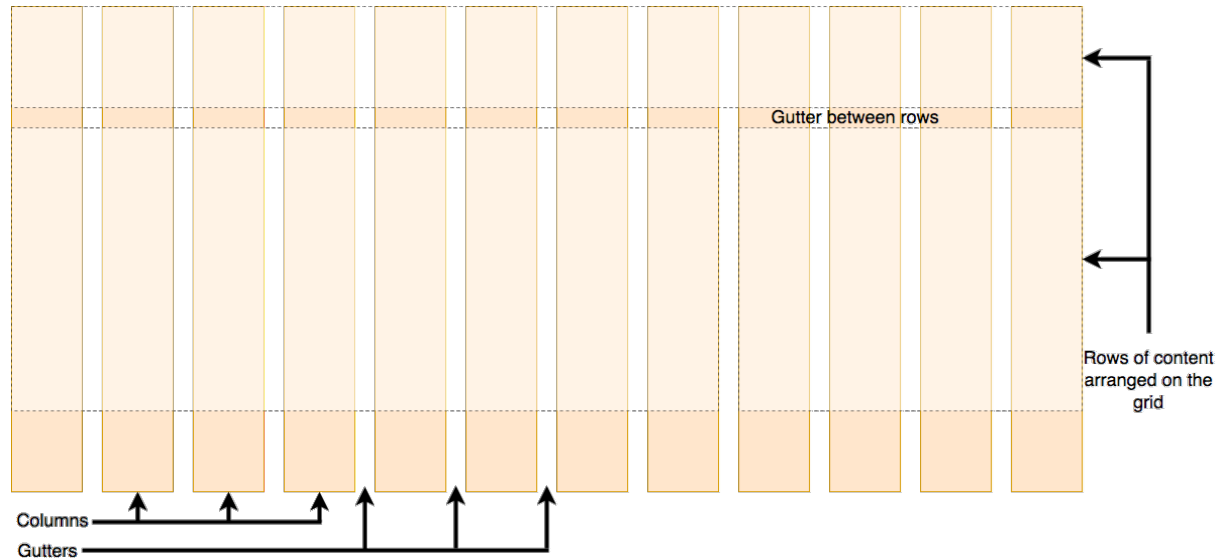
`flex-direction: column-reverse;`

grids

CSS Grid Layout es un sistema de diseño bidimensional para la web. Le permite diseñar el contenido en filas y columnas. Tiene características que hacen que los diseños complejos de construcción sean sencillos.

¿qué es el diseño de cuadrícula?

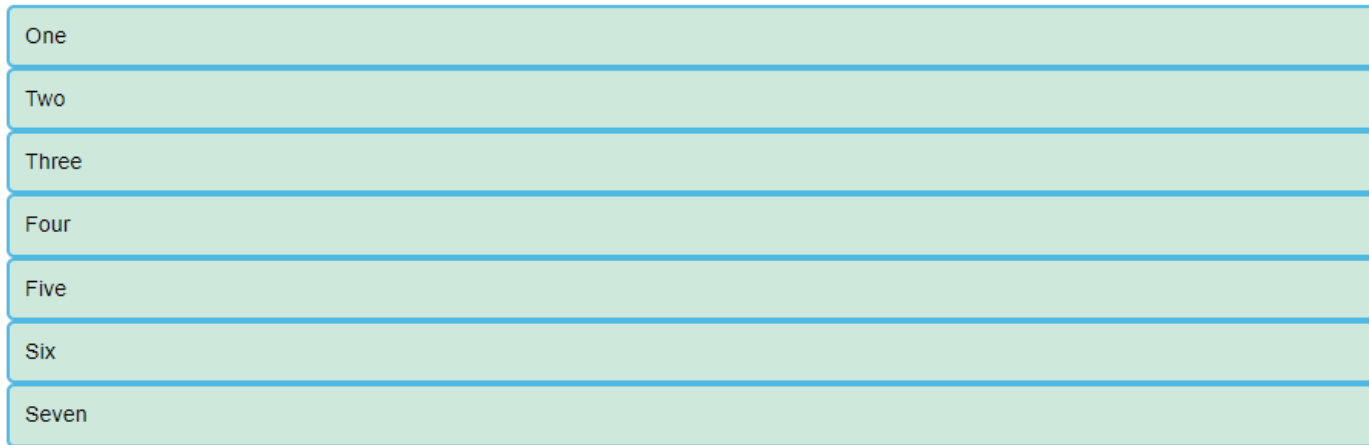
Una cuadrícula es una colección de líneas horizontales y verticales que crean un patrón contra el cual podemos alinear nuestros elementos de diseño. Nos ayudan a crear diseños en los que nuestros elementos no saltarán ni cambiarán de ancho a medida que avanzamos de una página a otra, proporcionando una mayor consistencia en nuestros sitios web. Una cuadrícula normalmente tendrá columnas, filas y, a continuación, espacios entre cada fila y columna. Los huecos se conocen comúnmente como canaletas.



Realizamos el siguiente ejemplo

```
.container {  
  display: grid;  
}
```

Simple grid example



Para lograr los bordes redondeados
usamos `border-radius`

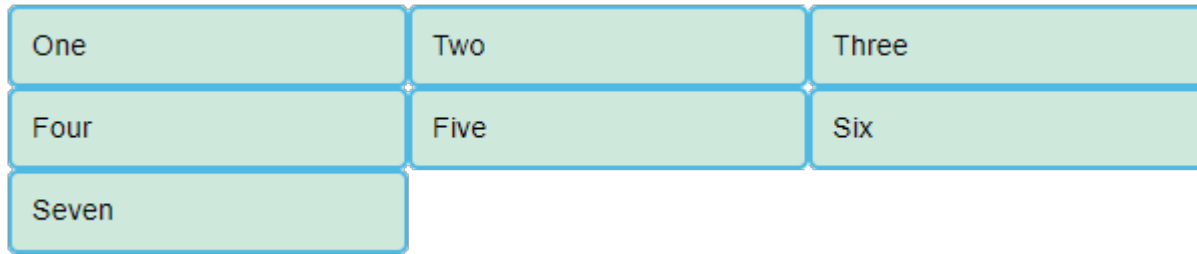


A diferencia de Flexbox, los elementos no se verán inmediatamente diferentes, sus elementos continuarán mostrándose uno debajo del otro como lo hacen en el flujo *normal*. `display: grid`

Para ver algo que se parezca más a una cuadrícula, necesitaremos agregar algunas columnas a la cuadrícula. Agreguemos tres columnas de 200 píxeles. Puede utilizar cualquier unidad de longitud o porcentaje para crear estas pistas de columna.

```
.container {  
  display: grid;  
  grid-template-columns: 200px 200px 200px;  
}
```

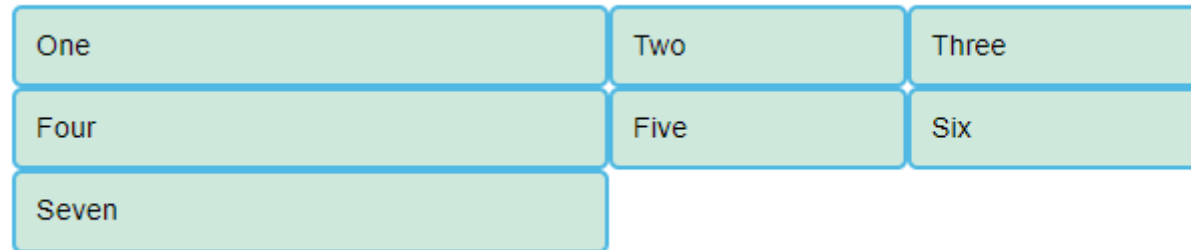
Luego de cargar dicho código, cargar la página. Debería ver que los elementos se han reorganizado de tal manera que haya uno en cada celda de la cuadrícula.



Rejillas flexibles con la unidad fr

Además de crear cuadrículas utilizando longitudes y porcentajes, podemos usar la unidad para dimensionar de manera flexible las filas y columnas de la cuadrícula. Esta unidad representa una fracción del espacio disponible en el contenedor de la cuadrícula.fr

```
.container {  
  display: grid;  
  grid-template-columns: 2fr 1fr 1fr;  
}
```



realizar el siguiente ejemplo en conjunto

```
header {  
  grid-column: 1 / 3;  
  grid-row: 1;  
  
}  
article {  
  grid-column: 2;  
  grid-row: 2;  
  
}  
  
aside {  
  grid-column: 1;  
  grid-row: 2;  
  
}  
  
footer {  
  grid-column: 1 / 3;  
  grid-row: 3;  
}
```

This is my lovely blog

Other things

Nam vulputate diam nec tempor bibendum. Donec luctus augue eget malesuada ultrices. Phasellus turpis est, posuere sit amet dapibus ut, facilisis sed est.

My article

Duis felis orci, pulvinar id metus ut, rutrum luctus orci. Cras porttitor imperdiet nunc, at ultricies tellus laoreet sit amet. Sed auctor cursus massa at porta. Integer ligula ipsum, tristique sit amet orci vel, viverra egestas ligula. Curabitur vehicula tellus neque, ac ornare ex malesuada et. In vitae convallis lacus. Aliquam erat volutpat. Suspendisse ac imperdiet turpis. Aenean finibus sollicitudin eros pharetra congue. Duis ornare egestas augue ut luctus. Proin blandit quam nec lacus varius commodo et a urna. Ut id ornare felis, eget fermentum sapien.

Nam vulputate diam nec tempor bibendum. Donec luctus augue eget malesuada ultrices. Phasellus turpis est, posuere sit amet dapibus ut, facilisis sed est. Nam id risus quis ante semper consectetur eget aliquam lorem. Vivamus tristique elit dolor, sed pretium metus suscipit vel. Mauris ultricies lectus sed lobortis finibus. Vivamus eu urna eget velit cursus viverra quis vestibulum sem. Aliquam tincidunt eget purus in interdum. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

Contact me@example.com

revolución*
digital_