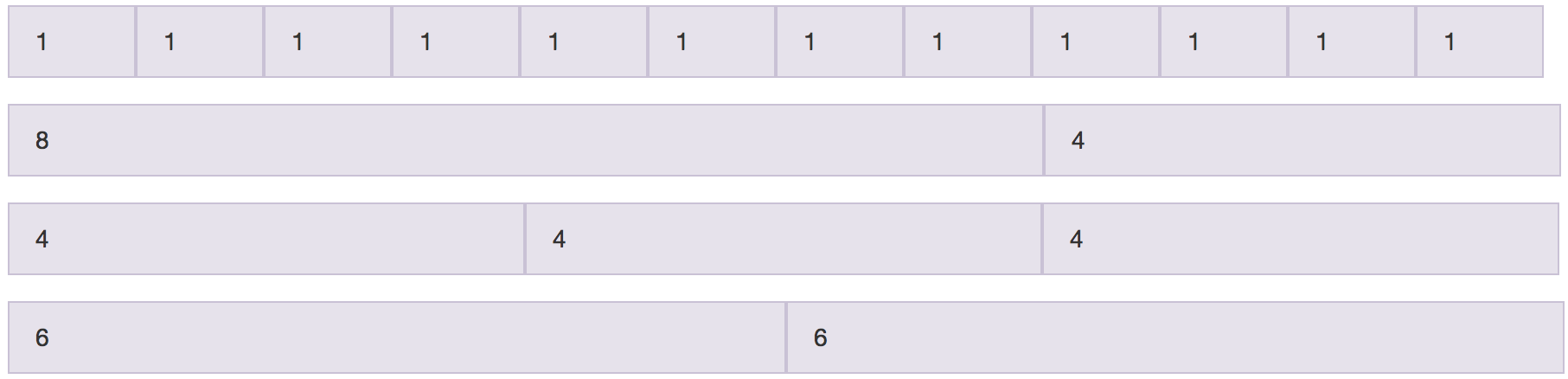
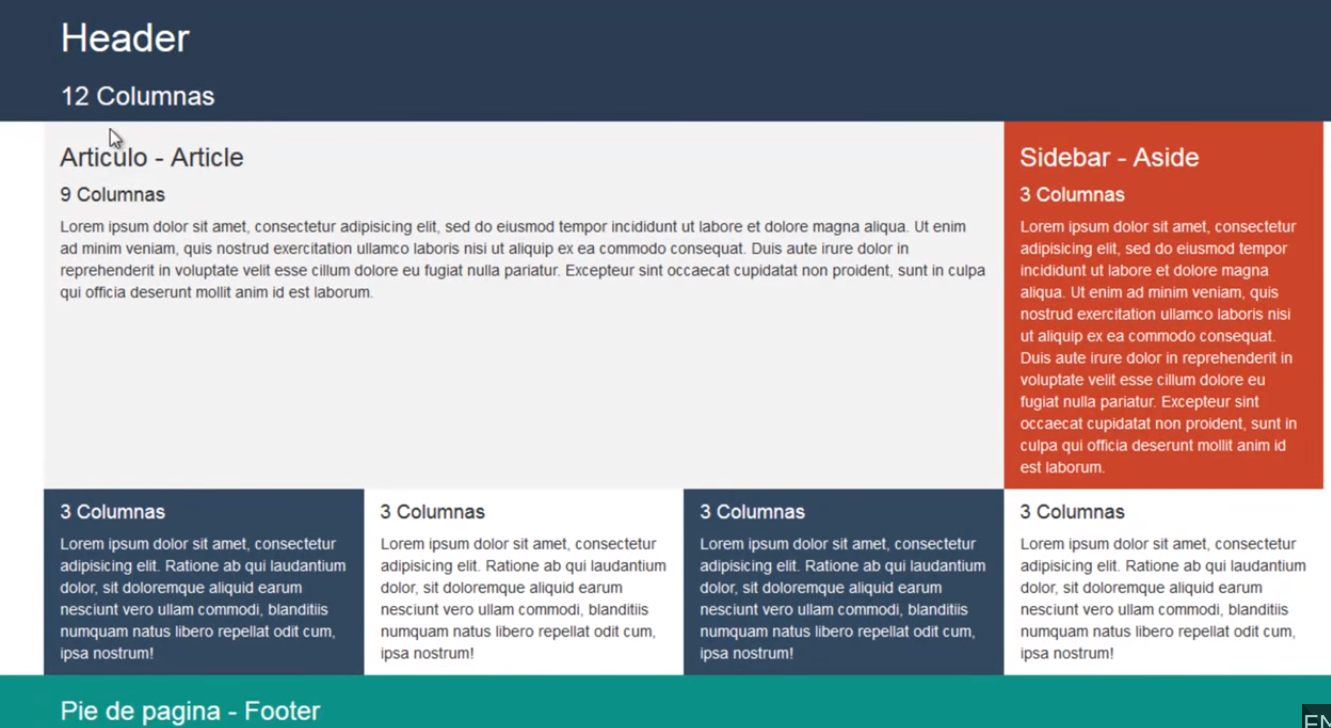
# Sistema de grillas en Bootstrap

* La grilla en Bootstrap se basa en 12 columnas (ya que 12 es divisible por muchos números: 6,4,3,2 y 1).
* La suma de columnas dentro de una fila siempre tiene que ser igual a 12. Por ejemplo:



### Ejemplo

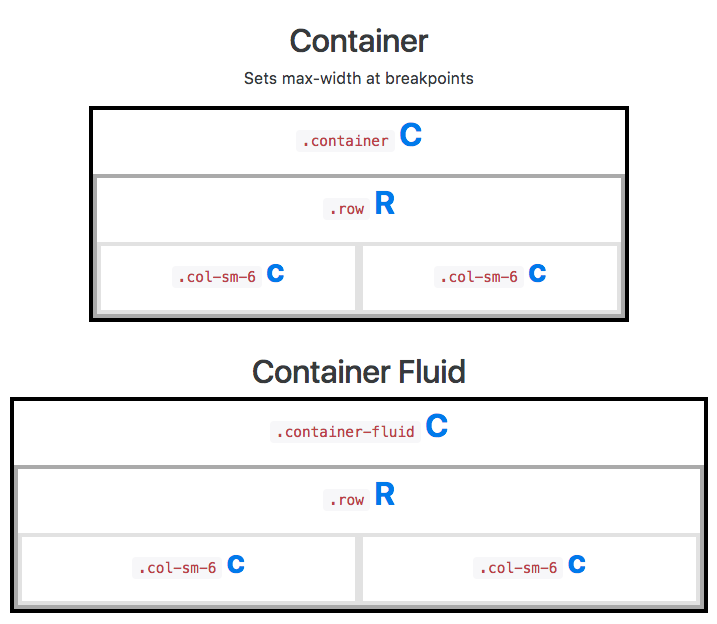


* La grilla de Bootstrap tiene tres elementos principales: Containers, Rows y Columns (C.R.C.)

## Container

Las grillas están contenidas dentro de un elemento que utilice las clases contenedoras. Existen dos tipos:

1. .container-fluid: tienen un ancho adaptado al ancho del navegador
2. .container: tiene un ancho fijo centrado



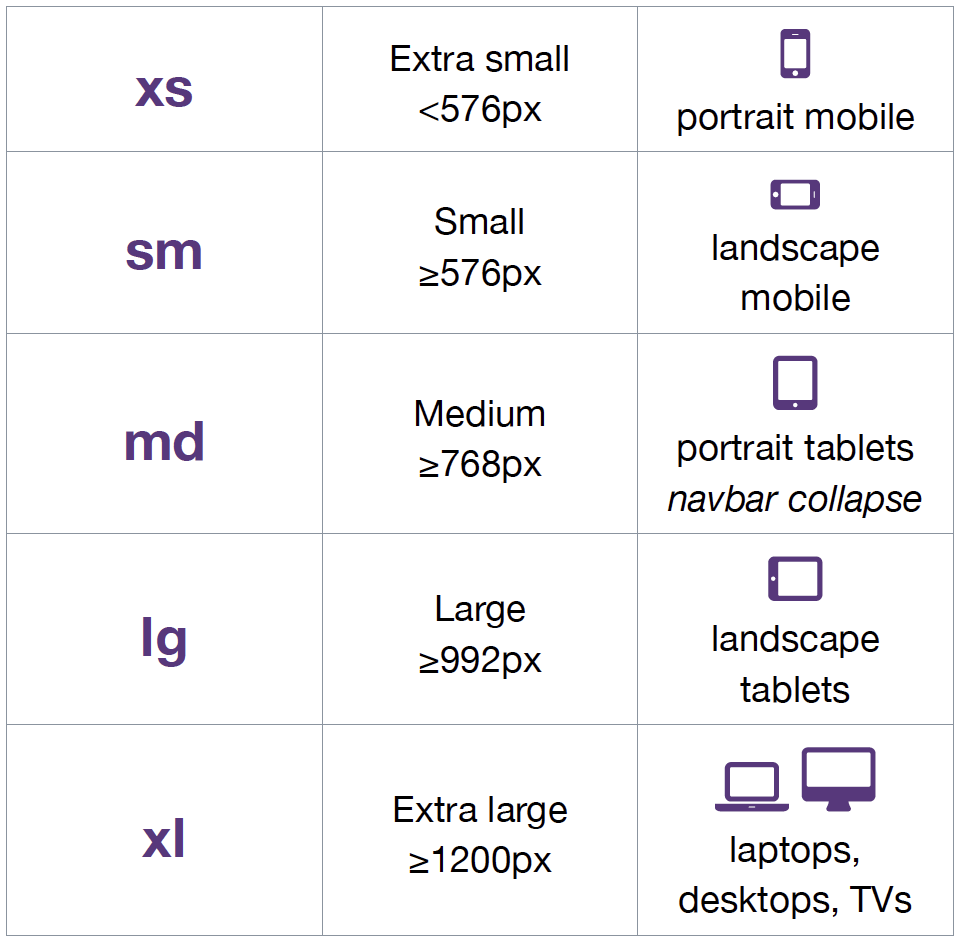
## Columnas

* Las columnas las vamos a definir mediante la clase .col.
* El nombre de la clase .col se forma de la siguiente manera:



Donde:

* + col: indica que se trata de una clase de columnas (esto siempre es así)
  + TamañoDePantalla: indica que esta clase se va aplicar solo cuando el navegador se encuentre en dicho tamaño de pantalla. Las opciones son: *xs, sm, md, lg* y *xl*.   
    Equivalen a los siguientes tamaños:



* + TamañoDeLaColumnaEnCantidadDeColumnas: indica el tamaño que va a tomar en columnas el elemento cuando se encuentre en el tamaño de pantalla definido anteriormente

### Responsive

Para un elemento podemos definir varias clases y se van a ir aplicando la correcta en función del tamaño de la pantalla.

### Ejemplo



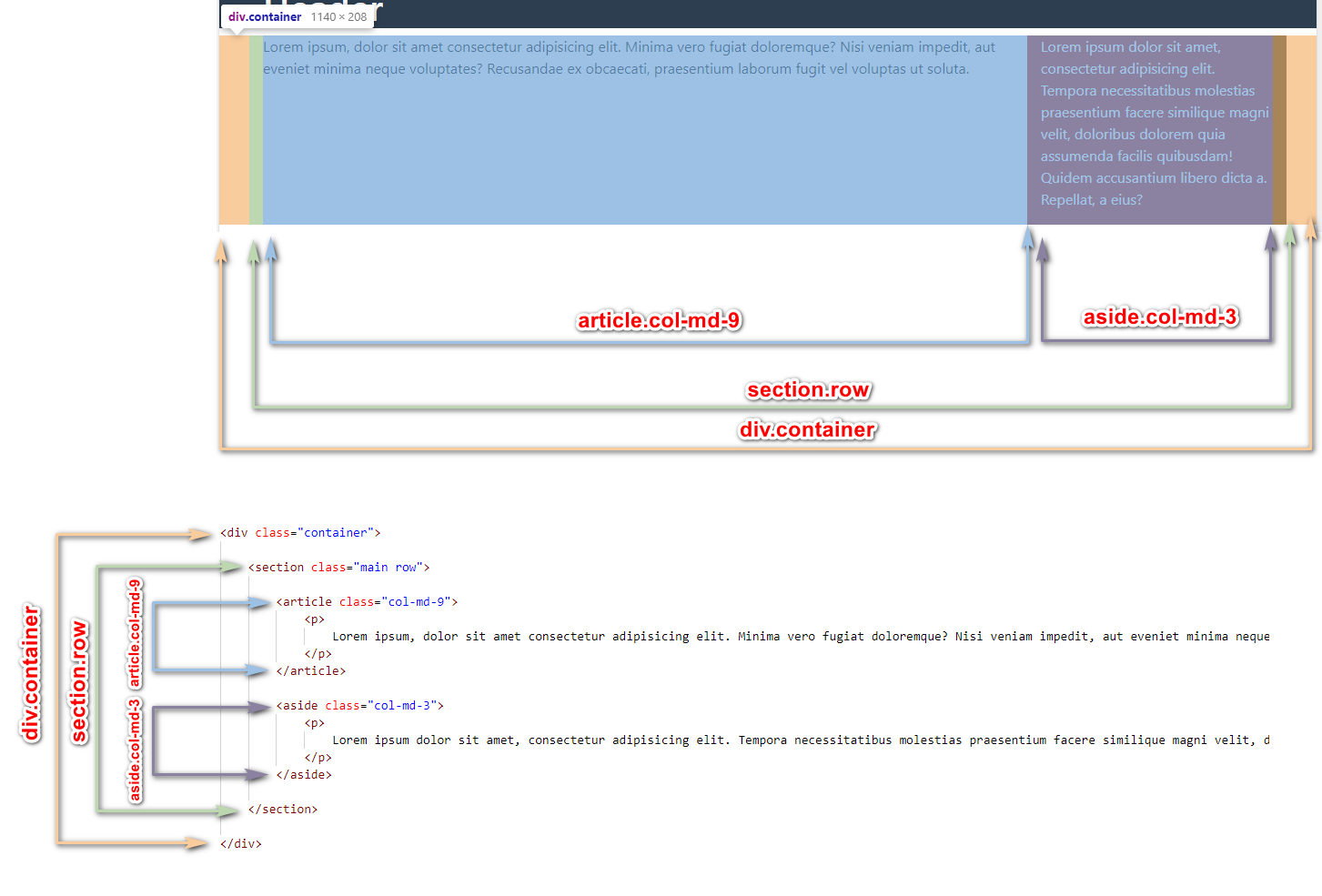
Los tamaños de pantalla que no definamos van a utilizar el próximo que hayamos definido para abajo, por ejemplo: tenemos definido *xs* y *sm*, si la pantalla es *lg* entonces se va a utilizar la medida *sm*.

## Filas

Para que los contenidos se ubiquen uno al lado del otro deben estar dentro de un elemento que posea la clase *row* y dentro de ese elemento contenedor se tienen que ubicar en columnas distintas.

Si queremos que los contenidos se encuentren uno debajo del otro los ponemos en distintos Rows.

### Ejemplo



## Diferencias de tamaño en una fila

Nos puede pasar que los elementos de una misma fila (y distinta columna) tengan diferentes tamaños de alto. En este caso puede ocurrir que se produzcan errores de visualización. Para solucionarlo entre cada columna agregamos un contenedor vacío que utilice la clase clearfix

## Mover una columna

(PROBAR)

En el caso que no completemos todo el ancho de la fila podemos desplazar una columna mediante la clase col-md-offset-3 (donde 3 seria la cantidad de columnas que queremos desplazar)

## Resumen

1. Creamos un elemento contenedor y le asignamos la clase *container*
2. Creamos un elemento contenedor y le asignamos la clase *row*
3. Creamos elementos contenedores para cada contenido que queramos que aparezca uno al lado del otro. Dividimos el ancho de la pantalla en la cantidad de contenidos que queremos mostrar y asignamos la proporción de tamaño.

