



# **Responsive, gradientes y paletas: el arte de que todo se vea bien en todos lados**



**I heard you want to be a web developer**



**Here are a few devices to test your site**

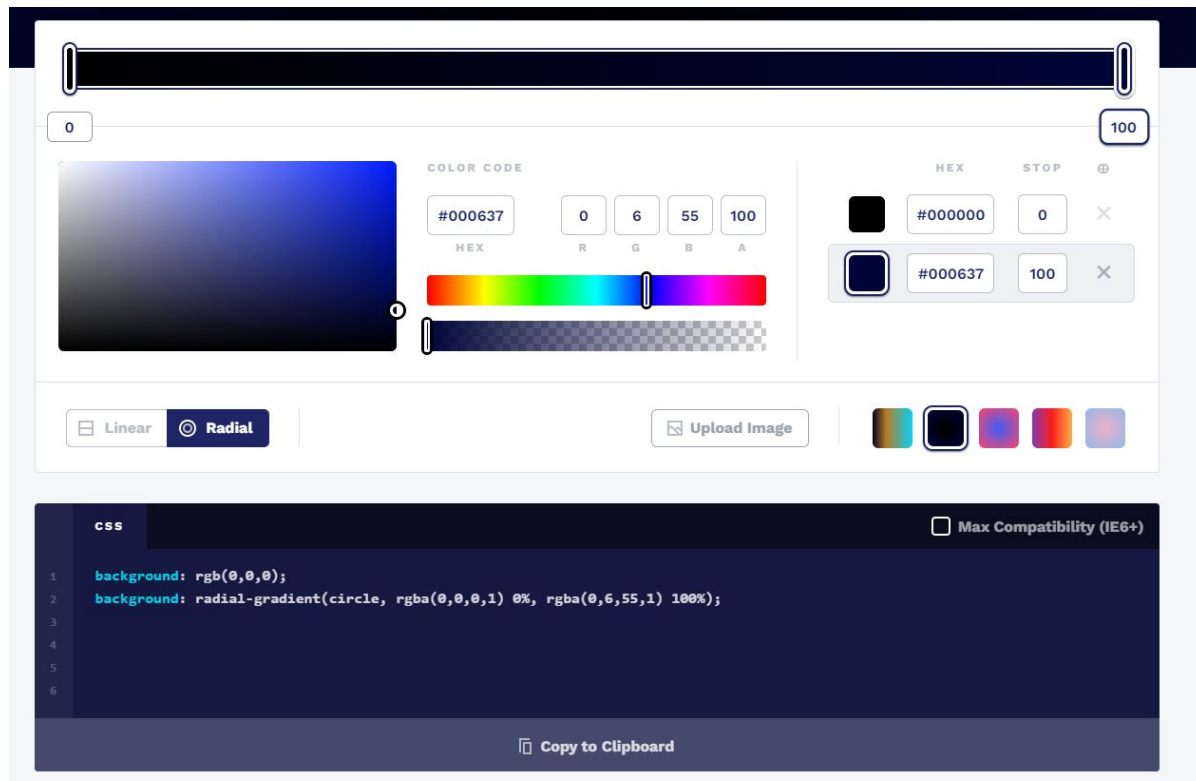
## Here are a few devices to test your site

# Empecemos por lo más fácil: gradientes

Tan fácil como ir de un color a otro. Seamos honestos, 90% de los gradientes los van a hacer aprovechando algún

sitio copado: <https://cssgradient.io/>





Noten que les da un background sólido... eso es en caso que por algún motivo el navegador no pueda interpretar el gradiente.



# Ahora... paletas!



Las paletas podemos generarlas nosotros, o podemos, nuevamente, usar un sitio generador de paletas.

Acá les dejo un par de sitios con consejos de diseño, pero igual vamos a hablar un poquito los conceptos:

<https://blog.aulaformativa.com/consejos-crear-una-paleta-de-colores-efectiva-disenos/>

<https://blog.aulaformativa.com/importancia-psicologia-del-color/>

<https://www.staffcreativa.pe/blog/psicologia-diseno-de-logotipos/>

<https://www.renderforest.com/es/blog/how-to-choose-your-logo-colors>

Muchas veces van a trabajar con la paleta que les dé el cliente, pero tienen que saber qué transmite cada color, y saber pensar las formas y tipografías.



**power  
sophistication  
mystery  
death**

**hope  
simplicity  
cleanliness  
goodness  
purity**

**love  
passion  
romance  
danger  
energy**

**intellect  
friendliness  
warmth  
caution  
cowardice**

**peace  
sincerity  
confidence  
integrity  
tranquility**

**authority  
maturity  
security  
stability**

**life  
growth  
nature  
money  
freshness**

**innovation  
creativity  
thinking  
ideas**

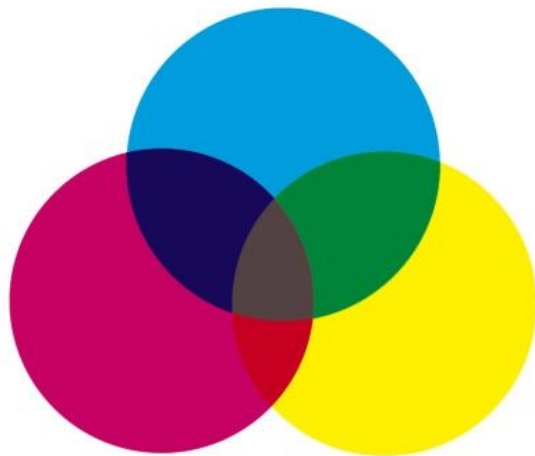
**royalty  
luxury  
wisdom  
dignity**



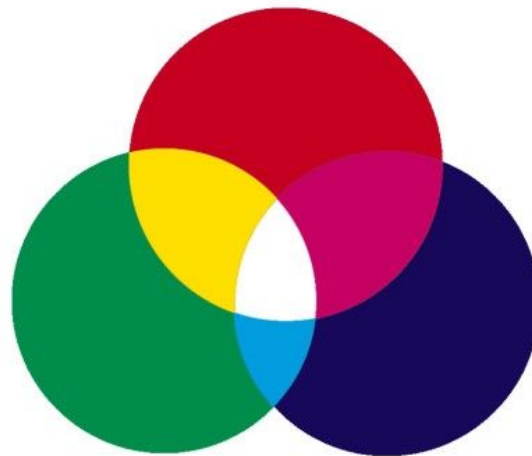
## Famous Brands' Colors

ebaqdesign





**Subtractive color (CMYK)**

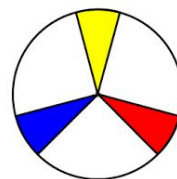
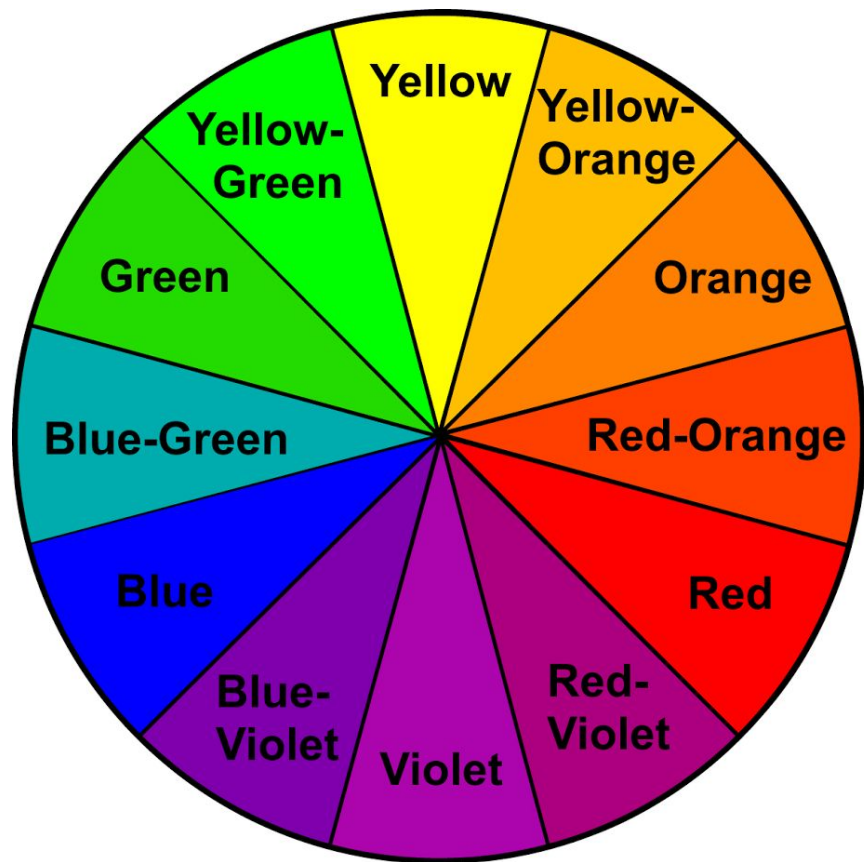


**Additive Color (RGB)**

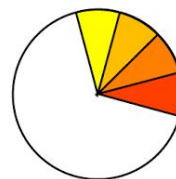
Nosotros trabajamos con RGB, porque son los colores de la emisión de luz. CMYK es el sistema que se usa para impresión, y como verán, sus colores primarios no son los mismos, ni la forma en que combinan.







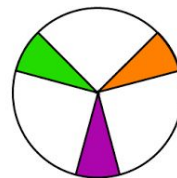
primary



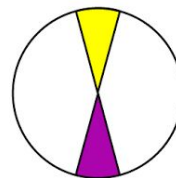
analogous



hue



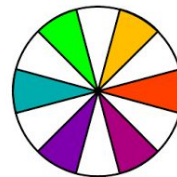
secondary



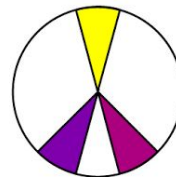
complementary



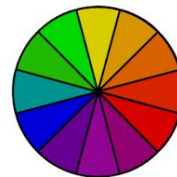
tint



intermediate



split complementary



shade





**Complementary**



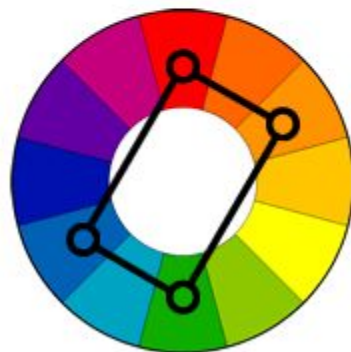
**Analogous**



**Triadic**



**Split Complementary**



**Tetradic**



# Bueno, pero y de dónde puedo sacar paletas?

Hay montones de sitios, y también cuentas de instagram que suben paletas copadas. Algunas sugerencias:

<https://coolers.co/> mi favorito. Las paletas generadas por usuarios son hermosas.

<https://www.canva.com/colors/color-palette-generator/>






<https://color.adobe.com/es/create/color-wheel> otro excelente. Pueden elegir el tipo de armonía de color.

<https://material.io/resources/color/#!/?view.left=0&view.right=0&primary.color=B39DDB> esta permite ver cómo quedan las combinaciones en una UI, definitivamente vale la pena.



# Che, y ahora voy a tener que copiarme esa paleta en un txt para no perderla?

Se pueden crear variables de color en css, y ya está, los tenés guardados, y además si lo cambiás, cambia automáticamente en todos los lugares en que lo uses. Un golazo.

```
:root {  
  --main-dark-color:  #252422;  
  --main-grey-color:  #403d39;  
  --main-dark-grey:  #ccc5b9;  
  --main-light-grey:  #fffcf2;  
  --main-orange:  #eb5e28;  
}
```

```
header a {  
  text-decoration: none;  
  color: var(--main-light-grey);  
  font-size: 1.7em;  
}
```



When you ask the backend developer to choose color shades for the website frontend



No me hagan sufrir, pero... bueno, ya vamos a ver este tema.





# Bueno, listo. Ya no sacamos de encima los colores. Ahora vamos a hablar de responsive.

Hoy en día un sitio tiene que verse bien en toda clase de dispositivos. Los mediaqueries son nuestros amigos para esto.

```
@media (min-width: 1024px) {  
  main {  
    flex-direction: row;  
    flex-wrap: wrap;  
    width: 100%;  
    justify-content: space-around;  
  }  
}
```



Cuenta la leyenda que en los viejos tiempos, podías cargar un CSS distinto para cada tipo de pantalla, y eso se llamaba adaptative design. Imagínense tener 4 CSS distintos de acuerdo a qué tamaño hay que renderizar. Todavía hoy existen de esos...

<https://www.idento.es/blog/desarrollo-web/disenio-responsive-vs-adaptive-cual-es-mejor/>





# ADAPTIVE VS RESPONSIVE



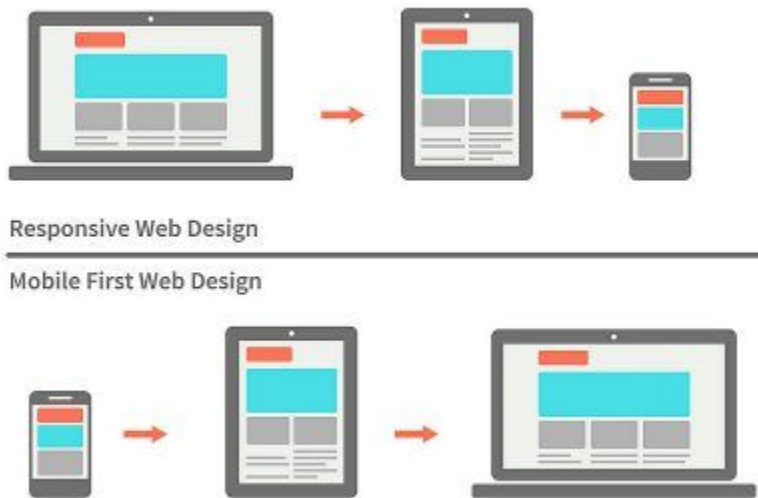
**KNOW THE DIFFERENCE!**



# Mobile first

Es el concepto (que como todos somos unos chantas no solemos aplicar) de que hay que diseñar para la pantalla más chica, y poner los queries para adaptar a los tamaños más grandes.

Es importante trabajar bien la versión mobile porque es la que google va a tener en cuenta para el posicionamiento.



# Ok, y cómo es esto de los media queries entonces?

Básicamente separan fragmentos de CSS para que solo se rendericen ante ciertas condiciones. Lo que más vamos a usar va a ser min-width y max-width.

Adentro de la etiqueta @media vamos a meter todos los estilos que necesitemos overridear para ese tamaño, *no todos los estilos. Por favor que me va a dar un ataque si meten una copia del CSS.*

```
@media (max-width: 600px) {  
  .facet_sidebar {  
    display: none;  
  }  
}
```



Podemos usar and, entre otras cosas:



En lo personal trato de tener mis queries lo más espartanos posibles. No está mal poner el screen, pero no suelo usarlo.

```
1  
2 /*===== Bootstrap 3 Media Queries =====  
3 = Bootstrap 3 Media Queries =  
4 =====*/  
5  
6 /*===== Mobile First Method =====*/  
7  
8 /* Custom, iPhone Retina */  
9 @media only screen and (min-width : 320px) {}  
10  
11 /* Extra Small Devices, Phones */  
12 @media only screen and (min-width : 480px) {}  
13  
14 /* Small Devices, Tablets */  
15 @media only screen and (min-width : 768px) {}  
16  
17 /* Medium Devices, Desktops */  
18 @media only screen and (min-width : 992px) {}  
19  
20 /* Large Devices, Wide Screens */  
21 @media only screen and (min-width : 1200px) {}  
22  
23 /*===== Non-Mobile First Method =====*/  
24  
25 /* Large Devices, Wide Screens */  
26 @media only screen and (max-width : 1200px) {}  
27
```



El único caso de uso de and que me suele gustar es poniendo min y max, así me aseguro que esos estilos existen exclusivamente en ese rango.

```
body {  
  background-color: #fff;  
}  
  
@media (min-width: 30em) and (max-width: 80em) {  
  body {  
    background-color: purple;  
  }  
}
```



Por otro lado podemos decidir qué estilos queremos de acuerdo a si la pantalla está vertical u horizontal con el orientation:

## orientation

**Valor:** landscape | portrait

**Medio:** [media/visual](#) (en-US)

**Acepta prefijos min/max:** no

Indica cuando el dispositivo esta en modo landscape (el ancho de la pantalla es mayor al alto) o modo portrait (el alto de la pantalla es mayor al ancho).

## Ejemplo

Para aplicar una hoja de estilo solo en orientación vertical (portrait):

```
@media all and (orientation: portrait) { ... }
```



Si estuviéramos en adaptative en vez de responsive, podemos usar los mediaquery para decirle qué css tomar para cada situación:

Para aplicar una hoja de estilo a un documento cuando este sea mostrado en una pantalla de menos de 800px de ancho, usted puede usar esto:

```
<link rel="stylesheet" media="screen and (max-device-width: 799px)" />
```



Y podemos combinar, aunque nunca tuve que hacerlo:

#### lista separada por comas

Las listas separadas por comas se comportan como el operador `or` cuando es usado en media queries.

Cuando utilice una lista separada por comas y alguno de los queries retorna verdadero, el estilo o la hoja de estilos sera aplicada. Cada query en una lista separada por comas es tratado como una query individual y cualquier operador aplicado a un medio no afectara a los demás. Esto significa que cada query en una lista separada por comas puede tener como objetivo diferentes medios, tipos y estados.

Si usted quiere aplicar una serie de estilos para un equipo con un ancho mínimo de 700px o si el dispositivo esta colocado en horizontal, usted puede escribir lo siguiente:

```
@media (min-width: 700px), handheld and (orientation: landscape) { ... }
```



# Y cómo decido qué breakpoints usar?

Según w3schools, podemos usar estos 5, aunque en lo personal me parece que el large requeriría dos, no un breakpoint:

```
/* Extra small devices (phones, 600px and down) */  
@media only screen and (max-width: 600px) {...}  
  
/* Small devices (portrait tablets and large phones, 600px and up) */  
@media only screen and (min-width: 600px) {...}  
  
/* Medium devices (landscape tablets, 768px and up) */  
@media only screen and (min-width: 768px) {...}  
  
/* Large devices (laptops/desktops, 992px and up) */  
@media only screen and (min-width: 992px) {...}  
  
/* Extra large devices (large laptops and desktops, 1200px and up) */  
@media only screen and (min-width: 1200px) {...}
```





Bootstrap nos da otros breakpoints. El xxl es nuevo, viene en la versión 5. De todos modos, la documentación nos dice que esos breakpoint pueden ser customizados.

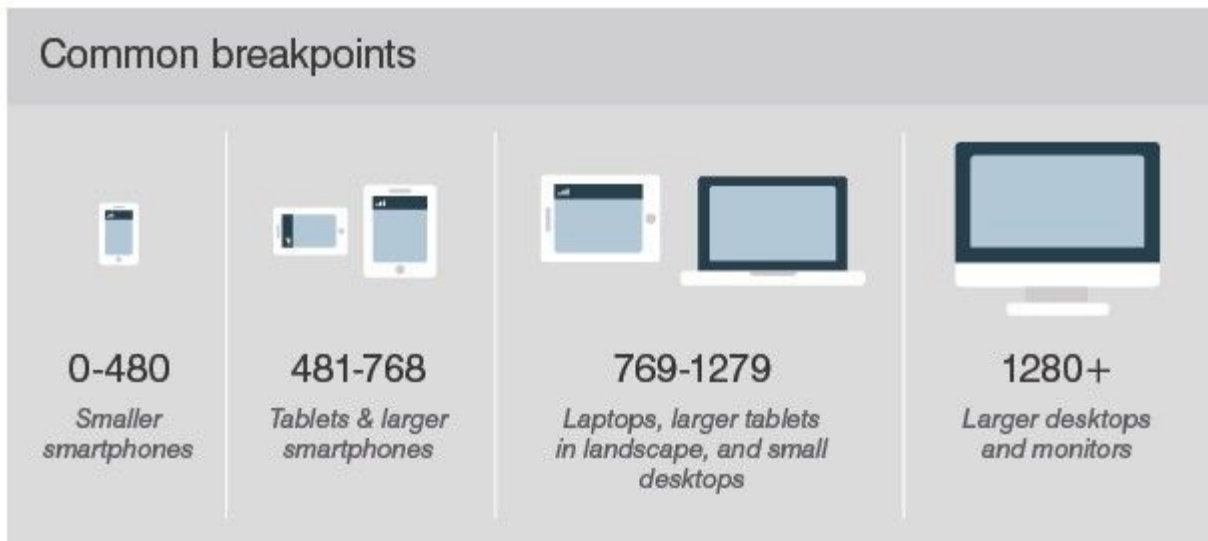
## Available breakpoints

Bootstrap includes six default breakpoints, sometimes referred to as *grid tiers*, for building responsively. These breakpoints can be customized if you're using our source Sass files.

Breakpoint	Class infix	Dimensions
X-Small	<i>None</i>	<576px
Small	<code>sm</code>	≥576px
Medium	<code>md</code>	≥768px
Large	<code>lg</code>	≥992px
Extra large	<code>xl</code>	≥1200px
Extra extra large	<code>xxl</code>	≥1400px



Yo suelo usar un esquema similar a este, solo que agrego queries en 1440 y 1600 si es necesario. Usualmente los uso para ajustes chiquitos, pero suelen necesitarse algunos ajustes entre 1280 y 1920.



Este sitio es genial, acá pueden verlo con más detalle:

<https://css-tricks.com/a-complete-guide-to-css-media-queries/>





**NEW THINGS?**

**I'M TERRIFIED OF THAT.**

makeameme.org

