

COLOR

UNCPBA/ FCE / 2020  
INTERFACES DE USUARIO E INTERACCIÓN | TUDAI

# Administrativos

Planilla (LU, Nombre, Mail): <https://cutt.ly/KfdDMpW>

Admitir x c/u: Correo oficial, evento con mails.

Git?

→ Puntualidad!

Horario del Miércoles:

Prog2, 185 alumnos → no way

VEI, 47 alumnos (vs. 18 aprox Prog2)

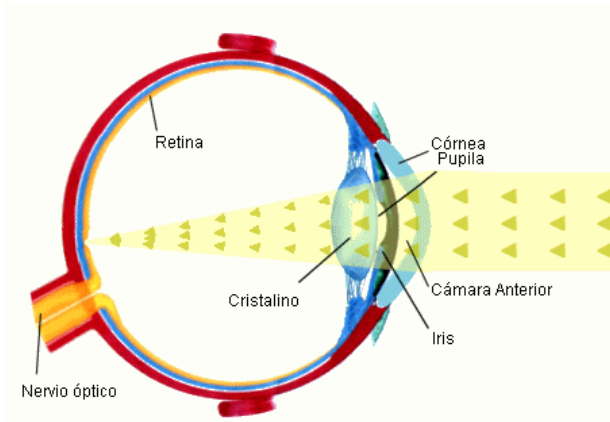
**2** opciones.

# ¿Qué es el color?

*El color es luz.*

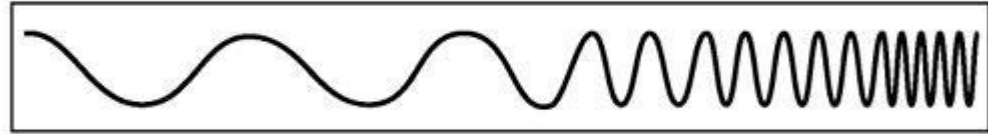
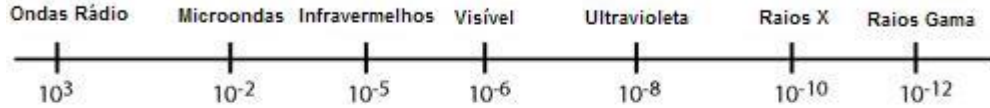
*Ondas electromagnéticas viajando en el espacio.*

*“La percepción de los colores es un fenómeno subjetivo; el color varía según la cantidad de luz del ambiente y la distancia a la que se encuentra el objeto”*

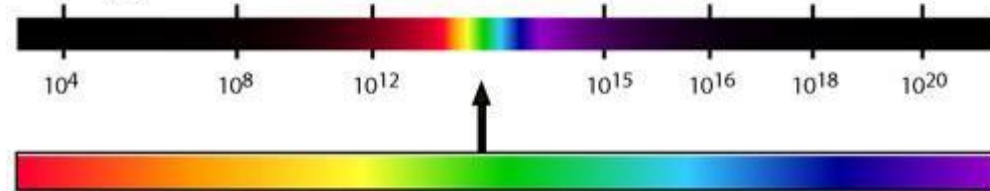


## Espectro Electromagnético

Comprimento de Onda  
(metros)



Frequência  
(Hz)

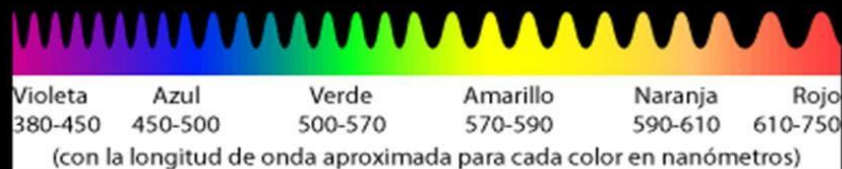


COLOR: Impresión que producen en la retina, los rayos de luz reflejados y absorbidos por un cuerpo, según la longitud de onda de estos rayos.

La Luz blanca  
se descompone  
al pasar por  
un prisma



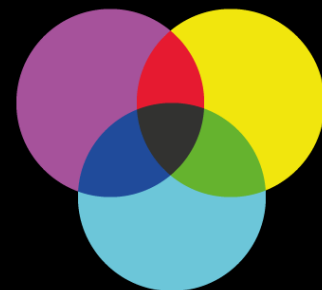
**LUZ: Las ondas electromagnéticas que el ojo humano percibe**



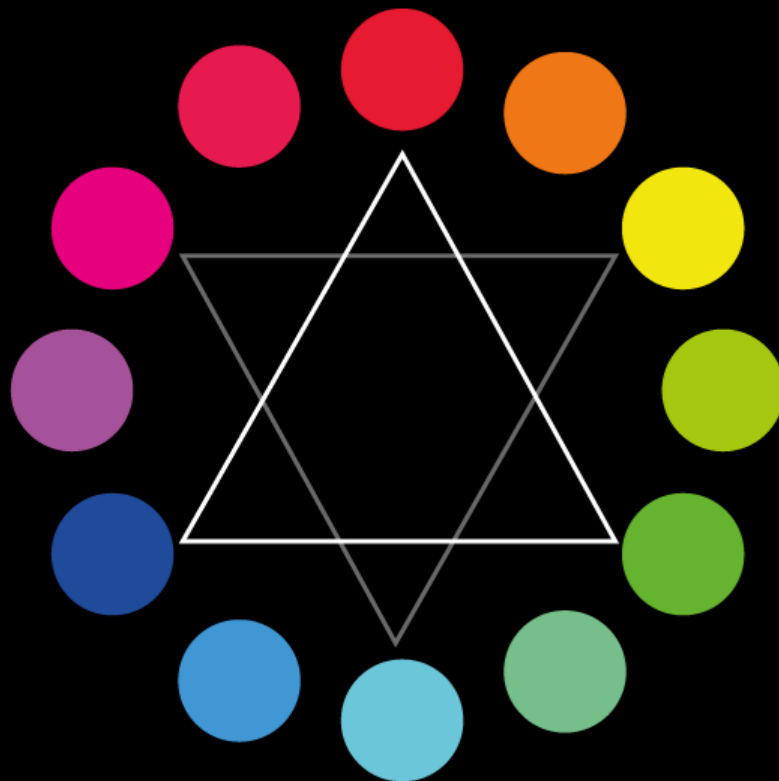
# Síntesis aditiva del color



**Síntesis sustractiva**



# Círculo cromático (RGB)



Primarios  
Secundarios

# Armonías tonales

The image shows a screenshot of a web-based color palette generator. The interface is dark-themed and features a large color wheel on the left side. The wheel is divided into segments of various colors, with a central circular area. Above the wheel, there are several circular icons representing different color schemes. To the right of the wheel, the text "Triad (3-colors)" is displayed, along with a radio button and the text "add complementary". Below the wheel, there are input fields for "Hue: 0°" (with "opposite" below it) and "Dist: 45°" (with "default: 30°" below it). At the bottom left, there is a "Base RGB:" field with the value "AA3939" and a "Fine Tune..." link. On the right side, there is a "My Palette:" section showing a horizontal row of color swatches. Below this, there is a large rectangular area displaying a complex arrangement of color swatches, including a large red area, a blue area, and a green area. At the bottom right, there is a "Share palette" link and a "Vision simulation" toggle.

<https://paletton.com/#uid=1000u0kllllaFw0g0qFqFg0w0aF>



# Existen distintos sistemas de representación

RGB = Red - Green - Blue

RGBA = Red - Green - Blue - Alpha

HSL = Hue - Saturation - Lightness

HSLA = Hue - Saturation - Lightness - Alpha

Hexadecimal = #RRGGBBAA

0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

Otros (CMYK, HSV, ARGB, YUV ...)

[https://es.wikipedia.org/wiki/Espacio\\_de\\_color](https://es.wikipedia.org/wiki/Espacio_de_color)

<https://htmlcolorcodes.com/es/nombres-de-los-colores/>

**¿Cómo se representa un color?**

**¿Cómo defino un color en particular?**

**Función JavaScript**

`rgba(255,255,255,1);`

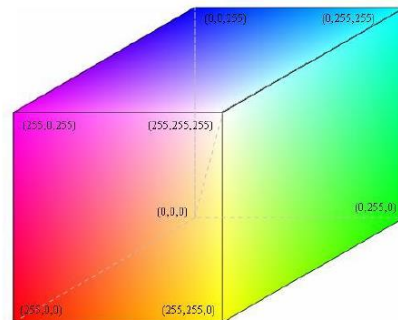
`hsla(0,0,0,0);`

# Colores hexadecimales

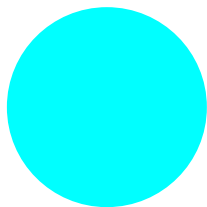
Los valores de color hexadecimales también son compatibles con todos los navegadores.

Un color hexadecimal se especifica con: # RR GG BB .

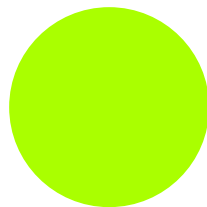
RR (rojo), GG (verde) y BB (azul) son enteros hexadecimales entre 00 y FF que especifican la intensidad del color.



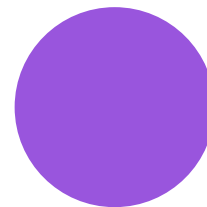
#00FFFF



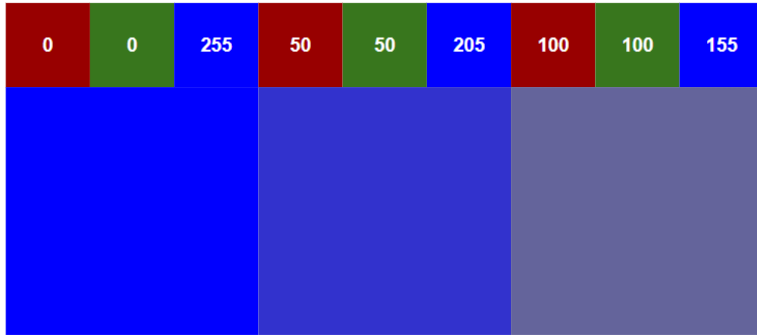
#AAFF00



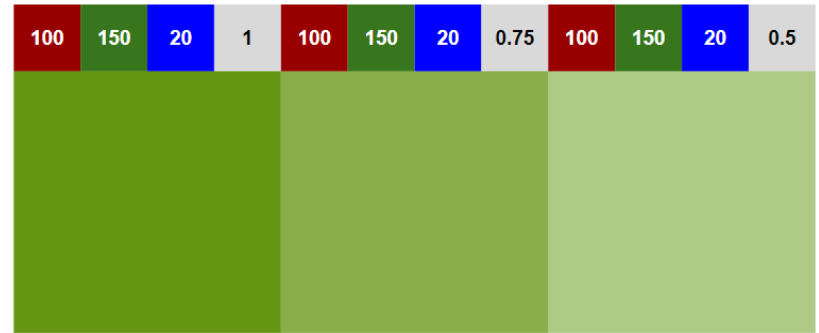
#9955DD



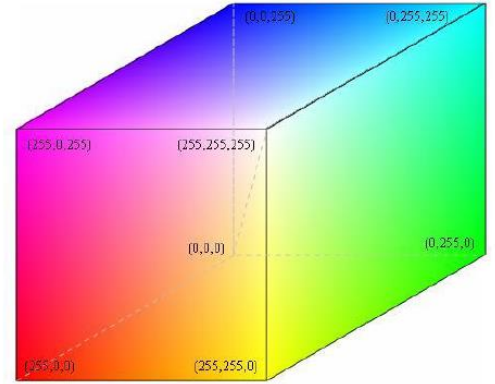
# Pixel RGB y RGBA



Un pixel de color se representa en el modelo RGB con 3 bytes



Un pixel de color se representa en el modelo RGBA con 4 bytes



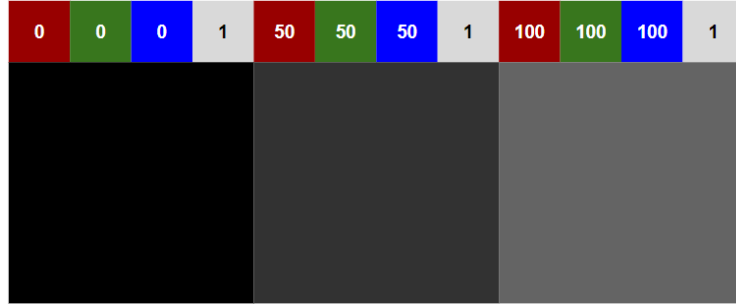
let color = rgba(255,100, 50, 0.5);



let color2 = rgba(255, 225, 175, 1);















# Alpha



## HTML color names [\[ edit \]](#)

The HTML 4.01 specification, ratified in 1999, defines 16 named colors,<sup>[6]</sup> as follows (names are defined in this context to be case-insensitive):

**CSS 1–2.0, HTML 3.2–4, and VGA color names**

	Name ↕	Hex (RGB) ↕	Red (RGB) ↕	Green (RGB) ↕	Blue (RGB) ↕	Hue (HSL/HSV) ↕	Satur. (HSL) ↕	Light (HSL) ↕	Satur. (HSV) ↕	Value (HSV) ↕	CGA number (name); alias ↕
	White	#FFFFFF	100%	100%	100%	0°	0%	100%	0%	100%	15 (white)
	Silver	#C0C0C0	75%	75%	75%	0°	0%	75%	0%	75%	07 (light gray)
	Gray	#808080	50%	50%	50%	0°	0%	50%	0%	50%	08 (dark gray)
	Black	#000000	0%	0%	0%	0°	0%	0%	0%	0%	00 (black)
	Red	#FF0000	100%	0%	0%	0°	100%	50%	100%	100%	12 (high red)
	Maroon	#800000	50%	0%	0%	0°	100%	25%	100%	50%	04 (low red)
	Yellow	#FFFF00	100%	100%	0%	60°	100%	50%	100%	100%	14 (yellow)
	Olive	#808000	50%	50%	0%	60°	100%	25%	100%	50%	06 (brown)
	Lime	#00FF00	0%	100%	0%	120°	100%	50%	100%	100%	10 (high green); green
	Green	#008000	0%	50%	0%	120°	100%	25%	100%	50%	02 (low green)
	Aqua	#00FFFF	0%	100%	100%	180°	100%	50%	100%	100%	11 (high cyan); cyan
	Teal	#008080	0%	50%	50%	180°	100%	25%	100%	50%	03 (low cyan)
	Blue	#0000FF	0%	0%	100%	240°	100%	50%	100%	100%	09 (high blue)
	Navy	#000080	0%	0%	50%	240°	100%	25%	100%	50%	01 (low blue)
	Fuchsia	#FF00FF	100%	0%	100%	300°	100%	50%	100%	100%	13 (high magenta); magenta
	Purple	#800080	50%	0%	50%	300°	100%	25%	100%	50%	05 (low magenta)

<http://www.workwithcolor.com/color-converter-01.htm>

<https://www.calculadoraconversor.com/conversores/modelos-de-color/>

[https://www.w3schools.com/TAGs/ref\\_colornames.asp](https://www.w3schools.com/TAGs/ref_colornames.asp)

# Dibujando en web

**<canvas>** es el elemento donde se dibuja en HTML5

**Context** provee la funcionalidad para dibujar.

**[!]** Todos los elementos visual de la página tiene un **Context**.



# Ejemplo de uso de Canvas

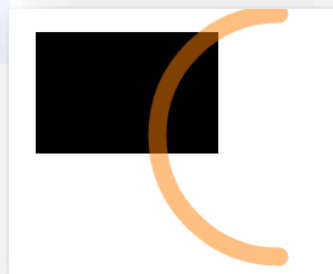
```
<canvas id="canvas" width="900" height="600" style="border:#000 solid 1px;" />
</body>
```

```
<script>
```

```
var ctx = document.getElementById("canvas").getContext("2d");
ctx.fillStyle = "#000000";
ctx.fillRect(250, 25, 150, 100);
ctx.beginPath();
ctx.arc(450, 110, 100, Math.PI * 1/2, Math.PI * 3/2);
ctx.lineWidth = 15;
ctx.lineCap = 'round';
ctx.strokeStyle = 'rgba(255, 127, 0, 0.5)';
ctx.stroke();
```

```
</script>
```

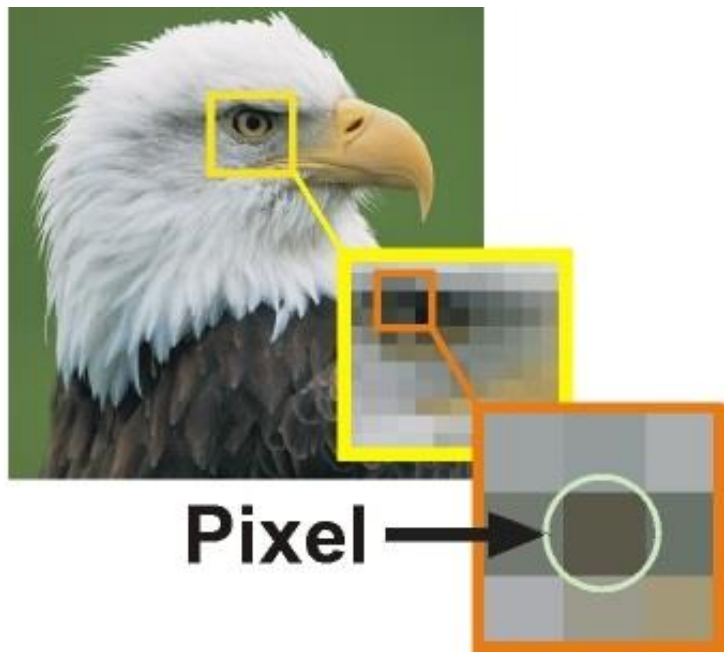
```
</html>
```

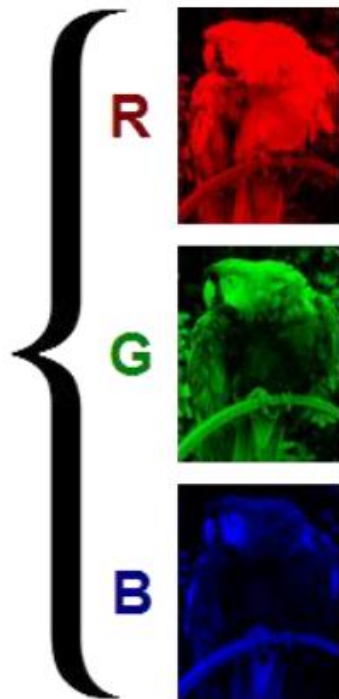
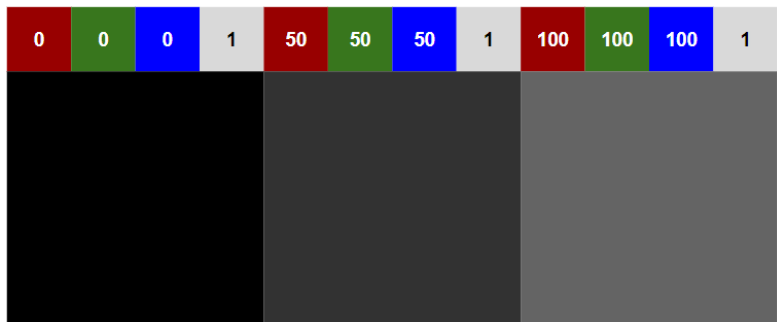


**To-Do:** → Buscar la referencia de los métodos de Context, y experimentar...

# Imágenes

Una imagen digital es una matriz 2D (WxH) de píxeles.





# Representar imágenes

- HTML5 provee un “*ImageData*” .
- La imagen se recorre en *Ancho* y *Alto* (*for*, *for*)
- “*ImageData*” almacena los pixeles en un arreglo de 1d
- Los colores se almacenan en un *Array de enteros* y se pueden acceder como si fueran una matriz
  
- Para dibujar la Imagen en Pantalla: **ctx.putImageData ( Imagen, x, y ) ;**
- Para convertir de matriz a arreglo?

*indice = ...*

*indice = (x + y \* imageData.width) \* 4;*

# Crear imagen desde cero

```
var imageData = ctx.createImageData(width, height);

for (x=0; x<width; x++){
    for (y=0; y<height; y++){
        setPixel(imageData, x, y, r, g, b, a);
    }
}

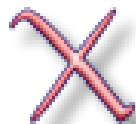
ctx.putImageData(imageData, 0, 0);
```

```
function setPixel(imageData, x, y, r, g, b, a)
{
    index = (x + y * imageData.width) * 4;
    imageData.data[index+0] = r;
    imageData.data[index+1] = g;
    imageData.data[index+2] = b;
    imageData.data[index+3] = a;
}
```

# Cómo cargar una imagen?

- La carga de imágenes es **asincrónica**
- El script se ejecuta secuencialmente línea por línea
- El tiempo de demora depende del tamaño de la imagen y de la latencia del servidor que la tiene

```
var imagen1 = new Image();  
  
imagen1.src = "imagen.jpg";  
  
ctx.drawImage(imagen1, 0, 0);
```



→ Puede que la imagen no esté cargada en memoria al momento de dibujarla

# Cargar una imagen y dibujarla

```
var imagen1 = new Image();  
imagen1.src = "imagen.jpg";
```

Ubicación del  
recurso



```
imagen1.onload = function() {  
    myDrawImageMethod(this);  
}
```

Evento de  
finalización de la  
carga del recurso



```
function myDrawImageMethod(image) {  
    ctx.drawImage(image, 0, 0);  
}
```

Dibujo la imagen  
usando el contexto  
del HTML5



# Por qué se usa .onload() ?

```
1  function myDrawImageMethod(image) {
2      ctx.drawImage(image, 0, 0);
3  }
4  .....
5  var image1 = new Image();
6  image1.src = "imagen1.jpg";
7  image1.onload = function() {
8      myDrawImageMethod(this);
9  }
10 var image2 = new Image();
11 image2.src = "imagen2.jpg";
12 image2.onload = function() {
13     myDrawImageMethod(this);
14 }
15 // Otra linea de código 1
16 // Otra linea de código 2
17 // Otra linea de código 3
18 // Otra linea de código 4
```

