MUEIB/MNUR
ETSEIB

Course on Medical Imaging
2023-2024
Q1

Session 2
Introduction to images part II

Dani Tost

Summary of last week

Tools to be used:

- Python
- Vtk
- PyQt
- Scikit-image
- Matplotlib
- git

Concepts

- **vectorial** geometry
- **raster** geometry

Homework from last week

Implement the class Polygon … doubts?

Vectorial images

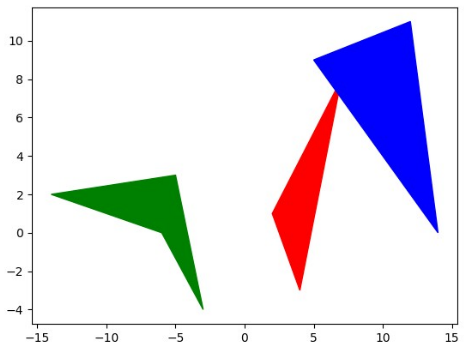
Second exercise: implement a vectorial image as a container of polygons (we'll make it more complex later) given the following specification.

```
Class vecimage.VecImage()

    Methods
        add(pol)
                adds the polygon pol to the vectorial image
        draw_matplotlib(plt
                renders all the polygons of the image into a matplotlib figure
      Operations
```

v[i]	returns the ith polygon of the vectorial image
len(v)	renders the number of polygons of v

Vectorial images



In the terminal, try this to get the result:

```
>>> import matplotlib.pyplot as plt
>>> fig = plt.Figure()
>>> from polygon import Polygon
>>> from vecimage import VecImage
>>> v = VecImage()
>>> v.add(Polygon('red', [2, 4, 7], [1, -3, 8]))
>>> v.add(Polygon('blue', [12, 14, 5], [11, 0, 9]))
>>> v.add(Polygon('green', [-6, -14, -5, -3], [0, 2, 3, -4]))
>>> v.draw_matplotlib(plt)
>>> plt.show()
```

Update the git project to download the test file

```
git pull
```

Pass the doctests

```
python3 -m doctest tests/tests_vecimage.txt
```

So far for the revision of python and coding procedure. We'll come back to vectorial images later.

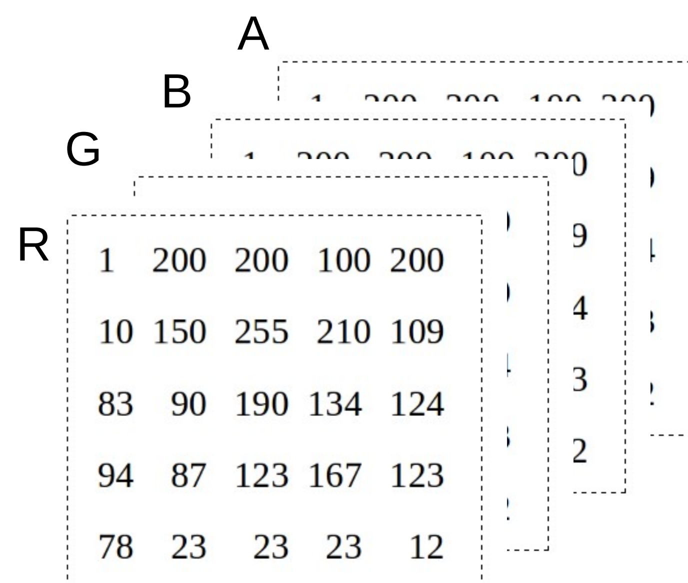
Raster images

Sources

Photos

Medical imaging devices

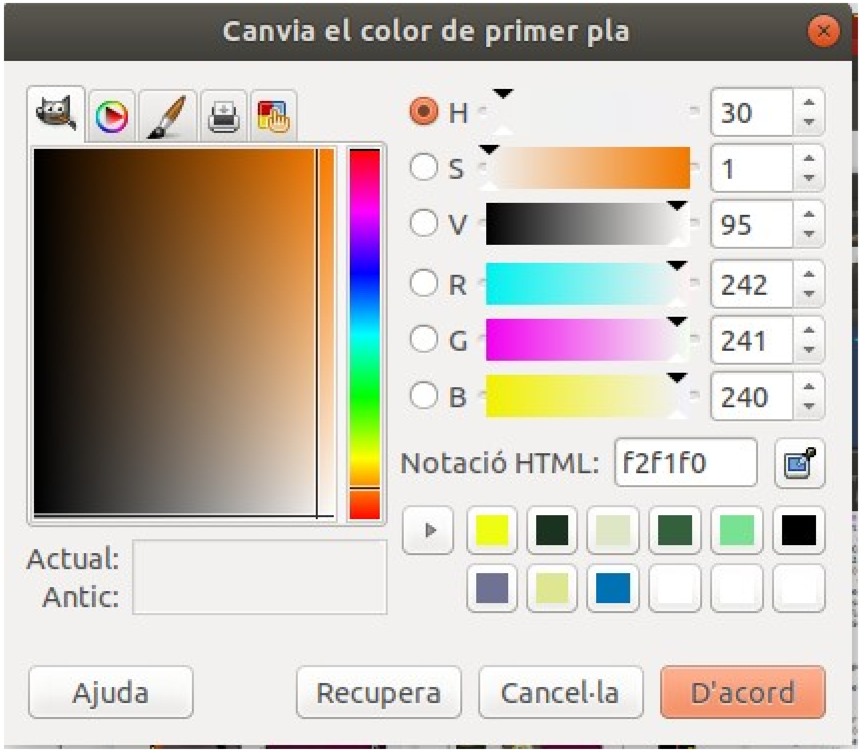
Editor



Matrix of data (1 to 4 channels)

Pixel value

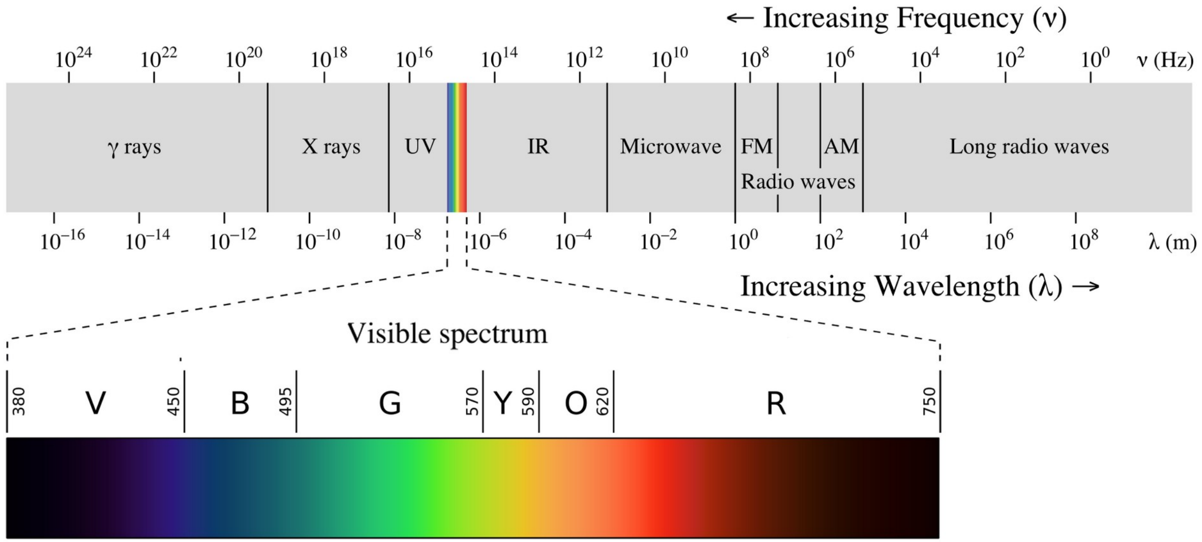
- Value:
  - W/B: (binary images)
  - Gray scale (e.g. 0-255)
  - Color
    - Direct
    - An index to a color table



- Codification
  - RGB
  - YCM
  - RGBA
  - ....
- Channels RGBA (A = alpha channel).

Light

- An electromagnetic wave
- Color is a categorization of light according to wavelength
- Spectral color have only one wavelength.
- Non spectral colors are a blend of various spectral colors

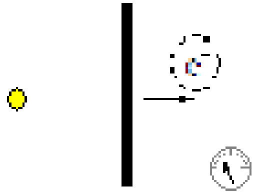


Color	Wavelength interval
Red	~ 700–635 nm
Orange	~ 635–590 nm
Yellow	~ 590–560 nm
Green	~ 560–520 nm
Cyan	~ 520–490 nm
Blue	~ 490–450 nm
Violet	~ 450–400 nm

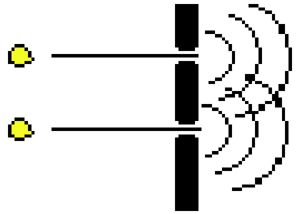
Nature of light

Theories:

- **geometrical optics**: light is a stream of particles (photons)
  explains photoelectric effect



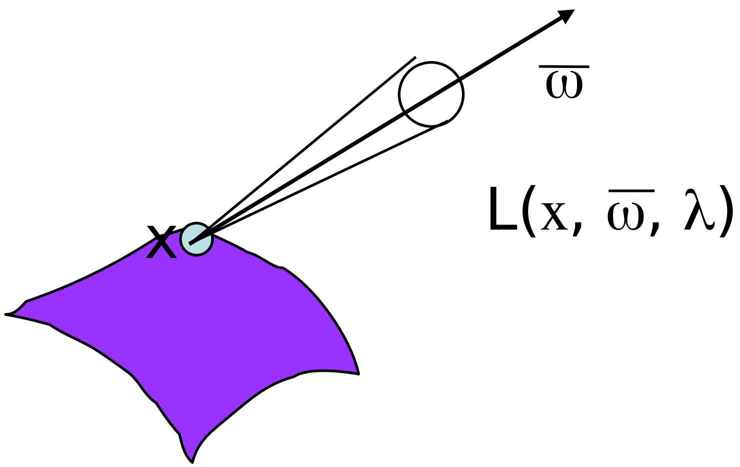
- **wave optics:** light is an electromagnetic wave
  explains diffraction

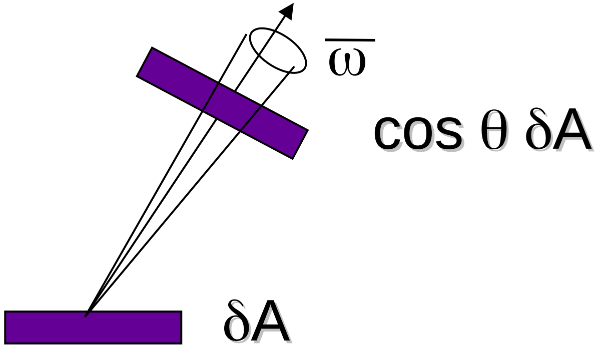


- **quantum optics:** dual nature of light
  photons are packet wave which do not extend infinitely

Measurement of light

Radiance: radiant power of a given wavelength λ in a point x and direction ω per solid angle unit $\delta\omega$ and unit projected area (Units: W/sr.m^2)

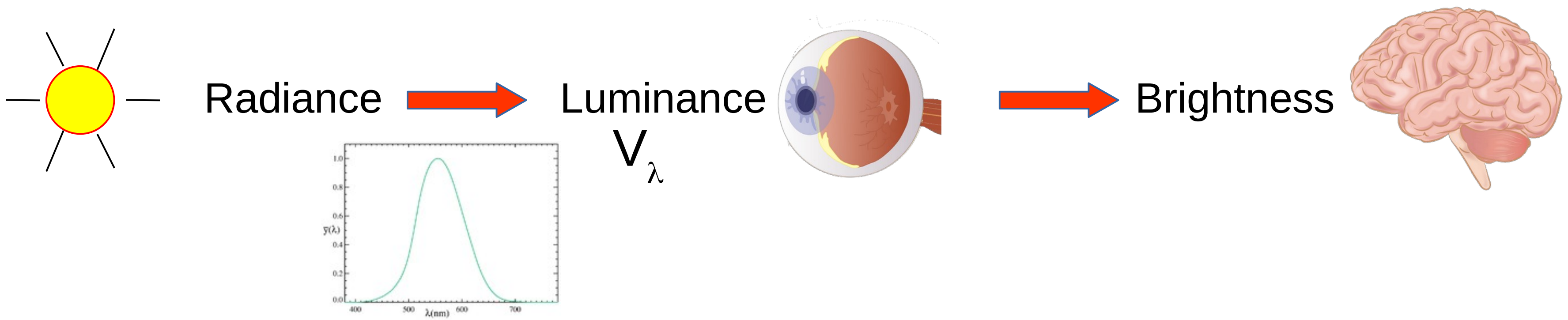




Luminance: equivalent measure in photometric units (cd/m2 o Lm/sr.m^2), obtained by integrating radiance for all wavelengths, weighted by the luminous efficacy function (V_λ) that represents the sensitivity of human visual system to different wavelengths.

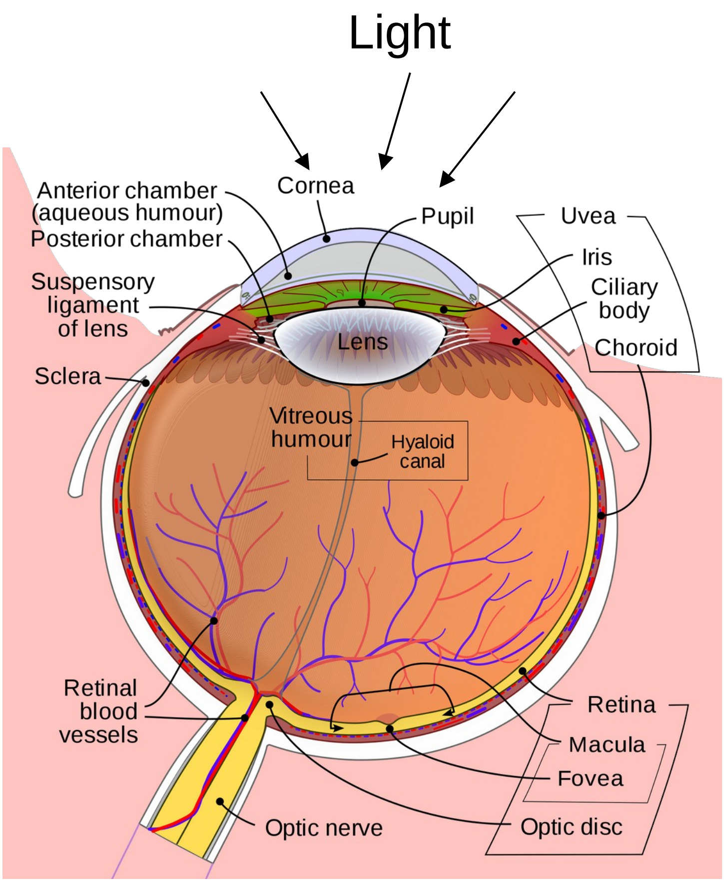
Measurement of light

Brightness: subjective visual experience of luminance (non-measurable)



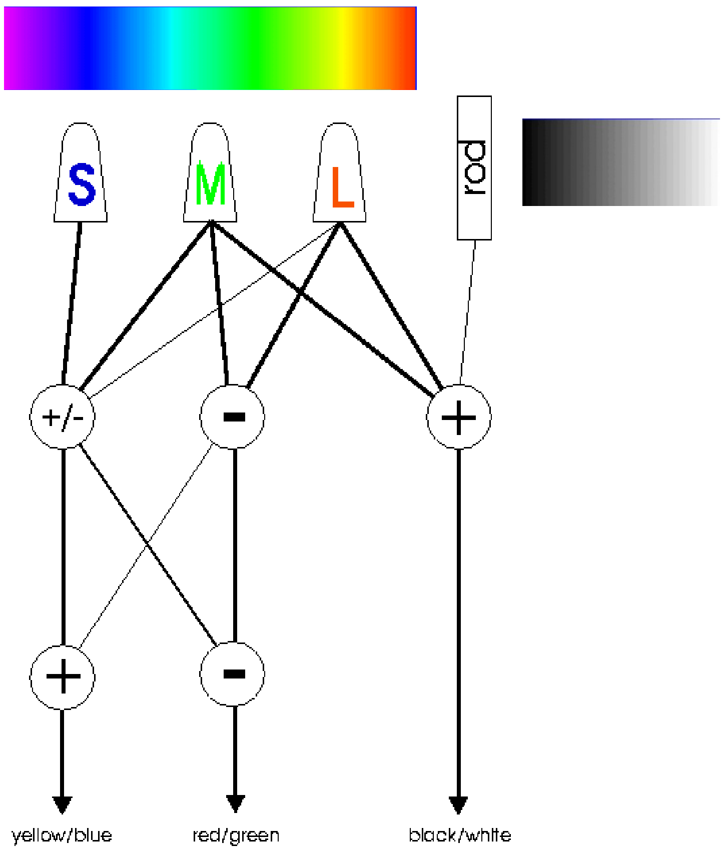
Intensity: often when coding to simplify light computations we use an adimentional RGB value of “intensity” (from 0 to 1 in each channel or 0 to 255) to represent the “luminance” of a source.

The human visual system



Visible light rays are transmitted through the cornea, then the vitreous humour and reaches the **retina** where they are converted into electrical signals by photoreceptors called **rods** and **cones** and sent to the brain with the optic nerve.

The human visual system



Rods detect gray level and low intensities

Cones detect high intensities and different wavelengths (colors)
. Small (S)
. Medium (M)
. Large (L)

The combination of response of the three types of cones allow us perceive a wide diversity of colors.

This process is called **trichromacy**

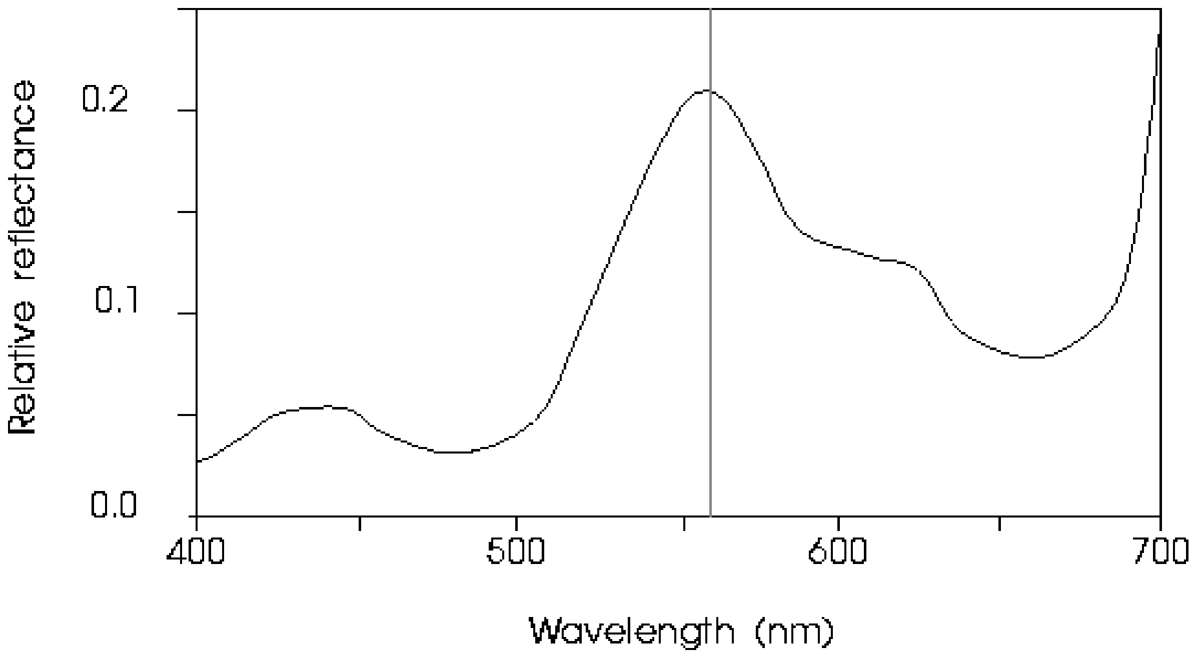
Color

The perception of color depends on the visual system and also on the way human brain processes the information. It varies from species and from individual to individual.

The Stomatopod (mantis shrimp) has an amazing visual system. They have between 12 and 16 types of cones. They can perceive wavelengths of light ranging from deep ultraviolet (UVB) to far-red (300 to 720 nm) and polarized light. Their eyes are independent and each has trinocular vision. Some species can tune the sensitivity of their long-wavelength colour vision to adapt to their environment. (https://youtu.be/kq_aqC-dZ6k)

Color

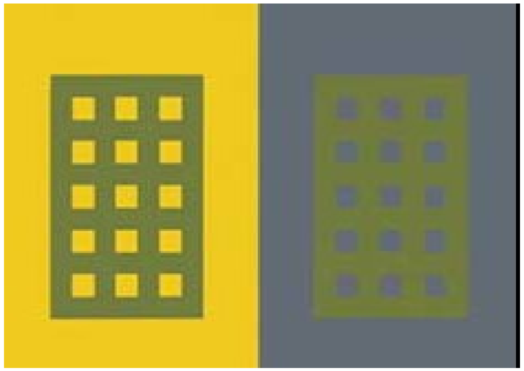
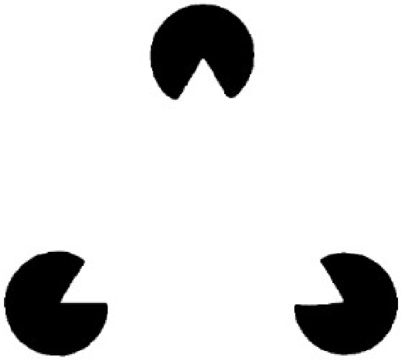
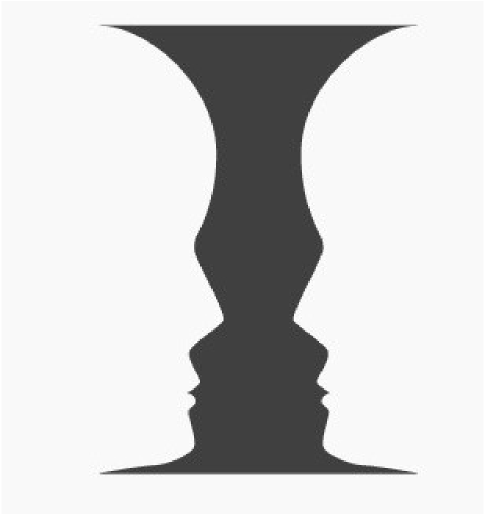
The perception of color depends on the visual system and also on the way human brain processes the information. It varies from species and from individual to individual.



Color

The perception of color depends on the visual system and also on the way human brain processes the information. It varies from species and from individual to individual. It depends on the context and the circumstances.

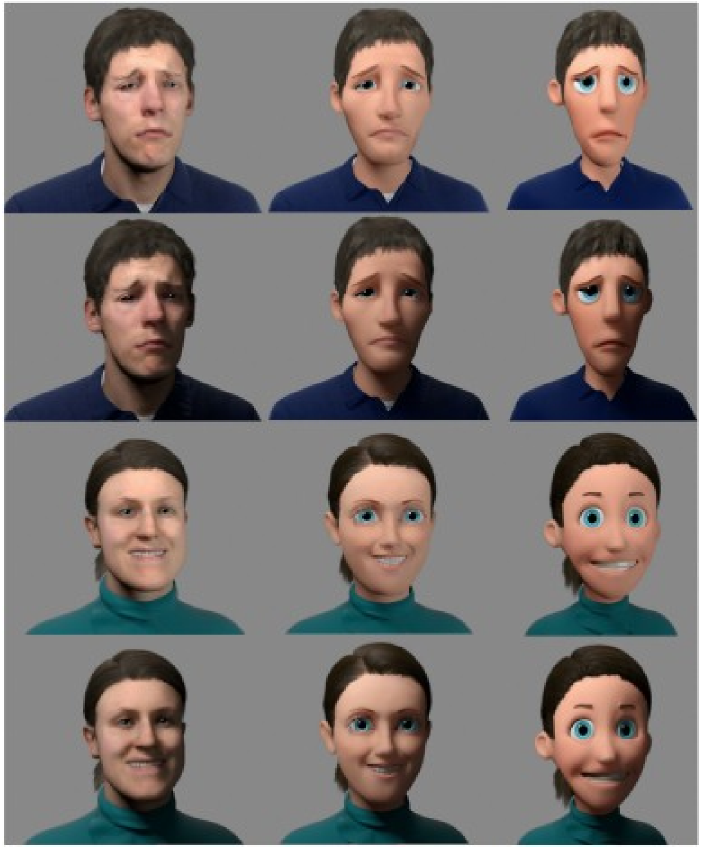
Contrast between the luminance of a spot of light and the background are essential in perception. Moreover the brain ability to group features (Gestalt Principles) yields to optical illusions.


https://www.interaction-design.org/literature/topics/gestalt-principles

Perception

Lighting changes not only our objective perception but also our subjective one. For instance bright light enhances the recognition of an expression joy and dark light the recognition of sadness.



Enlighten Me: Importance of Brightness and Shadow for Character Emotion and Appeal , P. Wisessing et at, 2020, ACM Trans. Graph., Vol. 39, No. 3, Article 19, Publication date: April 2020. DOI: https://doi.org/10.1145/3383195

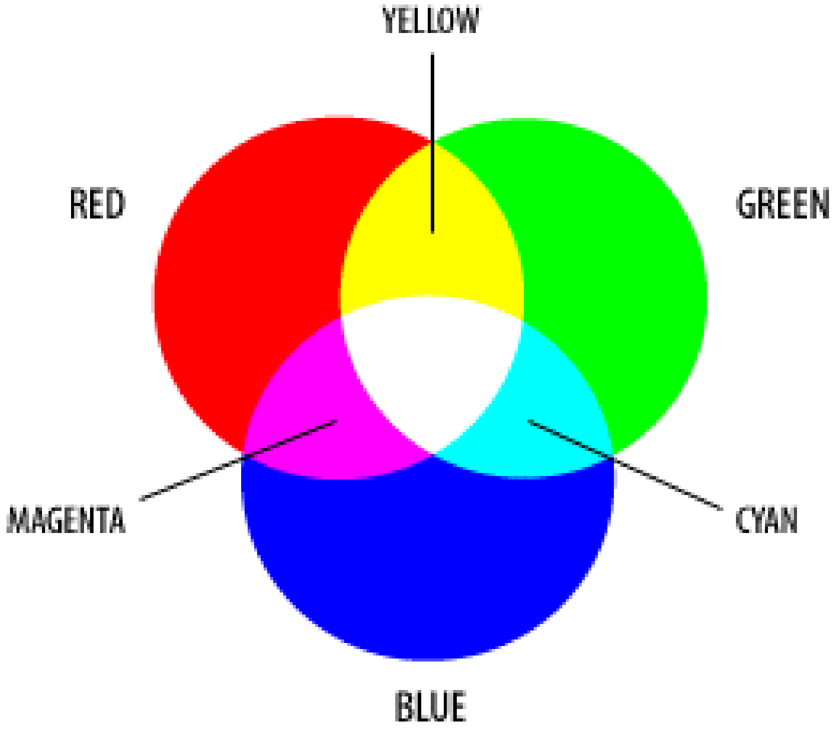
Perception

The perceived image of a real scene or a digital synthesis of the scene is neither the real scene nor the digital image

Modeling color

- **Additive models: RGB**
- **Substractive models: CMY**
- **Perceptual models (Munsell): HLS, HSV, LCH**
- CIE model (CIE XYZ, CIE Lab, CIE Luv)
- Color Naming System CNS
- Video/Photo models: Photo YCC, YCbCr ...

RGB and CYM(K) models



C = RGB - R

Y = RGB - B

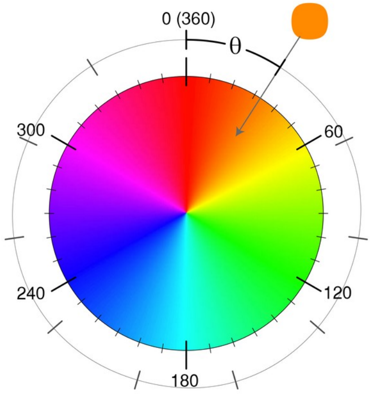
M = RGB - G

In general, R, G, and B (idem CYM) are **codified** with 1 byte per channel: from 0 to 255. They are usually *specified* as so (0 to 255) or in a scale of 0 to 100.

For shading computations, they are represented by floats, ranging from 0.0 to 1.0
In the CYM model K stands for Black and it is used to darken the color.

HSV model

Perceptual model



H = Hue ("gross color"), from 0 to 360º

S = Saturation, quantity of pure color (color = pure color+grey) from 0 to 1 or 0 to 100

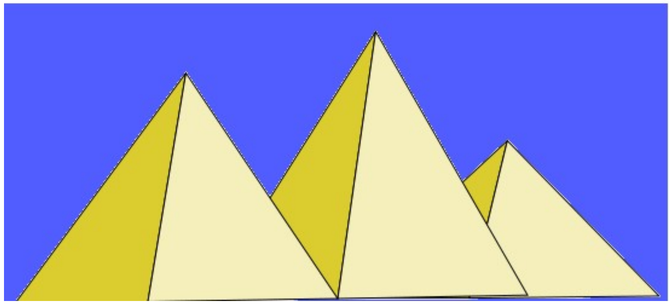
V = Value, luminance, brightness, from 0 to 1 or 0 to 100

HCL (Lch) Hue, Chroma and Luminance is alternative model

Color naming

11 “basic” color names: white, black, gray, red, yellow, green, blue, brown, pink, orange, purple

Blue is the last color has been named. In the Odyssey, Homer does not mention blue a single time. It describes the Mediterranean waters as wine-dark sea.



Some theories about language say that the first words related to color in a language are black and white (dark and light), then red, after green and yellow and only lastly blue. Some current tribes language still don’t have a word for blue. Several studies show that color naming has an impact on the way we differentiate colors.

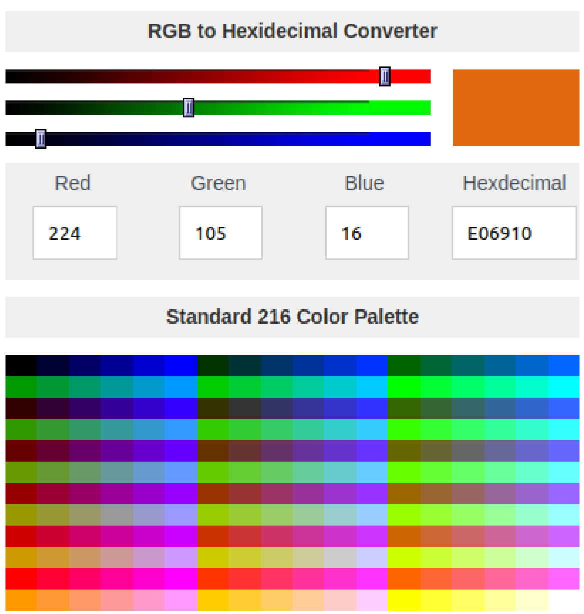
See: Colour categories and category acquisition in Himba and English
D. Roberson, J. Davidoff, I. Davies and L. Shapiro 2006

Color naming

Colors can be specified using hexadecimal codes. This is useful in the implementation of html pages.

Hexadecimal code: 6 characters (2 per channel RGB) each expressed in basis 16:
0 1 2 3 4 5 6 7 8 9 A B C D E F

Color Slider (RGB to hexidecimal converter)



Pink HTML Color Names

COLOR	NAME	HEX CODE	RGB CODE
	Pink	#FFC0CB	rgb(255, 192, 203)
	LightPink	#FFB6C1	rgb(255, 182, 193)
	HotPink	#FF69B4	rgb(255, 105, 180)
	DeepPink	#FF1493	rgb(255, 20, 147)
	MediumVioletRed	#C71585	rgb(199, 21, 133)
	PaleVioletRed	#DB7093	rgb(219, 112, 147)

224//16 = 14 224%16 = 0 → E0
105//16 = 6 105%16 =9 → 69
16//16 = 1 16%16=0 → 10

Conversion RGB HSV

1. Given (R, G, B) compute (R’, G’, B’) = (R/255, G/255, B/255)

2. m = min(R’, G’, B’) and M= max(R’, G’, B’) and D= M-m

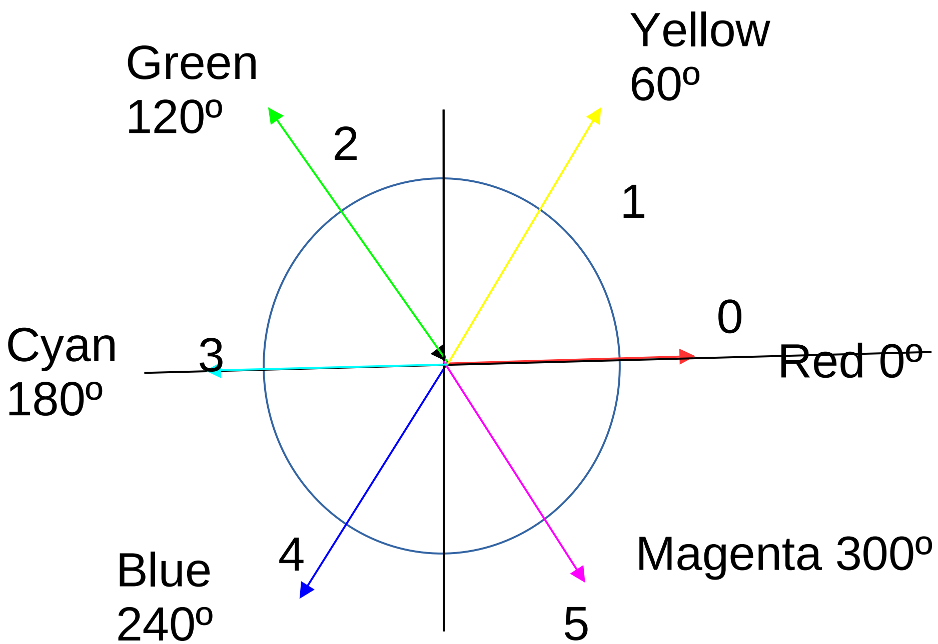
3. Color = (m, m, m) + (R-m, G-m, B-m) → indicates in which octant we are

4. Sat = D/M*100 (if M= 0 → Sat = 0)

5. Value = M*100

6. Hue:
   if M = R’ → Hue = (G-B)/D*60
   M = G’ → Hue = (2+ (B-R)/D)*60
   M = B’ → Hue = (4+(R-G)/D)*60

5



https://www.rapidtables.com/convert/color/rgb-to-hsv.html

Conversion RGB HSV

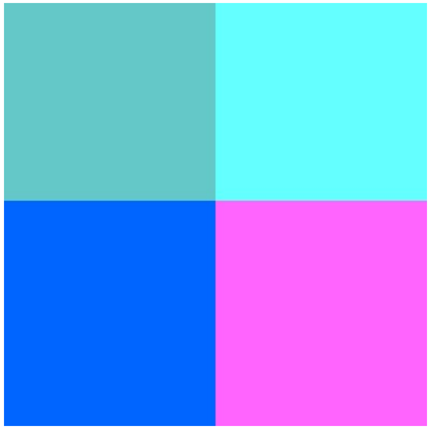
Example: RGB=(124, 30, 210)	RGB=(255, 255, 0)
1. RBG' =(0.49, 0.11, 0.82)	RGB'=(1, 1, 0)
2. m = 0.11, M= 0.82 D= 0.71	m = 0, M = 1, D= 1
3. Color = (0.11, 0.11, 0.11) + (0.38, 0, 0.71) Between blue and red	Color = (1, 1, 0) between red and green (yellow)
4. Sat = D/M*100 = 86.5	Sat = 1/1*100 = 100
5. Value = M*100 = 82	Value= 1*100 = 100
6. Hue = (4+ (0.49-0.11)/D)*60 = 271 L = (m+M)/2 *100 = 46	Hue = (0+(1-0)/1)*60 = 60

Conversion between models

1. Evaluate perceptually the RGB (100, 255, 0)

2. Evaluate perceptually the RGB(100, 50, 50) =(50, 0, 0) + (50, 50, 50)

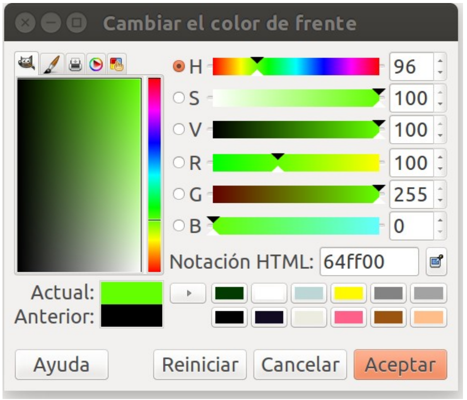
3. Which of the following colors is (100, 200, 200) = (0, 100, 100) + (100, 100, 100)
more likely to correspond?



Conversion between models

1. Evaluate perceptually the RGB (100, 255, 0)

- min(100, 255, 0) = 0 → it is a pure color, fully saturated, without white
- B= 0 → it's a mixture of red and green, more green than red
- G = 255 → maximum brightness.
==> It's a bright lemon green

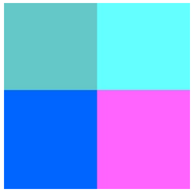


2. Evaluate perceptually the RGB(100, 50, 50)

- *(100, 50, 50) = (50, 0, 0) + (100, 100, 100) Its a dark red half saturated (dark pink)*
  *HSV = (0, 50/100*100, 100/255*100) = (0, 50, 39)*

3. Which of the following colors is (100, 200, 200) more likely to correspond?

(100, 200, 200) = (0, 100, 100) +(100, 100, 100). It's a cyan, thus either one or the other of the top colors. It's not the brightest possible value, thus its the topleft color.
Cyan ==> Hue = 180;
*Saturation= 100/200*100 = 50,*
*Valuee = max(100, 200, 200) / 255 *100 78*



Practice

An open source image editor: https://www.gimp.org/

RGB can be **specified** in a range of 0 to 100 or 0 to 255

H is in range 0-360 (angle in degrees)
S and V in a range 0 to 300 (could be 0 to 100)

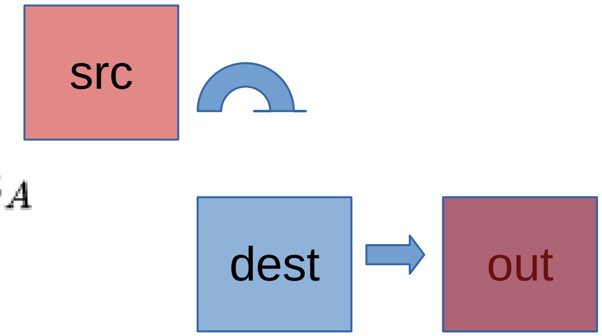
Transparency

The **transparency** (or inversely **opacity**) of a color measures how much of a source color remains when it is is blended with a background color. The opacity is measured in relative terms (from 0 to 1, 0 to 100 or 0 to 255): 0 fully transparent and 1 (or 100 or 255) fully opaque. It is denoted as **alpha (alpha-channel)** (represented with the letter A).

Colors supporting transparency are coded in RGBA, HSVA or CYMA among others.

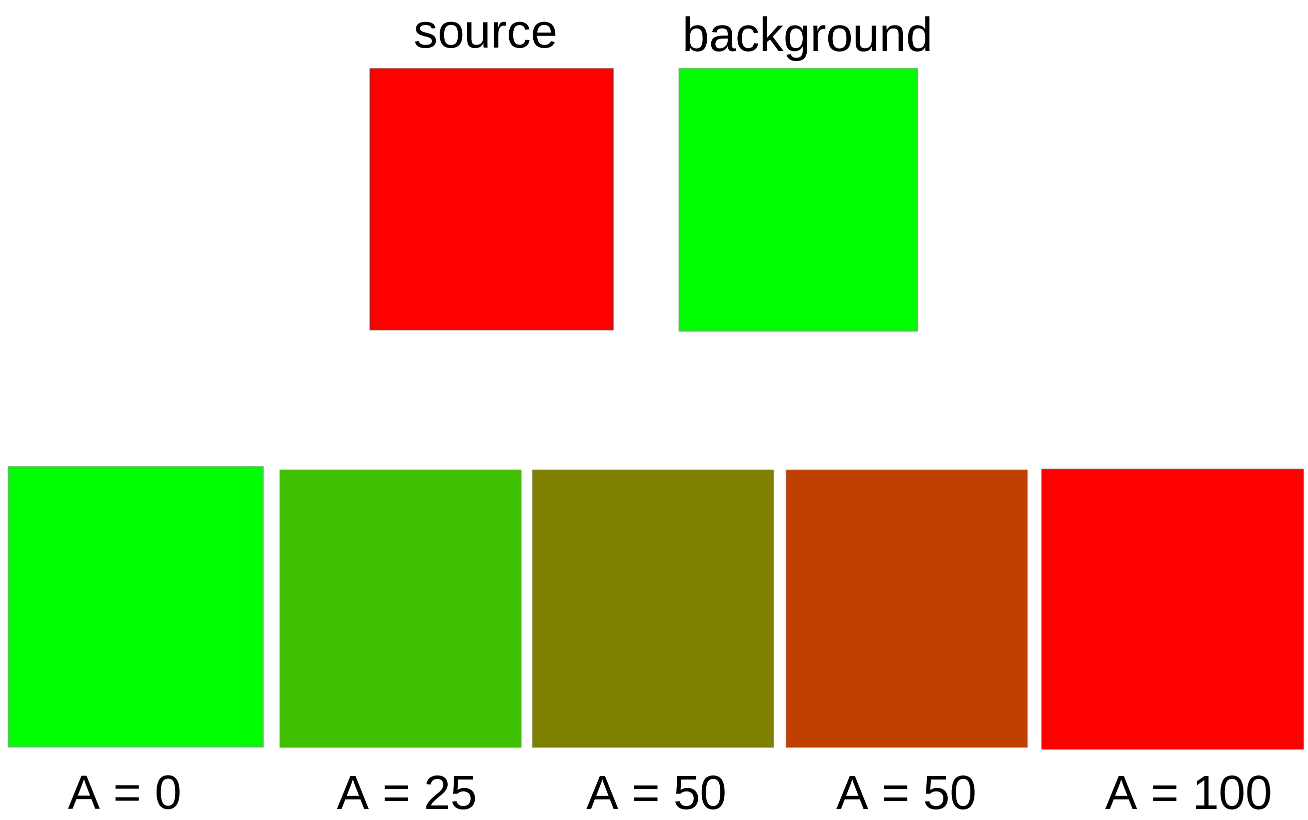
The composition (blending) of colors is done according to different arithmetic operations (over, in, out, atop, xor)[1]. The over operation called *alpha-blending* composition is generally used in images:

$$\begin{cases} \mathrm{out}_A = \mathrm{src}_A + \mathrm{dst}_A(1 - \mathrm{src}_A) \\ \mathrm{out}_{RGB} = \left(\mathrm{src}_{RGB}\mathrm{src}_A + \mathrm{dst}_{RGB}\mathrm{dst}_A\left(1 - \mathrm{src}_A\right)\right) \div \mathrm{out}_A \\ \mathrm{out}_A = 0 \Rightarrow \mathrm{out}_{RGB} = 0 \end{cases}$$



[1] Porter, Thomas; Duff, Tom (July 1984). "Compositing Digital Images" (PDF). SIGGRAPH 1984.

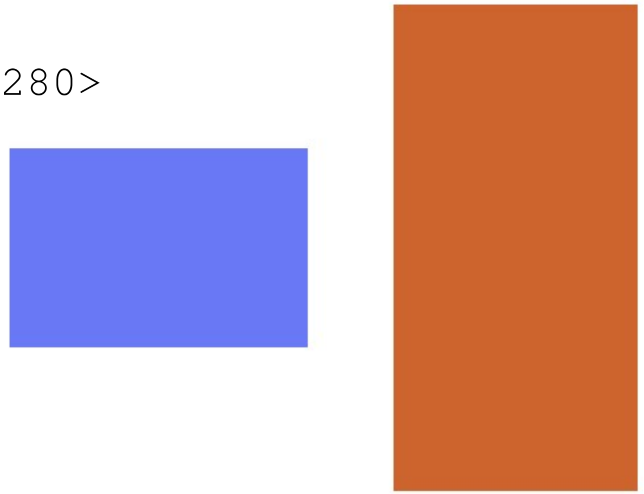
Transparency



Create an image with numpy

Numpy is a python library that provides several classes, in particular ndarray.
Matplotlib is a python library to plot images and graphics.

```
>>> from numpy import ndarray
>>> import numpy as np
>>> im1 = ndarray(shape =(200, 300, 3), dtype=np.uint8)
>>> im2 = ndarray(shape =(300, 150, 3), dtype=np.uint8)
>>> im1[:, :] = (105, 120, 245)
>>> im2[:, :] = (205, 100, 45)
>>> import matplotlib.pyplot as plt
>>> fig, ax = plt.subplots(1, 2)
>>> ax[0].imshow(im1)
<matplotlib.image.AxesImage object at 0x7fcd8c2356a0>
>>> ax[0].axis('off')
(-0.5, 299.5, 199.5, -0.5)
>>> ax[1].imshow(im2)
<matplotlib.image.AxesImage object at 0x7fcd8c234280>
>>> ax[1].axis('off')
(-0.5, 149.5, 299.5, -0.5)
>>> plt.show()
```



Observe the relationship between shape and width/height
Check if the filled value of the matrix coincides with the show color

The class RasImage

From the gitlab project, dowload a preliminary implementation of the class RasImage that represents a raster image composed by 1 to 4 uint8 channels. Analyze the code to see how we create a class method.

Complete the implementation by adding a class method that creates an image composed of a blending of colors

Class RasImage

- Attributes
  - name: the name of the image
- Methods/attributes
  - width()
    - The width of the image
  - height()
    - the height of the image
- Method
  - draw_matplotlib(sp, params)
    - draws the image in the plot sp given a dictionary of parameters

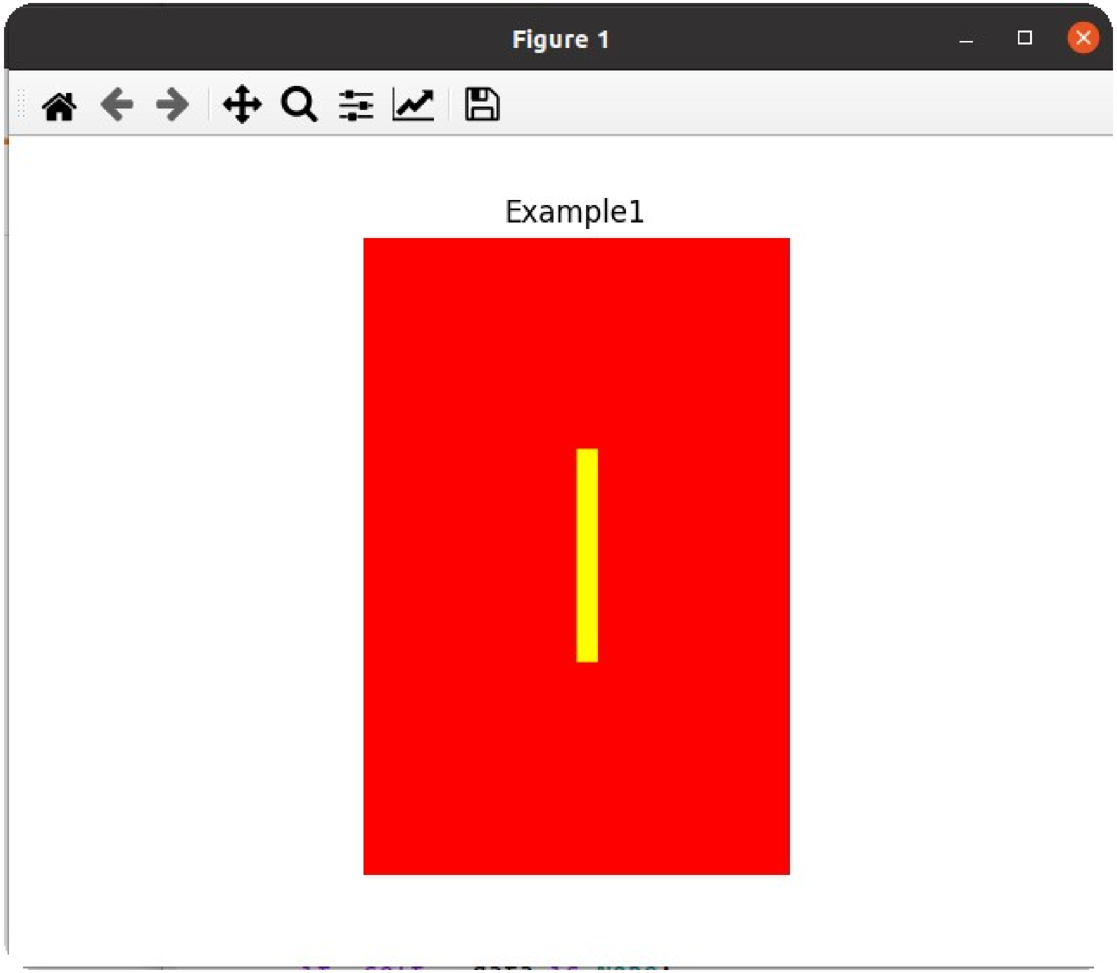
```
Operations

v[i,j]  returns the color of pixel (i, j)
len(v)        renders the number of polygons of
```

The class RasImage

Investigate the class RasImage. Observe how class methods work.

Homework: create a new class method that creates a fully opaque image with horizontal stipes blending hues from h1 to h2 (given parameters).



Next week...more