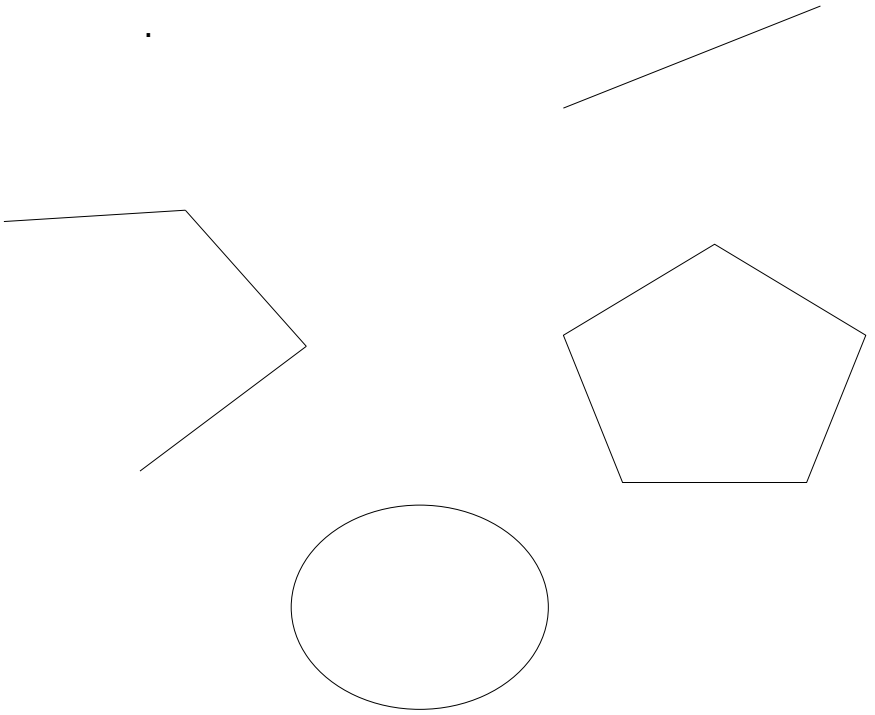
Primitivas Gráficas

Primitivas Gráficas

- Elementos básicos que formam um desenho
  - Ponto
  - Segmento
  - Polilinha
  - Polígono
  - circunferência



PIXEL

- Ponto em matemática → não tem dimensão
- Em computação gráfica → PIXEL
  - Picture element
- Possui posição e valor

Imagem

- Matriz de pixels
  - Resolução: xMáx * yMáx

$$\begin{bmatrix}
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\
1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\
1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\
1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\
1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0
\end{bmatrix}$$

Fig. 1

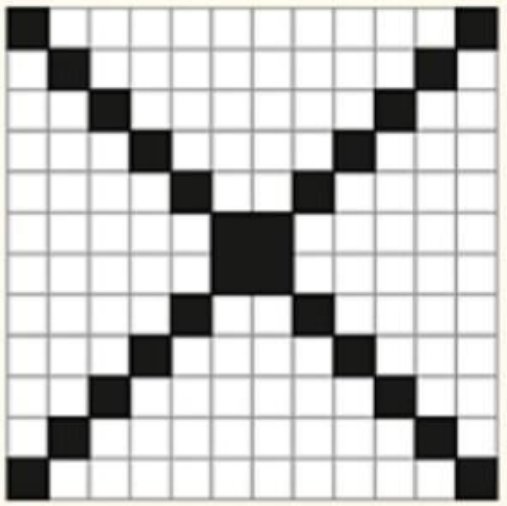


Fig. 2

Imagem

trinket Python3 Run Share

main.py

```
1  from PIL import Image, ImageDraw
2
3  im = Image.new('L',(200, 200),0)
4
5  im.save("a.png")
6
7
8
```

Powered by trinket

a.png

Imagem

Constructing images

`PIL.Image.new(mode, size, color=0) → Image` [source]

Creates a new image with the given mode and size.

PARAMETERS:

- **mode** – The mode to use for the new image. See: Modes.
- **size** – A 2-tuple, containing (width, height) in pixels.
- **color** – What color to use for the image. Default is black. If given, this should be a single integer or floating point value for single-band modes, and a tuple for multi-band modes (one value per band). When creating RGB or HSV images, you can also use color strings as supported by the ImageColor module. If the color is None, the image is not initialised.

RETURNS:

An `Image` object.

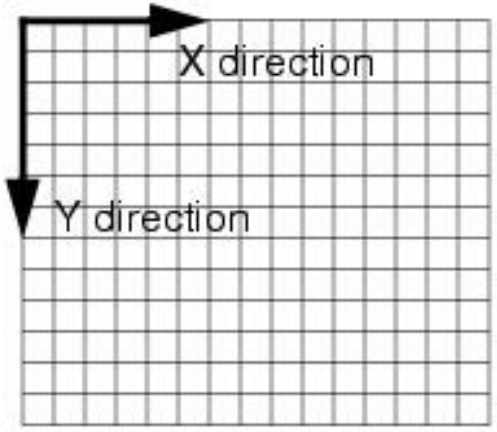
Imagem

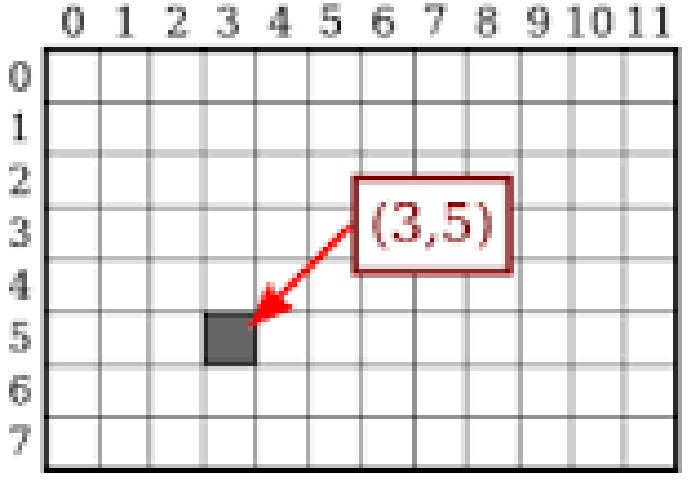
- Modes

  - `1` (1-bit pixels, black and white, stored with one pixel per byte)
  - `L` (8-bit pixels, grayscale)
  - `P` (8-bit pixels, mapped to any other mode using a color palette)
  - `RGB` (3x8-bit pixels, true color)
  - `RGBA` (4x8-bit pixels, true color with transparency mask)
  - `CMYK` (4x8-bit pixels, color separation)
  - `YCbCr` (3x8-bit pixels, color video format)
    - Note that this refers to the JPEG, and not the ITU-R BT.2020, standard
  - `LAB` (3x8-bit pixels, the L*a*b color space)
  - `HSV` (3x8-bit pixels, Hue, Saturation, Value color space)
    - Hue's range of 0-255 is a scaled version of 0 degrees <= Hue < 360 degrees
  - `I` (32-bit signed integer pixels)
  - `F` (32-bit floating point pixels)

Resolução

- Resolução: xMáx * yMáx





Resolução

`ImageDraw.point(xy, fill=None)` [source]

Draws points (individual pixels) at the given coordinates.

PARAMETERS:

- **xy** – Sequence of either 2-tuples like `[(x, y), (x, y), ...]` or numeric values like `[x, y, x, y, ...]`.
- **fill** – Color to use for the point.

Resolução

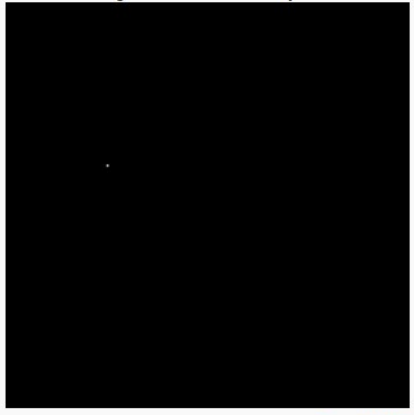
trinket Python3 Run Share

main.py

```
1  from PIL import Image, ImageDraw
2
3  im = Image.new('L',(200, 200),0)
4
5  w,h = im.size
6  print("resolução",w*h,"pixels")
7
8  draw = ImageDraw.Draw(im)
9  draw.point((50,80),255)
10
11 im.save("a.png")
12
13
```

Powered by trinket

```
resolução 40000 pixels
```



a.png

Resize

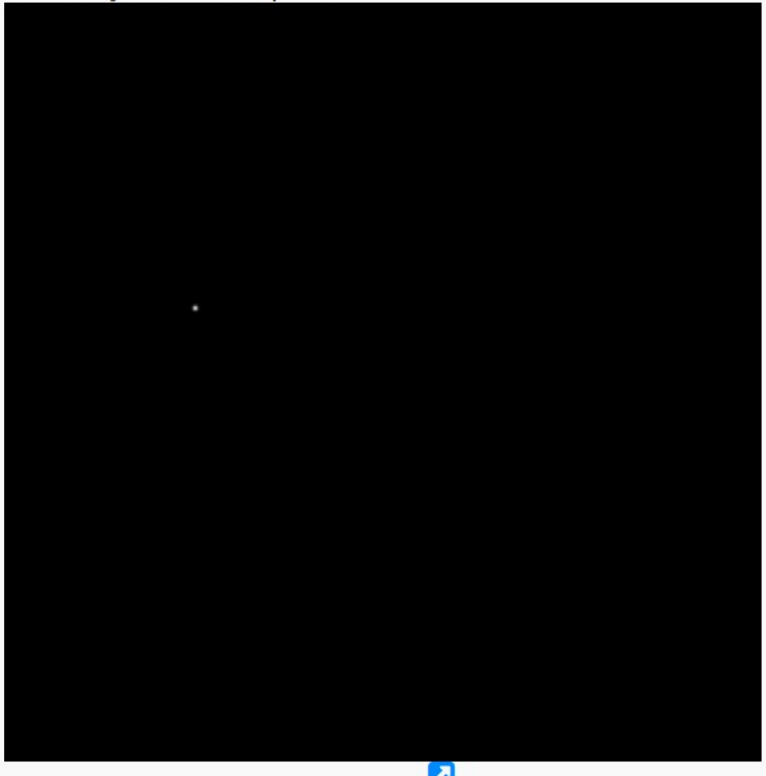
trinket Python3 Run Share

main.py

```
1  from PIL import Image, ImageDraw
2
3  im = Image.new('L',(200, 200),0)
4
5  w,h = im.size
6  print("resolução",w*h,"pixels")
7
8  draw = ImageDraw.Draw(im)
9  draw.point((50,80),255)
10
11
12 im2 = im.resize((400,400))
13 im2.show()
14 im2.save("a.png")
15
16
17
```

Powered by trinket

```
resolução 40000 pixels
```



a.png

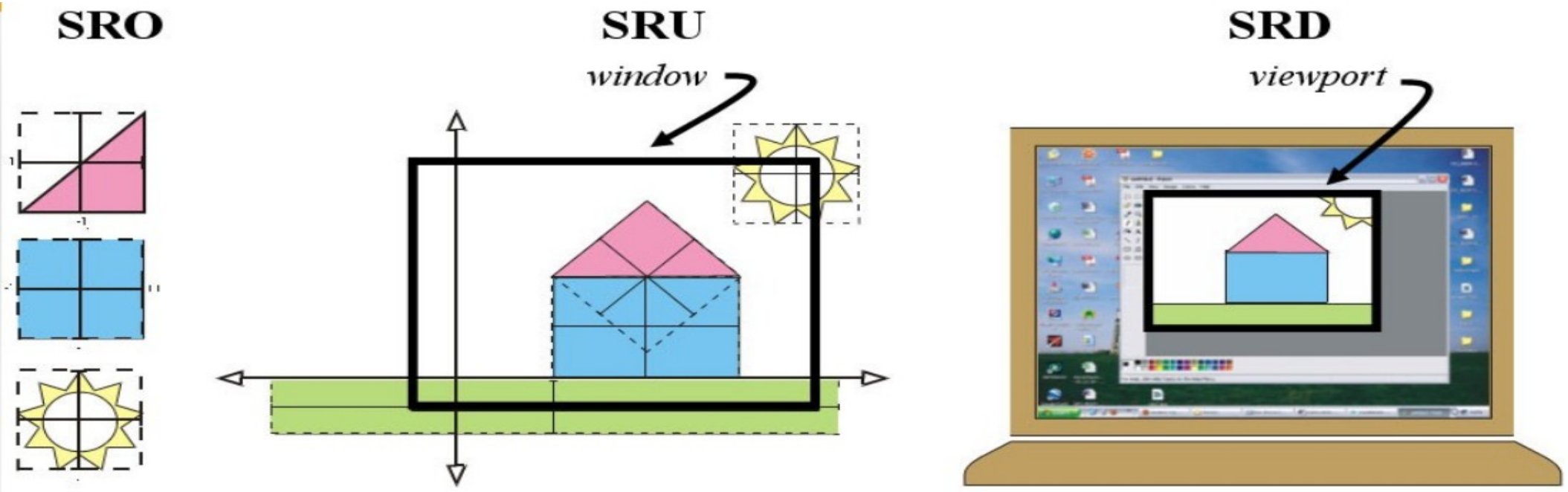
Sistemas de Referência

- SRU → Sist. de Ref. do Universo
  - Xr e Yr
  - Coordenadas reais de um ponto no espaço R2
- SRD → Sist. de Ref. do Dispositivo
  - Xp e Yp
  - Coordenada do pixel correspondente
    - Inteiros positivos na matriz gráfica
- SRO – Sist. de Ref. do Objeto
  - Cada objeto pode ser modelado em um universo próprio

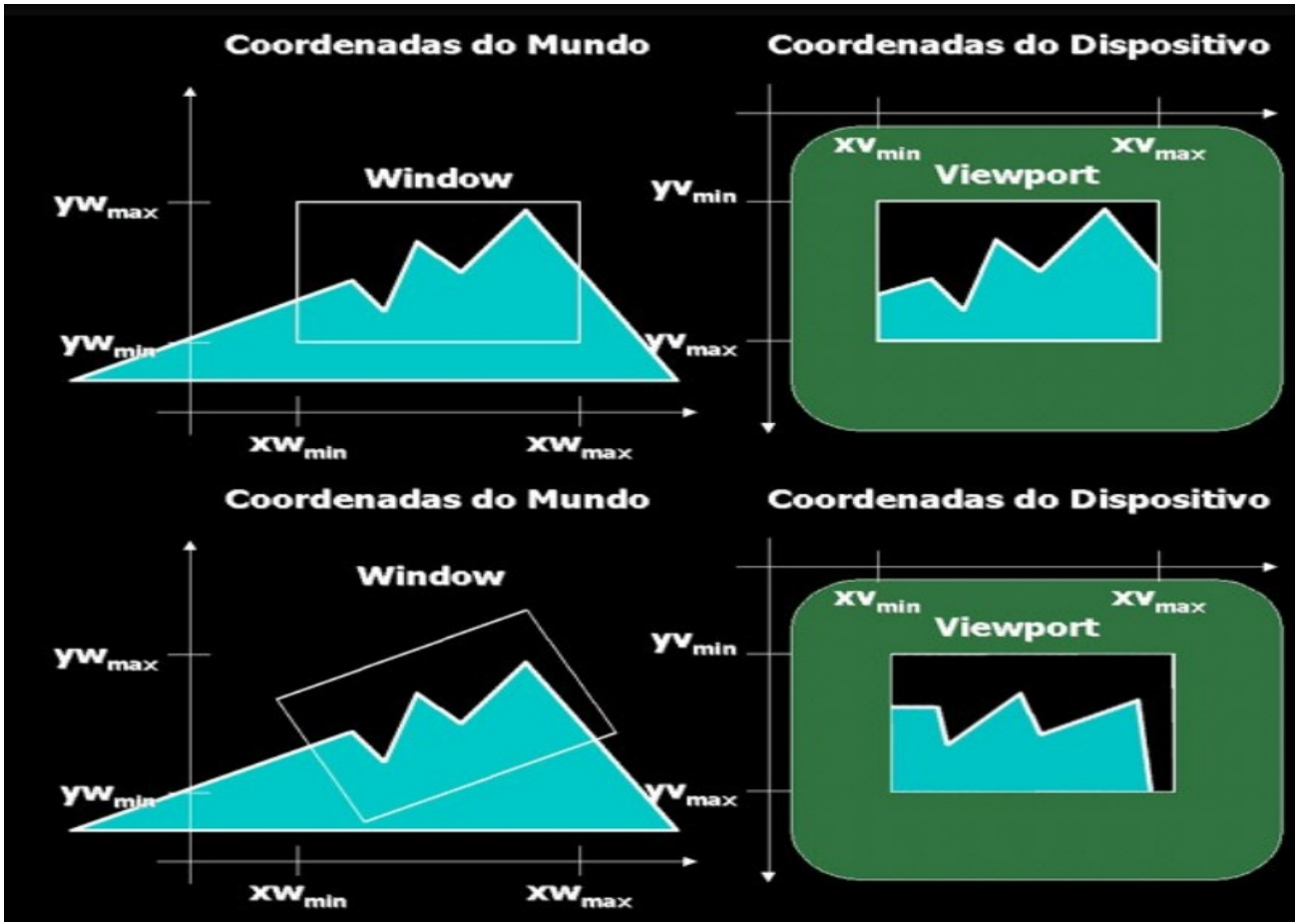
Window x viewport

- Plano cartesiano infinito
- Window
  - janela na qual vemos uma cena
- Viewport
  - Matriz de pixels
  - Resolução
  - Valores max e min

Sistemas de Referência



Sistemas de Referência



Rasterização

- Conversão: SRU → SRD
  - (Xr,Yr) → (Xp,Yp)

$$Xp = \frac{Xp_{max} - Xp_{min}}{Xr_{max} - Xr_{min}} * (Xr - Xr_{min}) + Xp_{min}$$

$$Yp = Yp_{max} - \frac{Yp_{max} - Yp_{min}}{Yr_{max} - Yr_{min}} * (Yr - Yr_{min}) + Yp_{min}$$

Exercício

- Considere o ponto: (Xr,Yr) = (-1.4,0.3)
- Window definida entre -2.0 e +2.0
  - Xrmin = Yrmin = -2
  - Xrmax = Yrmax = 2
- Viewport d 800x600 pixels
  - Xpmin = Ypmin = 0
  - Xpmax = 800
  - Ypmax = 600
- Qual o valor da coordenada do pixel: (Xp,Yp)

Exemplo

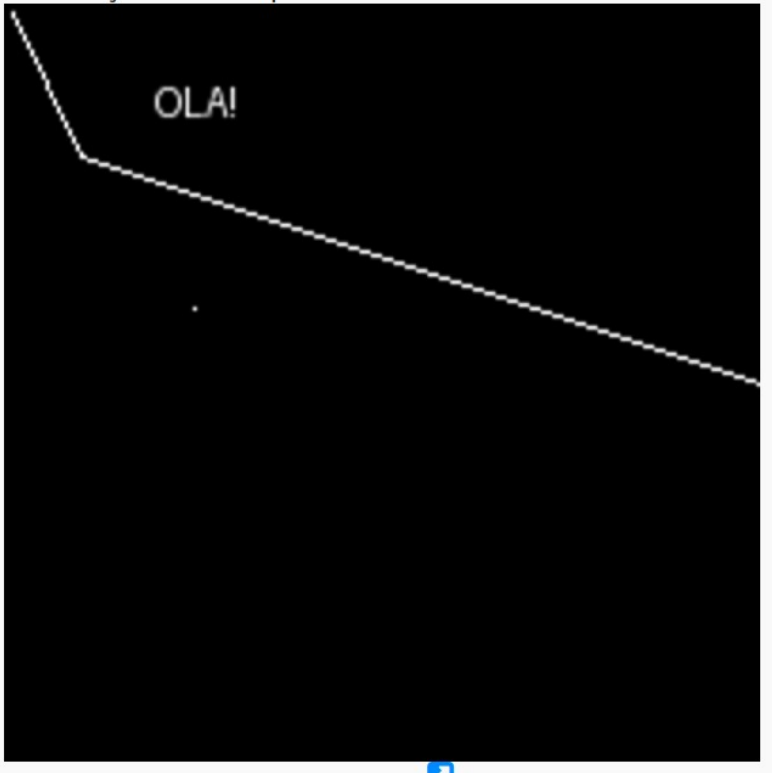
trinket Python3 Run Share

main.py

```
1  from PIL import Image, ImageDraw
2
3  im = Image.new('L',(200, 200),0)
4
5  w,h = im.size
6  print("resolução",w*h,"pixels")
7
8  draw = ImageDraw.Draw(im)
9  draw.point((50,80),255)
10
11 draw.line((2,2, 20,40), 255)
12 draw.line((20,40, 200,100), 255)
13
14 draw.text((40,20), "OLA!", 255)
15
16 im2 = im.resize((400,400))
17 im2.save("a.png")
18
19
20
```

Powered by trinket

resolução 40000 pixels



a.png

Exemplo

`ImageDraw.line(xy, fill=None, width=0, joint=None)` [source]

Draws a line between the coordinates in the `xy` list. The coordinate pixels are included in the drawn line.

PARAMETERS:

- **xy** – Sequence of either 2-tuples like `[(x, y), (x, y), ...]` or numeric values like `[x, y, x, y, ...]`.
- **fill** – Color to use for the line.
- **width** –

  The line width, in pixels.

  New in version 1.1.5.

  > Note
  >
  > This option was broken until version 1.1.6.

- **joint** –

  Joint type between a sequence of lines. It can be `"curve"`, for rounded edges, or `None`.

  New in version 5.3.0.

Exemplo

```
ImageDraw.text(xy, text, fill=None, font=None, anchor=None, spacing=4, align='left',
    direction=None, features=None, language=None, stroke_width=0, stroke_fill=None,
    embedded_color=False, font_size=None)                                    [source]
```

Draws the string at the given position.

PARAMETERS:

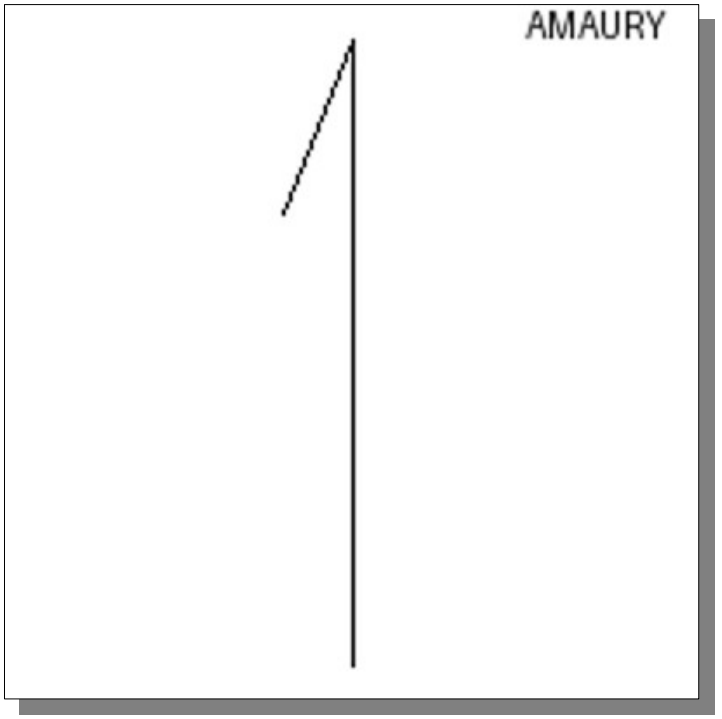
- **xy** – The anchor coordinates of the text.
- **text** – String to be drawn. If it contains any newline characters, the text is passed on to `multiline_text()`.
- **fill** – Color to use for the text.
- **font** – An `ImageFont` instance.
- **anchor** –

  The text anchor alignment. Determines the relative location of the anchor to the text. The default alignment is top left, specifically `la` for horizontal text and `lt` for vertical text. See Text anchors for details. This parameter is ignored for non-TrueType fonts.

  > **Note**
  >
  > This parameter was present in earlier versions of Pillow, but implemented only in version 8.0.0.

Lab 1

- Desenhe uma imagem com o último dígito do seu RA no centro e seu nome escrito no canto superior direito.
- Exemplo: A23621, Amaury



Exercício

Exercício: Considere um SRU com $Xu_{min}=0$ e $Xu_{max}=10$, $Yu_{min}=0$, $Yu_{max}=8$, e que o objeto definido pelos pontos abaixo foi mapeado para um dispositivo de 1280 X 1024.

Apresente as coordenadas do objeto no SRD.

$P_1=(3,2)$

$P_2=(4,7)$

$P_3=(5,2)$

$P_4=(2,6)$

$P_5=(6,6)$