



Python e Jogos

The background features several large, overlapping pastel shapes in shades of peach, light blue, and lavender. A thin, wavy yellow line meanders across the center of the image, passing behind the text.

01 Hello World!

Uma rápida introdução para colocarmos a mão na massa!

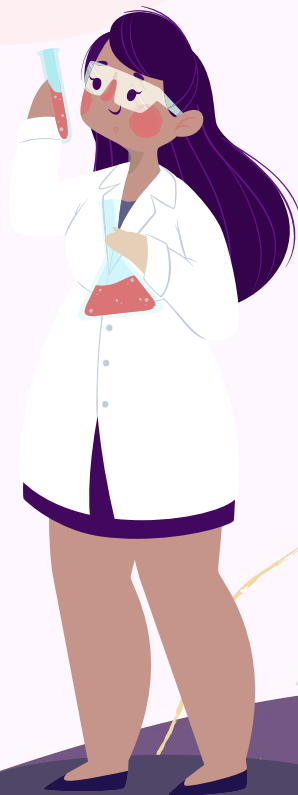
Quem somos

“PyLadies DF é um grupo de mulheres maravilhosas que gostam de programar em Python, se ajudar, e incentivar mais mulheres nas áreas de tecnologia”

– Bruna Moreira



pyladiesdf





O que queremos fazer

Aprender programação criando um jogo rápido

Como vamos fazer isso

Pegando o jogo pronto, e fazendo do nosso jeito

```
    }) done(function(response) {
      for (var i = 0; i < response.length; i++) {
        var layer = L.marker(
          [response[i].latitude, response[i].longitude]
          // , {icon: myIcon}
        );
        layer.addTo(group);

        layer.bindPopup(
          "<p>" + "Species: " + response[i].species + "<br>"
          "<p>" + "Description: " + response[i].description + "<br>"
          "<p>" + "Seen at: " + response[i].latitude + "<br>"
          "<p>" + "On: " + response[i].sighted_at + "<br>"
        );
      }

      $('select').change(function() {
        species = this.value;
      });
    });
  });
  $.ajax({
    url: queryURL,
    method: "GET"
  }) done(function(response) {
    for (var i = 0; i < response.length; i++) {
      var layer = L.marker(
        [response[i].latitude, response[i].longitude]
        // , {icon: myIcon}
      );
      layer.addTo(group);
    }
  });
}
```

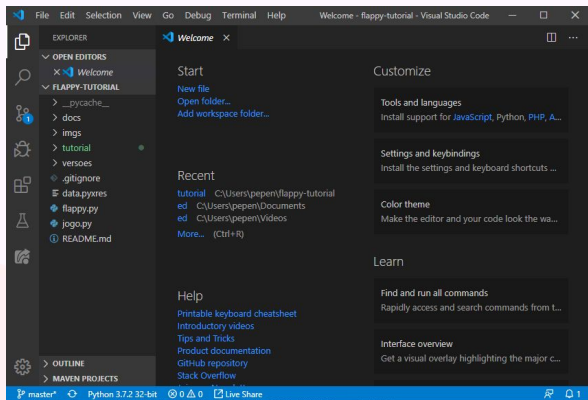


02

Ambiente de Programação

O que precisaremos?

O que precisaremos?



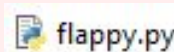
Editor de código

Onde escreveremos o código
fonte dos programas

Interpretador de Python


Programa que lê e executa
instruções codificadas

```
Python 3.7 (32-bit)
t (Intel) on win32
type "help", "copyright", "credits" or "license" for more information.
>>> a = 'Hello'
>>> print a
File "stdin", line 1
  print a
SyntaxError: Missing parentheses in call to 'print'. Did you mean print(a)?
>>> print a()
File "stdin", line 1
  print a()
SyntaxError: invalid syntax
>>> print(a)
Hello
>>> b = 'world!'
>>> print(a+b)
Helloworld!
>>> print(a+""+b)
Helloworld!
>>> print(a+""+b)
Hello world!
>>>
```



Módulos adicionais

Para aumentar as
funcionalidade

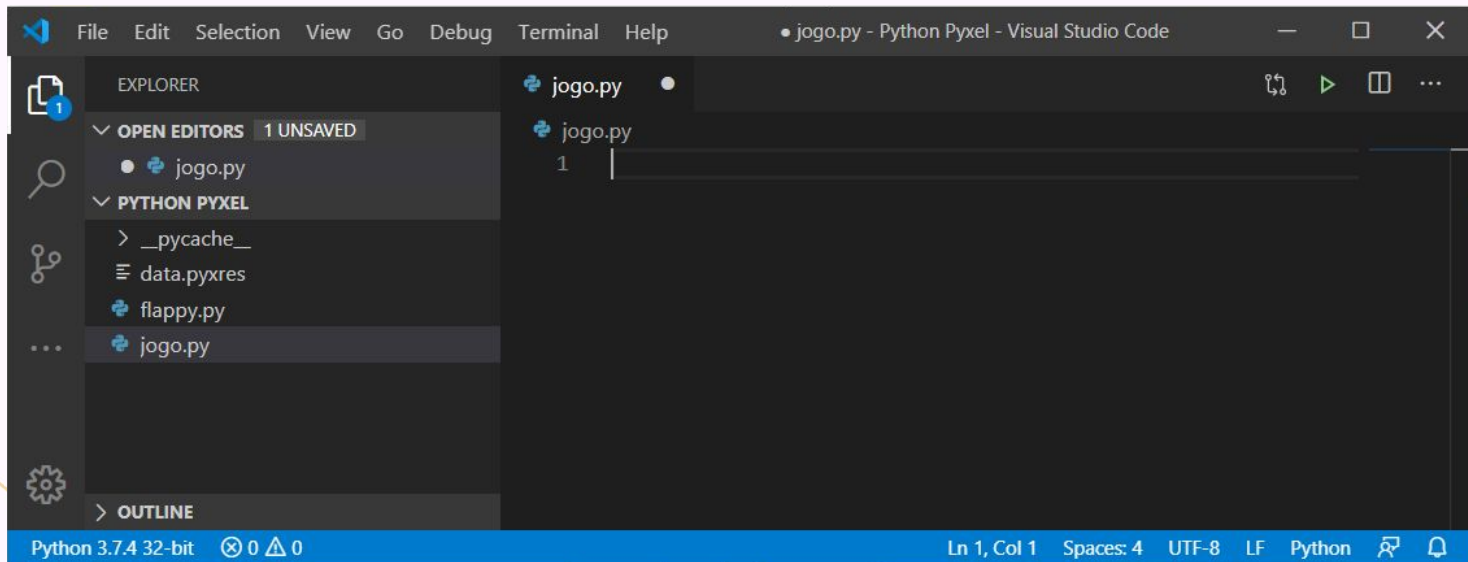
The background features a light lavender color with several large, soft-edged shapes in teal, light orange, and dark purple. Thin, hand-drawn gold lines are scattered across the page, some forming loops and others as straight strokes.

03

Iniciando o jogo!

Primeira parte do tutorial

Preparando o Editor...



Editor com o arquivo jogo.py

Executando o jogo

```
jogo.py x
jogo.py > ...
1 import flappy
2
3 flappy.comecar()
```

Nosso código

Comando no terminal

```
OUTPUT  TERMINAL  DEBUG CONSOLE  PROBLEMS  2: python
PS C:\Users\jununes\Documents\Python> python jogo.py
```

Variáveis

```
jogo.py  ×  
jogo.py > ...  
1  import flappy  
2  
3  largura_tela = 150  
4  altura_tela = 255  
5  gravidade = 1.0  
6  pulo = 8.0  
7  
8  flappy.comecar()
```

E se modificarmos esses valores?

Importante: os valores da tela devem estar entre 0 e 255.



04

Desenhando na Tela

Desenhar na tela

```
17 #=====
18 # Funções
19 #=====
20 def desenhar():
21     desenhar_fundo()
22     desenhar_nuvens()
23     desenhar_canos()
24     desenhar_chao()
25     desenhar_flappy()
26     desenhar_instrucoes()
27
```



Funções

```
def funcao():  
    instrucao1  
    instrucao2  
    # Comentário  
    instrucao3
```

Desenhando o fundo

```
import flappy
import pixel
```

Módulos adicionais

0	#000000 0, 0, 0	1	#2B335F 43, 51, 95	2	#7E2072 126, 32, 114	3	#19959C 25, 149, 156
4	#BB4B52 139, 72, 82	5	#395C9B 57, 92, 152	6	#A9C1FF 169, 193, 255	7	#EEEEEE 238, 238, 238
8	#D41B6C 212, 24, 108	9	#D3B441 211, 132, 65	10	#E9C35B 233, 195, 91	11	#70C6A9 112, 198, 169
12	#7696DE 118, 150, 222	13	#A3A3A3 163, 163, 163	14	#FF979B 255, 151, 152	15	#EDC7B0 237, 199, 176

Cores disponíveis

```
def desenhar_fundo():
    pixel.cls(12)
```

Usando cores

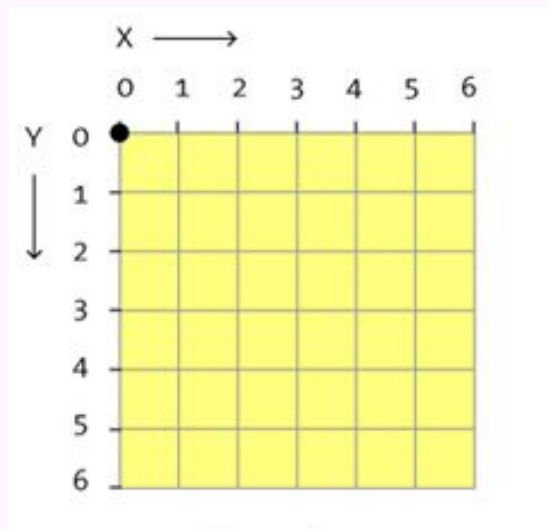
Desenhando figuras geométricas

```
def desenhar_flappy():  
    largura = 17  
    altura = 13  
    cor = 10  
    pixel.rect(flappy_x, flappy_y, largura, altura, cor)
```

`pixel.rect(x, y, largura, altura, cor)`

O que esse código desenha?

Sistemas de Coordenadas e Cores



Posição na tela

Pyxeledit

Abrir editor

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS 6

PS C:\Users\jununes\Documents\Python> **pyxeleditor** data.pyxres

```
def desenhar_flappy():  
    # Número da imagem  
    img = 0  
  
    # Posição inicial no Pyxeleditor  
    u = 0  
    v = 0  
  
    # Tamanho da imagem  
    largura = 17  
    altura = 13  
  
    # Cor tratada como transparente  
    mascara = 0  
  
    # Chamamos a função blt para desenhar tudo isso!  
    pyxel.blt(flappy_x, flappy_y, img, u, v, largura, altura, mascara)
```

Desenho na tela do jogo



Edição de desenhos

Animando o Flappy Bird

```
def desenhar_flappy():  
    # Fixamos o frame  
    frame = 0  
  
    # Posição inicial no Pyxeleditor agora depende do frame escolhido  
    u = 0  
    v = frame * 16  
  
    # O resto continua como antes...  
    img = 0  
    largura = 17  
    altura = 13  
    mascara = 0  
    pixel.blit(flappy_x, flappy_y, img, u, v, largura, altura, mascara)
```

Animação do personagem

Animando o Flappy Bird

```
frame = pyxel.frame_count % 3
```

Mudando o desenho

```
frame = (pyxel.frame_count // 4) % 3
```

```
def desenhar_flappy():  
    # Escolhemos qual dos 3 frames 0, 1 ou 2 utilizar  
    frame = (pyxel.frame_count // 4) % 3  
  
    # Número da imagem  
    img = 0  
  
    # Posição inicial no Pyxeleditor  
    u = 0  
    v = frame * 16  
  
    # Tamanho da imagem  
    largura = 17  
    altura = 13  
  
    # Cor tratada como transparente  
    mascara = 0  
  
    # Chamamos a função blt para desenhar tudo isso!  
    pyxel.blt(flappy_x, flappy_y, img, u, v, largura, altura, mascara)
```

Animação do personagem

Desenhando os Canos

```
def desenhar_canos():  
    cor = 11  
    largura = 25  
    altura = 135  
    x, y = canos[0]  
  
    # Cano superior  
    pyxel.rect(x, y, largura, altura, cor)  
  
    # Cano inferior  
    pyxel.rect(x, y + abertura_cano, largura, altura, cor)
```

Canos no cenário

Elemento de Repetição: For

```
for elemento in lista:  
    comandos que dependem de elemento
```

Passando por toda lista

```
# Percorre a posição x, y de cada cano  
for x, y in canos:  
    # Desenha o cano na posição x, y  
    pyxel.rect(x, y, largura, altura, cor)  
    pyxel.rect(x, y + abertura_cano, largura, altura, cor)
```

Desenhando cada cano na lista de canos

Desenhando os Canos

```
def desenhar_canos():  
    cor = 11  
    largura = 25  
    altura = 135  
    for x, y in canos:  
        pyxel.rect(x, y, largura, altura, cor)  
        pyxel.rect(x, y + abertura_cano, largura, altura, cor)
```

Canos no cenário

Canos em Pixel Art

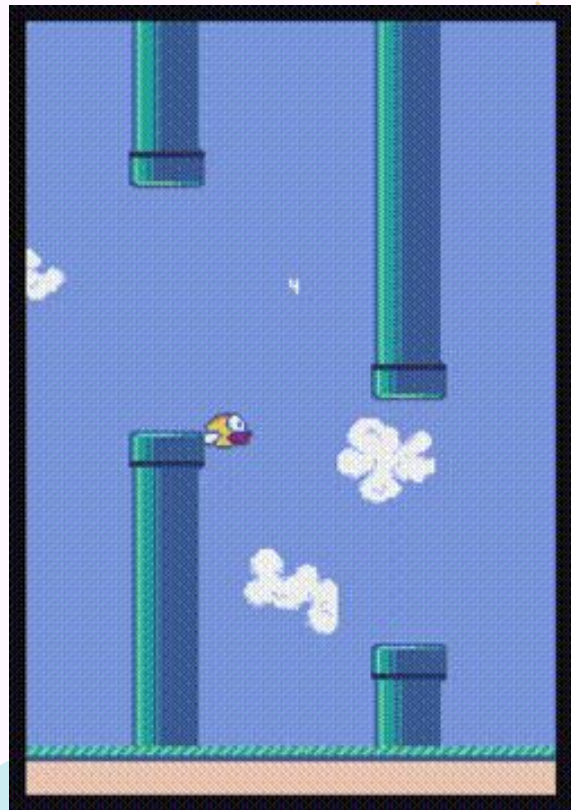
```
def desenhar_canos():  
    img = 1          # índice da imagem (0 a 3)  
    u = 0            # posição x no banco de imagens  
    v = 0            # posição y no banco de imagens  
    largura = 25     # largura em pixels  
    altura = 150     # altura em pixels  
    cor = 0          # cor considerada transparente ao desenhar a imagem (opcional)  
    for x, y in canos:  
        pixel.blit(x, y, img, u, v, largura, altura, cor)  
        pixel.blit(x, y + abertura_cano, img, u, v, largura, -altura, cor)
```

Colocando vários canos com o desenho

Lógica do Jogo

```
def atualizar_jogo():  
    atualizar_flappy()  
    atualizar_canos()  
    atualizar_colisoes()  
    atualizar_score()
```

Loop do jogo



Simulando Gravidade

```
flappy_x = largura_tela / 3  
flappy_y = altura_tela / 2  
velocidade = 0  
morto = False
```

```
def atualizar_flappy():  
    velocidade = velocidade + gravidade  
    flappy_y = flappy_y + velocidade
```

```
velocidade = velocidade + gravidade  
flappy_y = flappy_y + velocidade
```

```
def atualizar_flappy():  
    global velocidade, flappy_x, flappy_y  
  
    velocidade = velocidade + gravidade  
    flappy_y = flappy_y + velocidade
```

Pulos e Interação com o teclado

```
pulando = pyxel.btnp(pyxel.KEY_SPACE)
```

Pulando com espaço

```
pulando = pyxel.btnp(pyxel.KEY_SPACE) or pyxel.btnp(pyxel.KEY_UP)
```

Pulando com tecla para cima

```
if pulando:  
    velocidade = velocidade_de_pulo # escolhemos a velocidade do pulo
```

Velocidade do pulo

Pulos e Interação com o teclado

```
def atualizar_flappy():  
    global velocidade, flappy_x, flappy_y  
  
    pulando = pyxel.btnp(pyxel.KEY_SPACE) or pyxel.btnp(pyxel.KEY_UP)  
  
    # Atualiza a velocidade  
    velocidade += gravidade  
  
    # Verifica se está pulando antes de atualizar a posição  
    if pulando:  
        velocidade = -pulo  
  
    # Atualiza a posição  
    flappy_y += velocidade  
  
    # Limita altura  
    if flappy_y > altura_tela - 29:  
        flappy_y = altura_tela - 29
```

Pulando

Vivo ou Morto?

```
if morto:
    flappy_x = flappy_x - 1
```

Tá morto??

```
def atualizar_flappy():
    global velocidade, flappy_x, flappy_y

    pulando = pyxel.btnp(pyxel.KEY_SPACE) or pyxel.btnp(pyxel.KEY_UP)

    # Atualiza a velocidade
    velocidade += gravidade

    # Verifica se está pulando antes de atualizar a posição
    if pulando:
        velocidade = -pulo

    # Atualiza a posição
    flappy_y += velocidade

    # Limita altura
    if flappy_y > altura_tela - 29:
        flappy_y = altura_tela - 29

    # Desloca-se com o cenário, caso esteja morto
    if morto:
        flappy_x -= 1
```

The background features several large, overlapping pastel-colored shapes: a large light orange circle in the top left, a light purple circle in the center, a light teal circle in the bottom left, and a light orange circle in the bottom right. A thin, wavy yellow line starts from the top right and curves towards the center, passing behind the text.

05

Finalizando!

Pontuação

```
score = 0

def atualizar_score():
    global score

    # Percorre as coordenadas x e y de cada cano
    for (x, y) in canos:

        # Verifica se a posição do cano coincide com a do passarinho e se ele
        # não está morto
        if flappy_x == x and not morto:
            score = score + 1
```

Quantos pontos você fez?

The background features several large, soft-edged pastel shapes in shades of peach, light blue, and lavender. A thin, wavy yellow line meanders across the center of the image, passing behind the text.

06
Ε agora??

Links úteis

- Tutorial Completo Flappy Bird
- Pyxel
- PyGames
- Python para Zumbis



Avaliação do
treinamento



Dúvidas



Treinamento
completo

Conheça mais sobre as PyLadies!

“ O PyLadies é uma comunidade mundial que foi trazida ao Brasil com o propósito de incentivar mais mulheres a entrarem na área tecnológica”



pyladiesdf



pyladiesdf

Obrigada!



Entre em contato com a gente!

brasil@pyladies.com
df.pyladies.com

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**.

