

# Tutorial do VSCode

Python para Todos

May 30, 2025

# Sumário

Introdução

Como definir o tipo de um arquivo?

Criando um arquivo

Executando o arquivo

O que é o Pygame?

Vantagens do Pygame

O que iremos criar nesse curso?

Uma breve visão do Pygame

Criando o jogo da cobrinha

# Introdução

O Visual Studio Code (VSCode) é um editor de código gratuito da Microsoft, rápido e personalizável, com:

- ▶ Destaque de sintaxe e autocompletar
- ▶ Extensões para várias linguagens (Python, JavaScript, etc.)
- ▶ Suporte a Git
- ▶ Interface moderna

Ideal para programadores de todos os níveis.

# Como definir o tipo de um arquivo?

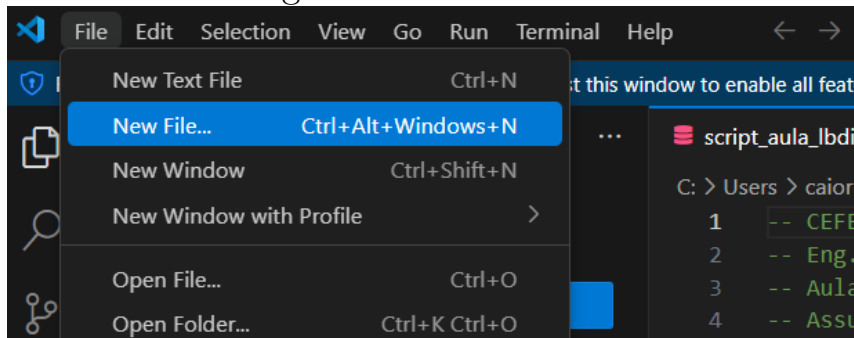
Para fazer isso no VSCode basta após inserir o nome do arquivo digitar “.” seguido do seu tipo.

Exemplo com o arquivo de nome “teste” em Python:

Nome do arquivo: teste.py

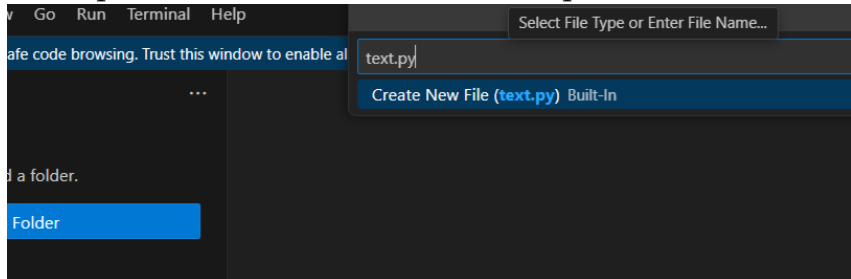
# Criando um arquivo

Primeiro passo: Clicar em "File", no canto superior esquerdo, e em seguida em "New File ..."



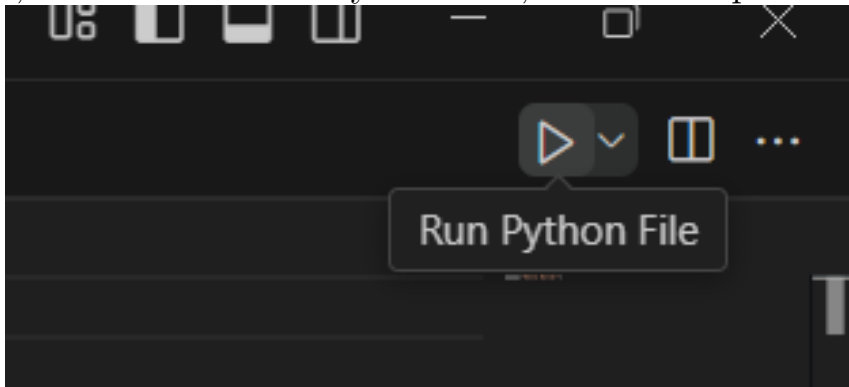
# Criando um arquivo

Segundo passo: Definir o nome do arquivo como "teste.py"



# Executando o arquivo

Digitar `print("Hello World!")` E em seguida vamos executar o código, clicando em Run Python File, no canto superior direito



# Executando o arquivo

```
PS C:\Users\caior> & C:/Users/caior/AppData/Local/Programs/Python/Python313/Python313.exe C:/Users/caior/AppData/Local/Programs/Python/Python313/Scripts/python.exe C:/Users/caior/AppData/Local/Programs/Python/Python313/Scripts/hello.py  
Hello World!  
PS C:\Users\caior> 
```



# O que é o Pygame?

O PyGame é uma biblioteca do Python para criação de jogos 2D, permitindo:



- ▶ Criar movimentação para personagens
- ▶ Gerar cenário
- ▶ Colocar música
- ▶ Colocar Sprites para os personagens

# Vantagens do Pygame

- ▶ Simples de aprender (utiliza Python)
- ▶ Não é uma Game Engine pesada para rodar
- ▶ Porta de entrada para outras plataformas de criação de jogos, como:
  - ▶ **Godot** (utiliza uma linguagem semelhante à Python)
  - ▶ **Unreal Engine**
  - ▶ **Game Maker Studio**

# O que iremos criar nesse curso?

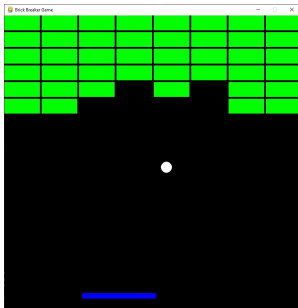
Nesse curso, iremos criar dois jogos simples:

- ▶ **Snake (Jogo da cobrinha):** O clássico jogo da cobrinha em que você deve coletar maçãs que aparecem aleatoriamente no mapa



# O que iremos criar nesse curso?

- ▶ **Brick Breaker:** O jogo em que você arremessa uma bola em tijolos para quebrá-los. Toda vez que um bloco é quebrado, a velocidade do jogo aumenta



# Uma breve visão do Pygame

Vamos mostrar alguns comandos básicos de Pygame

- ▶ **import pygame as pg** : importa a biblioteca do pygame. O final "as pg" significa que estamos dando um sobrenome "pg" para o pygame para facilitar a escrita
- ▶ **pg.init()** e **pg.quit()**: inicializa e fecha o pygame
- ▶ **tela = pg.display.set\_mode((largura, altura))**: cria a tela do pygame, dado um tamanho de largura e altura
- ▶ **clock = pg.time.Clock()** e **clock.tick(fps)**: define uma variável para ser o clock (tempo de atualização de cada tela do jogo) e quantas vezes por segundo será atualizado a tela

# Uma breve visão do Pygame

- ▶ **`pg.key.get_pressed()`** : Pega a tecla digitada pelo usuário no momento (usada para definir a movimentação do personagem)
- ▶ **`pg.draw.rect(tela, COR, (x, y, largura, altura))`**: cria um objeto retangular, que ficará na tela, com uma determinada cor, terá um posição (coordenada) específica e um tamanho (largura e altura)
- ▶ **`pygame.draw.circle(tela, COR, (x, y), raio)`**: cria um círculo, que ficará na tela, com uma determinada cor, terá uma posição específica e um raio
- ▶ **`pg.display.update()`**: atualiza a tela baseado no clock

# Uma breve visão do Pygame

```
1 import pygame as pg # importa a biblioteca do pygame
2
3 pg.init() # Inicializa o Pygame
4
5 largura, altura = 800, 600 # Tamanho da janela
6 tela = pg.display.set_mode((largura, altura)) #define o tamanho
    da tela
7 pg.display.set_caption("Teste") #Coloca uma legenda para a tela
8
9 x, y = 100,100 #define a posi o inicial do jogador
10 velocidade = 5 #vai definir a velocidade do personagem mexendo
    na tela
11 clock = pg.time.Clock() #cria a vari vel do clock do jogo
```

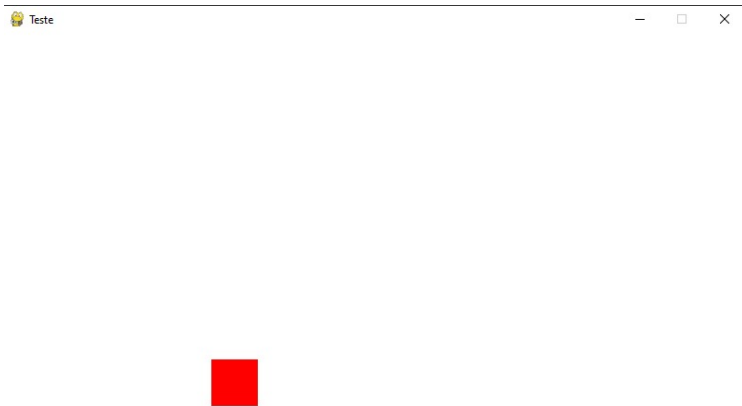
# Uma breve visão do Pygame

```
1 while True:
2     clock.tick(60) # 60 FPS
3     tela.fill('white') #preenche o fundo com a cor branca
4     for evento in pg.event.get():
5         if evento.type == pg.QUIT: #caso o jogo foi fechado
6             pg.quit()
7     teclas = pg.key.get_pressed() #pega tecla digitada
8     if teclas[pg.K_a]:
9         x -= velocidade #diminui a posicao do x
10    if teclas[pg.K_d]:
11        x += velocidade #aumenta a posicao do x
12    if teclas[pg.K_w]:
13        y -= velocidade #diminui a posicao do y
14    if teclas[pg.K_s]:
15        y += velocidade #aumenta a posicao do y
16    pg.draw.rect(tela,'red',(x, y, 50, 50))
17    pg.display.update() #atualiza a tela
```



# Uma breve visão do Pygame

Teremos algo assim:



# Criando o jogo da cobrinha (SETUP)

```
1 import pygame as pg
2 import random as rd
3 pg.init()
4 largura, altura = 720, 720
5 tela = pg.display.set_mode((largura, altura))
6 clock = pg.time.Clock()
```

# Criando o jogo da cobrinha (SETUP)

```
1 velocidade = 5 #define a velocidade(pixel por movimento) do
   jogo
2 tamanho_cobra = 15 #define quantos quadrados nossa cobra tera
   inicialmente
3 initial_control = pg.Vector2(velocidade,0)
4 tam = 30 # valor do tamanho dos objetos
5 cobra_pos = pg.Vector2(tela.get_width()/2, tela.get_height()/2)
6 maca_pos = pg.Vector2(rd.randint(tam, largura-tam), rd.randint(
   tam, altura-tam))
7 control_pos = pg.Vector2(velocidade,0) # ser o valor que ir
   controlar a posicao da maca
8 lista_cobra = [] #sera a lista que ficara a cobra
```

# Criando o evento de parada e os objetos principais

```
1 while True:
2     clock.tick(60)
3     tela.fill('white')
4
5     for event in pg.event.get():
6         if event.type == pg.QUIT:
7             pg.quit()
8
9     cobra = pg.draw.rect(tela, 'green', (cobra_pos.x, cobra_pos.
10         y, tam, tam))
11     maca = pg.draw.rect(tela, 'red', (maca_pos.x, maca_pos.y,
12         tam, tam))
```

# Movimentação da cobra

```
1 key = pg.key.get_pressed()
2 if key[pg.K_w]:
3     control_pos.y = -velocidade #define o valor e a direcao
4     control_pos.x = 0
5 if key[pg.K_s]:
6     control_pos.y = velocidade
7     control_pos.x = 0
8 if key[pg.K_a]:
9     control_pos.y = 0
10    control_pos.x = -velocidade
11 if key[pg.K_d]:
12     control_pos.y = 0
13     control_pos.x = velocidade
14 # Muda os valores de posicao da cobra
15 cobra_pos.x += control_pos.x
16 cobra_pos.y += control_pos.y
```

# Criação da cobra e reconhecendo colisão

```
1      #Atualiza o do corpo da cobra
2      lista_cabeca = []
3      lista_cabeca.append(cobra_pos.x)
4      lista_cabeca.append(cobra_pos.y)
5      lista_cobra.append(lista_cabeca)
6      #Limita o tamanho da cobra
7      if len(lista_cobra) > tamanho_cobra:
8          del lista_cobra[0]
9      for pos in lista_cobra: #desenha a cobra
10         pg.draw.rect(tela, 'green', (pos[0], pos[1], tam, tam))
11      #Colisao
12      if maca.colliderect(cobra):
13         maca_pos.x=rd.randint(tam, largura-tam)
14         maca_pos.y=rd.randint(tam, altura-tam)
15      pg.display.update()
```