

# PROGRAMAÇÃO PARA JOGOS I

George Gomes

Vamos começar a desenvolver nosso  
primeiro jogo!

E mais importante, conhecendo os  
fundamentos de desenvolvimento de jogos

Neste jogo, o foco será no Game Loop!

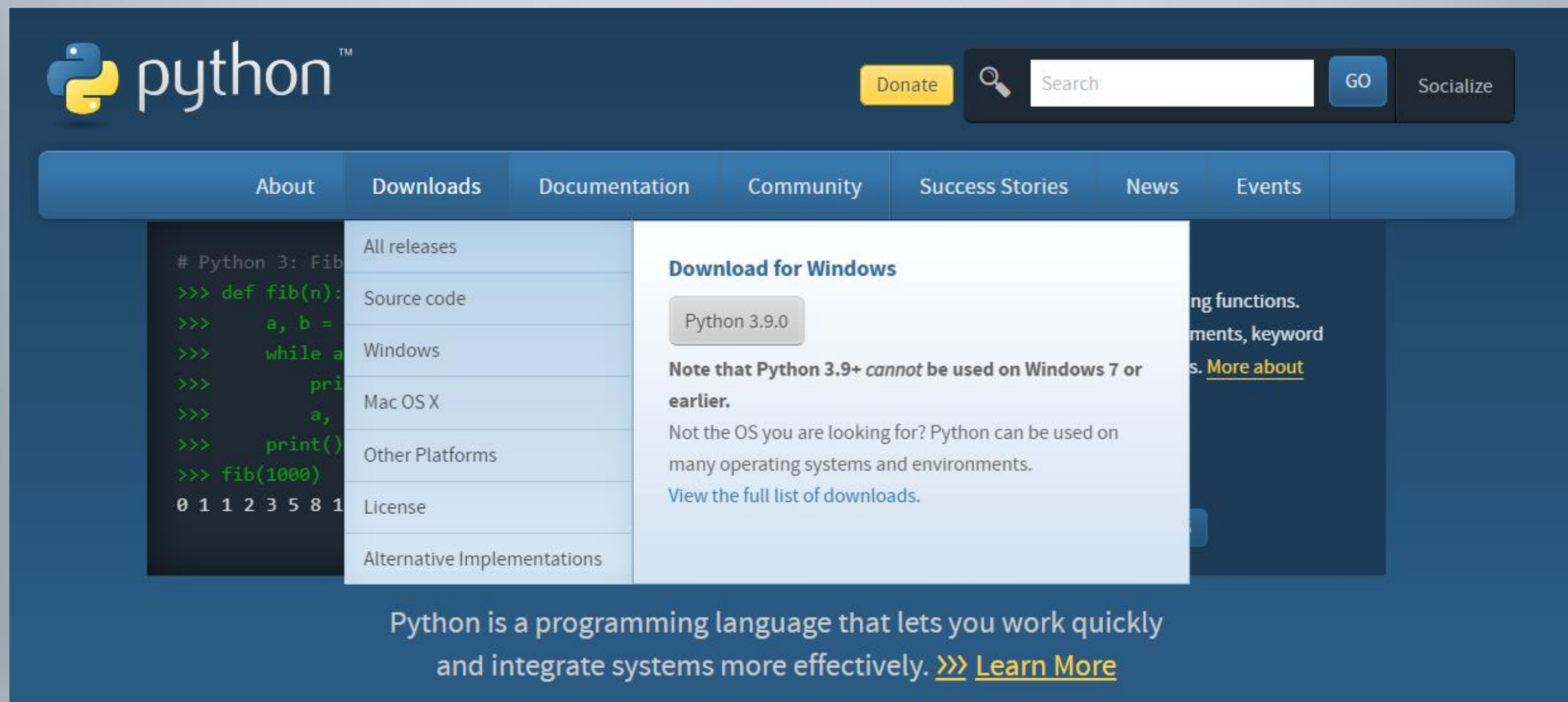
Qual plataforma usar?

# Pygames

Por que Pygames?

Simple e didática! Perfeita para construir um  
Game Loop!





# COMO COMEÇAR?

Instalando o Python em <https://www.python.org/>

Abra o prompt de comando e digite

```
python3 -m pip install -U pygame --user
```


Para ver se funcionou

```
python3 -m pygame.examples.aliens
```

Instalando o pygames em  
<https://www.pygame.org/wiki/GettingStarted>

Passo-a-passo para instalar python e pygame:

[www.youtube.com/watch?v=EKjALzLLgVs](http://www.youtube.com/watch?v=EKjALzLLgVs)



Version: 2020.3  
Build: 203.5981.165  
1 December 2020

[System requirements](#)  
[Installation Instructions](#)

## Download PyCharm

[Windows](#) [Mac](#) [Linux](#)

### Professional

For both Scientific and Web Python development. With HTML, JS, and SQL support.

[Download](#)

Free trial

### Community

For pure Python development

[Download](#)

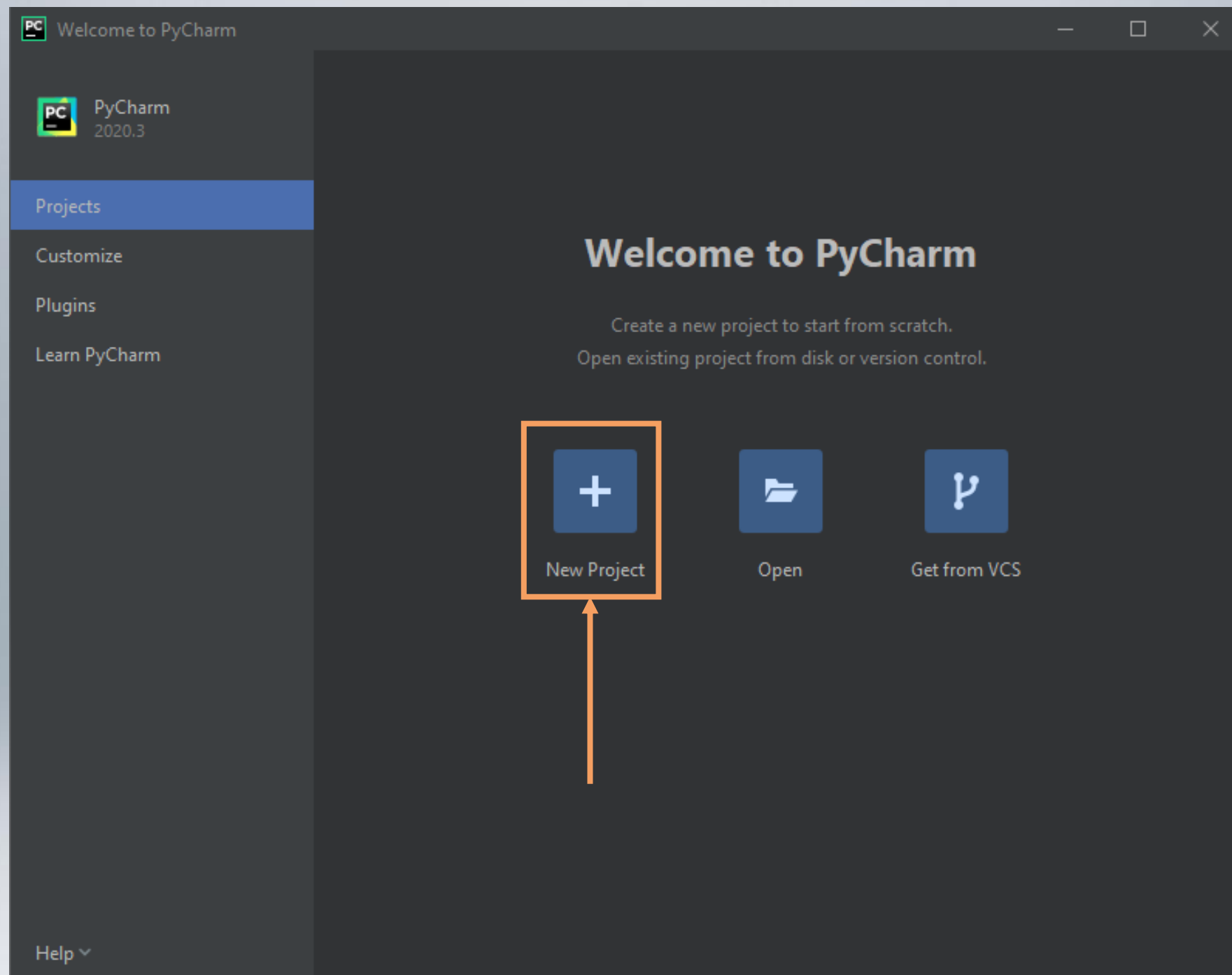
Free, open-source

Instalando o PyCharm em  
[www.jetbrains.com/pycharm/download](http://www.jetbrains.com/pycharm/download)

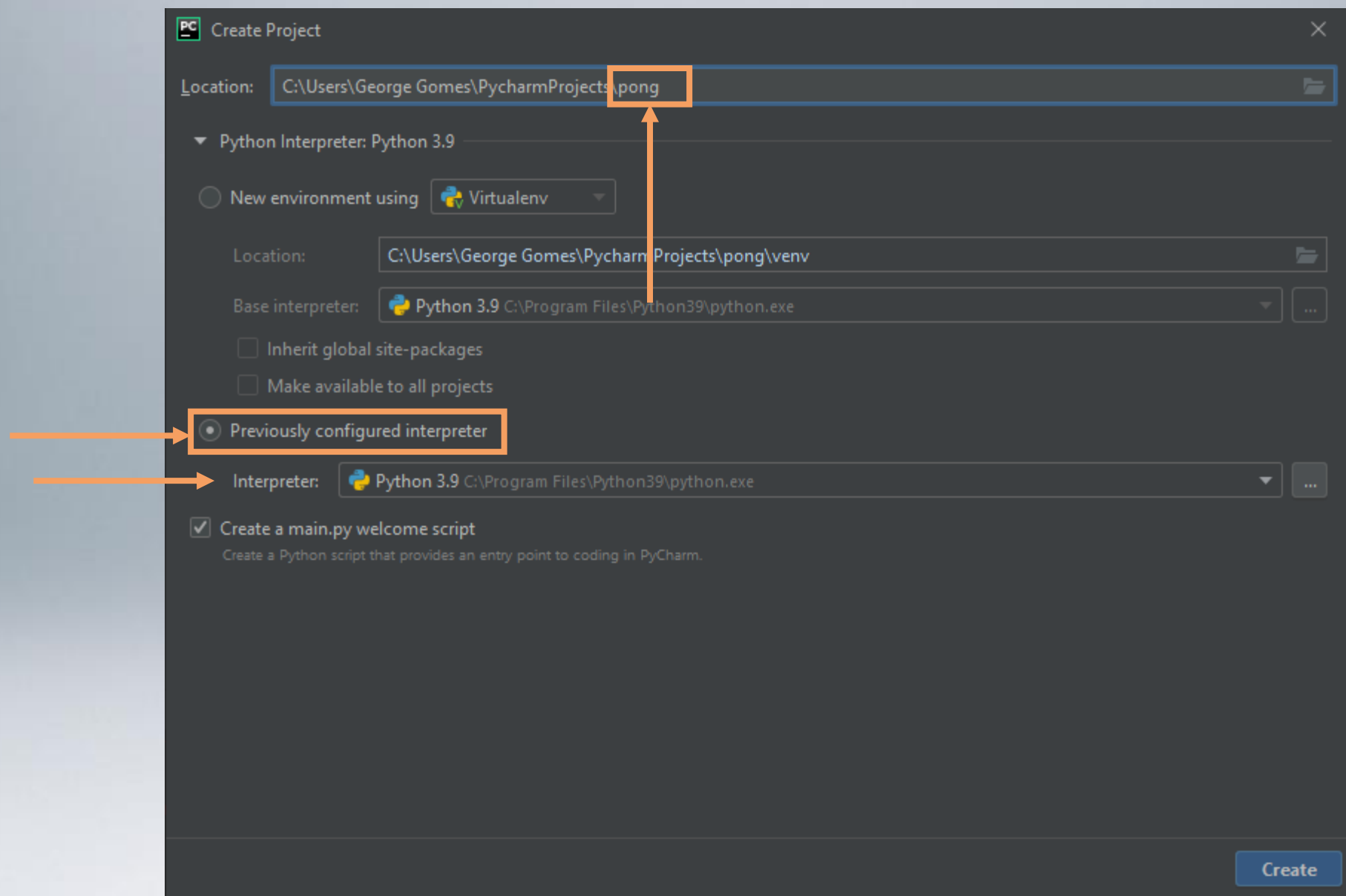
Qual jogo vamos criar?

O Pong!

Vamos começar!

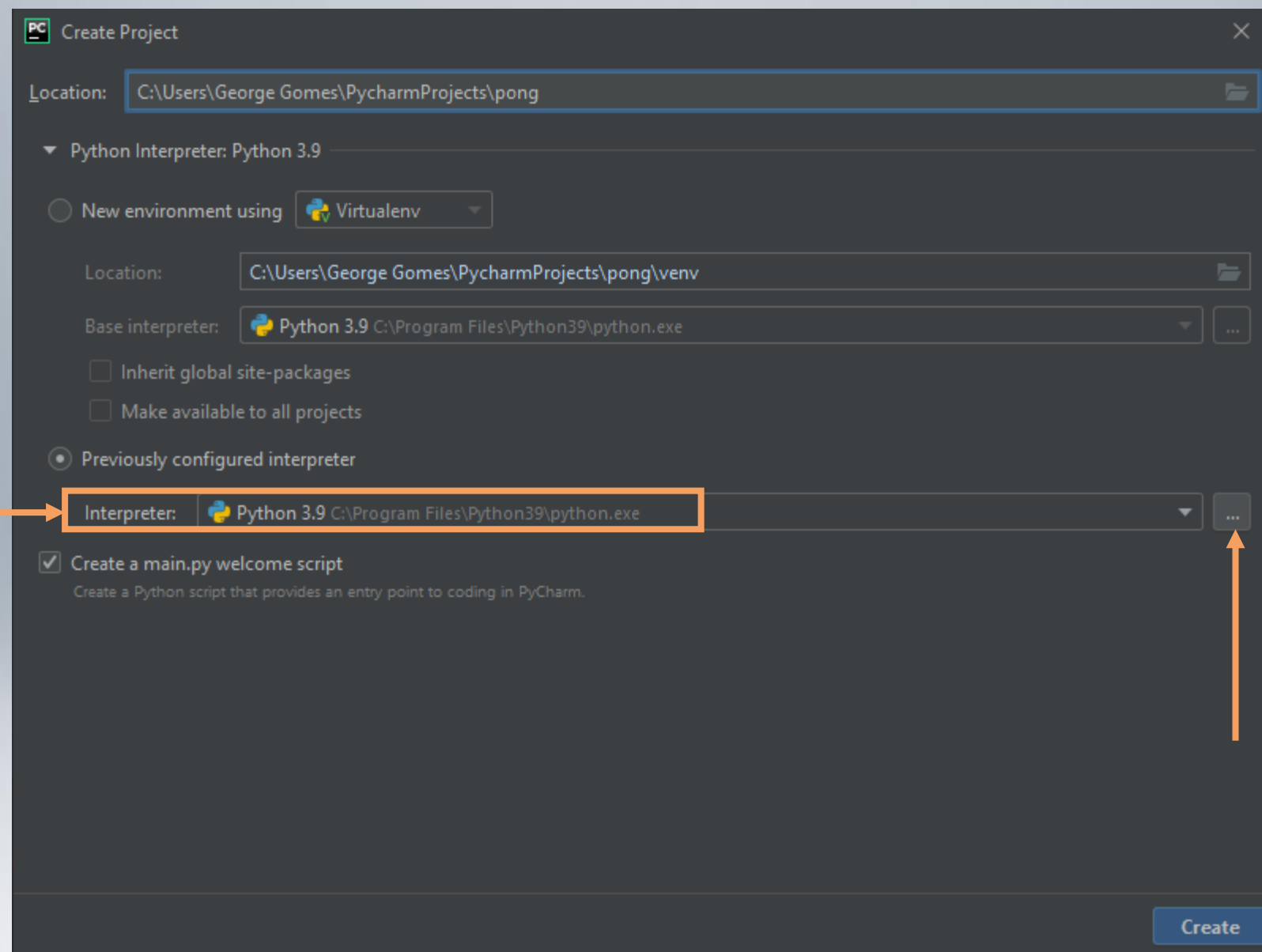


Novo projeto

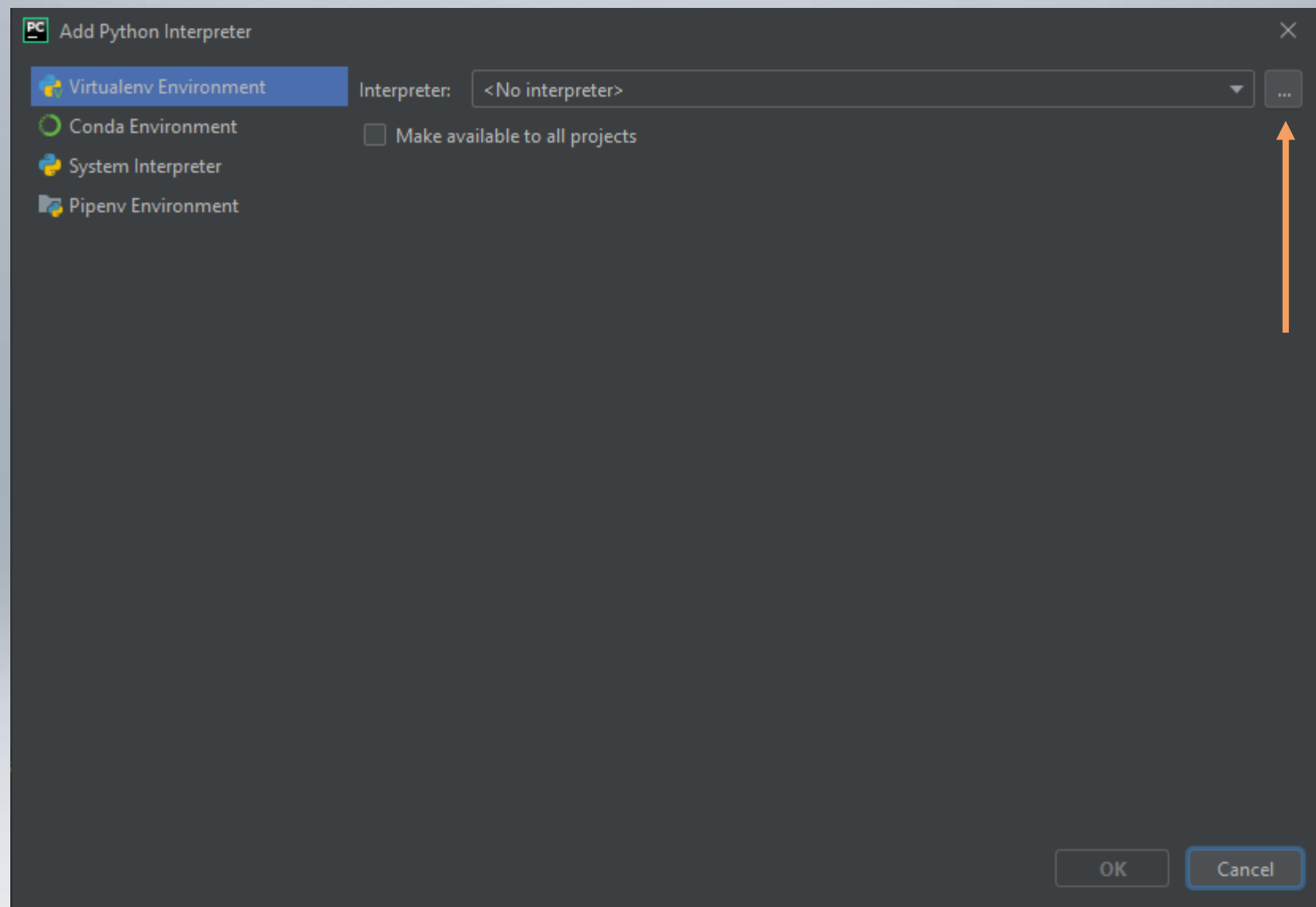


Configurando seu projeto

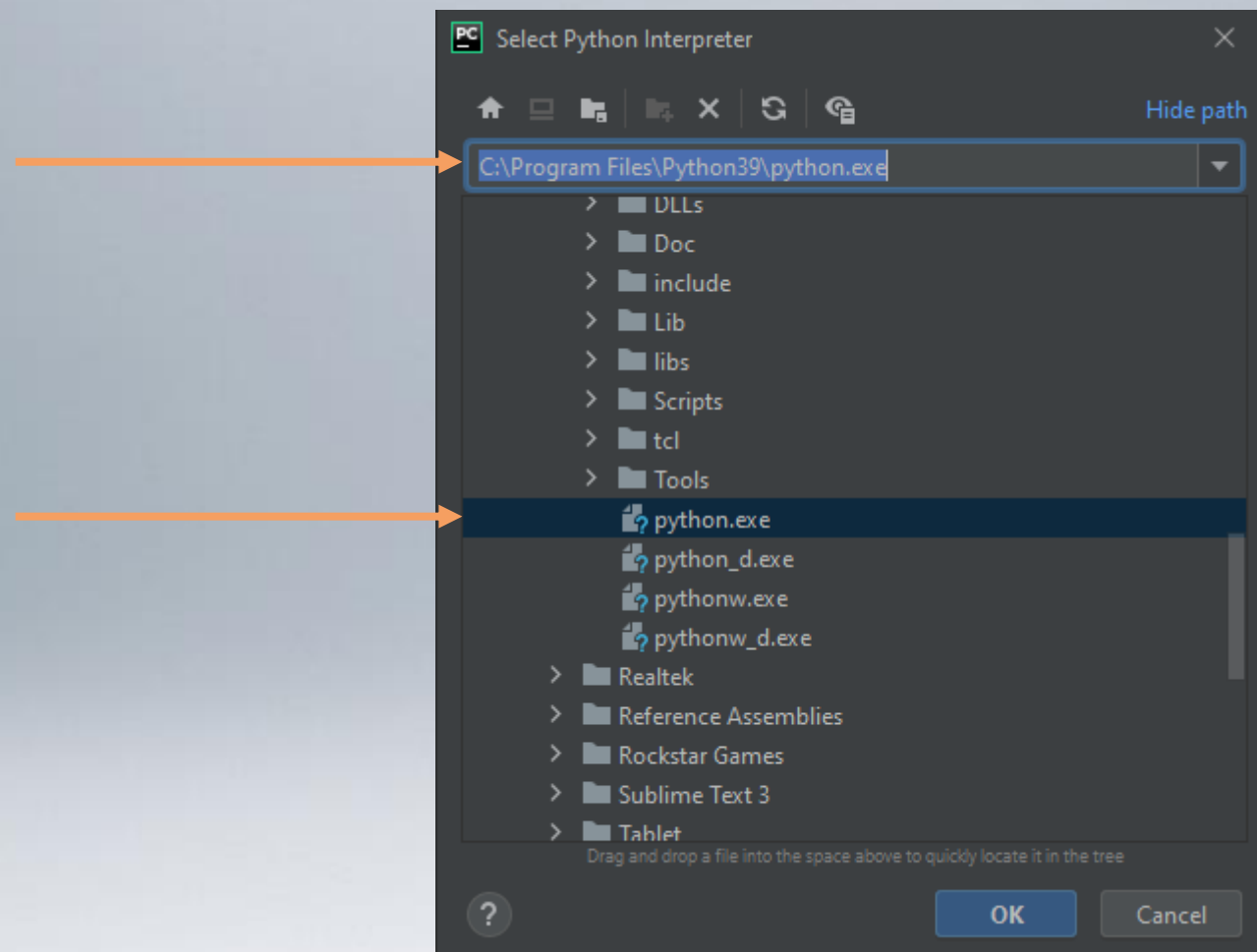




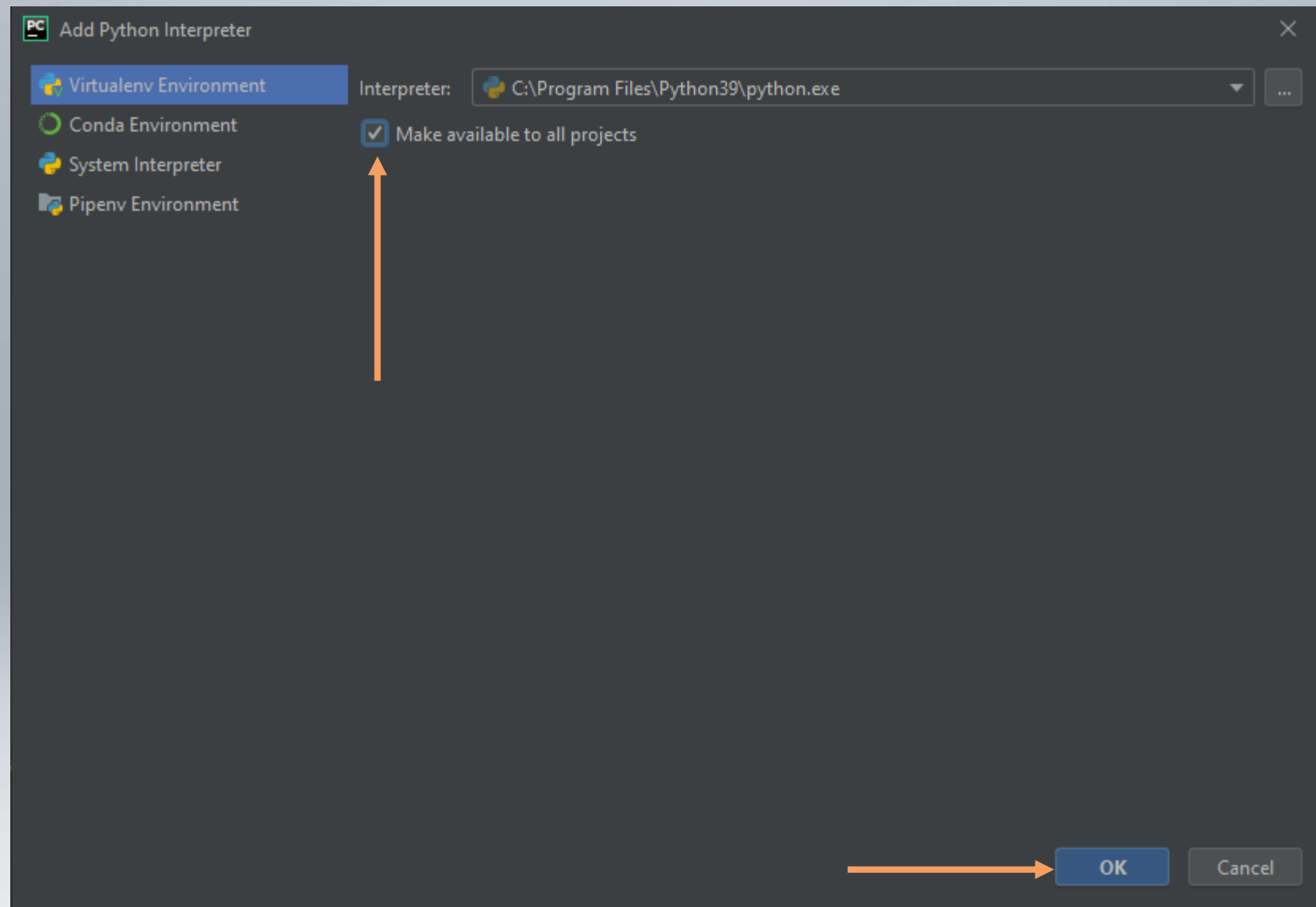
Se no interpretador apareceu '**No Interpreter**'. Vamos selecionar o Python.exe



Clique no botão ‘...’



Encontre onde o python.exe está



Marque essa opção para projetos futuros, depois OK

Project

- pong C:\Users\George Gomes\PycharmProjects\pong
  - main.py
- External Libraries
- Scratches and Consoles

Structure

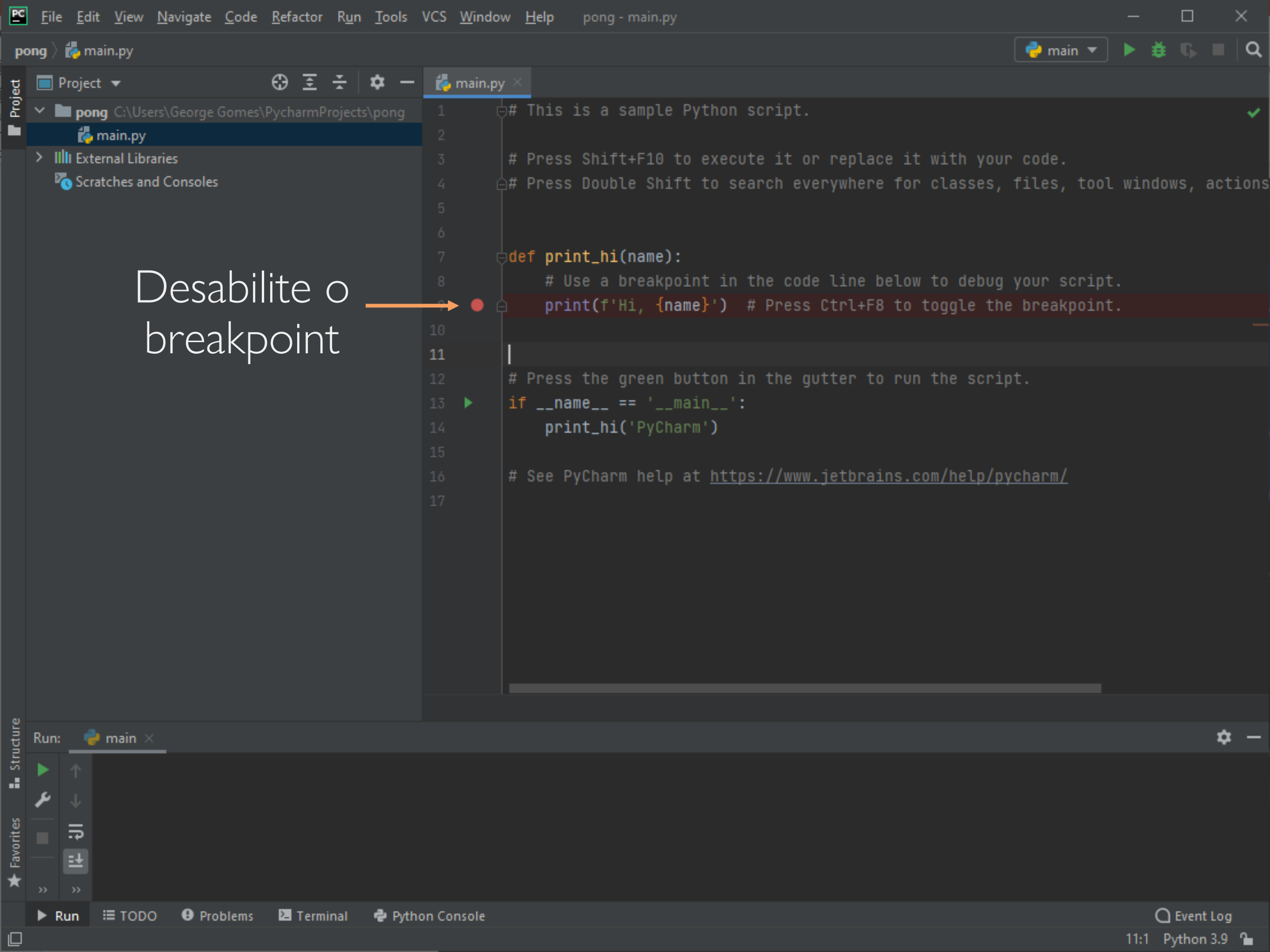
Favorites

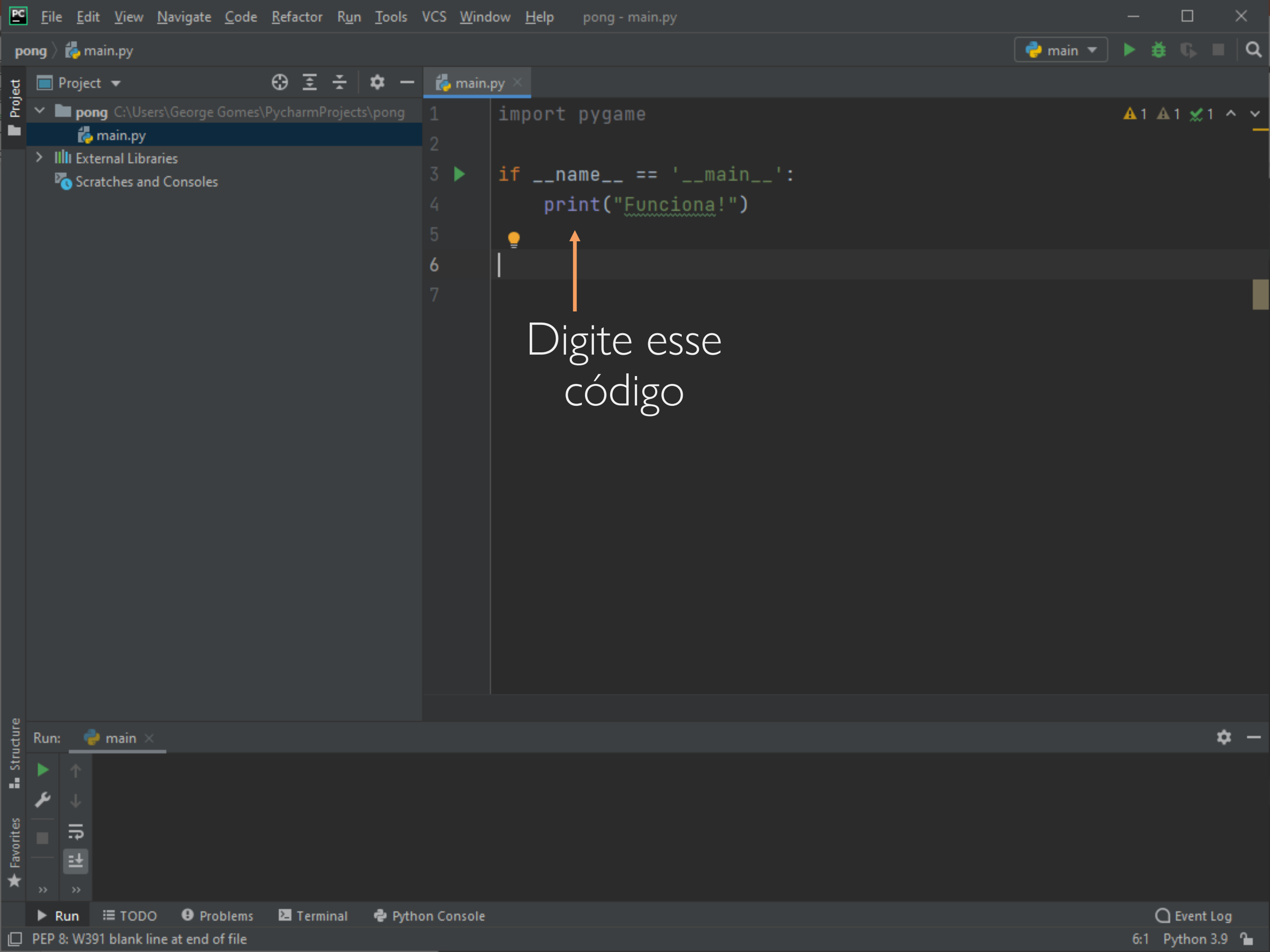
```

1  # This is a sample Python script.
2
3  # Press Shift+F10 to execute it or replace it with your code.
4  # Press Double Shift to search everywhere for classes, files, tool windows, actions, and settings.
5
6
7  def print_hi(name):
8      # Use a breakpoint in the code line below to debug your script.
9      print(f'Hi, {name}') # Press Ctrl+F8 to toggle the breakpoint.
10
11
12 # Press the green button in the gutter to run the script.
13 if __name__ == '__main__':
14     print_hi('PyCharm')
15
16 # See PyCharm help at https://www.jetbrains.com/help/pycharm/
17

```

Vamos limpar e criar um hello world!

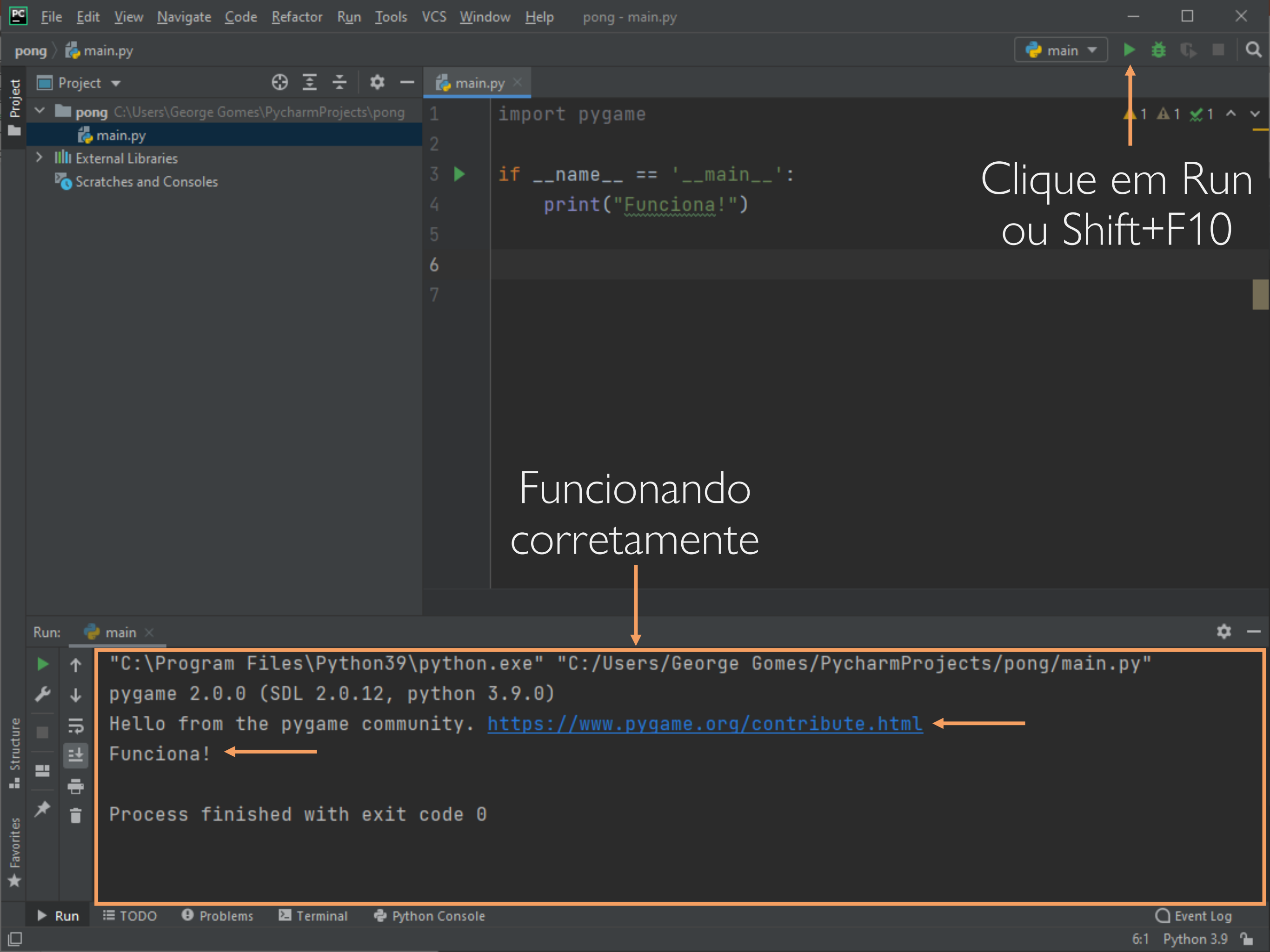






Ele testa se o nome do arquivo é main e escreve **'Funciona!'**

Além disso, testa se a biblioteca pygame está sendo importada corretamente



Clique em Run  
ou Shift+F10

Funcionando  
corretamente

Como um jogo funciona?

# Setup

Inicialização  
Variáveis

# Game Loop

Renderização (desenho)  
Atualização (física)

# Setup

Criar o retângulo

Definir posição e velocidade

# Game Loop

Desenhar o retângulo  
Atualizar sua posição

# Configurando nosso jogo Pong

# Digite o seguinte código

```
1  import pygame
2  import sys
3
4  # Inicialização
5  pygame.init()
6  clock = pygame.time.Clock()
7
8  # Configurando a janela
9  screen = pygame.display.set_mode((1280, 960))
10 pygame.display.set_caption('Pong')
11
12 while True:
13     # Processando as entradas (eventos)
14     for event in pygame.event.get():
15         if event.type == pygame.QUIT:
16             pygame.quit()
17             sys.exit()
18
19     # Atualizando a janela 60fps
20     pygame.display.flip()
21     clock.tick(60)
```



Importando as bibliotecas do pygame e de sistema para ter acesso aos eventos do sistema

```
1 import pygame  
2 import sys
```

Inicializando o pygame e criando um relógio para controle de quadros (*frames*) por segundo

```
4      # Inicialização
5      pygame.init()
6      clock = pygame.time.Clock()
```

Criando nossa janela com tamanho 1280x960 e atribuindo um nome

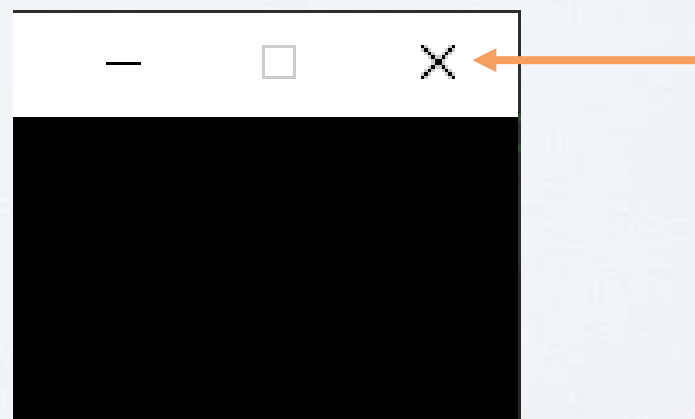
```
8      # Configurando a janela
9      screen = pygame.display.set_mode((1280, 960))
10     pygame.display.set_caption('Pong')
```

# Criando um *game loop* infinito (simples)

```
12 while True:
```

Processando as entradas (eventos) dentro do *loop*.  
No caso, está testando se clicou no botão fechar.  
Finaliza o pygame e fecha a janela.

```
12 while True:
13     # Processando as entradas (eventos)
14     for event in pygame.event.get():
15         if event.type == pygame.QUIT:
16             pygame.quit()
17             sys.exit()
```



Ainda dentro do *loop*. Atualiza a janela (*display.flip*) e determina a taxa de atualização para 60 quadros por segundo (*clock.tick*)

```
12  while True:
13      # Processando as entradas (eventos)
14      for event in pygame.event.get():
15          if event.type == pygame.QUIT:
16              pygame.quit()
17              sys.exit()
18
19      # Atualizando a janela 60fps
20      pygame.display.flip()
21      clock.tick(60)
```

# Desenhando a bola do Pong

# Display

1280

---

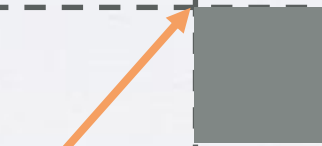


Display

960

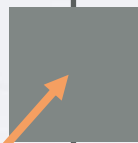
# Display

```
ball = pygame.Rect(1280/2, 960/2, 30, 30)
```



# Display

```
ball = pygame.Rect(1280/2-15, 960/2-15, 30, 30)
```



Desenhando os paddles

Player e Opponent

# Display

```
# Objetos  
ball = pygame.Rect(1280/2-15, 960/2-15, 30, 30)  
player = pygame.Rect(1280-20, 960/2-70, 10, 140)  
opponent = pygame.Rect(10, 960/2-70, 10, 140)
```



Se você tentou executar, nada vai aparecer  
ainda!

Precisamos chamar a função *draw* passando ‘o *quê*’  
será desenhado, *onde* será desenhado e qual *cor*

# Desenho dos objetos

```
17 while True:
18     # Processando as entradas (eventos)
19     for event in pygame.event.get():
20         if event.type == pygame.QUIT:
21             pygame.quit()
22             sys.exit()
23     # Desenho
24     pygame.draw.rect(screen, (200, 200, 200), ball)
25     pygame.draw.rect(screen, (200, 200, 200), player)
26     pygame.draw.rect(screen, (200, 200, 200), opponent)
27
28     # Atualizando a janela 60fps
29     pygame.display.flip()
30     clock.tick(60)
```

Cor cinza claro



Próximo passo?

Animação (próxima aula)

# Exercícios

Para reforçar a aula de hoje, indico dois tutoriais básicos para consolidar os conceitos do Pygame:

<https://humberto.io/pt-br/blog/desbravando-o-pygame-1-conhecendo-a-biblioteca/>

<https://humberto.io/pt-br/blog/desbravando-o-pygame-2-desenhando-na-tela/>

Já conhecia o Pygame? Conhece algum tutorial interessante? Se não, pesquise um tutorial interessante (Youtube e Google) e compartilhe no [#2-discussão-game-loop](#)

Você sabe o que é o termo *surface* no pygame? O que isso tem a ver com imagens e desenho?

Comenta lá no [#2-game-loop](#)

# PROGRAMAÇÃO PARA JOGOS I

George Gomes