

Arcade

Orientação a Objetos



Arcade

Orientação a Objetos

O Arcade é uma biblioteca orientada a objetos.

Podemos escrever código em Arcade de forma procedural, como fizemos até agora.

No entanto, o verdadeiro poder do Arcade é mostrado quando você cria programas completamente orientados a objetos.

Arcade

Introdução

Quando utilizamos `arcade.open_window()` em nossos exemplos, é criado um objeto `arcade.Window` nos bastidores para gerenciar a janela.

Depois disso, temos que criar nossa própria classe com base em `arcade.Window` para escrever um jogo completo em Python.

Arcade

Introdução

```
# Programa básico arcade usando objetos  
# Exibe uma janela azul com um círculo amarelo no meio  
  
# Importando o arcade  
import arcade  
  
# Constantes  
LARGURA = 600  
ALTURA = 800  
TITULO = "Bem-vindo ao Arcade"  
RAIO = 150
```



Arcade

Introdução

```
# Aqui está sendo criada uma classe chamada Jogo baseada na classe arcade.Window  
# Isso permite que você sobrescreva métodos da classe pai se necessário  
class Jogo(arcade.Window):  
    """Janela principal"""  
  
    def __init__(self):  
        """Inicializa a janela"""  
  
        # Utilizamos super() para chamar o método  
        # .__init__() da classe pai para criar a janela  
        super().__init__(LARGURA, ALTURA, TITULO)  
  
        # Em seguida, definimos a cor de fundo da janela  
        arcade.set_background_color(arcade.color.BLUE)
```

Arcade

Introdução

O método on_draw() é um dos vários métodos da classe arcade.Window que você pode # sobrescrever para customizar o comportamento do seu programa arcade

```
def on_draw(self):
```

```
    """Este método é chamado sempre que você  
    precisar desenhar na janela  
    """
```

```
    # Primeiro limpamos a tela para começar a desenhar  
    arcade.start_render()
```

```
    # Neste exemplo vamos desenhar um círculo amarelo  
    arcade.draw_circle_filled(  
        LARGURA / 2, ALTURA / 2, RAIO, arcade.color.YELLOW  
    )
```

```
    # Não é necessário chamar finish_render() porque o arcade chama implicitamente este método  
    # quando o on_draw() finaliza
```



Arcade

Introdução

```
# Ponto de entrada, código principal  
if __name__ == "__main__":  
    app = Jogo() # Criando um objeto da classe Jogo  
    arcade.run() # Executando o arcade
```

FIM

