



Trabalhando com cores

## Trabalhando com cores

O PyGame possui um módulo denominado ***Color*** para representação de cores. Veja alguns métodos deste módulo.

Método	Descrição
<code>pygame.Color.r</code>	Obtém ou define o valor vermelho da cor.
<code>pygame.Color.g</code>	Obtém ou define o valor verde da cor.
<code>pygame.Color.b</code>	Obtém ou define o valor azul da cor.
<code>pygame.Color.a</code>	Obtém ou define o valor alfa da cor.
<code>pygame.Color.cmy</code>	Obtém ou define a representação CMY da cor.
<code>pygame.Color.hsva</code>	Obtém ou define a representação de HSVA da cor.
<code>pygame.Color.hsla</code>	Obtém ou define a representação de HSLA da cor.
<code>pygame.Color.normalize</code>	Retorna os valores RGBA normalizados da cor.
<code>pygame.Color.set_length</code>	Define o número de elementos na cor para 1,2,3 ou 4.

## Trabalhando com cores

Para instanciar um objeto da classe **Color** devemos informar o nome da cor que é uma **String** ou os números que representam o padrão RGB.

Na aula 01, no exemplo 05, desenhemos um retângulo na tela informando uma tupla com os valores RGB para a cor azul, desta forma (0, 0, 255).

Agora nós vamos usar um objeto **Color** para informar a cor do retângulo.

```
# Definindo o objeto Color com a cor azul
cor = Color(0, 0, 255)
# Desenhando o retângulo usando o objeto Color
pygame.draw.rect(screen, cor, [150, 10, 50, 20])
```

## Trabalhando com cores

Veja agora um exemplo informando o nome da cor em vez do padrão **RGB**.

```
# Definindo o objeto Color com a cor azul  
cor = Color("blue")  
# Desenhando o retângulo usando o objeto Color  
pygame.draw.rect(screen, cor, [150, 10, 50, 20])
```

Para visualizar as cores disponíveis, utilize esse código abaixo

```
import pygame.color as color  
print(color.THECOLORS.keys())
```

## Trabalhando com cores

Em nosso próximo exemplo nós vamos trabalhar com ***Color.hsva***.

**HSV** é a abreviatura para o sistema de cores formadas pelas componentes *hue* (matiz ou tonalidade), *saturation* (saturação ou “pureza”) e *value* (valor ou brilho). Os valores de matiz vão de 0 a 360, nos demais de 0 a 100. O último parâmetro do método ***HSVA*** é o ***Alpha***, que é transparência.

Mais detalhes sobre HSV em: <https://pt.wikipedia.org/wiki/HSV>

# PyGame

## Trabalhando com cores

Nós vamos criar um objeto **Color** da cor azul, e vamos manipular a matiz da cor utilizando **HSVA**

```
import pygame
from pygame.locals import *

pygame.init()
screen = pygame.display.set_mode((640,480))
cor = Color('blue')
cor.hsva = (100, 100, 100, 50)
nova_cor = Color(cor.r, cor.g, cor.b)
running = True
imagem = 'esquerda'

...
```

# PyGame

## Trabalhando com cores

```
...
pygame.draw.rect(screen, nova_cor, [0, 0, 50, 50])
nova_cor.hsva = (50, 100, 100, 50)
pygame.draw.rect(screen, nova_cor, [0, 50, 50, 50])
nova_cor.hsva = (0, 100, 100, 50)
pygame.draw.rect(screen, nova_cor, [0, 100, 50, 50])
nova_cor.hsva = (30, 30, 30, 50)
pygame.draw.rect(screen, nova_cor, [0, 150, 50, 50])
pygame.display.flip()

while running:
    for event in pygame.event.get():
        if event.type == KEYDOWN:
            if event.key == K_ESCAPE:
                running = False
        elif event.type == QUIT:
            running = False

pygame.quit()
```

FIM