

| | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> 26 txt3 = font.render("SPACE"+str (key[pygame.K_SPACE])+" ENTER"+str(key[py game.K_RETURN]), True, WHITE, RED) 27 28 screen.fill(BLACK) 29 screen.blit(txt1, [100, 100]) 30 screen.blit(txt2, [100, 200]) 31 screen.blit(txt3, [100, 300]) 32 pygame.display.update() 33 clock.tick(10) 34 35 if __name__ == '__main__': 36 main() </pre> | <p>スペースキーとEnterキーのリストの値を描いたSurface</p> <p>指定した色でスクリーン全体をクリアする 文字列を描いたSurfaceをスクリーンに転送 // // 画面を更新する フレームレートを指定</p> <p>このプログラムが直接実行された時に main()関数を呼び出す</p> |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

このプログラムを実行すると、方向キー、スペースキー、**Enter**キーを押した時に1と表示されます。キーの組み合わせにもよりますが、2つ以上のキーの同時入力を判定できます。

図10-5-1 pygame_key.pyの実行結果



最新のPygameでは以下のような画面になります。
PythonではTrueと1、Falseと0が同等に扱われます。

図10-5-2 pygame_key.pyの実行結果



Pygameでは23行目にある「key = pygame.key.get_pressed()」という1行の記述で全てのキーの状態を取得できます。キーが押されている時には、key[pygame.キーボード定数]の値が1もしくはTrueになります。