

# Graafika ja Python

## Moodul

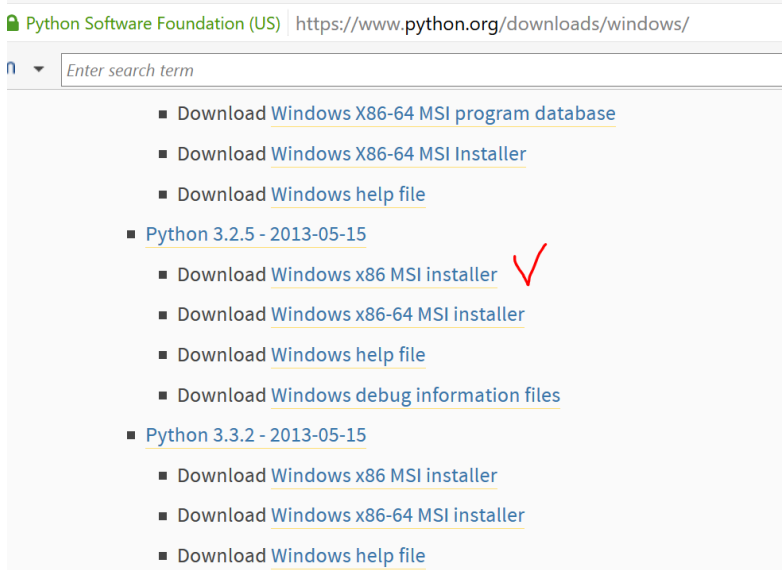


Pygame — moodulite kogum, mis annab võimalus luua mängud ja multimeedia rakendused Pythoni keeles.

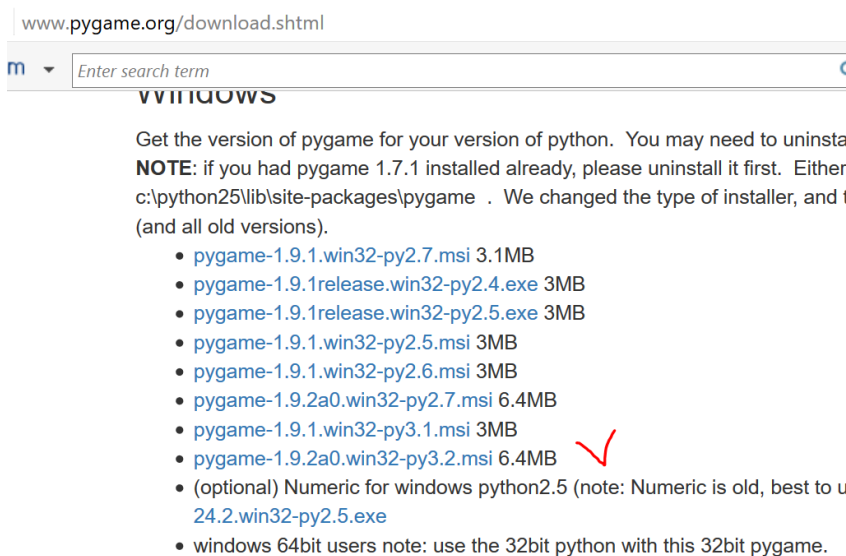
Pygame moodul loob graafikat, mis töötab erinevates operatsiooni süsteemides.

## Kuidas paigutada?

Pygame mooduli lihtsama installimiseks on vaja Python 3.2 kasutada. Algusel **laadime alla ja installeerime** sobiva Python versiooni

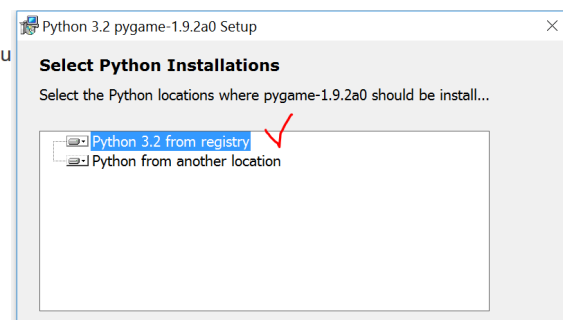


Edasi laadime Pygame [www.pygame.org](http://www.pygame.org) saitist alla



Hakkame instaleerima. Teisel sammul valime Python 3.2 (ta oli varem paigutatud)

Как проверить, что все работает. Käivitame **Python (command line)** ja kirjutame **import pygame** ning klõpsame Enter klahvil.

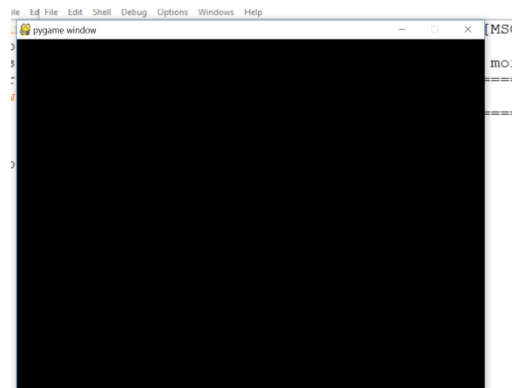


(vaata pildil)

```
C:\Python32\python.exe
Python 3.2.5 (default, May 15 2013, 23:06:03) [MSC v.1500 32
Type "help", "copyright", "credits" or "license" for more inf
>>> import pygame
>>>
```

## Graafika joonistamise aken

```
example1.py
1 import pygame
2 pygame.init()
3 screen = pygame.display.set_mode([640,480])
4 running=True
5 while running:
6     for event in pygame.event.get():
7         if event.type==pygame.QUIT:
8             running=False
9     pygame.quit()
```



pärast mooduli importimist (1 rida) on vaja seda initsialiseerida (2 rida)

3 rida – akna loomine, kus me formeerime kadri, **pygame.display.set\_mode(resolution=(0,0),flags=0, depth=0)**, loob akna, kus **resolution** – arvude loend, kus hoiame akna suurus x ja y teljel, **flags** – akna liik (täis ekraan, suurused muudetakse ja muud), **depth** – värvi sügavus. Me loome musta akna suurusega 640x480, (teised parameetrid me ei kasuta).

Pygame mooduliga programmis peab olema sündmuste tsükkel (event loop), mis kogu aeg kontrollib, kas kasutaja teeb midagi või mitte, näiteks: hiire kursori liikumine, klaviatuuri klahvi vajutus või paneme akna kinni. Tsükkel peab kogu aeg töötama, kuna mängib kasutaja. 4 ... 8 read – sündmuste tsükkel, tsükkel lõpetab oma töö, kui paneme akna kinni

9 rida – väljame programmist.

## Рисование графических примитивов на поверхности

Draw-joonistamise moodul:

- pygame.draw.rect**(aken, värv, ristkülik, paksus=0) — ristkülik\*
- pygame.draw.polygon**(aken, värv, punktide loend, paksus =0) — hulknurk
- pygame.draw.circle**(aken, värv, keskpunkt, raadius, paksus =0)— ringjoon
- pygame.draw.ellipse**(aken, värv, ristkülik, paksus =0)—ellips
- pygame.draw.arc**(aken, värv, ristkülik, algpunkt, lõppunkt, paksus =0)—kaar\*\*
- pygame.draw.line**(aken, värv, algpunkt, lõppunkt, paksus =0)— joon
- pygame.draw.lines**(aken, värv, suletud, punktide loend, paksus =0)— murdjoon
- pygame.draw.aaline**(aken, värv, algpunkt, lõppunkt, blend=1)— мелкая сглаженная линия
- pygame.draw.aalines**(aken, värv, suletud, punktide loend, blend=1)— мелкие сглаженные линии

\* ristkülik [x,y, külje pikkus, külje pikkus] paksus =0 –värvitud objekt,

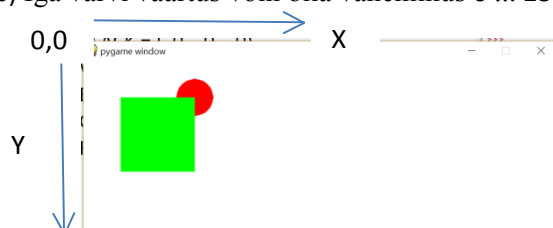
\*\* nurk radiaanites

Määrame värvi RGB formaadis (punane,roheleni,sinine) Iga värvi väärtus võin olla vahemikus 0 ... 255

BLACK = ( 0, 0, 0) WHITE = (255, 255, 255)

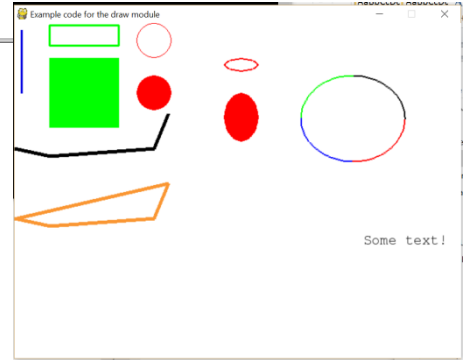
BLUE = ( 0, 0, 255) GREEN = ( 0, 255, 0)

RED = (255, 0, 0)



## Näide

```
1 import pygame
2 pygame.init()
3 screen=pygame.display.set_mode([640,480])
4
5 pygame.display.set_caption("Example code for the draw module")
6 screen.fill([255,255,255]) #valge
7
8 pygame.draw.circle(screen, [255, 0, 0], [200, 100], 25, 0)
9 pygame.draw.circle(screen, [255, 0, 0], [200, 25], 25, 1)
10
11 pygame.draw.rect(screen, [0, 255, 0], [50,50,100,100], 0)
12 pygame.draw.rect(screen, [0, 255, 0], [50,3,100,30], 3)
13
14 pygame.draw.line(screen, [0, 0, 225], [10,10], [10,100], 3)
15
16 pygame.draw.lines(screen, [0, 0, 0], False, [[0, 180], [50, 190], [200, 180], [220, 130]], 5)
17 pygame.draw.lines(screen, [250, 150, 50], True, [[0, 280], [50, 290], [200, 280], [220, 230]], 5)
18
19 pygame.draw.ellipse(screen, [255, 0, 0], [300, 50, 50, 20], 2)
20 pygame.draw.ellipse(screen, [255, 0, 0], [300, 100, 50, 70])
21
22 BLACK = ( 0, 0, 0)
23 WHITE = (255, 255, 255)
24 BLUE = ( 0, 0, 255)
25 GREEN = ( 0, 255, 0)
26 RED = (255, 0, 0)
27 pi=3.14
28
29 pygame.draw.arc(screen, BLACK,[410, 75, 150, 125], 0, pi/2, 2)
30 pygame.draw.arc(screen, GREEN,[410, 75, 150, 125], pi/2, pi, 2)
31 pygame.draw.arc(screen, BLUE, [410, 75, 150, 125], pi,3*pi/2, 2)
32 pygame.draw.arc(screen, RED, [410, 75, 150, 125], 3*pi/2, 2*pi, 2)
33
34 myfont = pygame.font.SysFont("monospace", 20)
35 # render text
36 label = myfont.render("Some text!", True, (0,0,0))
37 screen.blit(label, (500, 300))
38
39
40 pygame.display.flip()
41
42 running=True
43 while running:
44     for event in pygame.event.get():
45         if event.type==pygame.QUIT:
46             running=False
47 pygame.quit()
```



5 reas loome vorme pealkiri, 6 rida - **screen.fill(värv)** määrab akna värvi, 22-26 reas deklareerime muutujad värvi koodi hoidmiseks, kasutame **pygame.font.SysFont(font, tähe suurus)** (34 rida) teksti fondi ja suuruse määramiseks ja loome värvilise teksti **myfont.render(tekst,pehmendus,värv)** abil . **screen.blit(tekst,punkti koordinaadid)** lisab teksti valitud punktis

40 rida - Pygame loob ekraani pildi ja **pygame.display.flip()** uundab ekraani, puudub joonistamise vibratsioon

## Ülesanne

Joonista õnnekaart