Drawing objects: rectangle

```
#!/usr/bin/env python3

import pygame, sys
from pygame.locals import *

dark_blue = (25, 50, 150)
black = (0, 0, 0)

height = 300
width = 400
```

Drawing objects: rectangle

```
pygame.init ()
screen = pygame.display.set_mode([width, height])
pygame.draw.rect (screen, dark_blue, (50, 50, 60, 60), 0)
pygame.display.flip()

while True:
    for event in pygame.event.get ():
        if event.type == pygame.QUIT:
            sys.exit(0)
        if event.type == KEYDOWN:
            if event.key == K_ESCAPE:
                sys.exit (0)
```

Drawing objects: rectangle

- the line

- ```
  pygame.draw.rect (screen, dark_blue, (50, 50, 60, 60), 0)
  ```

- means:
  - call the `rect` method and draw on `screen` a rectangle of colour `dark_blue`
  - which has a top left corner of `50, 50`
  - and a bottom right corner of `60, 60`
  - this rectangle will be completely filled (boarder size of `0`)
  ```

Drawing objects: rectangle

- notice that in both the circle and rectangle examples nothing is displayed until you flip the buffer

- ```
  pygame.display.flip()
  ```

- using a common technique of double buffering
  - your application draws everything off screen and then it is flipped onto the screen, giving the illusion everything is drawn at once
  ```

Drawing objects: polygon

```
#!/usr/bin/env python3

import pygame, sys
from pygame.locals import *

dark_blue = (25, 50, 150)
black = (0, 0, 0)

height = 300
width = 400
```

Drawing objects: polygon

```
pygame.init ()
screen = pygame.display.set_mode([width, height])
pygame.draw.polygon (screen, dark_blue, [[50, 50], [100, 100], [50, 100]], 0)
pygame.display.flip()

while True:
    for event in pygame.event.get ():
        if event.type == pygame.QUIT:
            sys.exit(0)
        if event.type == KEYDOWN:
            if event.key == K_ESCAPE:
                sys.exit (0)
```

Drawing objects: polygon

- the line

- ```
  pygame.draw.polygon (screen, dark_blue, [[50, 50],
                                           [100, 100],
                                           [50, 100]], 0)
  ```

- means:
  - call the `polygon` method and draw on `screen` a polygon of colour `dark_blue`
  - which has a top left corner of `50, 50`
  - a corner at `100, 100`
  - and a final corner at `50, 100`
  - this polygon will be completely filled (boarder size of `0`)
  ```

Drawing objects: polygon

- the list of corners is often called a list of vertices as it also describes the lines (verticies of the polygon)