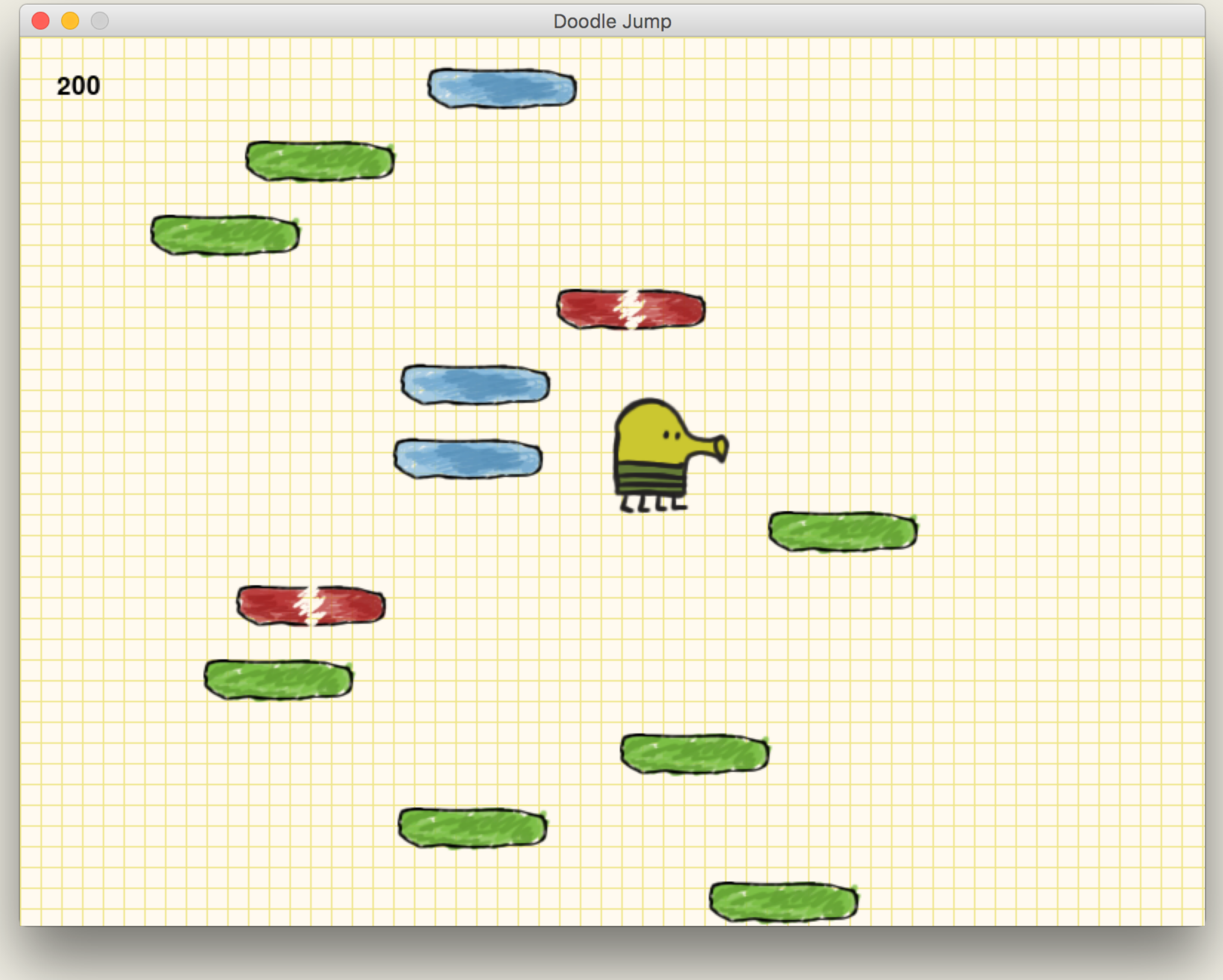
PYGAME(2)

2020資訊之芽Python班

Rilak@2020/05/31

Our Goal

- 教會你完成作業需要的所有語法
- 影像 Image
- 字型 Font
- 顯示型態 Surface
- 作業: Doodle Jump



IMAGE

影像

example8.py

載入影像

- image = pygame.image.load(filename)
  - filename：影像檔案

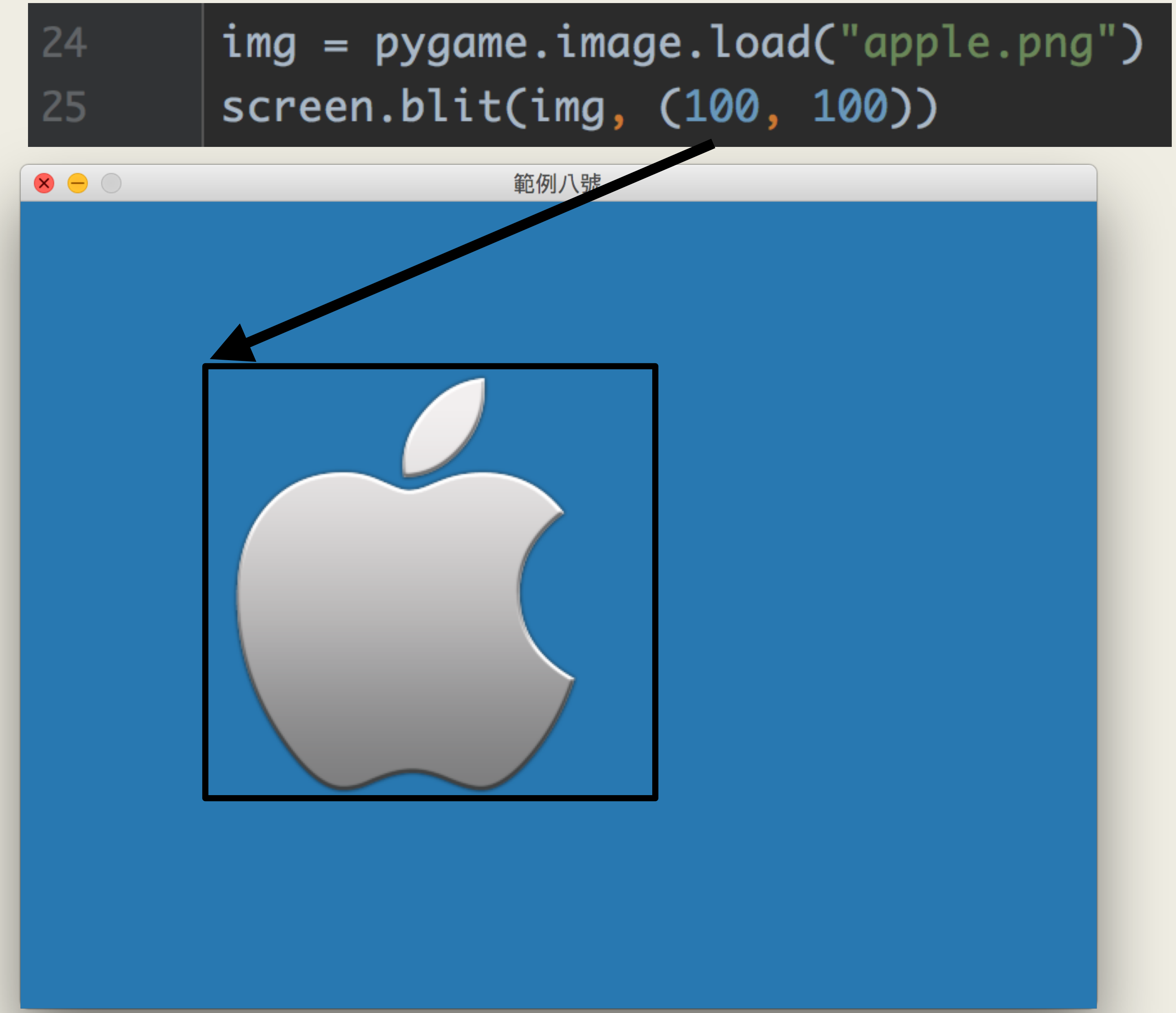
- type(image)

  => <class 'pygame.Surface'>

繪製

- dstn.blit(src, pos, area, flags)
  - destination：被貼的surface
  - source：貼過去的surface
  - position：貼在dstn上的位置(左上角)
  - area：src要貼過去的區域
  - flags：特殊設定

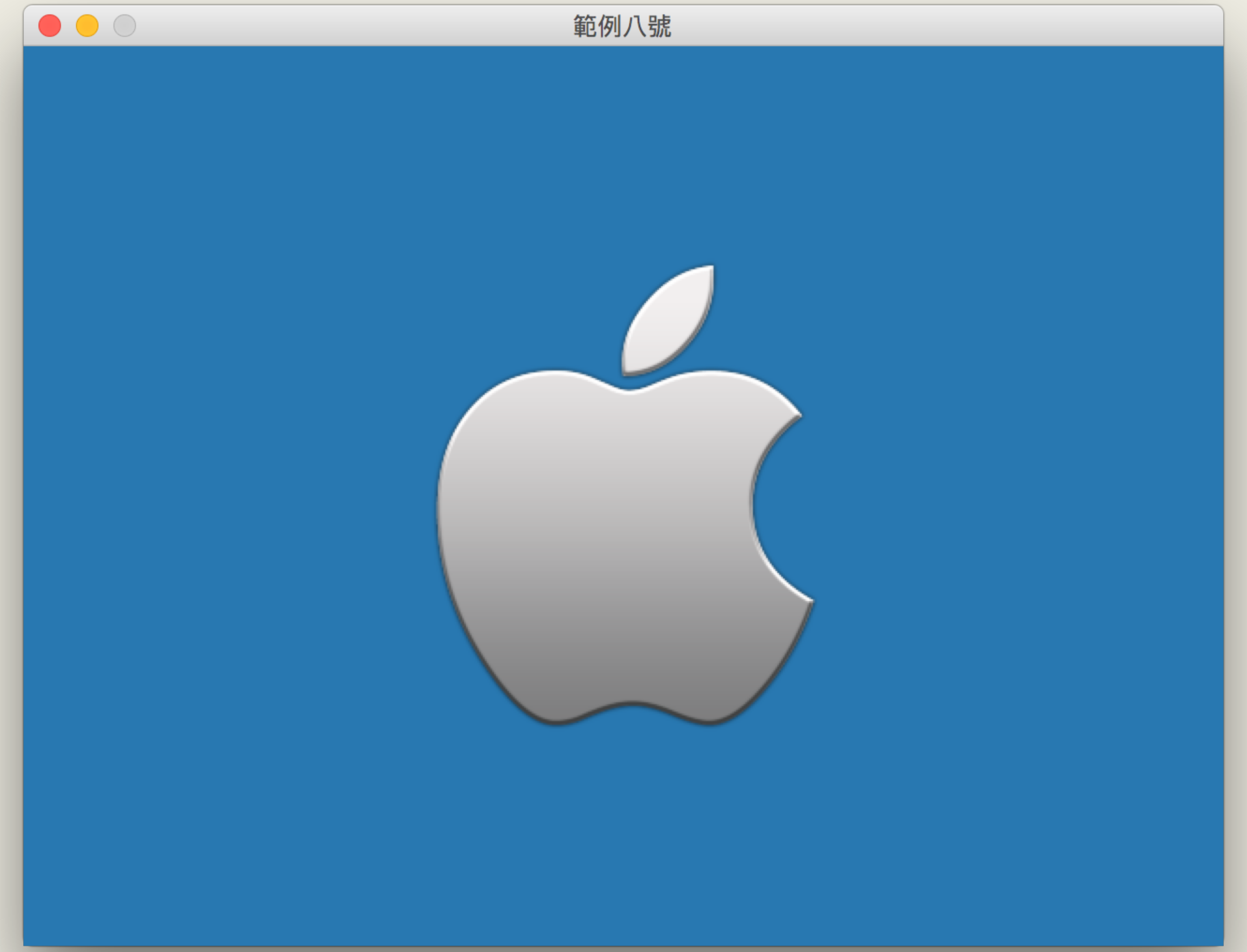
example8.py



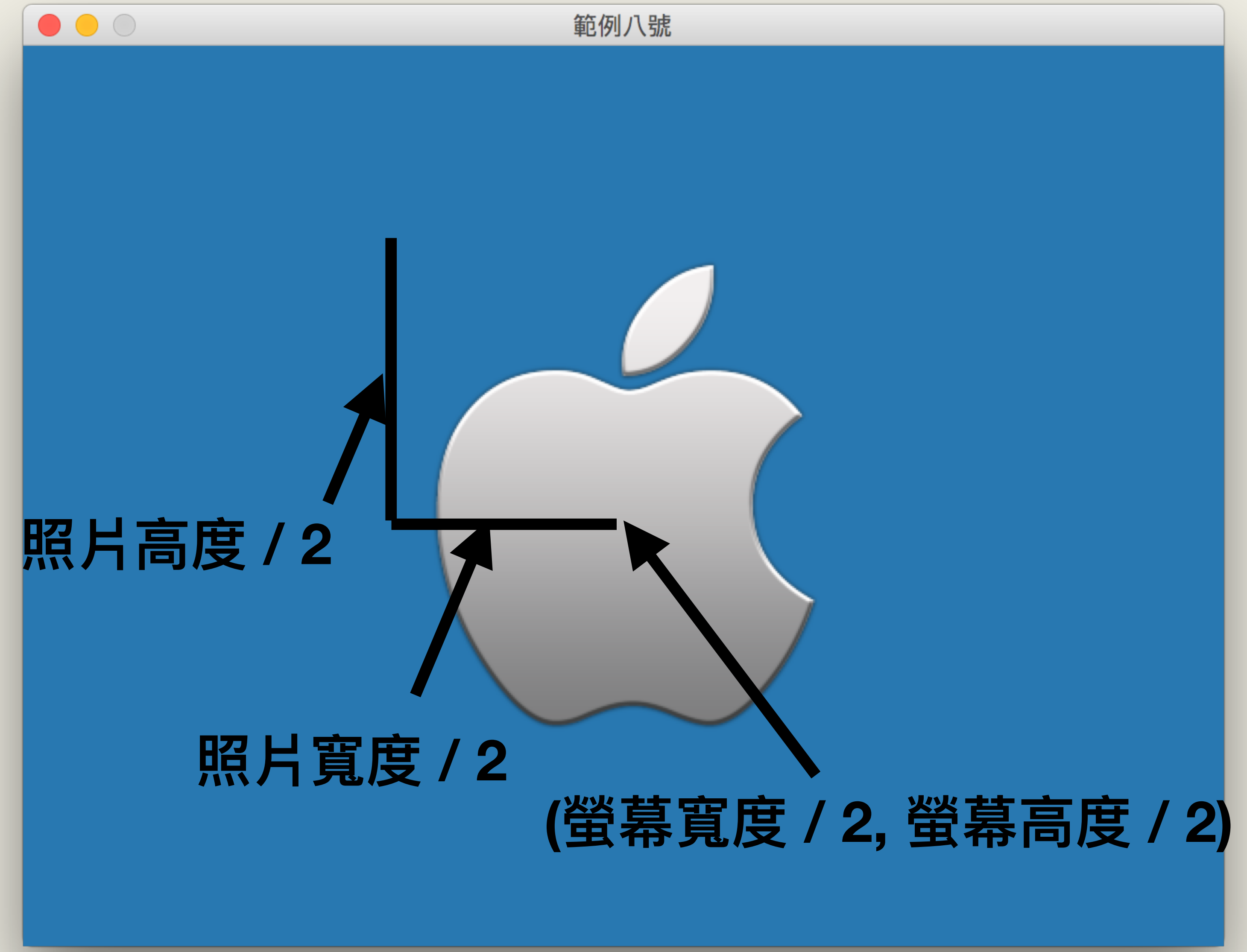
```python
img = pygame.image.load("apple.png")
screen.blit(img, (100, 100))
```

Practice

- 如何讓圖片顯示在畫面正中央?
- Hint: 善用 get_width() 和 get_height() 組合公式



example8_ans.py



FONT

字型

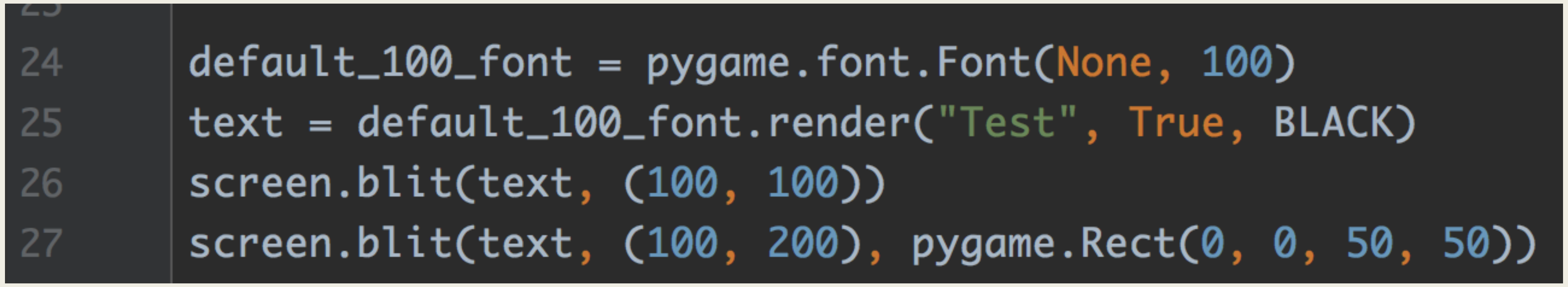
設定字型與大小

- font = pygame.font.Font(filename, size)
  - filename：字體檔案；若為None，則使用預設字體
  - size：字的大小
- word = font.render(text, antialias, color, background)
  - text, antialias, color：內容, 平滑, 顏色
  - background：背景顏色

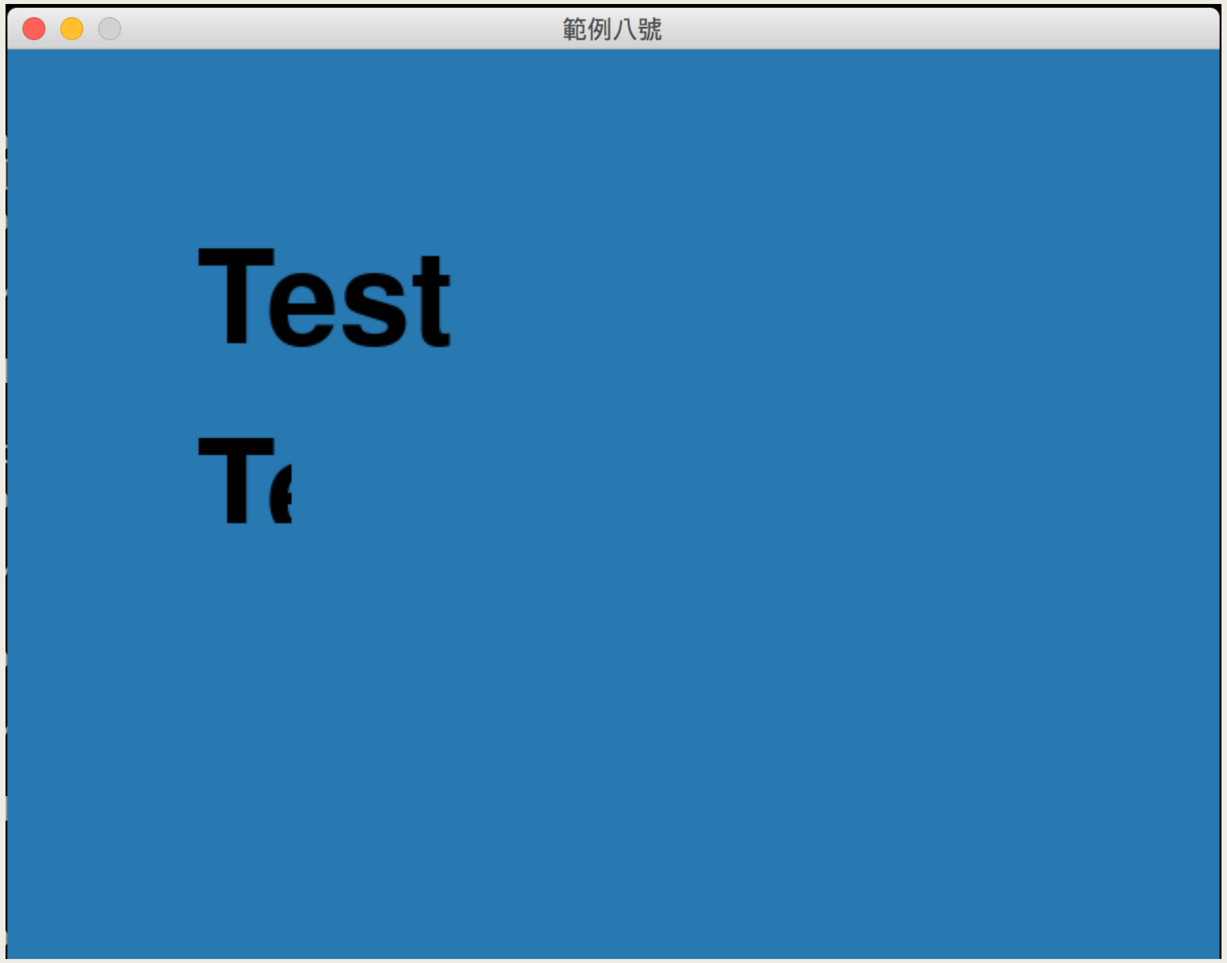
型態?

- `screen  <class ‘pygame.Surface'>`
- `font    <class ‘pygame.font.Font'>`
- `word    <class ‘pygame.Surface'>`

example9.py



```python
default_100_font = pygame.font.Font(None, 100)
text = default_100_font.render("Test", True, BLACK)
screen.blit(text, (100, 100))
screen.blit(text, (100, 200), pygame.Rect(0, 0, 50, 50))
```



SURFACE

顯示型態

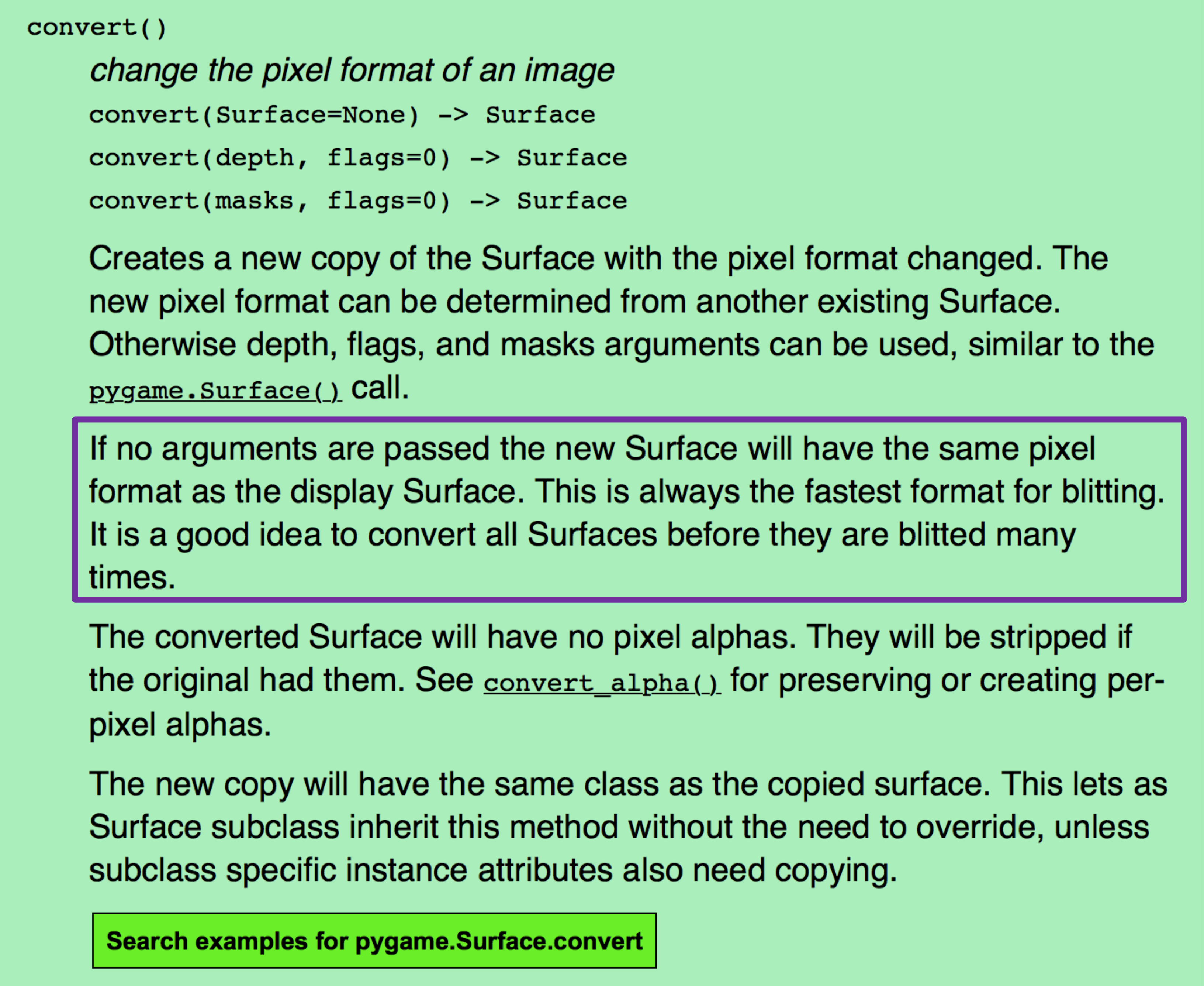
操作

- screen.fill(color)
- word.get_size()
  - 得到surface的寬度及高度
- word.get_width()
  - 得到surface的寬度
- word.get_height()
  - 得到surface的高度

繪製

- dstn.blit(src, pos, area, flags)
  - destination：被貼的surface
  - source：貼過去的surface
  - position：貼在dstn上的位置(左上角)
  - area：src要貼過去的區域
  - flags：特殊設定

Convert



```
convert()
```

change the pixel format of an image

```
convert(Surface=None) -> Surface
convert(depth, flags=0) -> Surface
convert(masks, flags=0) -> Surface
```

Creates a new copy of the Surface with the pixel format changed. The new pixel format can be determined from another existing Surface. Otherwise depth, flags, and masks arguments can be used, similar to the `pygame.Surface()` call.

If no arguments are passed the new Surface will have the same pixel format as the display Surface. This is always the fastest format for blitting. It is a good idea to convert all Surfaces before they are blitted many times.

The converted Surface will have no pixel alphas. They will be stripped if the original had them. See `convert_alpha()` for preserving or creating per-pixel alphas.

The new copy will have the same class as the copied surface. This lets as Surface subclass inherit this method without the need to override, unless subclass specific instance attributes also need copying.

Q&A