

Soluzioni degli esercizi

Queste soluzioni sono proposte soprattutto per favorire un'acquisizione progressiva delle conoscenze. Bisogna partire dall'assunto che esse *non* siano le uniche o le migliori. Prima di studiarle, ognuno deve cercare in autonomia le *proprie*, che potranno anche essere molto diverse da quelle proposte. Alcune delle soluzioni seguenti potrebbero essere incomplete e presentare solo alcune idee per risolvere gli aspetti più critici del problema.

In queste proposte noterete che i nomi delle variabili, i commenti ecc. sono in inglese. Un suggerimento è quello di provare a operare sul codice per esempio "*traducendolo*" in italiano in modo da riflettere sulla sua logica e il suo contenuto.

Esercizi capitolo 2 - Programmare con Python

Hello, admin!

```
user = input("User? ")
print("Welcome, ", user, "!")
if user == "admin":
    print("At your command")
```

 https://fondinfo.github.io/play/?exs/c02_admin.py

 Attenzione a distinguere i nomi delle variabili, che sono visibili solo nel codice del programma, dai messaggi testuali per l'utente che sono racchiusi tra virgolette.

Cerchio

```
import g2d
from math import pi

g2d.init_canvas((400, 400))
r = float(g2d.prompt("Radius? "))

if 0 <= r <= 200:
    g2d.draw_circle((200, 200), r)
    area = pi * r ** 2
    perimeter = 2 * pi * r
    g2d.draw_text("Area: " + str(area),
                  (CX, CY - r - TY), TH)
    g2d.draw_text("Perimeter: " + str(perimeter),
                  (CX, CY + r + TY), TH)
else:
    g2d.alert("Error: out of range")

g2d.main_loop()
```

 https://fondinfo.github.io/play/?exs/c02_circle.py

Per generalità, è preferibile usare `float` anziché `int`.

I confronti possono essere concatenati, omettendo un `and` implicito: `0 <= r and r <= 200`.

L'anno del drago

```
year = int(input("What's your birth year? "))
if (2024 - year) % 12 == 0:
    print("Your sign is Dragon")
else:
    print("Your sign is not Dragon")
```

 https://fondinfo.github.io/play/?exs/c02_dragon.py

The Bridge of Death

```
name = input("What is your name? ")
quest = input("What is your quest? ")
color = input("What is your favorite color?")

if name == "Lancelot" and quest == "Holy Grail" and color == "Blue":
    print("Right. Off you go.")
else:
    print("Down into the Gorge of Eternal Peril!")
```

 https://fondinfo.github.io/play/?exs/c02_bridge.py

Calcolo dell'età

```
b_year = int(input("Birth year? "))
b_month = int(input("Birth month? "))
b_day = int(input("Birth day? "))

c_year = int(input("Current year? "))
c_month = int(input("Current month? "))
c_day = int(input("Current day? "))

age = c_year - b_year
if c_month < b_month or (c_month == b_month and c_day < b_day):
    age -= 1

print("Your age is", age)
```

 https://fondinfo.github.io/play/?exs/c02_age.py

Minore e maggiore

```
from random import randint

a = randint(1, 6)
b = randint(1, 6)
c = randint(1, 6)
smallest = 0
largest = 0

if a < b and a < c:
    smallest = a
elif b < c:
    smallest = b
else:
    smallest = c

if a > b and a > c:
    largest = a
elif b > c:
    largest = b
else:
    largest = c

print("Values:", a, b, c)
print("Min:", smallest)
print("Max:", largest)
```

 https://fondinfo.github.io/play/?exs/c02_min3.py

Dopo il primo confronto, possiamo escludere che a sia il minore. Confrontiamo poi solo b con c .

Quadrati casuali

```
import g2d
from random import randrange

W, H = 640, 480
SIDE = 100
g2d.init_canvas((W, H))

n = int(g2d.prompt("n? "))
for i in range(n):
    color = randrange(255), randrange(255), randrange(255)
    pos = randrange(W - SIDE), randrange(H - SIDE)
    g2d.set_color(color)
    g2d.draw_rect(pos, (SIDE, SIDE))

g2d.main_loop()
```

 https://fondinfo.github.io/play/?exs/c02_randsquares.py

Quadrati in diagonale

```
import g2d
from random import randrange

L = 500
g2d.init_canvas((L, L))

n = int(g2d.prompt("How many squares?"))
l = L / n
for i in range(n):
    g2d.set_color((randrange(256), randrange(256), randrange(256)))
    pos = i * l
    g2d.draw_rect((pos, pos), (l, l))

g2d.main_loop()
```

 https://fondinfo.github.io/play/?exs/c02_diagsquares.py

La posizione è proporzionale a i . Lo spazio va diviso tra gli n quadrati. Il lato dei quadrati è dunque pari a $\frac{L}{n}$.

Segmenti casuali

Ci viene chiesto di generare n segmenti, per ognuno dei quali è necessario individuare due punti: un punto di partenza e un punto di arrivo.

Sarà dunque sufficiente un ciclo `for`, all'interno del quale genereremo due punti per ogni iterazione, per poi collegarli con un segmento.

Per generare i punti in posizioni visibili nel canvas, è molto comodo usare la funzione `randrange`, con parametro `W` o `H`.

Visto che la variabile di iterazione (che spesso chiamiamo `i`) in questo caso non è usata per i calcoli, possiamo sostituirla con un anonimo carattere di sottolineatura (*underscore*).

```
import g2d
from random import randrange

W, H = 480, 360
g2d.init_canvas((W, H))
g2d.set_color((0, 0, 0))

n = int(g2d.prompt("How many segments? "))
for _ in range(n):
    src = (randrange(W), randrange(H))
    dst = (randrange(W), randrange(H))
    g2d.draw_line(src, dst)

g2d.main_loop()
```

 https://fondinfo.github.io/play/?exs/c02_segments.py

Linea spezzata

In questo caso, segmenti successivi condividono un punto. In tutto, sarà necessario generare $n + 1$ punti casuali.

Ogni segmento, userà come punto di partenza, il punto di arrivo del segmento precedente. Ad ogni iterazione, basterà quindi generare un nuovo punto di arrivo.

Prima di iniziare il ciclo, però, andrà generato il punto di partenza iniziale.

Inoltre, al termine di ogni iterazione, bisogna fare in modo che l'attuale punto di arrivo diventi il punto di partenza alla successiva iterazione, con un semplice assegnamento. Senza di ciò, erroneamente i segmenti partirebbero sempre dall'unico punto di partenza definito inizialmente.

```
import g2d
from random import randrange

W, H = 480, 360
g2d.init_canvas((W, H))
g2d.set_color((0, 0, 0))

n = int(g2d.prompt("How many segments? "))

src = (randrange(W), randrange(H))
for _ in range(n):
    dst = (randrange(W), randrange(H))
```

```
g2d.draw_line(src, dst)
src = dst

g2d.main_loop()
```

 https://fondinfo.github.io/play/?exs/c02_polyline.py