[ITADINFO]

CONVEGNO ITALIANO
SULLA DIDATTICA DELL'INFORMATICA

Insegnare a Programmare Componendo Grafiche con PyTamaro



Luca
Chiodini



Joey
Bevilacqua



Matthias
Hauswirth

10 Ottobre 2026 · Padova

Lugano, Ticino



USI

Università della Svizzera italiana

swissuniversity.ch

Software Institute







Matthias
Hauswirth



Luca
Chiodini



Joey
Bevilacqua

Facciamo un’**attività di riscaldamento** e **conosciamoci!**

Domanda?

Sì
⇒ In piedi

No
⇒ Seduti

Facciamo un'**attività di riscaldamento** e **conosciamoci!**

Chi è a Padova in questo momento?

Sì
⇒ In piedi

No
⇒ Seduti

Facciamo un'**attività di riscaldamento** e **conosciamoci!**

Chi insegna in una scuola secondaria di secondo grado?

Sì
⇒ In piedi

No
⇒ Seduti

Facciamo un'**attività di riscaldamento** e **conosciamoci!**

Chi insegna (anche) programmazione?

Sì
⇒ In piedi

No
⇒ Seduti

Facciamo un'**attività di riscaldamento** e **conosciamoci!**

Chi insegna programmazione (anche) usando Python?

Sì
⇒ In piedi

No
⇒ Seduti

Creare Grafiche in Python con PyTamaro

PAC-MAN

A sensational, full color video game for 1 or 2 players that tickles vision and challenges reflexes. Adding to the fun are musical refrains, chomping and action sounds along with amusing cartoon shows between racks.



HIGH SCORE — Retained and displayed daily.

PLAYERS' SCORE

DOTS — 10 Points Each.

POWER CAPSULE — 50 Points Each.

GHOST MONSTERS

PAC-MAN — The 'main' man.

ESCAPE 'Warp' TUNNEL — Out one side– reappear on other.

BONUS FRUIT TARGET — Appear below Monster's Den twice during each rack.

NUMBER OF SCREENS CLEARED — Fruit indicates how many times player has cleared the screen of dots.

- Screen cleared once.
- Screen cleared twice.
- Screen cleared 3 times.
- ? Screen cleared 4 times.
- ? Screen cleared 5 times.
- — etc. —

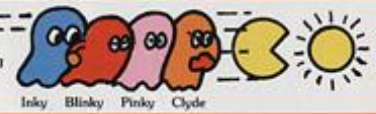
PAC-MAN REMAINING



The player, using a single handle control guides the PAC-MAN about the maze, scoring points by munching up the Dots in his path. Four Ghost Monsters—Inky, Blinky, Pinky and Clyde— chase after the PAC-MAN trying to capture and deflate him. The PAC-MAN can counterattack by eating the big, Power Capsule that enables him to overpower the Monsters for additional score. After all the Dots are gobbled up, the screen is cleared, and PAC-MAN continues for another round. Each rack features a special Fruit Target in the maze, which if eaten, earns Bonus Points. Players start with three PAC-MEN. An additional PAC-MAN is awarded for 10,000 points.

SCORE BIG ON THE COUNTERATTACK

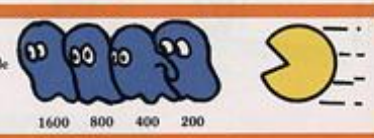
1 Quick, eat the flashing Power Capsule.



2 When the PAC-MAN powers up, the Monsters start to run away.



3 Points double for each Monster caught.



BONUS POINT FRUIT TARGETS!

The player is awarded extra Bonus Points for eating the Fruit Targets that appear in the maze.



100 Points

300 Points

500 Points

Or up to a maximum of 5,000 Bonus Points.

PAC-MAN is available in 3 Midway models: New Mini-Myte, Cocktail Table and Standard Arcade.



HEIGHT	WIDTH	DEPTH
58½" 149 cm	19½" 49.5 cm	24" 61 cm
29" 74 cm	32" 81.25 cm	22" 56 cm
73" 185.5 cm	26½" 67.25 cm	34" 86.25 cm

MIDWAY MFG. CO.
A BALLY COMPANY
10750 West Grand Avenue
Franklin Park, Illinois 60131
Phone: (312) 451-1360
For service information—call toll free 800-323-7182

DISTRIBUTED BY:

©Copyright 1980 by MIDWAY MFG. CO. All rights reserved.

Printed in U.S.A.

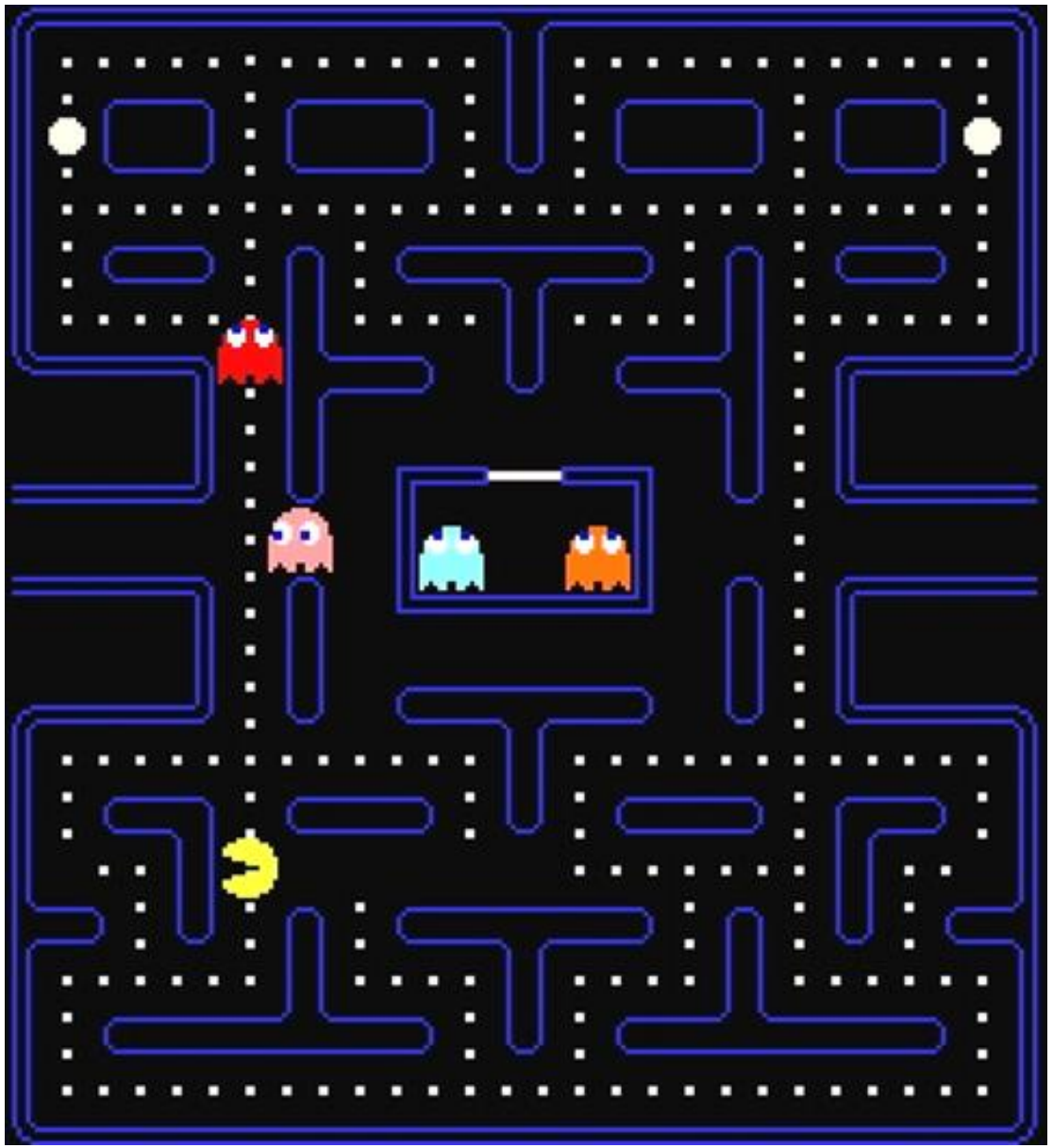

1UP
220
HIGH SCORE
1000
2UP
290

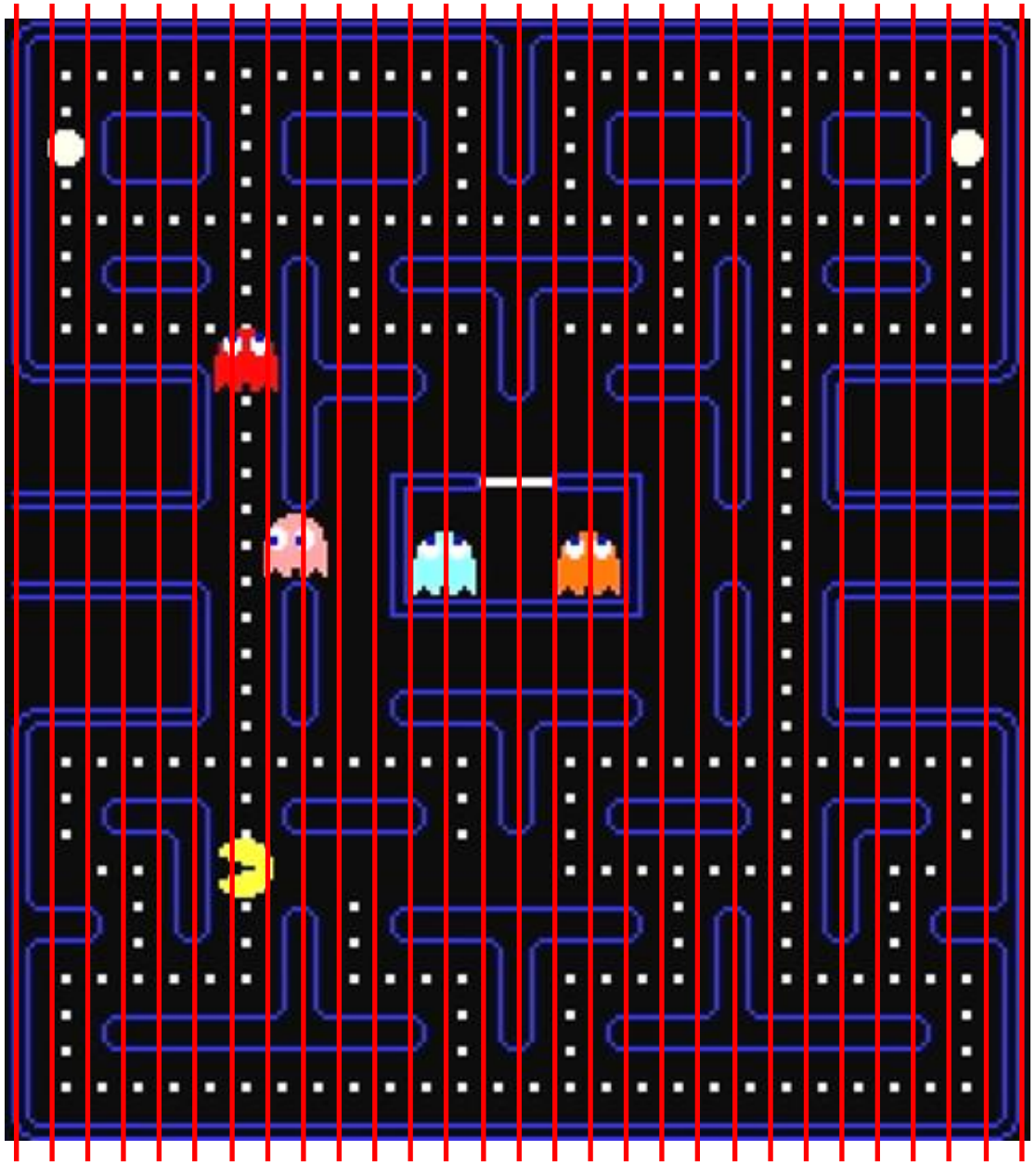


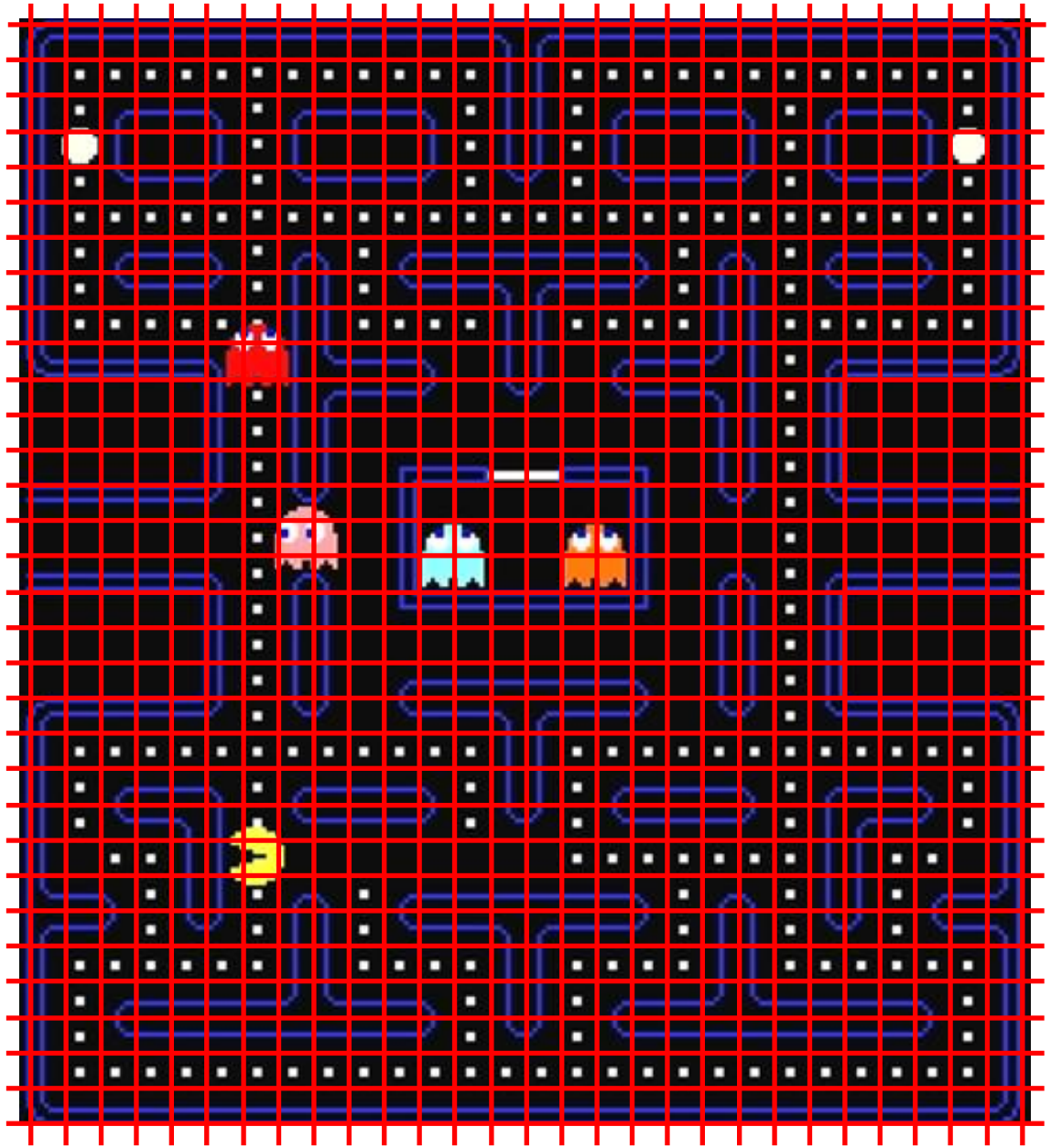
Punteggio

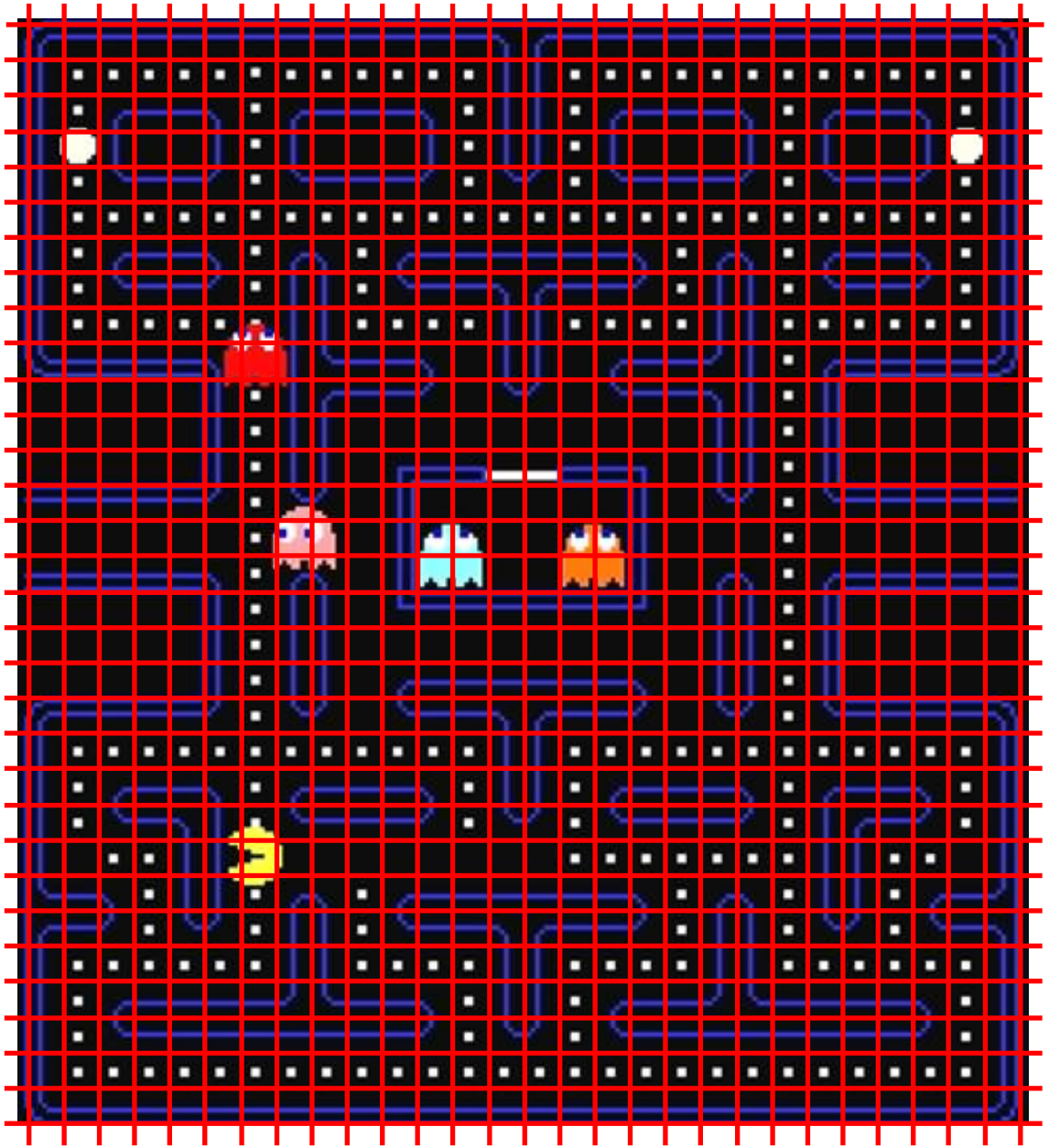
Labirinto

Stato

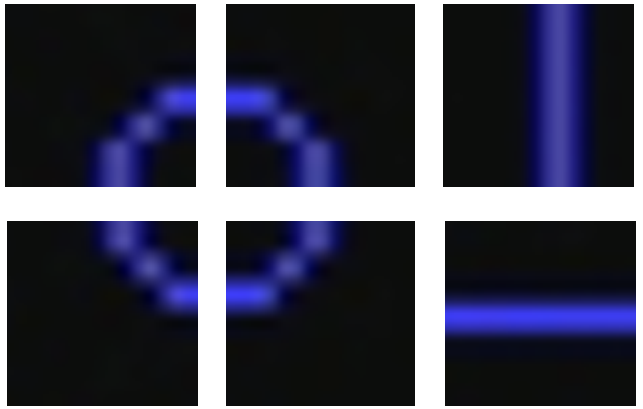








Muri



Pavimento con Puntino

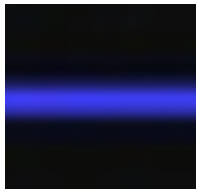


Pavimento

Pavimento con Pillola Speciale

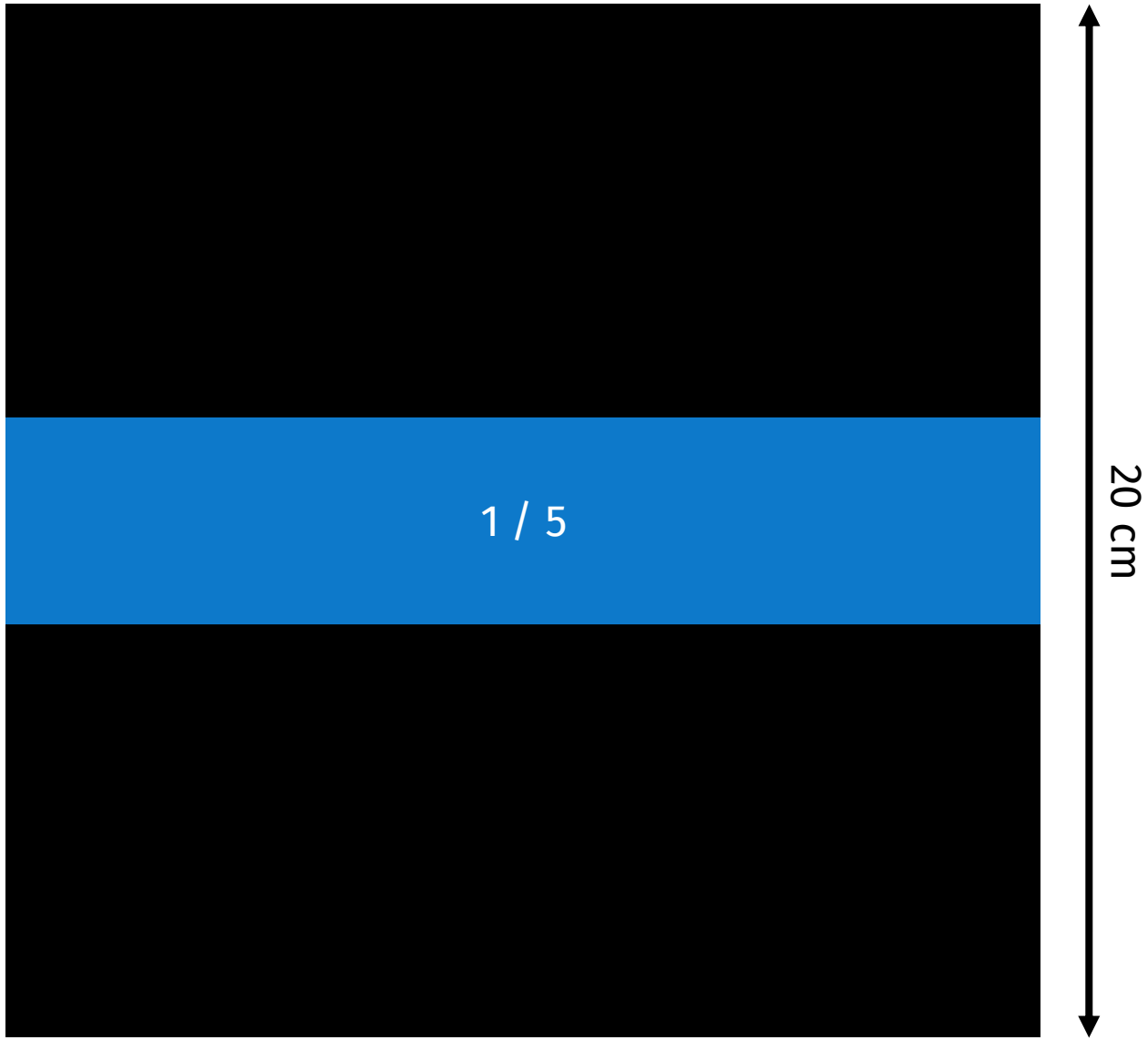


Muro



Costruire una piastrella, su carta

Tocca a te!



Soluzioni diverse







Creiamo il nostro Pacman!

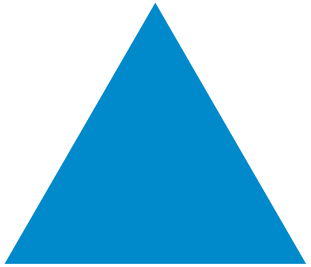
PyTamaro

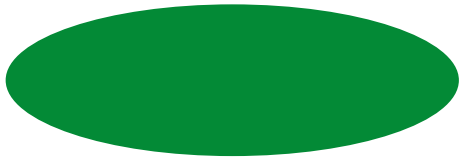
Gli **ingredienti** per una soluzione



Grafiche Primitive – “Atomi”



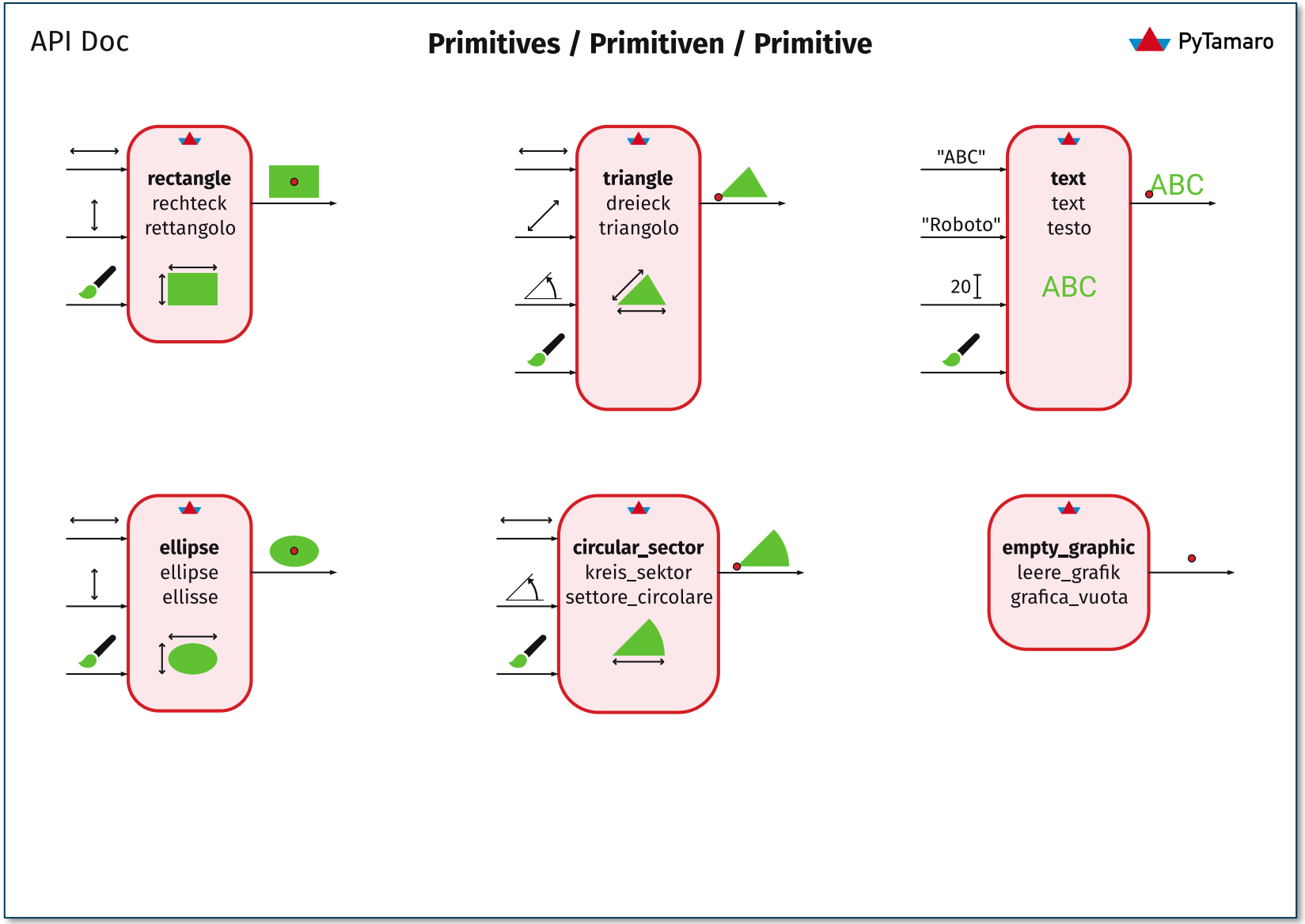






Ciao

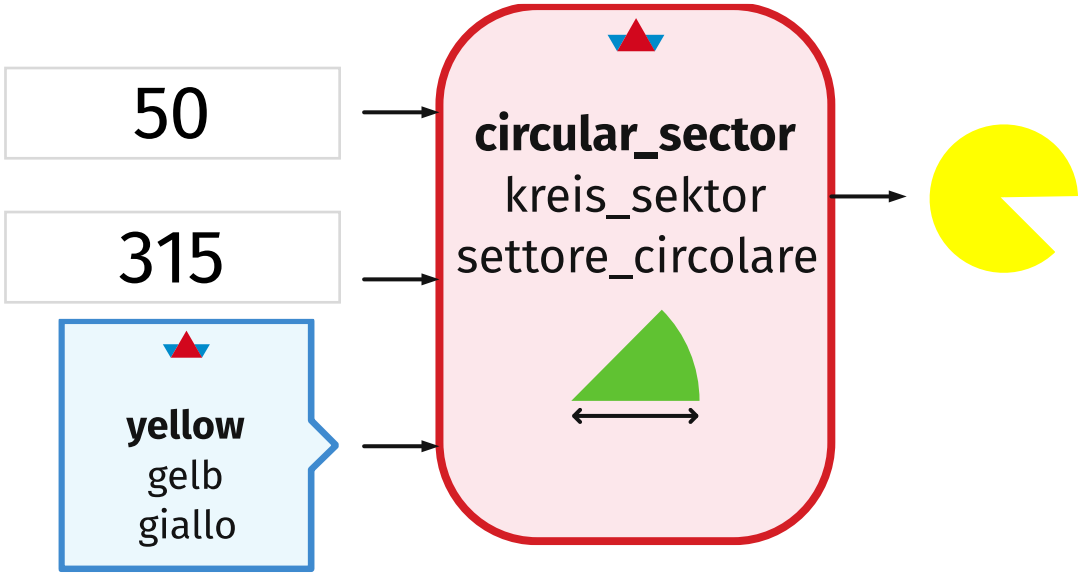
Documentazione API



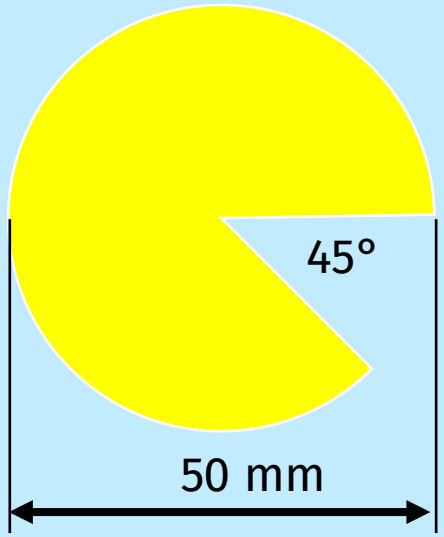
Per cominciare, un obiettivo più semplice...

Programma questo (quasi) Pacman





Grafica Desiderata

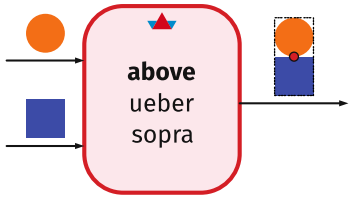


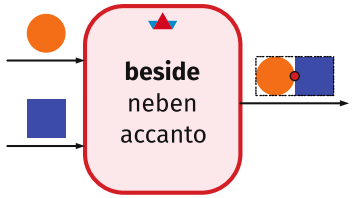
Documentazione API (sul retro)

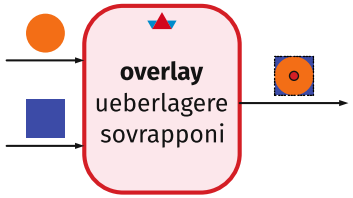
API Doc

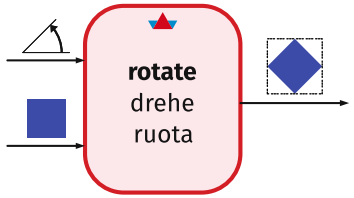
Combinators / Kombinatoren / Combinatori

PyTamaro

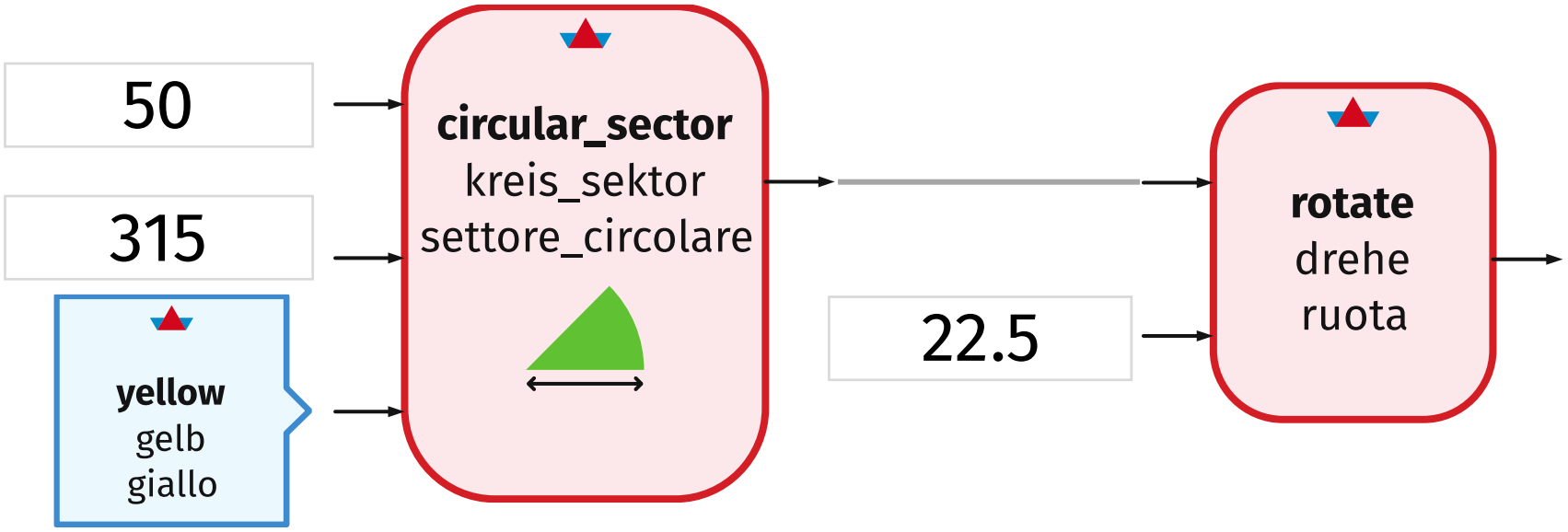






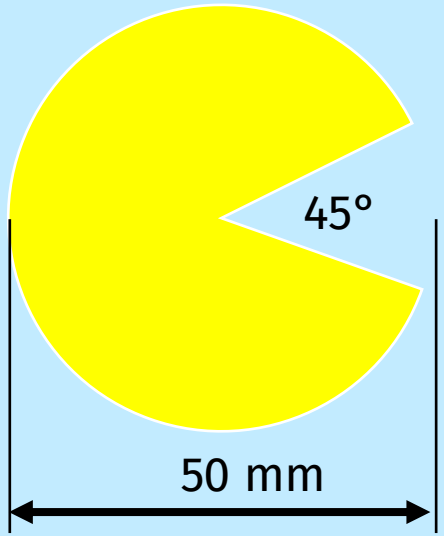


Componi un Pacman!



Tocca a te!

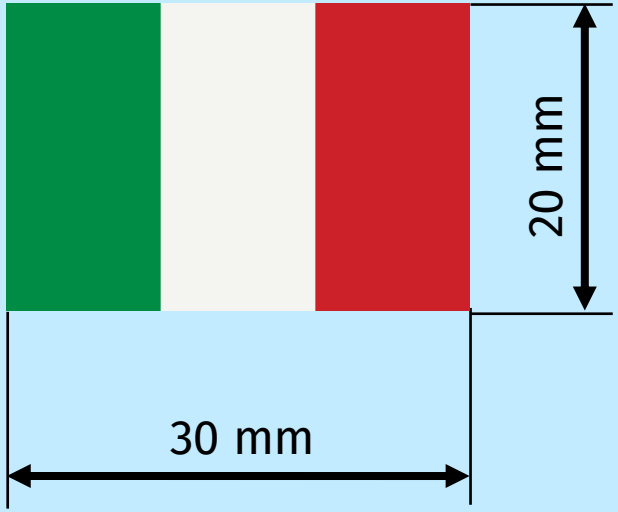
Grafica Desiderata



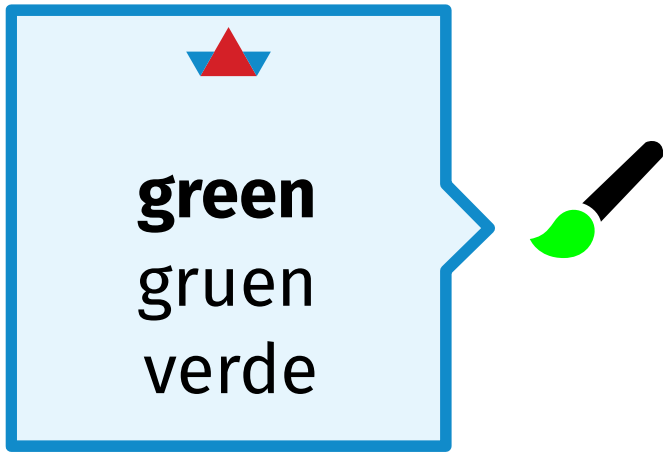
Componi una bandiera Italiana!

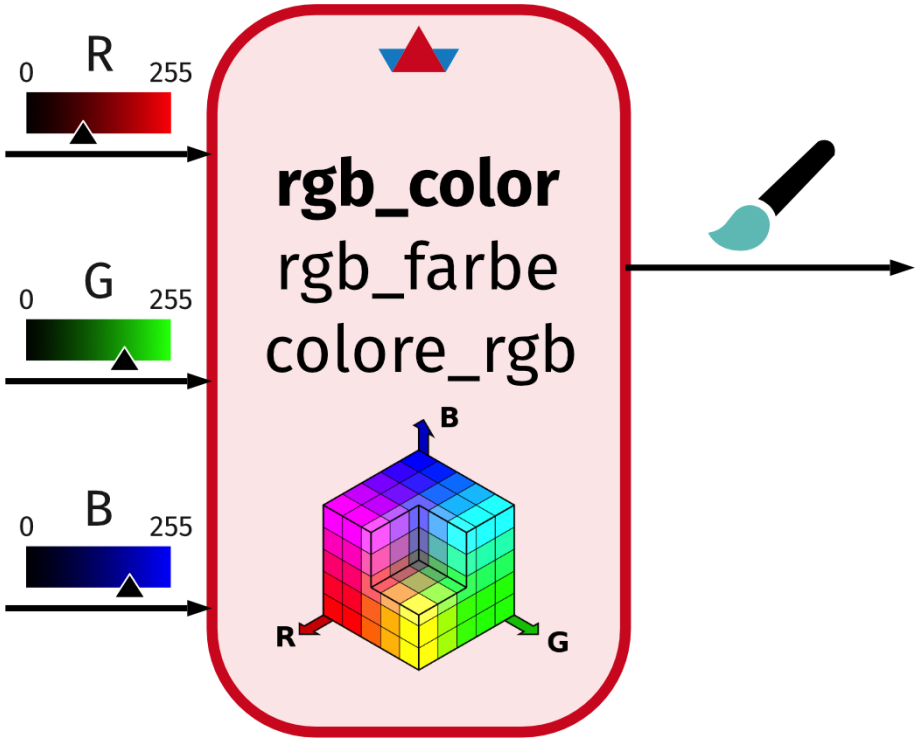
Tocca a te!

Grafica Desiderata



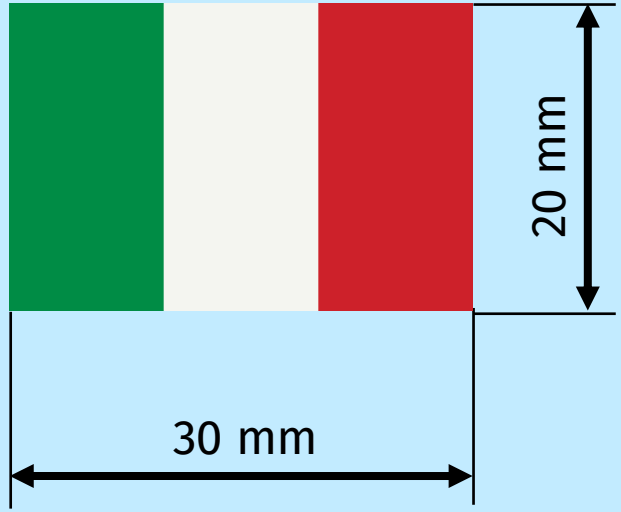
Colori personalizzati



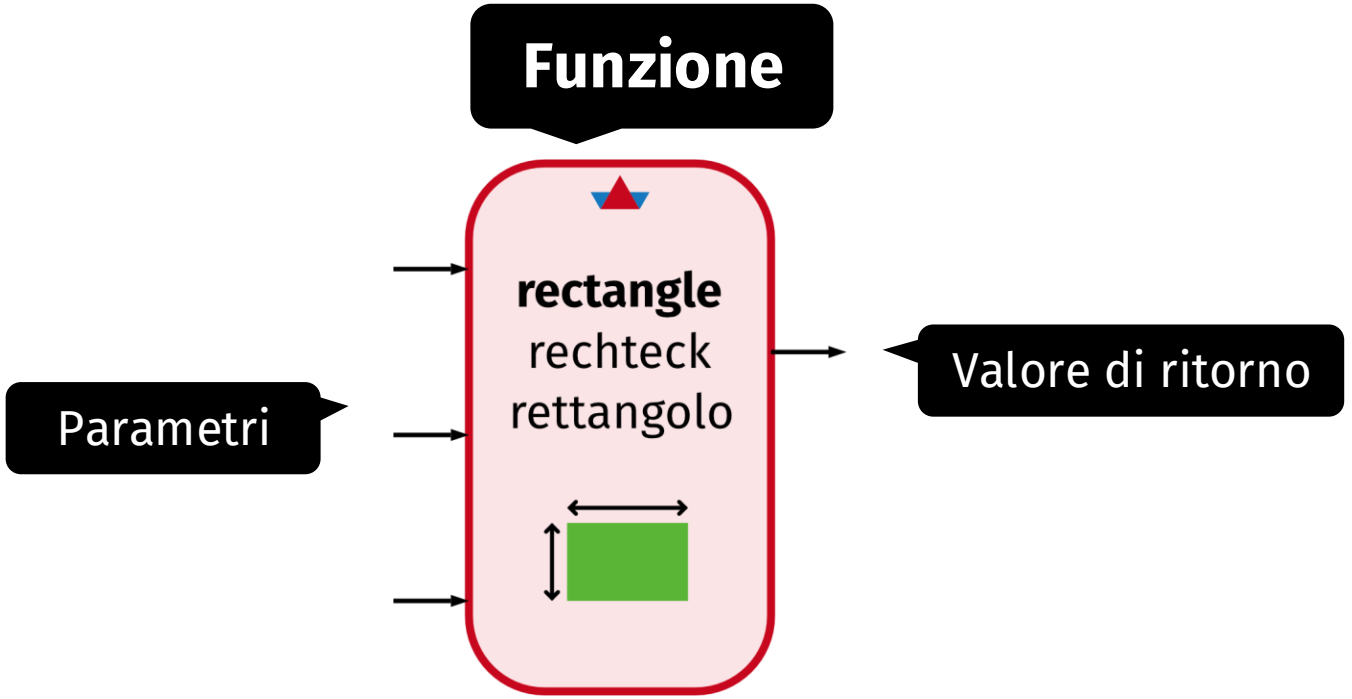


Abbiamo composizioni *diverse*?

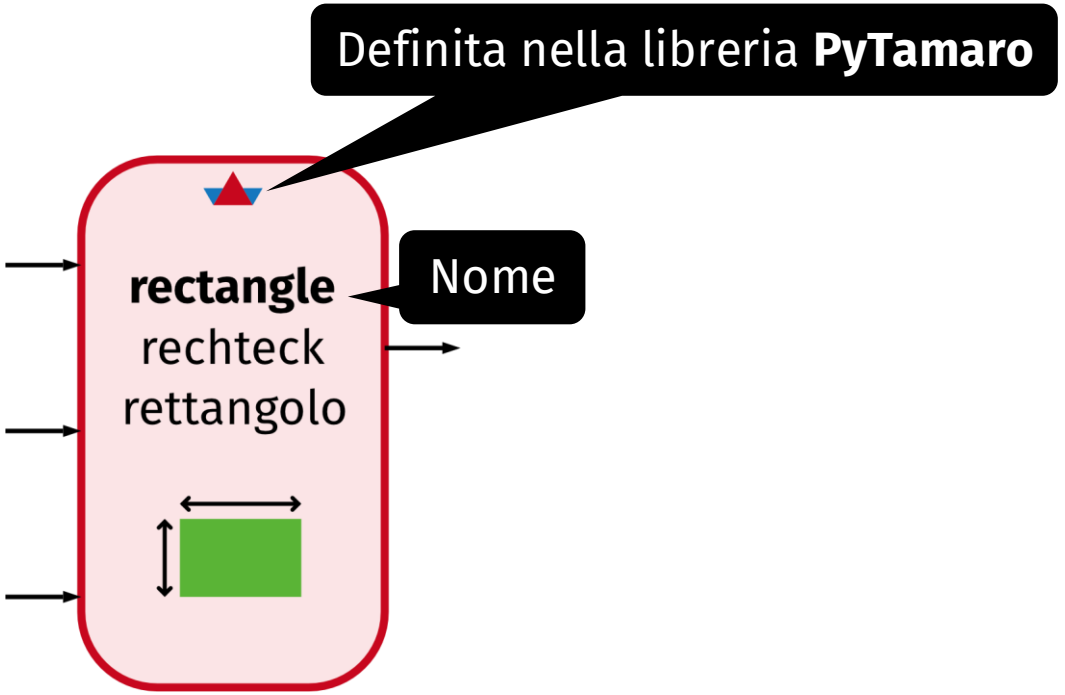
Grafica Desiderata



Concetti di programmazione



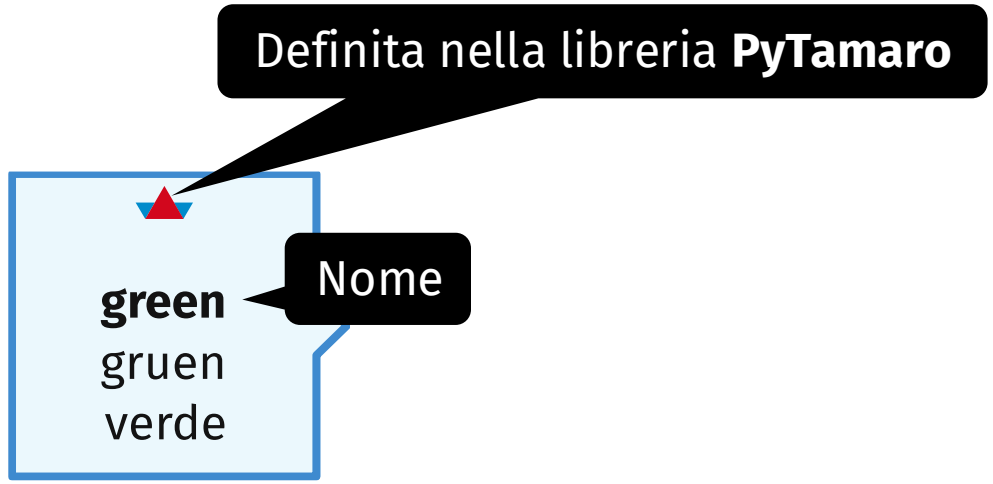
Concetti di programmazione



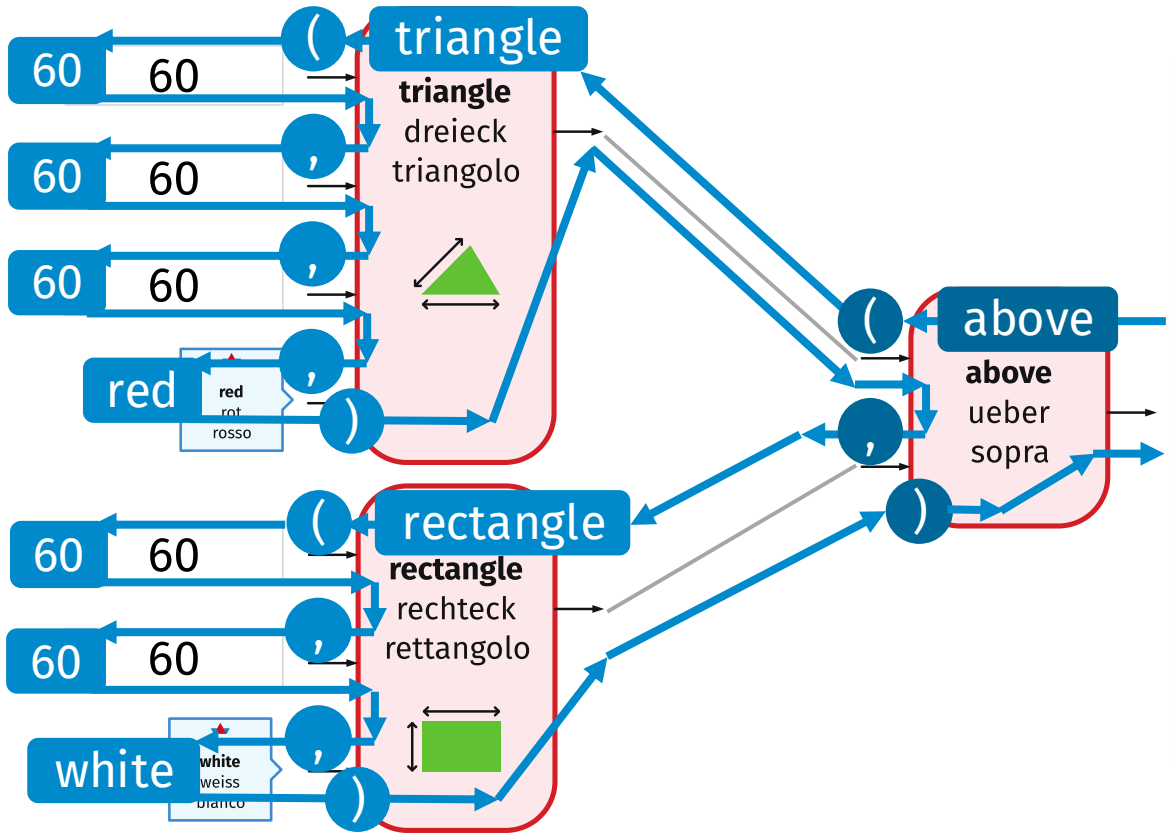
Concetti di programmazione



Concetti di programmazione



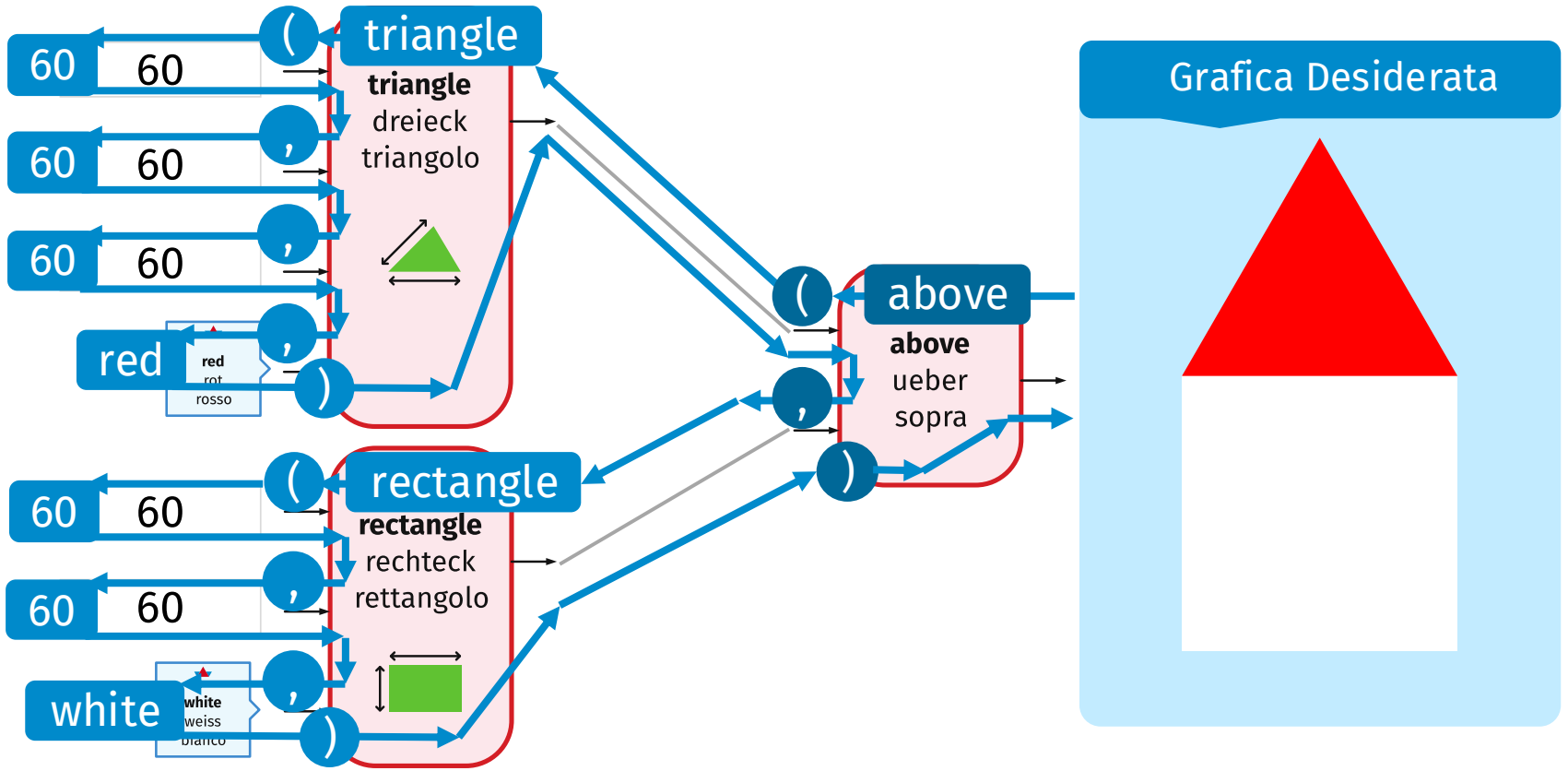
Dalle carte al codice



Python

```
above(
    triangle(60,60,60,red),
    rectangle(60,60,white)
)
```

Chiamate di funzioni annidate

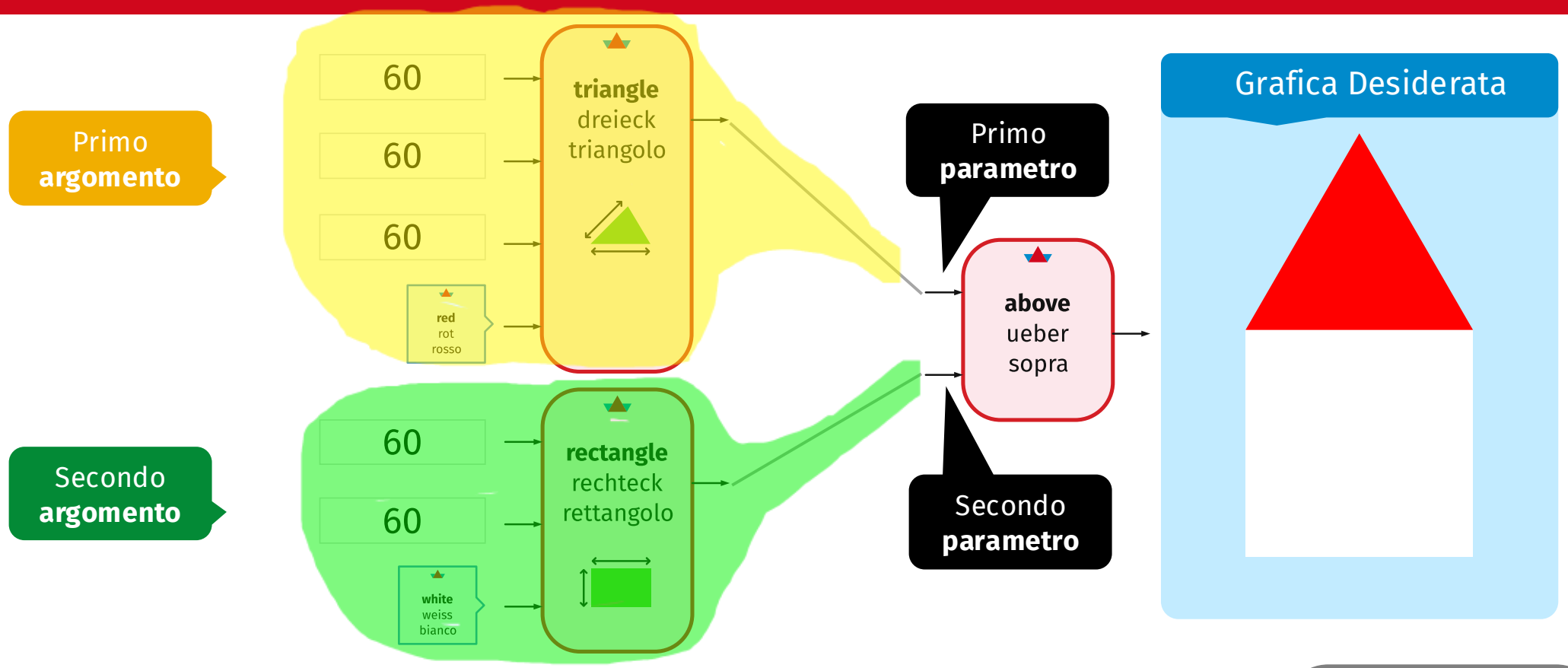


Python

Chiamate **annidate**

```
above(
    triangle(60, 60, 60, red),
    rectangle(60, 60, white)
)
```

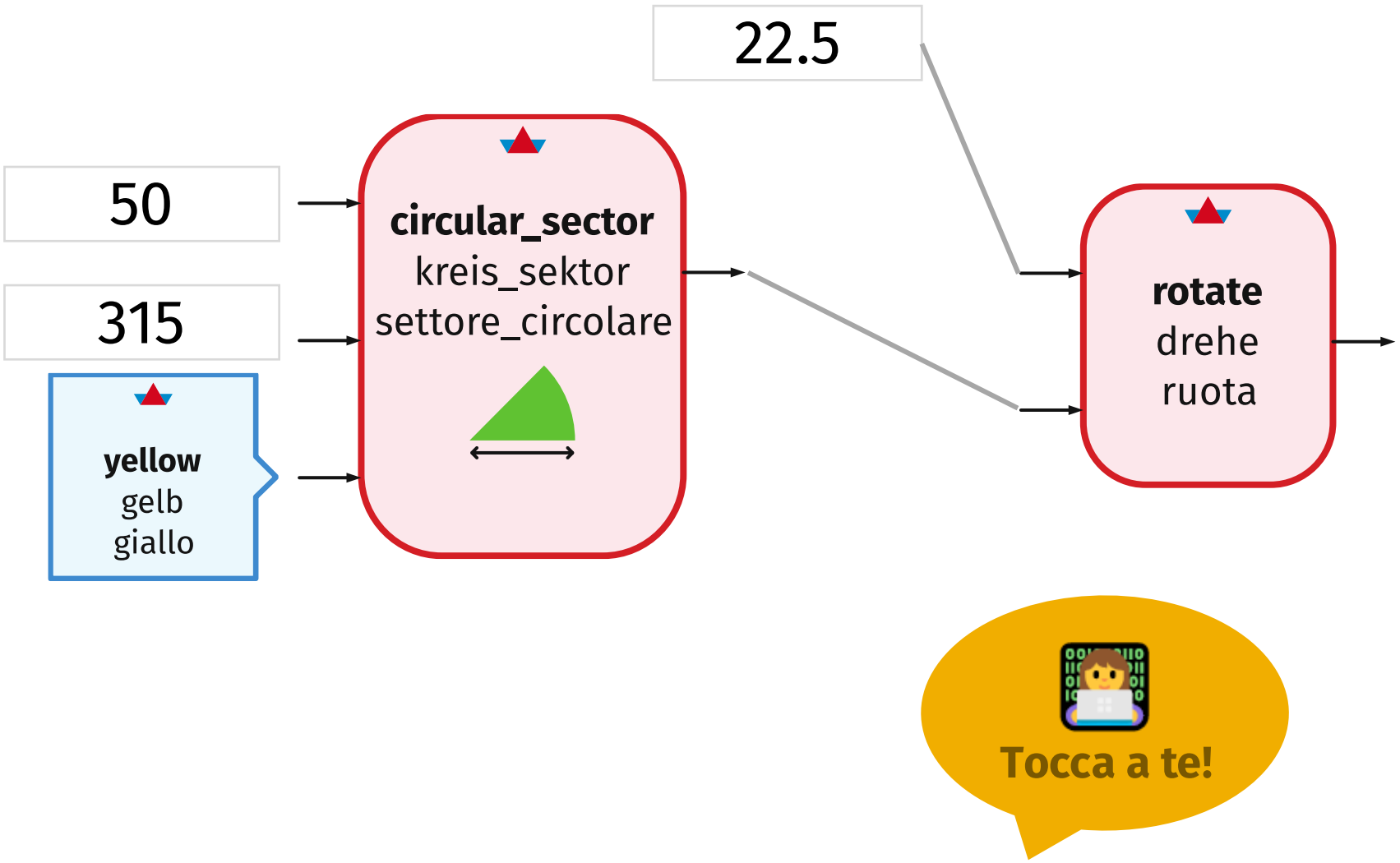
Argomenti (Espressioni) != Parametri (Variabili)



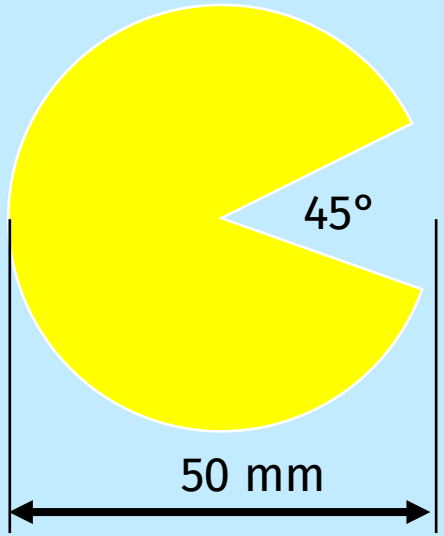
Python

```
above(
    triangle(60, 60, 60, red),
    rectangle(60, 60, white)
)
```

Programma in Python – Traduci carte in codice



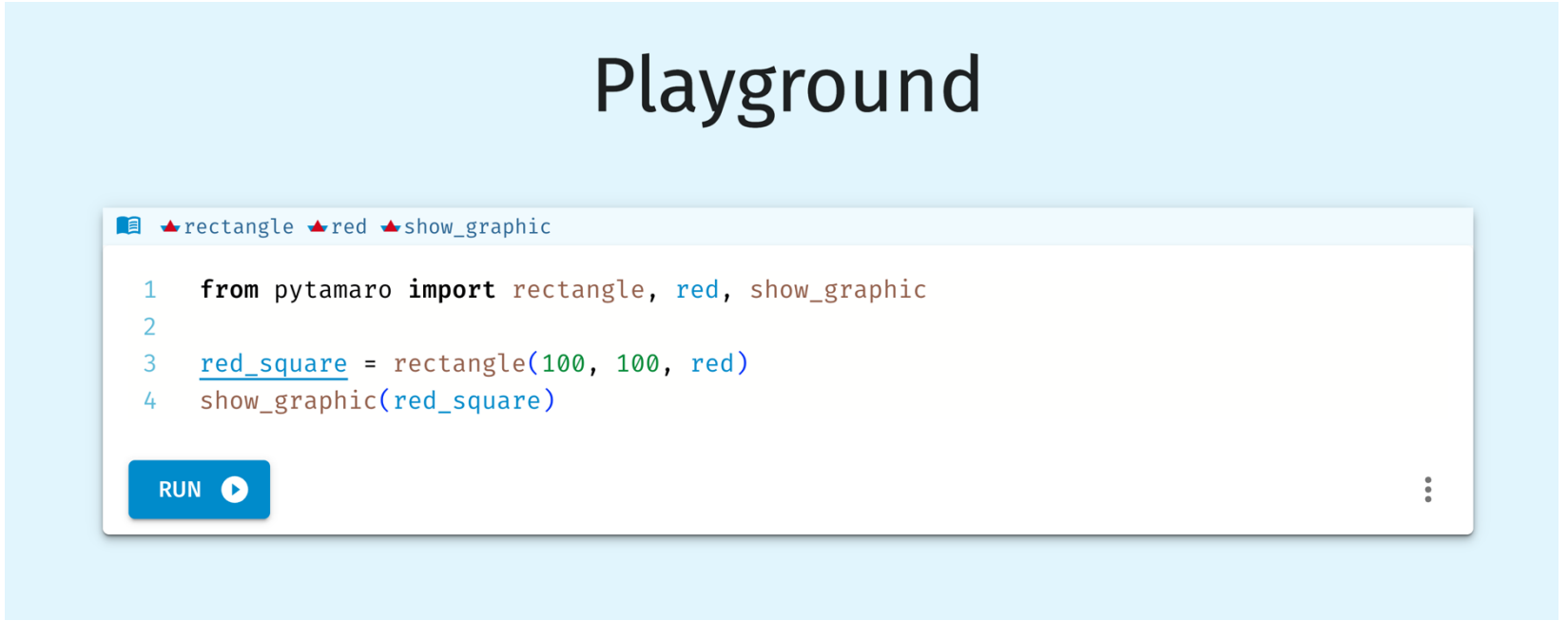
Grafica Desiderata



Programmiamo in Python

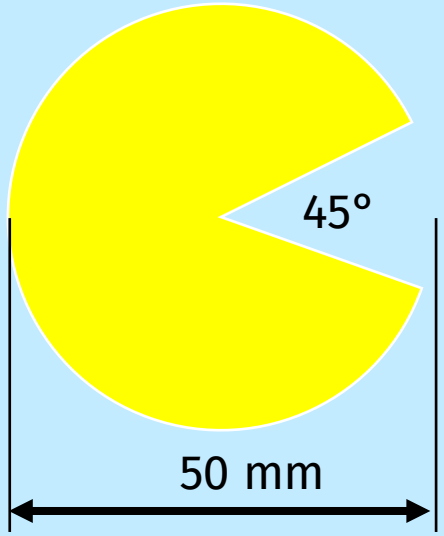
pytamaro.si.usi.ch





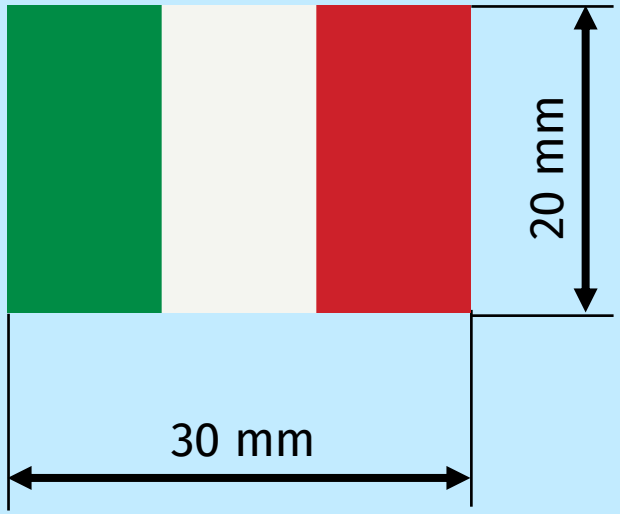
Programmiamo il nostro Pacman

Grafica Desiderata

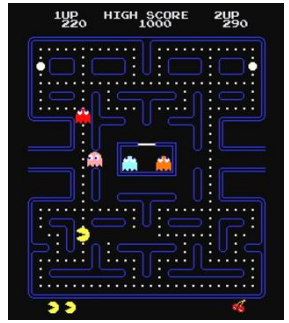


Programmiamo una bandiera Italiana

Grafica Desiderata



De(composizione)



Problema

Decomponi

Sotto-problema

Sotto-problema

Sotto-problema







"Atomi"

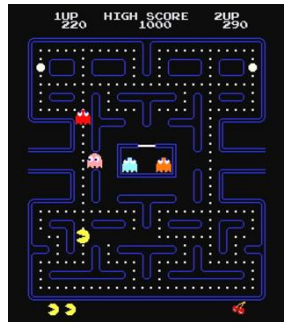
Sotto-soluzione

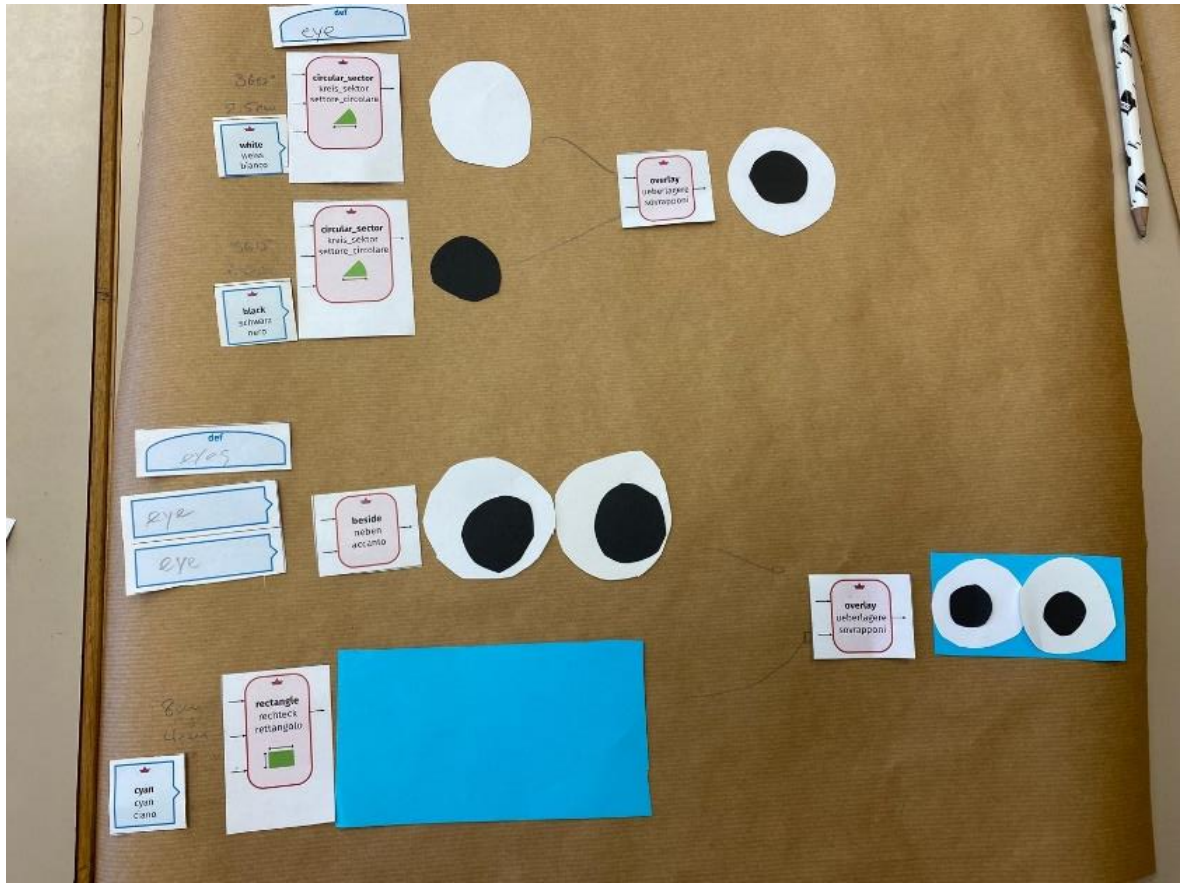
Sotto-soluzione

Sotto-soluzione

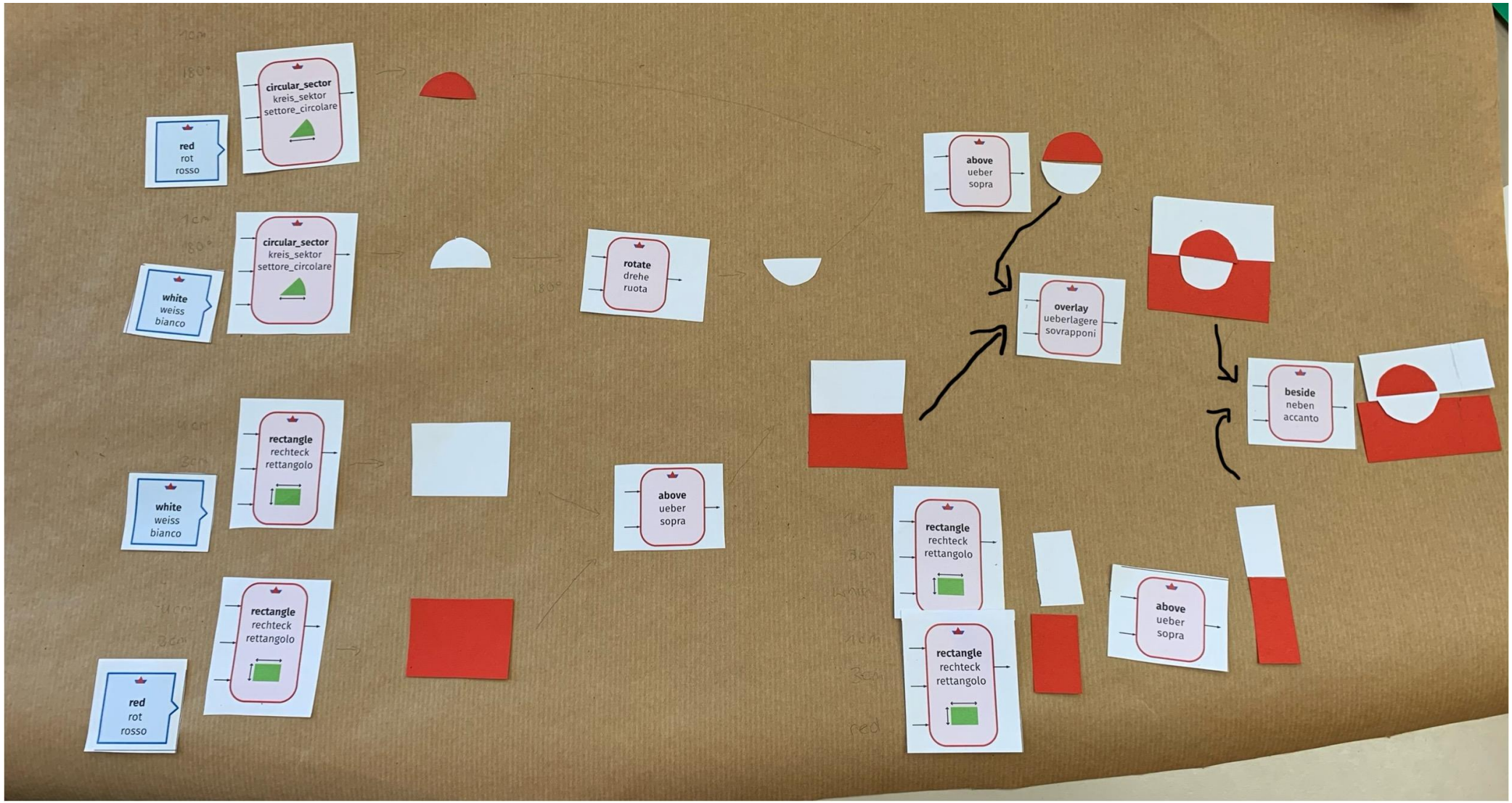
Componi

Soluzione

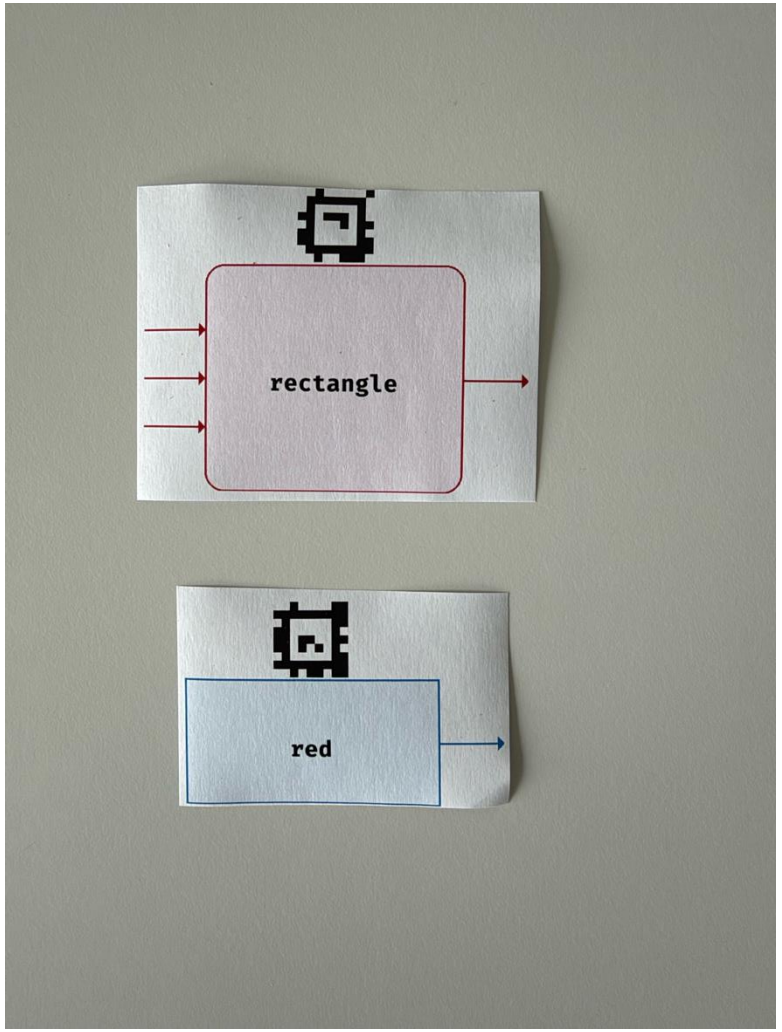



def
eye
circular_sector
kreis_sektor
settore_circolare
white
weiss
bianco
overlay
ueberlagere
sovrapponi
circular_sector
kreis_sektor
settore_circolare
black
schwarz
nero
def
eyes
eye
eye
beside
neben
accanto
overlay
ueberlagere
sovrapponi
rectangle
rechteck
rettangolo
cyan
cyan
ciano


black
schwarz
nero
rectangle
rechteck
rettangolo
above
ueber
sopra
red
rot
rosso
rectangle
rechteck
rettangolo
above
ueber
sopra
yellow
gelb
giallo
rectangle
rechteck
rettangolo


red
rot
rosso
circular_sector
kreis_sektor
settore_circolare
white
weiss
bianco
circular_sector
kreis_sektor
settore_circolare
rotate
drehe
ruota
above
ueber
sopra
overlay
ueberlagere
sovrapponi
beside
neben
accanto
rectangle
rechteck
rettangolo
white
weiss
bianco
above
ueber
sopra
rectangle
rechteck
rettangolo
rectangle
rechteck
rettangolo
above
ueber
sopra
red
rot
rosso

EyeTamaro (Early Prototype)







Dalle Grafiche alle Animazioni

Trova le 10 differenze





Trova le 10 differenze





Differenze = Buchi (Parametri)





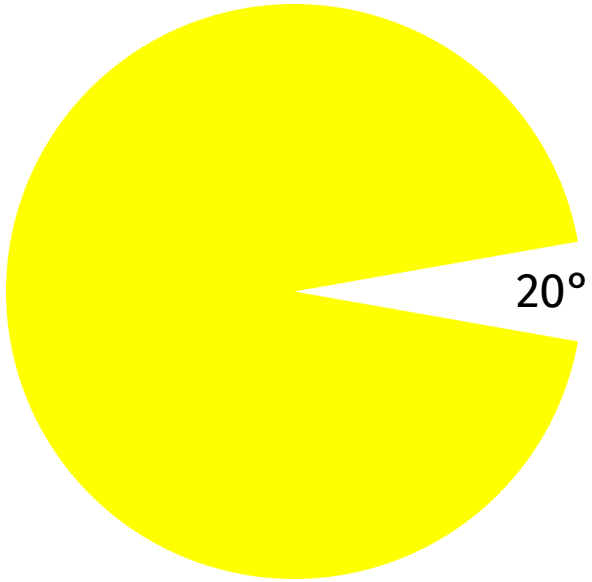
Astrazione

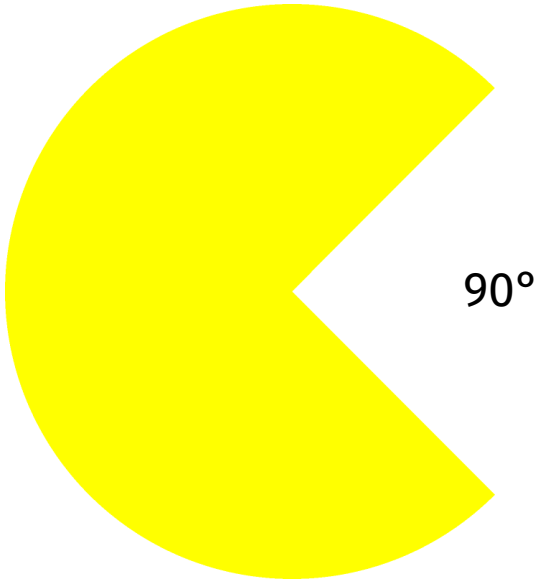
1. Trova le **Somiglianze**
2. Rendi le **Differenze** dei parametri



Cosa c’è di diverso?

Programmiamo entrambi i Pacman!



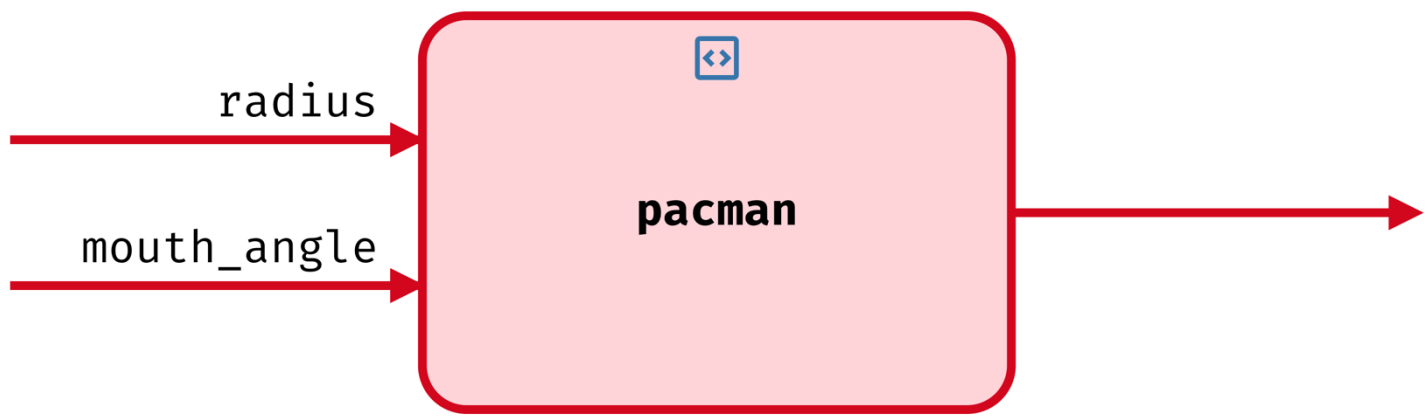


Hai fatto copia-incolla?

```
def pacman(radius, mouth_angle):
    return rotate(mouth_angle / 2,
                  circular_sector(radius, 360 - mouth_angle, yellow))
```

Function defined in your code

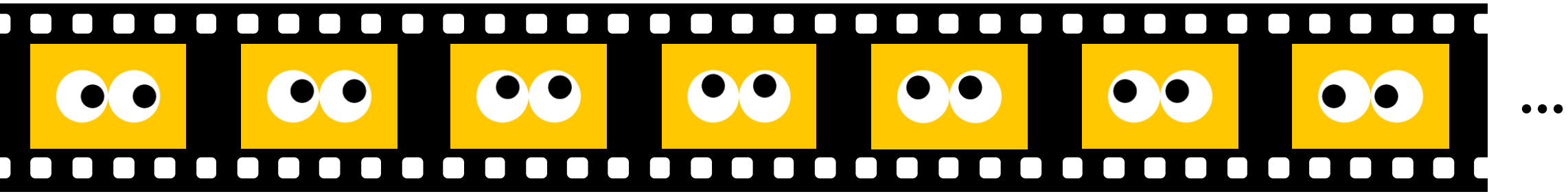
pacman(radius, mouth_angle)



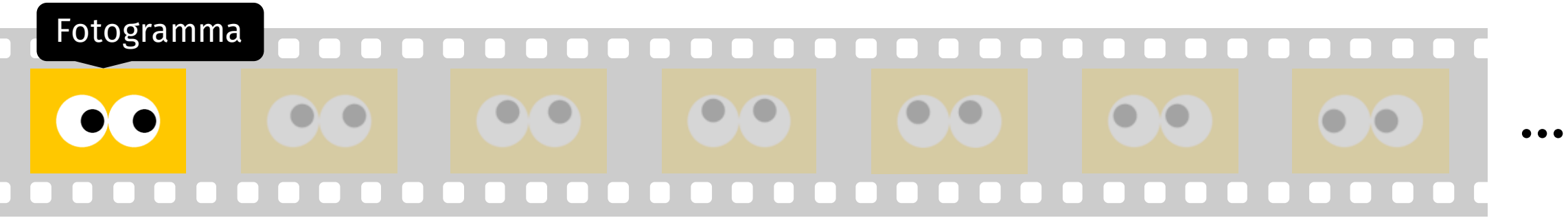
Una striscia di pellicola



Una striscia di pellicola

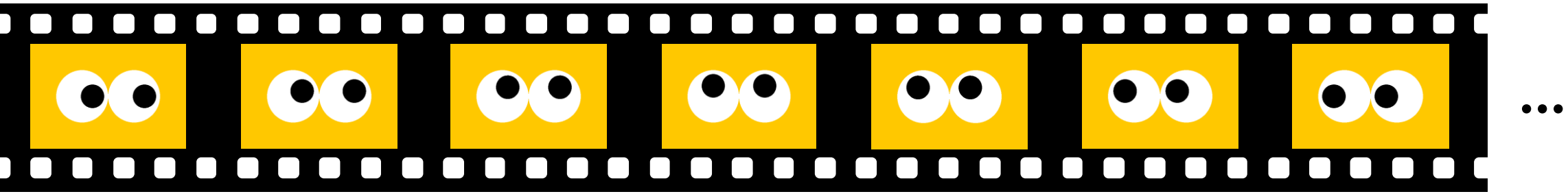


Una striscia di pellicola



...

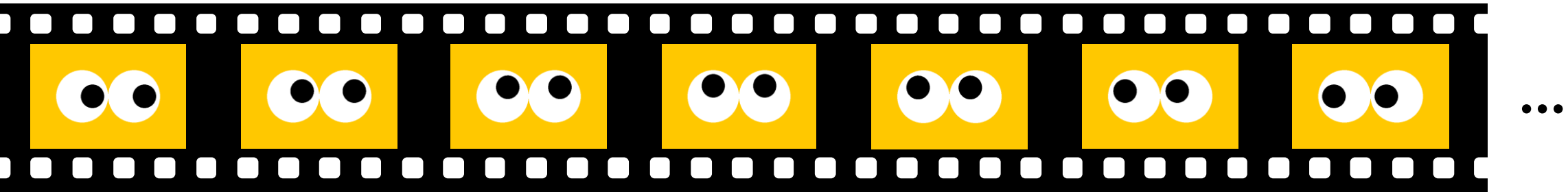
Una striscia di pellicola si MUOVE



...

? fotogrammi/s

Una striscia di pellicola si MUOVE



...

25 fotogrammi/s

0.04 s/fotogramma

40 ms/fotogramma

Una lista di grafiche (Dati ripetitivi)



Una lista di grafiche (Dati ripetitivi)



Python

Grafiche

```
frames = [
  rolling_eyes_frame0,
  rolling_eyes_frame1,
  rolling_eyes_frame2,
  rolling_eyes_frame3,
  ...
]
show_animation(frames)
```

Una lista di grafiche (Dati ripetitivi)



Python

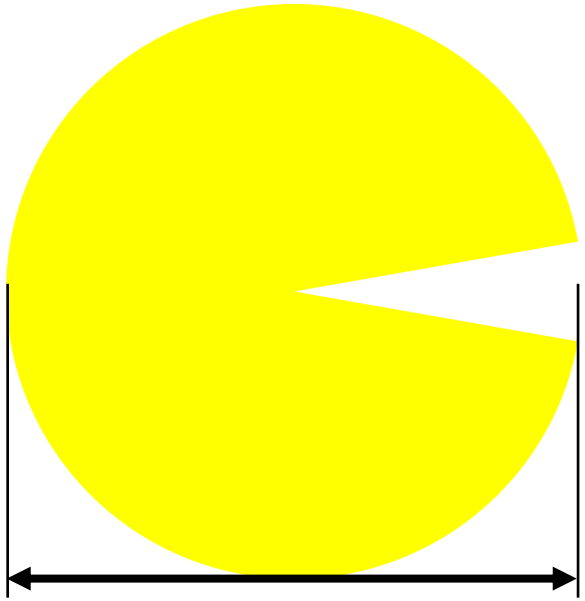
```
frames = [
    rolling_eyes_frame0,
    rolling_eyes_frame1,
    rolling_eyes_frame2,
    rolling_eyes_frame3,
    ...
]
show_animation(frames)
```

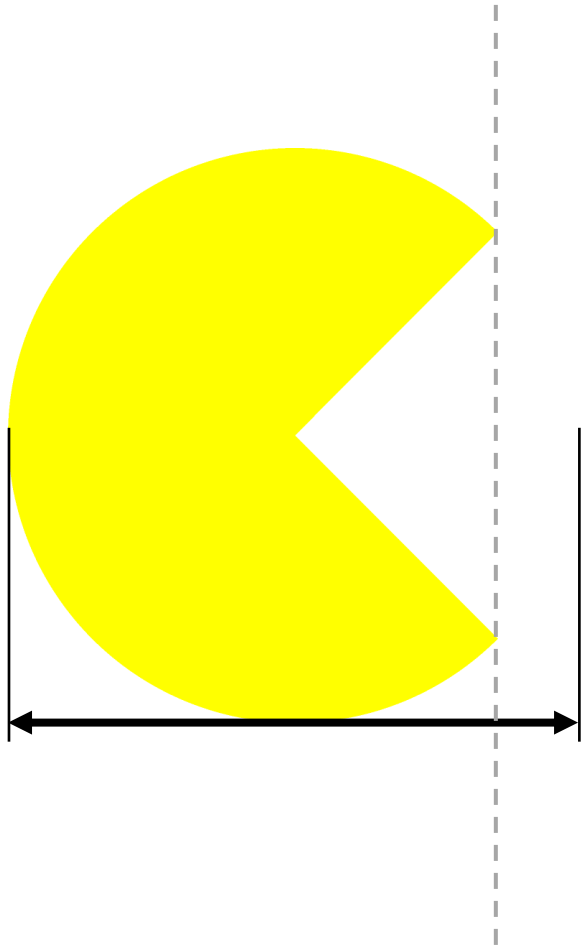
list[Graphic]

Creiamo i fotogrammi e animiamo il nostro Pacman

Fotogramma 1	Fotogramma 2	Fotogramma 3	Fotogramma 4	Fotogramma 5	Fotogramma 6
15°	30°	45°	60°	75°	90°

Abbiamo un problema con la larghezza dei fotogrammi!





Una lista di grafiche (*Dati ripetitivi*)



...

somiglianze?

differenze?

Una lista di grafiche (*Dati ripetitivi*)



Python

```
frames = [
  rolling_eyes_frame(0),
  rolling_eyes_frame(30),
  rolling_eyes_frame(60),
  rolling_eyes_frame(90),
  ...
]
show_animation(frames)
```

list[Graphic]

somiglianze?

differenze?

Una lista di grafiche (*Dati ripetitivi*)



Python

```
frames = [
  rolling_eyes_frame(0),
  rolling_eyes_frame(30),
  rolling_eyes_frame(60),
  rolling_eyes_frame(90),
  ...
]
show_animation(frames)
```

stesso

diverso

list[Graphic]

somiglianze?

differenze?

Refactor: Una lista di numeri

0 30 60 90 120 150 180 ...



...

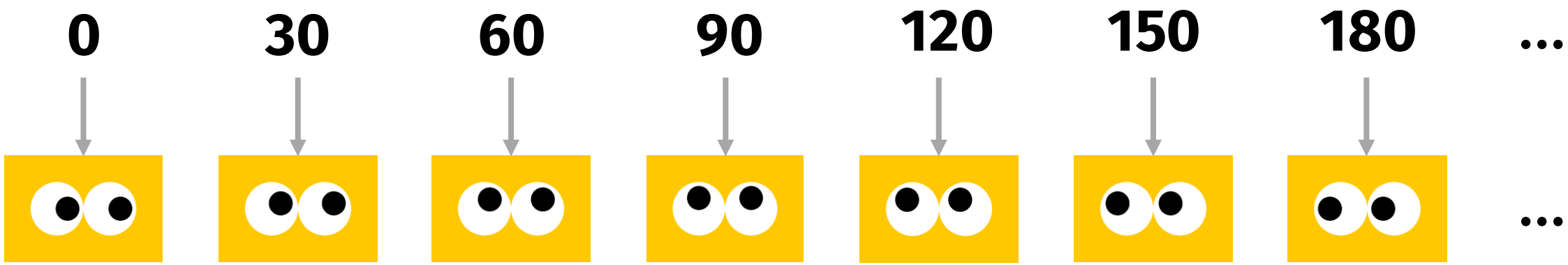
Python

```
angles = [
  0,
  30,
  60,
  90,
  ...
]
```

list[int]

Zero duplicazione di codice: non c'è più nulla di simile da rimuovere

Mappare: Calcolo Ripetitivo



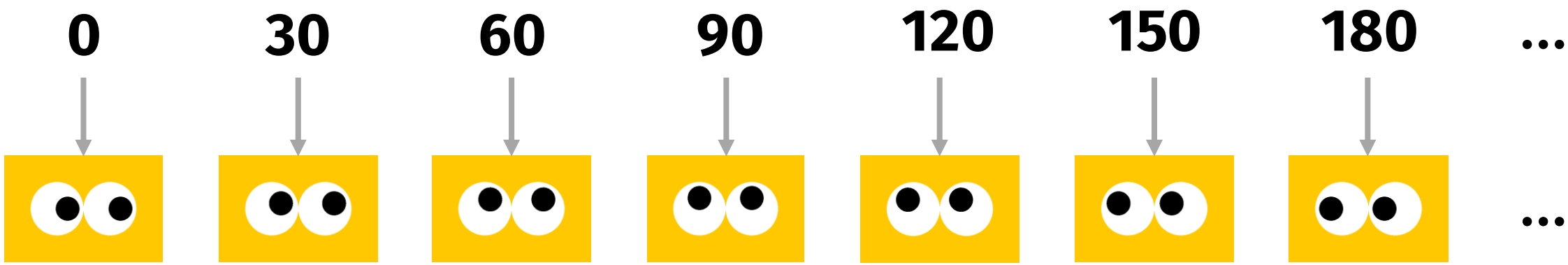
Python

```
angles = [
  0,
  30,
  60,
  90,
  ...
]
frames = [                                    ]
show_animation(frames)
```

Per scrivere [su tastiera italiana:
AltGr + è

mappare
da **angoli**
a **grafiche**

Mappare: Calcolo Ripetitivo



Python

```
angles = [
  0,
  30,
  60,
  90,
  ...
]
frames = [rolling_eyes_frame(angle) for angle in angles]
show_animation(frames)
```

Matematica

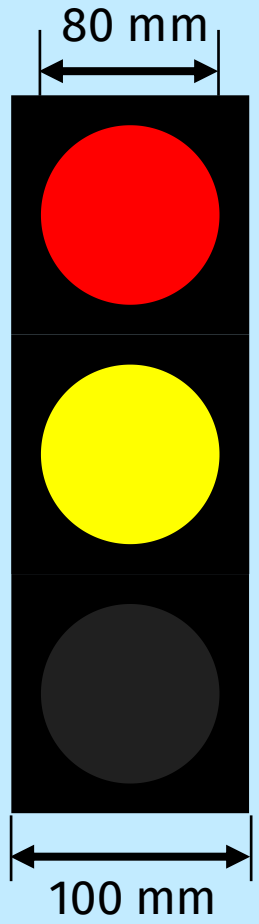
$$\{ f(a) \mid a \square A \}$$

mappare
da **angoli**
a **grafiche**

Componi un semaforo

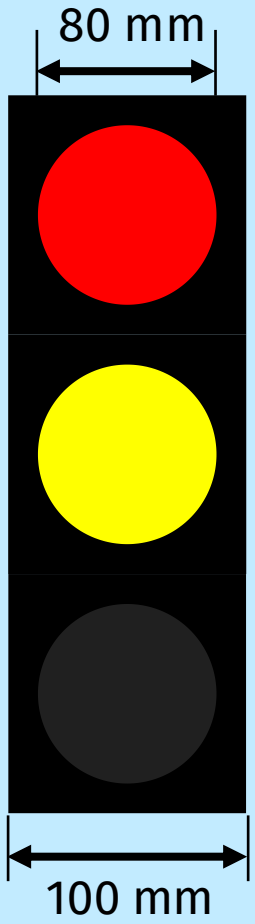


Grafica Desiderata



Motivi? Somiglianze & Differenze?

Desired Graphic

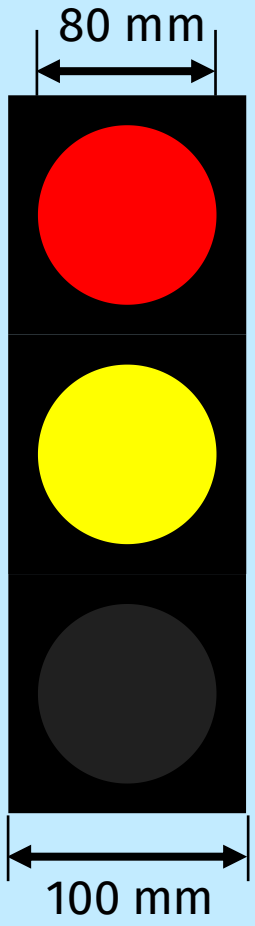


Astrarre in una funzione

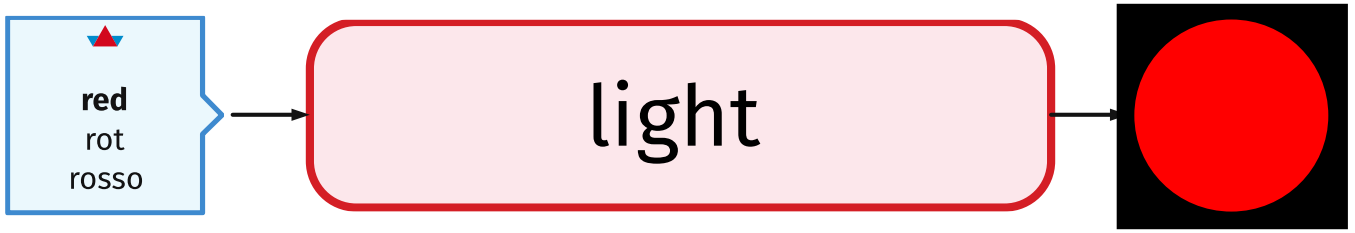
Python

```
def light(color: Color) -> Graphic:
  return ellipse(80, 80, color)
```

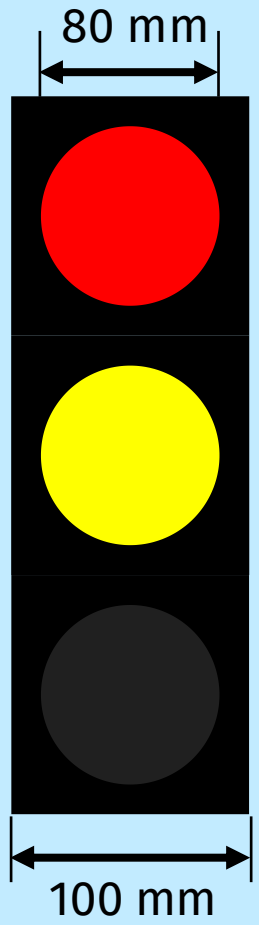
Grafica Desiderata



Chiamare la funzione



Grafica Desiderata

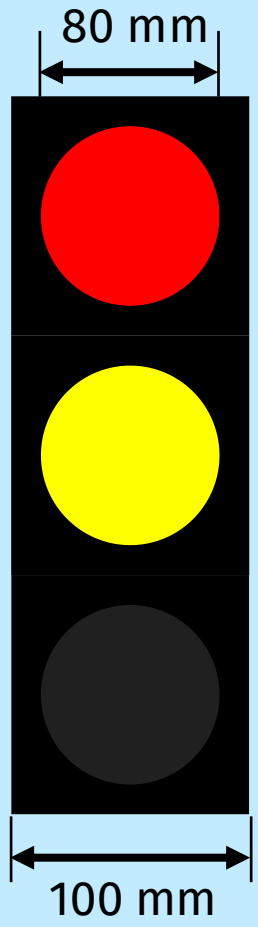


Chiamare la funzione

Python

```
above(
  light(red),
  above(
    light(yellow),
    light(black)
  )
)
```

Grafica Desiderata



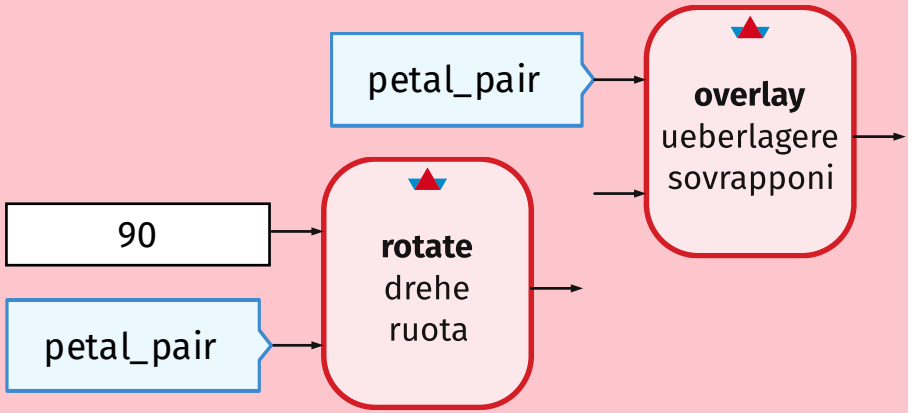
Codice – Carte – Grafica

Interpretare la soluzione

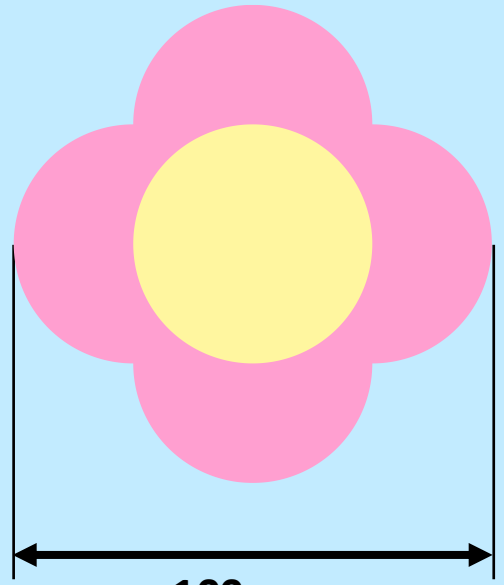
Codice
soluzione

```
overlay(
  petal_pair,
  rotate(
    90,
    petal_pair
  )
)
```

Carte
notional machine



Grafica
problema



Risolvere il problema

Fine!

[ITADINFO]

CONVEGNO ITALIANO
SULLA DIDATTICA DELL'INFORMATICA

Faculty of Informatics

LüCE
Lugano Computing Education
Research Lab

Insegnare a programmare componendo grafiche con PyTamaro

luce.si.usi.ch



Luca
Chiodini



Joey
Bevilacqua



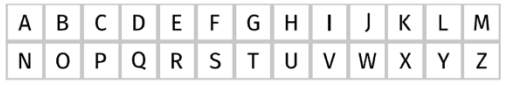
Matthias
Hauswirth

Iscriviti alla nostra mailing list / lascia il tuo feedback!

https://usi.qualtrics.com/jfe/form/SV_0oaTikCNq0DahCe

e un'altra cosa...

Costruisci il tuo modello di linguaggio ("Programmare per Imparare" LLM)



A Trivial Language Model

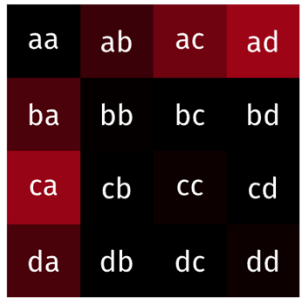


N-Gram

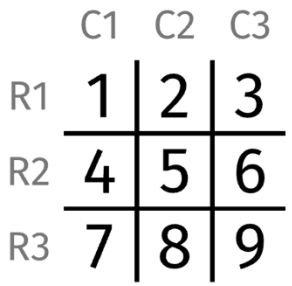


Rossi

Prompt



Pairs of Characters



Tic-Tac-Toe