



Disegnare forme con p5.js

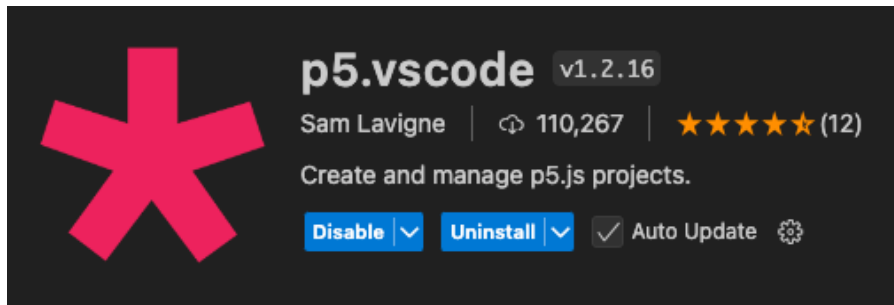
Recap: cos'è p5.js?

- Una libreria JavaScript per il coding creativo
- Creata per artisti, designer, educatori e principianti
- Si concentra sul rendere la programmazione accessibile agli studenti visivi
- Perfetta per creare grafica e animazioni interattive

Hai installato l'editor Vs Code?

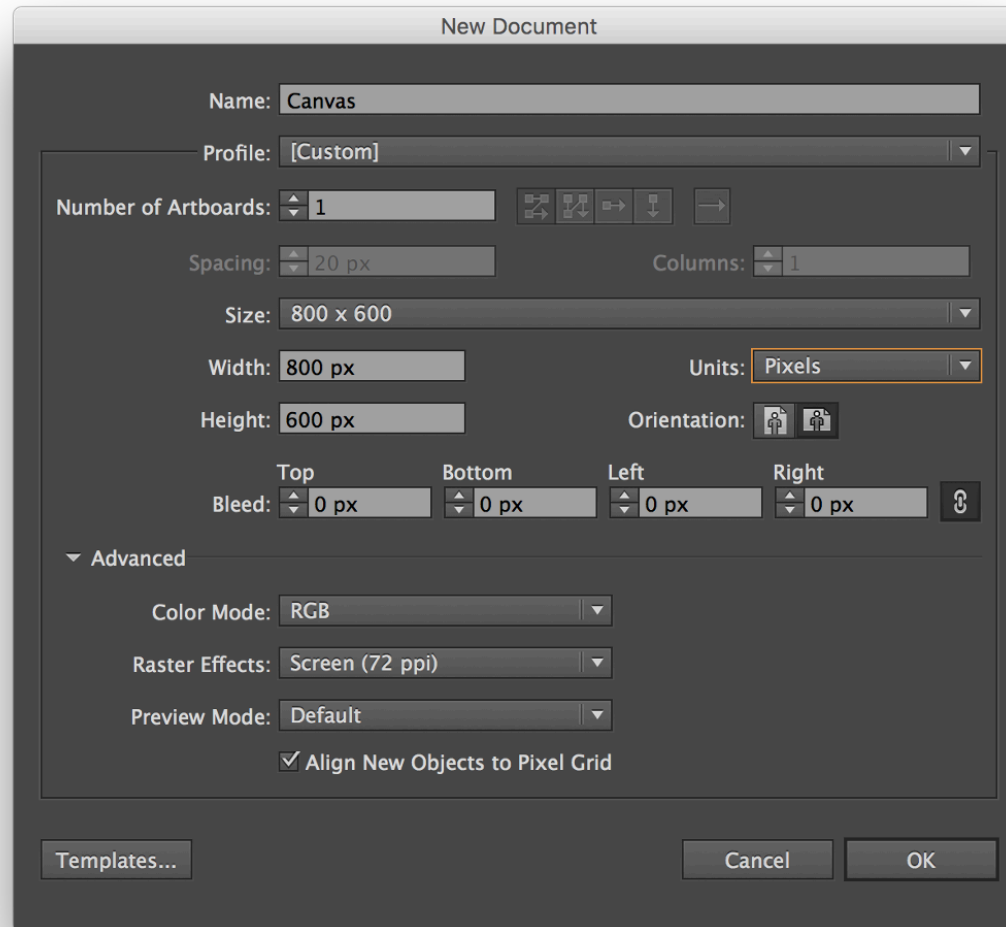
<https://code.visualstudio.com/download>

Plugin p5.js per VS Code





p5*.js



Il Tuo Primo Canvas

```
function setup() {  
  createCanvas(800, 600);  
}  
  
function draw() {  
  // Il tuo codice di disegno va qui  
}
```

- `createCanvas(width, height)` crea il tuo spazio di disegno
- Le dimensioni del canvas sono misurate in pixel
- Il canvas predefinito è 100x100 se non specificato

setup() viene eseguito una volta sola

```
function setup() {  
  createCanvas(800, 600);  
  background(255);  
}
```

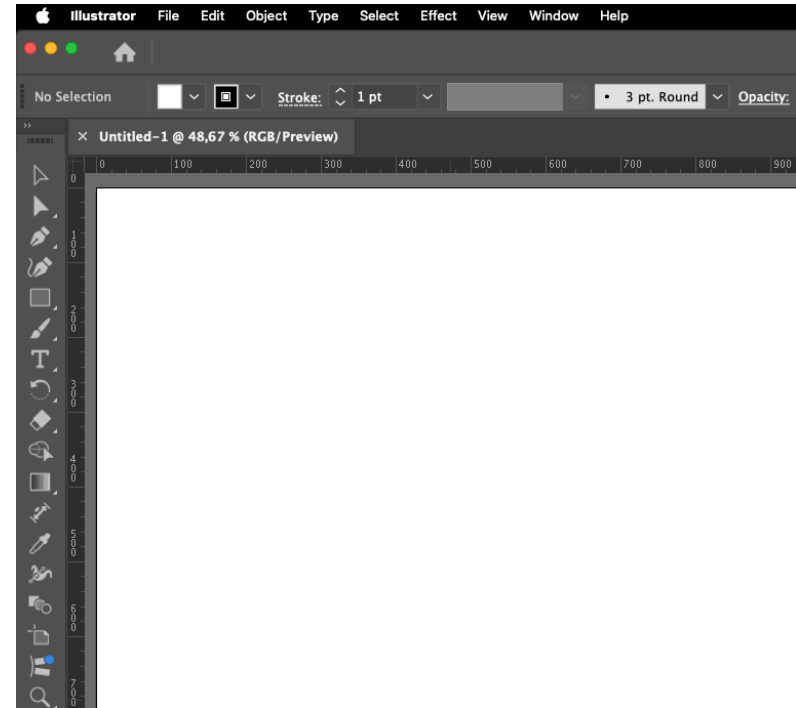
draw() viene eseguito continuamente

```
function draw() {  
  ellipse(mouseX, mouseY, 100, 100);  
}
```


Comprendere il Sistema di Coordinate



- L'origine $(0,0)$ è nell'angolo in alto a sinistra
- X aumenta verso destra
- Y aumenta verso il basso
- La posizione è misurata in pixel



Disegnare Forme Base

Forme Base

Disegna forme con:

- rettangolo
- cerchio
- / linea
- ◆ altri poligoni

Attributi delle forme

Modifica l'aspetto delle forme con:

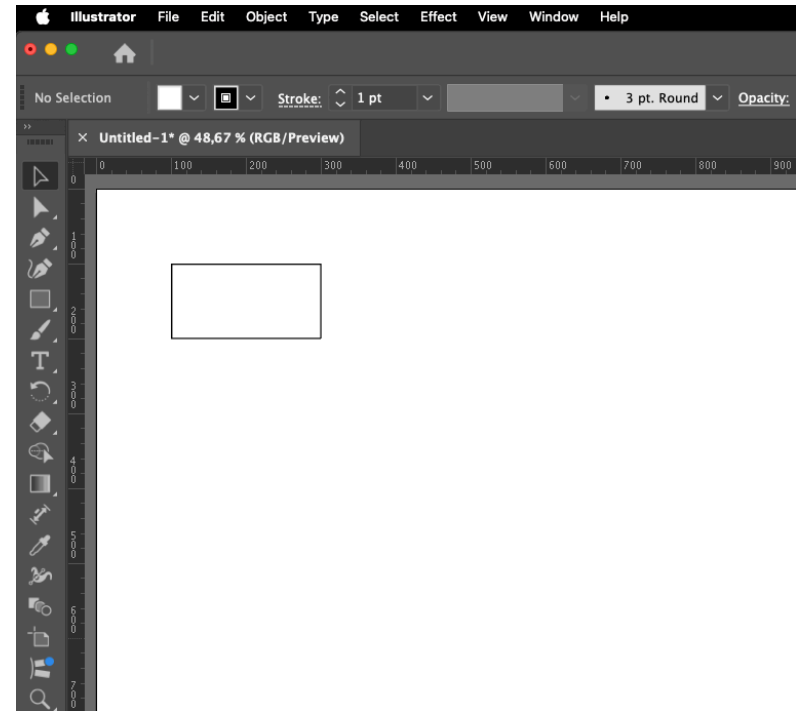
- Stroke (bordo)
- Fill (riempimento)
- StrokeWeight (spessore del bordo)
- Colore

Forme Base: I Rettangolo

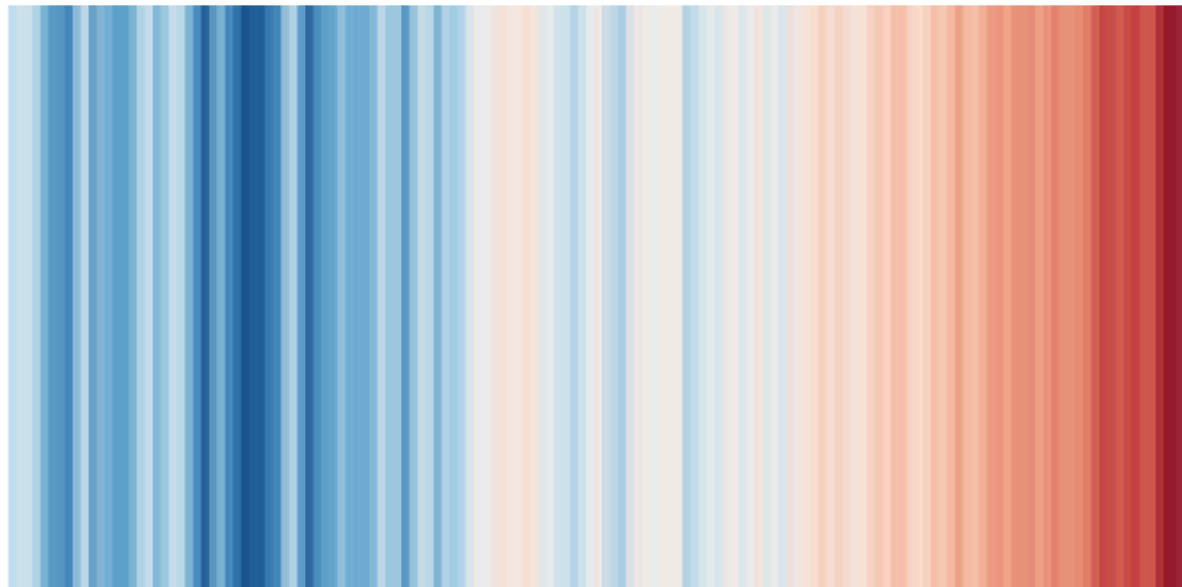
```
// rect(x, y, larghezza, altezza)  
rect(100, 100, 200, 100);
```

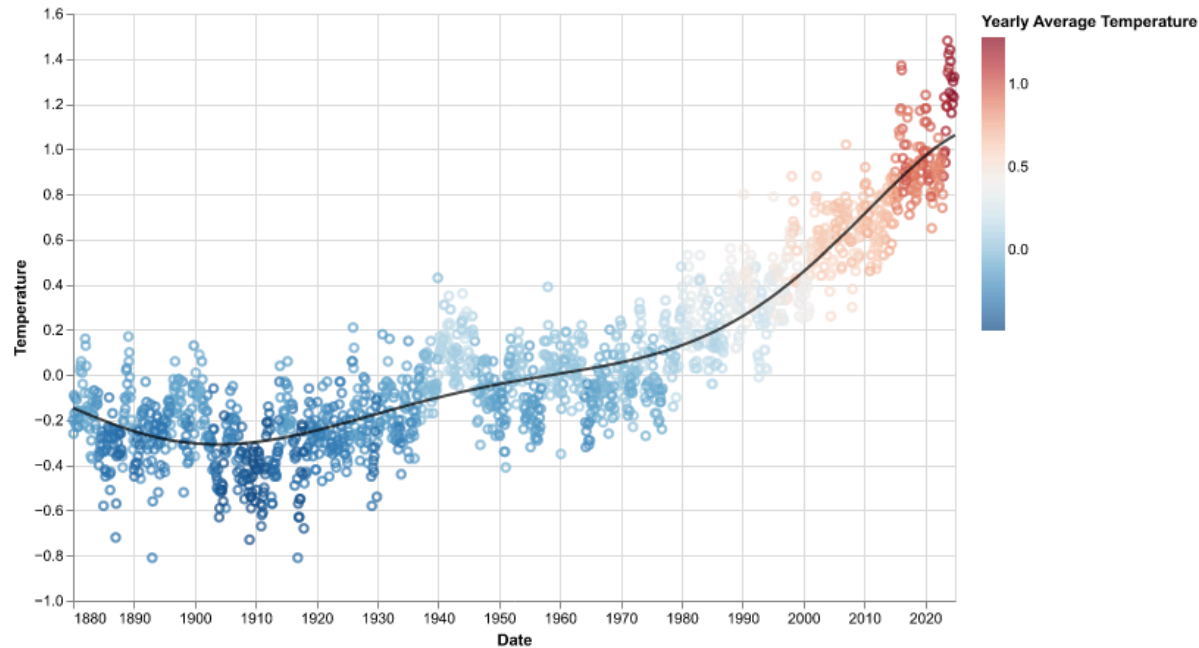
Parametri:

- **x** : posizione orizzontale dell'angolo in alto a sinistra
- **y** : posizione verticale dell'angolo in alto a sinistra
- **larghezza** : larghezza del rettangolo
- **altezza** : altezza del rettangolo



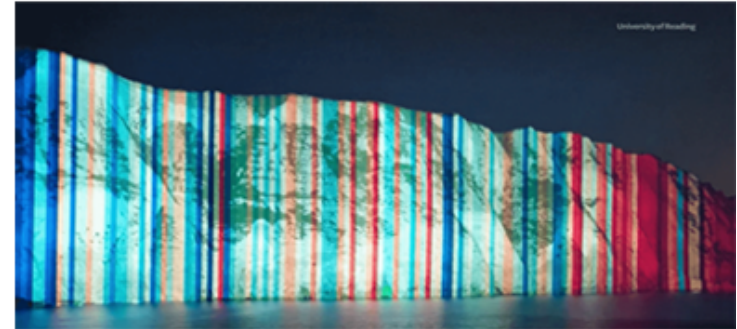
Un esempio di coding utilizzando solo rettangoli colorati: Climate stripes





Ed Hawkins, Università di Reading UK, 2018

https://en.wikipedia.org/wiki/Climate_stripes



In un contesto meno tecnico, le grafiche sono state adottate dagli attivisti per il clima, utilizzate come immagini di copertina di libri e riviste, utilizzate nel design della moda, proiettate su monumenti naturali e utilizzate su uniformi di squadre sportive, palchi di festival musicali e infrastrutture pubbliche.

Modalità del Rettangolo

```
rectMode(CORNER); // Modalità predefinita  
rect(100, 100, 50, 50);  
  
rectMode(CENTER);  
rect(100, 100, 50, 50); // Ora centrato in (100,100)
```

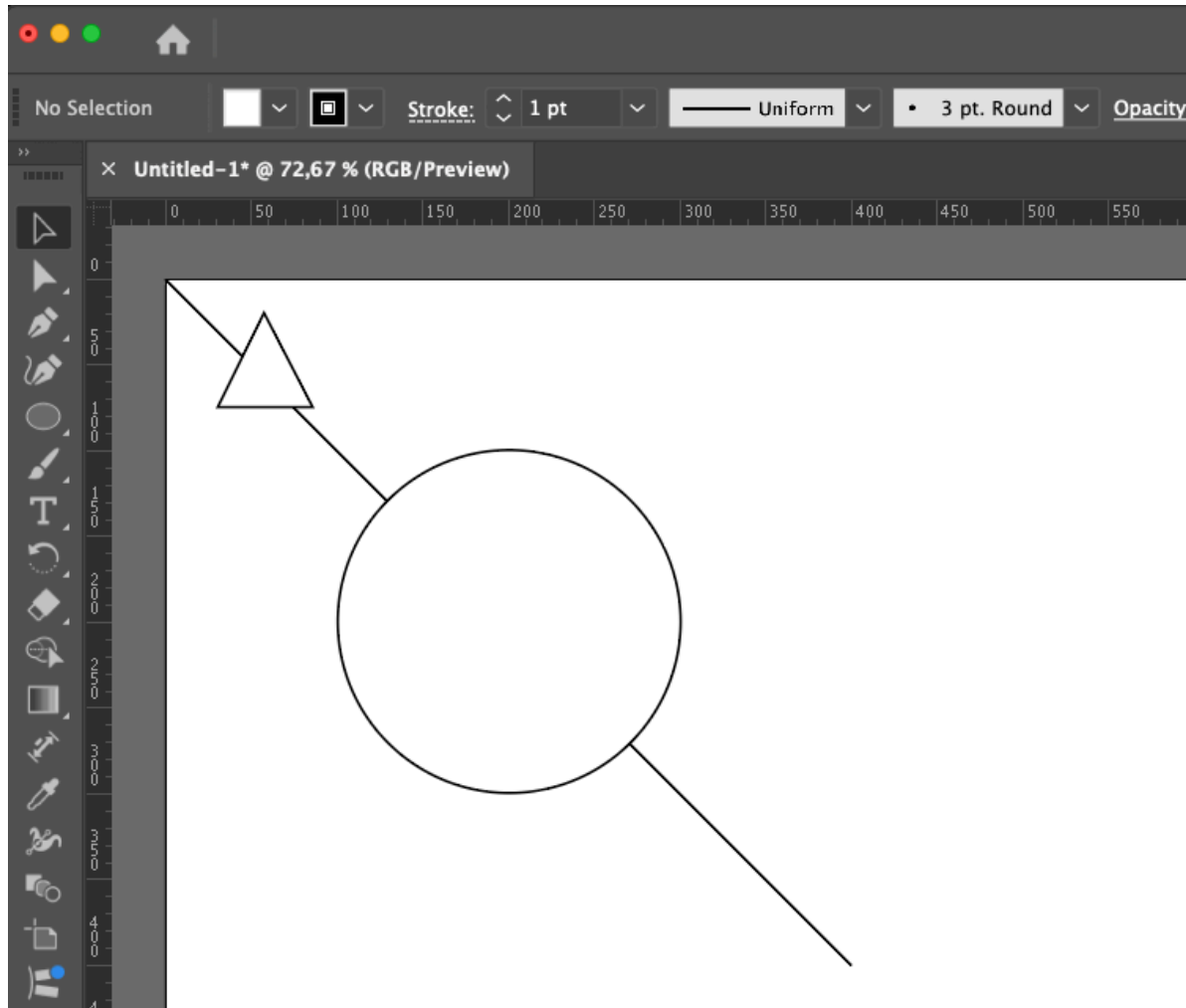
- **CORNER** : La posizione definisce l'angolo in alto a sinistra (predefinito)
- **CENTER** : La posizione definisce il centro del rettangolo
- Cambia come vengono interpretate le coordinate

Altre Forme Base

```
// Linea: line(x1, y1, x2, y2)
line(0, 0, 400, 400); // Linea diagonale

// Cerchio: circle(x, y, diametro)
circle(200, 200, 100);

// Triangolo: triangle(x1, y1, x2, y2, x3, y3)
triangle(30, 75, 58, 20, 86, 75);
```

Ordine di Disegno

- Le forme vengono disegnate nell'ordine in cui appaiono nel codice
- Le forme successive appaiono sopra quelle precedenti
- Pensa a come disporresti dei fogli di carta

```
rect(10, 10, 50, 50); // Disegnato per primo  
line(0, 0, 100, 100); // Disegnato secondo, appare sopra
```

Trovare Aiuto: Documentazione

- Visita p5js.org/reference
- Ogni funzione ha:
 - Descrizione
 - Lista dei parametri
 - Codice di esempio
 - Demo dal vivo
- Ottima risorsa per imparare nuove funzionalità

p5.js ^

Reference
Tutorials
Examples
Contribute
Community
About

</> Start Coding

♥ Donate

Jump to v

Shape
Color
Typography
Image
Transform
Environment
3D
Rendering
Math
IO

Filter by keyword

Looking for p5.sound? Go to the [p5.sound reference!](#)

Shape

2D Primitives

arc() Draws an arc.	circle() Draws a circle.	ellipse() Draws an ellipse (oval).	line() Draws a straight line between two points.
point() Draws a single point in space.	quad() Draws a quadrilateral (four-sided shape).	rect() Draws a rectangle.	square() Draws a square.
triangle() Draws a triangle.			

3D Models

createModel()	loadModel()	model()
----------------------	--------------------	----------------

Looking for the old p5.js site? Find it here! X

<https://p5js.org/reference/>

Buone Pratiche

1. Mantieni il codice ordinato (Usa Tab nell'editor per indentare)
2. Commenta il tuo codice per spiegare cosa fa usando //
3. Salva il tuo lavoro regolarmente
4. Testa le modifiche in modo incrementale
5. Usa la funzione di auto-aggiornamento durante lo sviluppo

Esercizio Iniziale

Crea un disegno semplice usando:

- `rect()`
- `line()`
- `circle()`
- Prova diverse posizioni e dimensioni
- Sperimenta con l'ordine di disegno

Ricorda: Inizia in modo semplice e aumenta gradualmente la complessità!

Prossimi Passi

- Esplorare il colore
- Aggiungere animazioni
- Gestire l'input dell'utente
- Creare interattività
- Costruire forme più complesse

Iniziamo a creare! 🎨

Il Colore in p5.js

Comprendere e Utilizzare i Colori nel Coding Creativo

Il Sistema RGB

Il colore digitale si basa su tre componenti fondamentali:

- **Red** (Rosso)
- **Green** (Verde)
- **Blue** (Blu)

Ogni colore è creato mescolando diverse quantità di questi tre colori base.

Range dei Valori RGB

```
// Formato: background(R, G, B)  
background(220, 0, 200); // Rosa  
background(255, 0, 0);   // Rosso pieno  
background(0, 255, 0);   // Verde pieno  
background(0, 0, 255);   // Blu pieno
```

- Ogni componente va da 0 a 255
- 0 = assenza del colore
- 255 = massima intensità
- Totale: 256 livelli per componente (da 0 a 255)

Scale di Grigi

Un caso speciale: quando R, G e B sono uguali

```
// Questi sono equivalenti:  
background(100, 100, 100); // RGB uguale  
background(100);          // Versione compatta
```

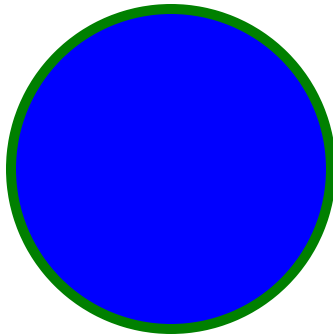
- 0 = nero
- 255 = bianco
- Valori intermedi = tonalità di grigio

Funzioni Principali del Colore

```
// Sfondo  
background(R, G, B);  
  
// Riempimento  
fill(R, G, B);  
  
// Contorno  
stroke(R, G, B);
```

Tutte le funzioni di colore accettano sia valori RGB che singoli valori per il grigio.

Riempimento e Contorno



```
fill(0, 0, 255)
```

```
stroke(0, 255, 0)
```

- `fill()` controlla il colore interno
- `stroke()` controlla il colore del contorno
- L'ordine è importante: vanno impostati prima di disegnare la forma

Rimuovere Riempimento o Contorno

```
// Rimuove il riempimento  
noFill();
```

```
// Rimuove il contorno  
noStroke();
```

Nota: `fill(0)` o `stroke(0)` non rimuovono il colore, ma lo impostano a nero!

Trasparenza (Alpha)

```
// Formato: fill(R, G, B, A)  
fill(255, 0, 0, 127); // Rosso semi-trasparente
```

- Aggiunge un quarto valore: Alpha
- Range: 0-255
- 0 = completamente trasparente
- 255 = completamente opaco

Spessore del Contorno

```
strokeWeight(8); // Contorno spesso 8 pixel  
rect(50, 50, 100, 100);
```

```
strokeWeight(1); // Contorno sottile 1 pixel  
circle(150, 150, 80);
```

`strokeWeight()` controlla lo spessore del contorno in pixel

Riepilogo dei Formati Colore

```
// Scala di grigi  
fill(100);           // Un valore  
  
// RGB  
fill(255, 0, 0);     // Tre valori  
  
// RGB con alpha  
fill(255, 0, 0, 127); // Quattro valori  
  
// Scala di grigi con alpha  
fill(100, 127);      // Due valori
```

Esercizio Pratico

Crea un disegno che utilizzi:

1. Diversi colori RGB
2. Trasparenze
3. Diverse dimensioni di contorno
4. Forme con e senza riempimento/contorno

Sperimenta con le combinazioni di colori! 🎨