



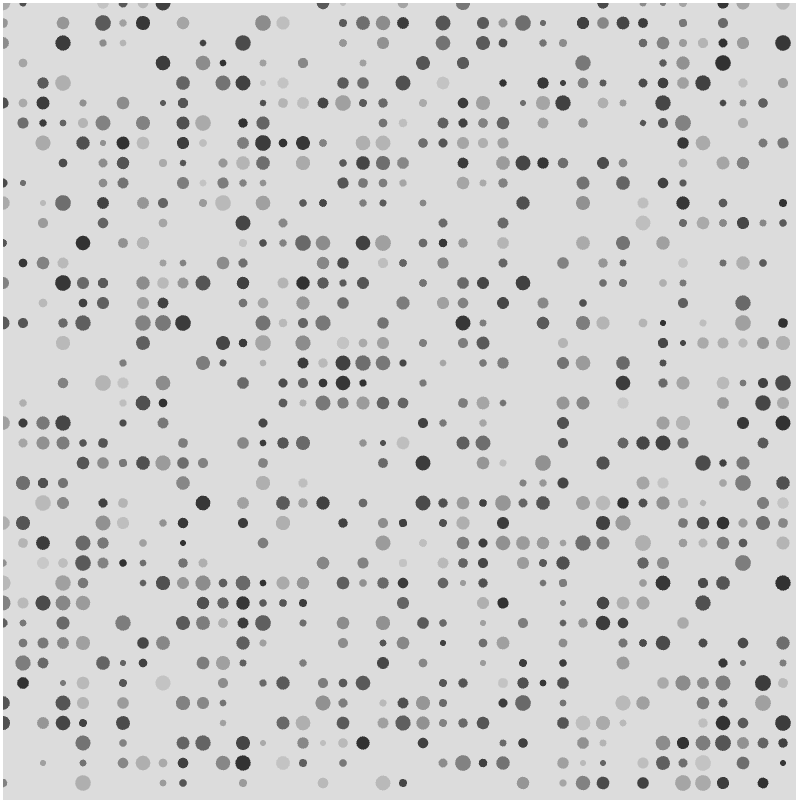
## Esercitazioni con Pattern

# Concetti Fondamentali

Per creare texture interessanti in P5.js utilizzeremo:

- Cicli `for` per ripetizione di pattern
- Istruzioni `if` per variazione
- Funzione `random()` per casualità controllata

## Esempio 1: Pattern di Punti Casuali



## Esempio 1: Pattern di Punti Casuali

- Crea una griglia di punti
- Ogni punto ha 50% di probabilità di apparire
- Dimensione e colore casuali

# Analizziamo il Primo Esempio

## 1. Loop Annidati:

- Loop esterno: controlla la posizione X
- Loop interno: controlla la posizione Y
- `x += 4` e `y += 4` : crea una griglia 4x4 pixel

## 2. Condizione Random:

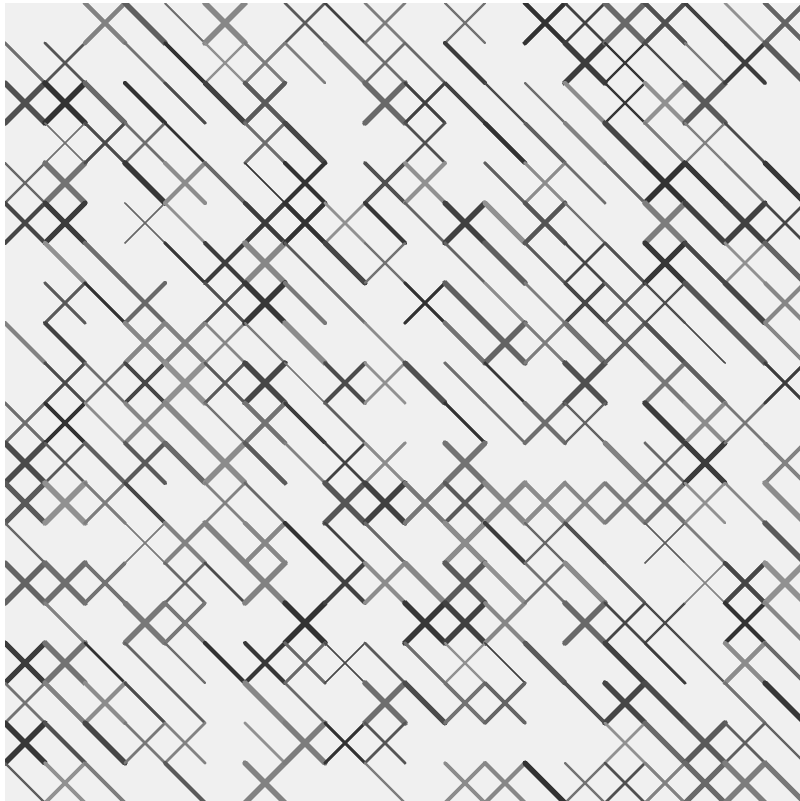
- `if (random() > 0.5)` : 50% probabilità
- Crea un pattern non uniforme

## 3. Variazioni Casuali:

- `random(3, 8)` : dimensione variabile
- `random(50, 200)` : tonalità di grigio variabile

```
function setup() {  
  createCanvas(400, 400);  
  background(220);  
  
  for (let x = 0; x < width; x += 4) {  
    for (let y = 0; y < height; y += 4) {  
      if (random() > 0.5) {  
        let size = random(3, 8);  
        fill(random(50, 200));  
        noStroke();  
        ellipse(x, y, size, size);  
      }  
    }  
  }  
}
```

## Esempio 2: Griglia con Linee Intersecate



# Analizziamo il Secondo Esempio

## 1. Struttura della Griglia:

- Celle di 20x20 pixel
- Maggiore spaziatura per un pattern più aperto

## 2. Linee Diagonali:

- Prima linea: da in alto a sinistra a in basso a destra
- Seconda linea (opzionale): direzione opposta
- Crea intersezioni interessanti

## 3. Variazioni Visive:

- Spessore linee variabile
- Tonalità di grigio variabile
- Probabilità diverse per controllo densità



```
function setup() {  
  createCanvas(400, 400);  
  background(240);  
  
  for (let x = 0; x < width; x += 20) {  
    for (let y = 0; y < height; y += 20) {  
      if (random() < 0.8) {  
        stroke(random(50, 150));  
        let weight = random(1, 3);  
        strokeWeight(weight);  
        line(x, y, x + 20, y + 20);  
  
        if (random() > 0.5) {  
          line(x + 20, y, x, y + 20);  
        }  
      }  
    }  
  }  
}
```

## Esempio 3: TV Anni '80 (non funzionante)



# Analizziamo il Terzo Esempio

## 1. Colore e Trasparenza:

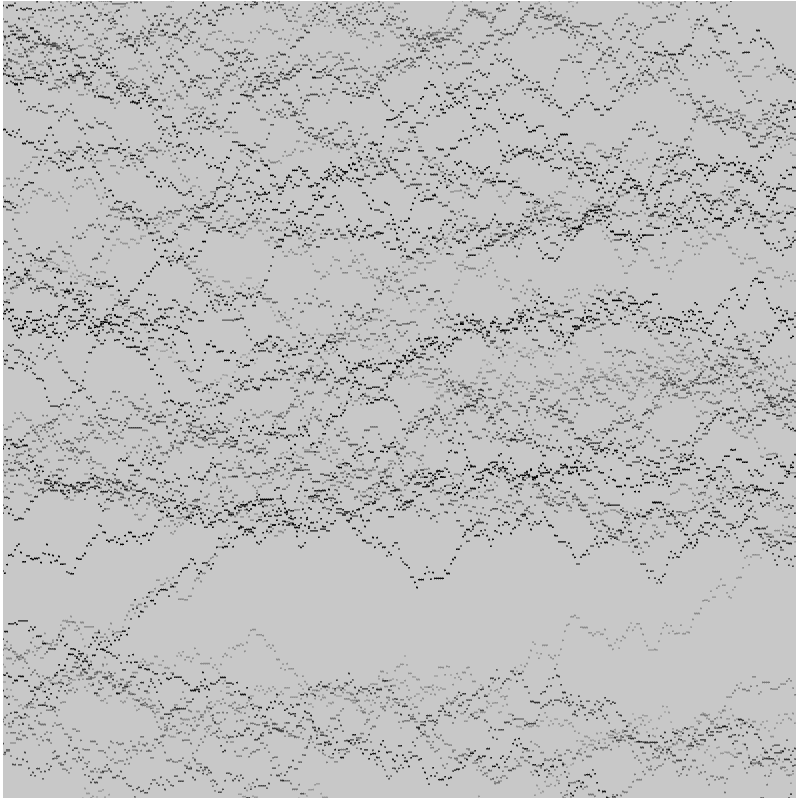
- RGB casuale per ogni quadrato
- Alpha fisso (150) per trasparenza
- Quadrati bianchi per contrasto

## 2. Probabilità di Colore:

- 70% probabilità di colore
- 30% probabilità di bianco

```
function setup() {  
  createCanvas(400, 400);  
  noStroke();  
  
  for (let x = 0; x < width; x += 4) {  
    for (let y = 0; y < height; y += 4) {  
      if (random() > 0.3) {  
        fill(random(255), random(255), random(255), 150);  
        rect(x, y, 4, 4);  
      } else {  
        fill(240);  
        rect(x, y, 4, 4);  
      }  
    }  
  }  
}
```

## Esempio 4: Linee Organiche



# Analizziamo il Quarto Esempio

## 1. Struttura Diversa:

- Loop esterno per numero di linee
- While loop per disegno continuo
- Movimento organico con random

## 2. Caratteristiche Uniche:

- Linee semi-trasparenti
- Movimento verticale casuale
- Effetto "capelli" o "fili"

```
function setup() {  
  createCanvas(400, 400);  
  background(255);  
  stroke(0, 50);  
  
  for (let i = 0; i < 40; i++) {  
    let x = 0;  
    let y = random(height);  
  
    while (x < width) {  
      if (random() > 0.5) {  
        y += random(-5, 5);  
      }  
      point(x, y);  
      x++;  
    }  
  }  
}
```

# Tecniche di Variazione

Per ogni pattern possiamo modificare:

## 1. Dimensioni:

- Spaziatura tra elementi
- Dimensione degli elementi
- Area coperta dal pattern

## 2. Probabilità:

- Frequenza degli elementi
- Probabilità di variazioni
- Densità del pattern

## 3. Colore e Stile:

- Range di colori
- Opacità
- Stili di disegno (stroke/fill)



## Considerazioni generali

- I pattern sono combinazioni di ripetizione e casualità
- Le variazioni creano interesse visivo
- Sperimentare con i parametri porta a risultati unici
- La casualità controllata è la chiave per texture naturali