



Trasformazioni 2D in p5.js

Il Sistema di Coordinate

- Origine (0,0) nell'angolo superiore sinistro
- X aumenta verso destra
- Y aumenta verso il basso
- Le unità sono in pixel
- Gli angoli sono in radianti

Funzione – translate()

```
translate(x, y)
```

- Sposta l'origine del sistema di coordinate
- Influenza tutti i disegni successivi
- È cumulativo con altre trasformazioni
- Non muove effettivamente gli oggetti disegnati, ma...

Cambia il sistema di riferimento!!!

Esempio Base di translate()

```
function draw() {  
  background(220);  
  
  // Disegna un cerchio rosso nell'origine  
  fill(255, 0, 0);  
  ellipse(0, 0, 50, 50);  
  // Sposta l'origine e disegna un cerchio blu  
  translate(100, 100);  
  fill(0, 0, 255);  
  ellipse(0, 0, 50, 50); // stesso cerchio, ma spostato  
}
```

Risultato: Due cerchi, uno nell'origine, uno spostato di 100px in x e y

rotate()

```
rotate(angolo) // angolo in radianti
```

- Ruota il sistema di coordinate
- Il centro di rotazione è l'origine corrente
- Gli angoli sono in radianti (PI)
- È cumulativo con altre trasformazioni

PI (π) e Radianti

- $0^\circ = 0$ radianti
- $90^\circ = \pi/2$ radianti
- $180^\circ = \pi$ radianti
- $270^\circ = 3\pi/2$ radianti
- $360^\circ = 2\pi$ radianti

Il valore di PI è approssimativamente 3.14159

Un radiante corrisponde a un arco lungo quanto il raggio (r) di una circonferenza.

Formula del perimetro di una circonferenza: $p = 2\pi r$

Esempio Base di rotate()

```
function draw() {  
  background(220);  
  
  // Disegna un rettangolo non ruotato  
  rect(0, 0, 100, 50);  
  
  // Ruota di 45 gradi (PI/4 radianti)  
  rotate(PI/4);  
  fill(255, 0, 0);  
  rect(0, 0, 100, 50);  
}
```

push() e pop()

```
push();    // Salva lo stato corrente  
// Applica trasformazioni  
pop();     // Ripristina lo stato salvato
```

- Isola le trasformazioni
- Previene effetti indesiderati
- Fondamentale per trasformazioni complesse

Combinare `translate()` e `rotate()`

```
function draw() {  
  background(220);  
  
  push(); // Salva lo stato  
  
  translate(width/2, height/2); // Sposta l'origine  
  
  rotate(frameCount * 0.02); // Ruota attorno al nuovo centro  
  
  rect(-25, -25, 50, 50); // Disegna un rettangolo  
  
  pop(); // Ripristina lo stato  
}
```

Esempio: Orologio

```
function draw() {  
  background(220);  
  translate(width/2, height/2);  
  // Lancetta delle ore  
  push();  
  rotate(hour() * (TWO_PI/12));  
  line(0, 0, 0, -50);  
  pop();  
  // Lancetta dei minuti  
  push();  
  rotate(minute() * (TWO_PI/60));  
  line(0, 0, 0, -70);  
  pop();  
}
```

Best Practices

1. Ordine delle Trasformazioni

- L'ordine è importante!
- Prima translate, poi rotate
- Usa `push()/pop()` per isolare

2. Origine e Centro

- Sposta l'origine per rotazioni più intuitive
- Considera il punto di riferimento degli oggetti

3. Debug

- Disegna punti di riferimento
- Usa colori diversi per debug
- Commenta le trasformazioni complesse

Pattern Comuni

1. Rotazione Attorno a un Punto

```
translate(x, y); // Sposta l'origine  
rotate(angle);   // Ruota  
translate(-x, -y); // Torna indietro
```

2. Orbita

```
translate(centerX, centerY);  
rotate(angle);  
translate(radius, 0);  
ellipse(0, 0, size, size);
```

Esercizi Pratici

1. Mulino a Vento

- Usa `translate()` per il centro
- `rotate()` per le pale
- `push()/pop()` per ogni pala

2. Sistema Solare

- Pianeti che orbitano
- Lune che orbitano i pianeti
- Trasformazioni annidate