



Istruzioni condizionali: if then else

Come un programma prende decisioni?

Finora abbiamo:

- Memorizzato informazioni in variabili
- Eseguito `setup()` una volta
- Eseguito `draw()` in loop

Ma come possiamo far prendere percorsi diversi al nostro codice?

Boolean Expressions

- Espressioni che risultano in vero o falso (true/false)
- Anche conosciute come 0 (false) o 1 (true)
- Esempi:

```
5 > 6 // false
```

```
7 < 210 // true
```

Sintassi dell'if Statement

```
if (espressione_booleana) {  
    // codice da eseguire se vero  
}
```

- L'espressione booleana va tra parentesi tonde
- Il codice da eseguire va tra parentesi graffe
- Il codice viene indentato per leggibilità
- Il codice viene eseguito solo se l'espressione è vera

mouseX e mouseY

- Variabili globali che contengono la posizione del mouse
- mouseX: posizione orizzontale
- mouseY: posizione verticale

Esempio: Rettangolo che segue il mouse

```
function setup() {  
  createCanvas(400, 400);  
}  
  
function draw() {  
  background(220);  
  
  rect(mouseX, mouseY, 50, 100);  
}
```

Esercizio: Cerchio che Cambia Colore

Creare un canvas di 600 x 300 e disegnare un cerchio agganciato alla posizione del mouse.

Il cerchio dovrà cambiare colore quando il mouse supererà la metà del canvas.

```
function setup() {  
  createCanvas(600, 400);  
}  
function draw() {  
  background(220);  
  // width è una variabile globale di sistema  
  if (mouseX > width / 2) {  
    fill(255, 0, 0); // rosso (RGB)  
  } else {  
    fill(0, 0, 255); // blu (RGB)  
  }  
  ellipse(mouseX, mouseY, 50, 50); // cerchio  
  line(width / 2, 0, width / 2, height); // linea verticale  
}
```


Cerchio Rimbalzante

Creiamo un'animazione di un cerchio che rimbalza sui bordi del canvas.

Variabili Globali

```
let circleX = 100; // posizione X iniziale  
let circleY = 300; // posizione Y iniziale  
let diameter = 50; // diametro del cerchio  
speedX = 2;       // velocità orizzontale  
speedY = 2;       // velocità verticale
```

Setup di Base

- Creiamo un canvas 400x400
- Disegniamo lo sfondo grigio (220)
- Disegniamo il cerchio nella posizione corrente

```
function setup() {  
  createCanvas(400, 400);  
}  
  
function draw() {  
  background(220);  
  circle(circleX, circleY, diameter);  
}
```

Movimento Base

```
function draw() {  
  // ... codice precedente ...  
  circleX = circleX + speedX;  
  circleY = circleY + speedY;  
}
```

Ad ogni frame:

- Aggiorniamo la posizione X aggiungendo speedX
- Aggiorniamo la posizione Y aggiungendo speedY

Rimbalzo sulle Pareti Verticali

```
if (circleX > width-diameter/2) {  
    speedX = -speedX;  
}  
if (circleX < diameter/2) {  
    speedX = -speedX;  
}
```

Se il cerchio tocca il bordo destro o sinistro:

- Invertiamo la direzione ($\text{speedX} = -\text{speedX}$)

Rimbalzo sulle Pareti Orizzontali

```
if (circleY > height-diameter/2 || circleY < diameter/2) {  
    speedY = -speedY;  
}
```

- Se il cerchio tocca il bordo superiore o inferiore: invertiamo la direzione verticale
- Usiamo OR `||` per combinare le due condizioni

Interazione col Mouse - Dimensione

```
function mousePressed() {  
  if (mouseX > width/2) {  
    diameter += 10;  
  } else {  
    diameter -= 10;  
  }  
}
```

Interazione col Mouse – Velocità

```
function mousePressed() {  
  if (mouseY > height/2) {  
    speedX += 1;  
    speedY += 1;  
  } else {  
    speedX -= 1;  
    speedY -= 1;  
  }  
}
```


Concetti Chiave Utilizzati

1. **Variabili** per posizione e movimento
2. **Movimento lineare** con velocità costante
3. **Collision detection** con i bordi
4. **Eventi del mouse** per l'interazione
5. **Conditional statements** per:
 - Rimbalzi
 - Interazione col mouse
 - Cambio colore

Easter Egg: Salvare la gif animata

```
// Salva 5 secondi di gif quando l'utente preme il tasto 's'.  
function keyPressed() {  
  if (key === 's') {  
    saveGif('bouncing_ball', 5);  
  }  
}
```

Estensioni Possibili

Idee per migliorare il progetto

- **Aggiungere gravità** (velocità verticale aumenta col tempo)
- Aggiungere più palline
- Aggiungere collisioni tra palline
- Aggiungere effetti sonori ai rimbalzi

Continua...