



Esercizio: Termometro

Scrivere un programma che disegna un termometro.

Il termometro avrà una linea che si sposta in base alla temperatura (t).
Il colore cambia in base a delle soglie

$t > 30$: hot (rosso)

$20 < t < 30$: warm (giallo)

$10 < t < 20$: cold (blu)

$0 < t < 10$: very cold (azzurro)

$t < 0$: freezing (bianco)



Termometro Interattivo in p5.js

Un'analisi dettagliata del codice



Obiettivo dell'esercizio

Creare un termometro interattivo che:

- Visualizza temperature da -10°C a 40°C
- Cambia colore in base alla temperatura
- Risponde al click del mouse
- Include indicatori visivi e emoji



Struttura Base del Codice

```
let temperature = 0;

function setup() {
  createCanvas(400, 400);
}

function draw() {
  // Codice per il disegno
}

function mousePressed() {
  // Gestione interazione
}
```



Mappatura della Temperatura

```
let mappedH = map(temperature, -10, 40, 350, 50);
```

- Input: temperatura da -10 a 40
- Output: posizione verticale da 350px a 50px
- Permette una visualizzazione proporzionale



Disegno del Contenitore

```
rectMode(CENTER)  
noStroke()  
fill(180)  
rect(width/2, height/2, 38, 355, 10);
```

Il contenitore grigio del termometro



Gestione dei Colori

```
if (temperature > 30) {  
    stroke(255, 0, 0);    // Rosso - Molto caldo  
} else if (temperature > 20) {  
    stroke(255, 255, 0);  // Giallo - Caldo  
} else if (temperature > 10) {  
    stroke(0, 0, 255);    // Blu - Freddo  
} else if (temperature > 0) {  
    stroke(0, 255, 255);  // Azzurro - Molto freddo  
} else {  
    stroke(255, 255, 255); // Bianco - Gelido  
}
```



Come riportiamo la posizione delle linee sull'asse y?

- Prendiamo la calcolatrice
- Calcoliamo la posizione verticale in base alla temperatura per ogni linea

Oppure... usiamo la funzione `map()`



→ documentazione `map()`.



HIGHLIGHT: La Funzione `map()` in p5.js

Trasformare i numeri con eleganza



Cos'è la funzione map()?

La funzione map() è come un traduttore di numeri. Prende un numero da un intervallo e lo trasforma nel suo equivalente in un altro intervallo.

```
map(value, start1, stop1, start2, stop2)
```

È come dire: "Se questo numero si trova al 25% del primo intervallo, trovami il numero che si trova al 25% del secondo intervallo"



Un esempio della vita reale

Pensate a una scala di valutazione scolastica:

Alle superiori usiamo i voti da 0 a 10, in università usiamo i voti da 0 a 30.

Se volessimo convertire un 7:

```
votoMio = 7;  
rangeSuperioriStart = 0;  
rangeSuperioriStop = 10;  
rangeUniStart = 0;  
rangeUniStop = 30;  
  
votoPreso = map(votoMio, rangeSuperioriStart, rangeSuperioriStop, rangeUniStart, rangeUniStop);
```

Il risultato sarà 21, perché 7 è il 70% di 10 e 21 è il 70% di 30.



La matematica dietro map()

Definizione: `map(value, start1, stop1, start2, stop2)`

Quando scriviamo: `map(value, 0, 10, 0, 30)` significa:

- Se il valore è 0, restituisci 0
- Se il valore è 10, restituisci 30
- Se il valore è 5, restituisci 15

`map()` calcola la proporzione tra i due intervalli e applica la stessa proporzione all'input, in linguaggio algebrico sarebbe:

$$7 : 10 = x : 30$$

$$x = 7 * 30 / 10$$

$$x = 21$$



La differenza chiave

La funzione `map()` è più flessibile perché gestisce gli offset:

```
let risultato1 = (5 * 100) / 10; // Da 0-10 a 0-100  
let risultato2 = map(5, 0, 10, 0, 100); // Identico risultato
```

N.B. entrambi partono da 0, ma `map()` può gestire offset diversi!



Analizziamo la formula di `map()`

La formula completa di `map()` è:

```
formulaCompletaDiMap = (value - start1) * (stop2 - start2) / (stop1 - start1) + start2
```

Quando `start1` e `start2` sono zero, si riduce a:

```
Proporzione = value * stop2 / stop1
```

Come una proporzione semplice!



Un esempio pratico: Il termometro

Nel nostro termometro:

la proporzione semplice NON funzionerebbe correttamente:

```
let posizione = (temperatura * 50) / 40;
```

Con map (funziona correttamente):

```
let posizione = map(temperatura, -10, 40, 350, 50);
```




La proporzione non può gestire:

- Temperature negative
- Il punto di partenza a 350px
- La direzione invertita (da 350 a 50)

Casi d'uso tipici - Proporzione semplice:

- Scalare dimensioni proporzionalmente
- Calcolare percentuali
- Conversioni lineari da zero

Casi d'uso tipici - `map()`:

- Gestire input con range negativi
- Convertire tra scale con offset (che non partono da zero)
- Invertire range di valori



Nel nostro termometro `map()` ci permette di:

- Gestire temperature negative (-10°C)
- Iniziare il disegno dal basso (350px)
- Muoverci verso l'alto (fino a 50px)
- Mantenere le proporzioni corrette

Tutto questo sarebbe molto più complesso con una proporzione semplice!



Linee Graduate

```
// Linee graduate  
stroke(0);  
strokeWeight(1)  
line(width/2 - 20, map(-10, -10, 40, 350, 50),  
      width/2 + 20, map(-10, -10, 40, 350, 50));
```



Interattività

```
function mousePressed() {  
  temperature = random(-10, 40);  
  console.log(temperature);  
}
```

- Click del mouse genera una temperatura casuale
- Range: da -10°C a 40°C
- Aggiornamento immediato della visualizzazione



Highlight: funzione random()

La funzione `random()` genera un numero casuale tra due valori.

```
let numeroCasuale = random(min, max);
```

- `min` : il valore minimo (incluso)
- `max` : il valore massimo (escluso)

N.B. Se passo un solo valore a `random()`, il valore minimo è 0.
-> documentazione Random().



Label della Temperatura

Il testo è posizionato utilizzando la funzione `text()` che accetta tre parametri:

```
text('testo da scrivere', posizioneX, posizioneY);
```

nel nostro caso:

```
// Testo con emoji  
text('😬 40', width/2 + 30, map(40, -10, 40, 350, 50));  
text('❄️ 0', width/2 + 30, map(0, -10, 40, 350, 50));
```

N.B. è possibile utilizzare emoji come testo!



Dettagli Implementativi Importanti

1. Sistema di coordinate centrato
2. Uso di `map()` per conversione valori
3. Gestione dello stroke per il "mercurio"
4. Aggiunta di elementi testuali
5. Feedback visivo con emoji



Grazie per l'attenzione!

Domande?