



# Esercitazione 2 (09-30)



## 2.1 Funzione, Fahrenheit

- Definire una funzione `cels_to_fahr`
  - Parametro: temperatura in gradi Celsius, in `float`
  - Risultato: temperatura in gradi Fahrenheit, in `float`
- Invocare la funzione dalla shell interattiva
- Definire poi una funzione `main`
  - *Procedura, senza parametri e senza risultato*
  - Chiedere all'utente la temperatura in gradi Celsius
  - Poi chiamare `cels_to_fahr` con questo parametro
  - Infine mostrare all'utente il risultato

*Partire dalla formula*  $fahr = cels * 1.8 + 32$



## 2.2 Box

- Classe che modella un riquadro
  - Due campi privati, per larghezza ed altezza: `_width`, `_height`
  - Due metodi, per ottenere area e perimetro: `area`, `perimeter`
- Nel corpo principale del programma, creare e usare un riquadro
  - Chiedere all'utente le dimensioni del riquadro
  - Passare queste dimensioni al costruttore, come parametri
  - Mostrare area e perimetro del riquadro creato



## 2.3 Triangolo di cifre

- Leggere un numero intero positivo  $n$  ( $< 10$ )
- Per ciascun valore  $y$  tra 1 ed  $n$ ...
- Stampare una riga con le cifre da 1 ad  $y$

```
1
12
123
1234
```

OUTPUT

*Usare due cicli for annidati*

*All'inizio non considerare  $n$ , ma fissare  $y$  e scrivere una sola riga:  $y = 3 \rightarrow "123"$*

*Poi racchiudere tutto in un ciclo for esterno*



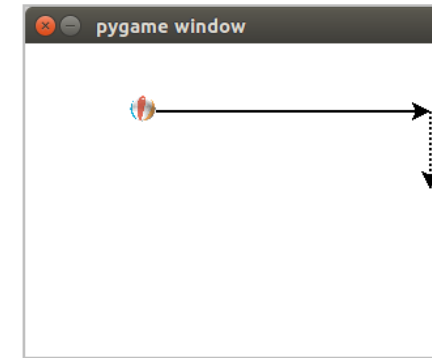
## 2.4 Triangolo di cerchi colorati

- Chiedere all'utente un numero **n**
- Generare **n** righe di cerchi, allineate a sinistra
- Il numero di riga corrisponde al numero di cerchi da disegnare
- Ogni cerchio ha un colore casuale



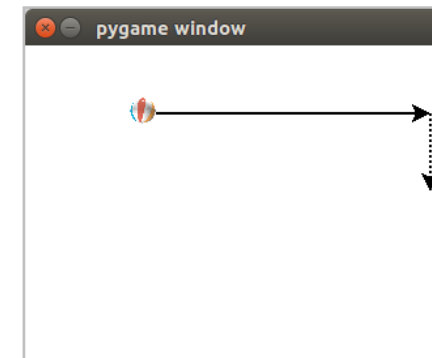
## 2.5 Animazione pallina

- Partire dall'esempio dell'animazione orizzontale, senza classi
  - Quando la pallina arriva ad uno dei 4 bordi...
  - Cambia direzione di 90°, in senso orario
  - Es. Arrivata al bordo destro, vira verso il basso
- Mostrare l'animazione di una pallina su un canvas



## 2.6 Pallina con svolta

- Creare una classe `TurningBall`
  - A partire dalla classe `Ball` vista a lezione
- Modificare il metodo `move`
  - Quando la pallina arriva ad uno dei 4 bordi...
  - Vira la direzione di 90°, in senso orario
  - Es. Arrivata al bordo destro, vira verso il basso



## 2.7 Rimbalzi con gravità

- Creare una classe `FallingBall`
  - A partire dalla classe `Ball` vista a lezione
- Nel metodo `move`, aggiungere l'effetto della gravità
  - Accelerazione: aggiungere un piccolo valore *costante* alla velocità verticale
  - $g = 0.4$
  - $dy += g$
  - Quando si tocca il fondo, cambiare il segno di  $dy$
- Mostrare l'animazione di una pallina, con l'effetto della gravità





## 2.8 Palline colorate

- Aggiungere ai dati della pallina (es. 2.7) anche un colore
  - Generare una tupla **RGB** casuale, nell'inizializzatore
- Aggiungere un corrispondente metodo *getter* denominato **color**
  - Per ottenere dall'esterno il colore della pallina
  - Restituire il colore come una tupla (tripla RGB)
- Creare due palline e mostrarne graficamente il movimento
  - Per ogni frame, chiamare il metodo **move** di ogni pallina
  - Rappresentare un rettangolo o un cerchio colorato nella posizione aggiornata di ogni pallina



## 2.9 Albero di cerchi colorati

- Chiedere all'utente un numero  $n$
- Generare  $n$  righe di cerchi, allineate al centro
- Il numero di riga corrisponde al numero di cerchi da disegnare
- Ogni cerchio ha un colore casuale
- Aggiungere in fondo una riga con un solo cerchio

