

# Esempi ed esercizi

Facciamo un po' di pratica



# Esempi



Esempi ed esercizi - Facciamo un po' di pratica

# Confronto tra stringhe

- ▶ Confronto tra stringhe, in ordine **lessicografico**
  - ▶ <, <=, >, >=, ==, !=
  - ▶ Confronto deciso dal primo carattere diverso
- ▶ Primi 128 codici Unicode == **ASCII**
- ▶ Prima le cifre, poi le maiuscole, infine le minuscole

```
>>> "art" < "arc"  
False  
>>> "first" < "second"  
True  
>>> "Second" < "first"  
True
```



# Tabella ASCII

| Decimal | Hex | Char                   | Decimal | Hex | Char    | Decimal | Hex | Char | Decimal | Hex | Char  |
|---------|-----|------------------------|---------|-----|---------|---------|-----|------|---------|-----|-------|
| 0       | 0   | [NULL]                 | 32      | 20  | [SPACE] | 64      | 40  | @    | 96      | 60  | `     |
| 1       | 1   | [START OF HEADING]     | 33      | 21  | !       | 65      | 41  | A    | 97      | 61  | a     |
| 2       | 2   | [START OF TEXT]        | 34      | 22  | "       | 66      | 42  | B    | 98      | 62  | b     |
| 3       | 3   | [END OF TEXT]          | 35      | 23  | #       | 67      | 43  | C    | 99      | 63  | c     |
| 4       | 4   | [END OF TRANSMISSION]  | 36      | 24  | \$      | 68      | 44  | D    | 100     | 64  | d     |
| 5       | 5   | [ENQUIRY]              | 37      | 25  | %       | 69      | 45  | E    | 101     | 65  | e     |
| 6       | 6   | [ACKNOWLEDGE]          | 38      | 26  | &       | 70      | 46  | F    | 102     | 66  | f     |
| 7       | 7   | [BELL]                 | 39      | 27  | '       | 71      | 47  | G    | 103     | 67  | g     |
| 8       | 8   | [BACKSPACE]            | 40      | 28  | (       | 72      | 48  | H    | 104     | 68  | h     |
| 9       | 9   | [HORIZONTAL TAB]       | 41      | 29  | )       | 73      | 49  | I    | 105     | 69  | i     |
| 10      | A   | [LINE FEED]            | 42      | 2A  | *       | 74      | 4A  | J    | 106     | 6A  | j     |
| 11      | B   | [VERTICAL TAB]         | 43      | 2B  | +       | 75      | 4B  | K    | 107     | 6B  | k     |
| 12      | C   | [FORM FEED]            | 44      | 2C  | ,       | 76      | 4C  | L    | 108     | 6C  | l     |
| 13      | D   | [CARRIAGE RETURN]      | 45      | 2D  | -       | 77      | 4D  | M    | 109     | 6D  | m     |
| 14      | E   | [SHIFT OUT]            | 46      | 2E  | .       | 78      | 4E  | N    | 110     | 6E  | n     |
| 15      | F   | [SHIFT IN]             | 47      | 2F  | /       | 79      | 4F  | O    | 111     | 6F  | o     |
| 16      | 10  | [DATA LINK ESCAPE]     | 48      | 30  | 0       | 80      | 50  | P    | 112     | 70  | p     |
| 17      | 11  | [DEVICE CONTROL 1]     | 49      | 31  | 1       | 81      | 51  | Q    | 113     | 71  | q     |
| 18      | 12  | [DEVICE CONTROL 2]     | 50      | 32  | 2       | 82      | 52  | R    | 114     | 72  | r     |
| 19      | 13  | [DEVICE CONTROL 3]     | 51      | 33  | 3       | 83      | 53  | S    | 115     | 73  | s     |
| 20      | 14  | [DEVICE CONTROL 4]     | 52      | 34  | 4       | 84      | 54  | T    | 116     | 74  | t     |
| 21      | 15  | [NEGATIVE ACKNOWLEDGE] | 53      | 35  | 5       | 85      | 55  | U    | 117     | 75  | u     |
| 22      | 16  | [SYNCHRONOUS IDLE]     | 54      | 36  | 6       | 86      | 56  | V    | 118     | 76  | v     |
| 23      | 17  | [ENG OF TRANS. BLOCK]  | 55      | 37  | 7       | 87      | 57  | W    | 119     | 77  | w     |
| 24      | 18  | [CANCEL]               | 56      | 38  | 8       | 88      | 58  | X    | 120     | 78  | x     |
| 25      | 19  | [END OF MEDIUM]        | 57      | 39  | 9       | 89      | 59  | Y    | 121     | 79  | y     |
| 26      | 1A  | [SUBSTITUTE]           | 58      | 3A  | :       | 90      | 5A  | Z    | 122     | 7A  | z     |
| 27      | 1B  | [ESCAPE]               | 59      | 3B  | ;       | 91      | 5B  | [    | 123     | 7B  | {     |
| 28      | 1C  | [FILE SEPARATOR]       | 60      | 3C  | <       | 92      | 5C  | \    | 124     | 7C  |       |
| 29      | 1D  | [GROUP SEPARATOR]      | 61      | 3D  | =       | 93      | 5D  | ]    | 125     | 7D  | }     |
| 30      | 1E  | [RECORD SEPARATOR]     | 62      | 3E  | >       | 94      | 5E  | ^    | 126     | 7E  | ~     |
| 31      | 1F  | [UNIT SEPARATOR]       | 63      | 3F  | ?       | 95      | 5F  | _    | 127     | 7F  | [DEL] |

```
>>> chr(65)
'A'
>>> ord('A')
65
```



Esempi ed esercizi - Facciamo un po' di pratica

# Ordiniamo le parole

```
prima = input("Inserisci una parola: ")
seconda = input("Inserisci un'altra parola: ")

if prima < seconda:
    print("Le parole che hai inserito sono ordinate!")
elif prima > seconda:
    print("Le parole che hai inserito sono in ordine inverso!")
else:
    print("Le due parole sono uguali")
```

# Elenchi

- ▶ Possiamo creare elenchi di stringhe...
- ▶ ... e poi chiederci se una stringa è nell'elenco

```
segni_zodiacali = [  
    "ariete",  
    "toro",  
    "gemelli",  
    "cancro",  
    "leone",  
    "verGINE",  
    "bilancia",  
    "scorpione",  
    "sagittario",  
    "capricorno",  
    "acquario",  
    "pesci"  
]
```

```
prima = input("Di che segno sei? ")  
  
if prima in segni_zodiacali:  
    ...
```

# Maiuscole o minuscole?

- ▶ Abbiamo notato che per Python "Toro" != "toro"
- ▶ Quindi rischiamo di non riconoscere una risposta corretta!
- ▶ Niente paura, possiamo rendere tutto minuscolo
- ▶ Oppure tutto maiuscolo
- ▶ Attenzione! Il valore della variabile `saluto` resta uguale!

```
>>> "Hello, world!".lower()
'hello, world!'
>>> "Hello, world!".upper()
'HELLO, WORLD!'
```

```
>>> saluto = "Hello, world!"
>>> saluto.upper()
'HELLO, WORLD!'
>>> saluto
'Hello, world!'
```



# Attenzione a salvare le modifiche

- ▶ Le funzioni `lower` e `upper` ci forniscono la stringa tutta minuscola e tutta maiuscola
- ▶ Ma il contenuto della variabile resta uguale!
- ▶ Risolviamo con l'assegnamento:

```
prima = input("Di che segno sei? ")  
prima = prima.lower()
```

- ▶ Oppure:

```
prima = prima.upper()
```



# La parola di Quello

```
print("Pronto?!")

prima = input("Di che segno sei? ")
prima = prima.lower()

if prima in segni_zodiacali:

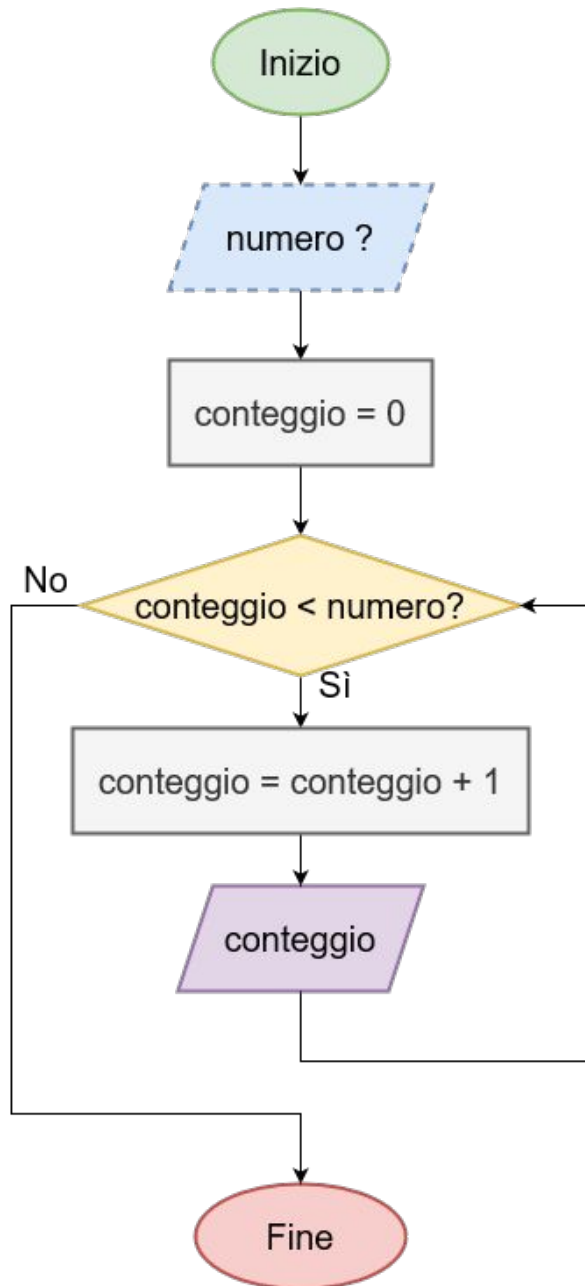
    seconda = input("Sennò? ")
    seconda = seconda.lower()

    if seconda in segni_zodiacali:
        print("La seconda che hai detto.")
    else:
        print("Tu come la vedi?")

else:
    print("La domanda è mal posta.")
```

Provate a disegnare il diagramma di flusso di questo programma!

# Contare, fare la somma, fare la media...



```
numero = input("Dimmi un numero: ")
numero = int(numero)

conteggio = 0

while conteggio < numero:
    conteggio = conteggio + 1
    print(conteggio)
```

# Contare, fare la somma, fare la media...

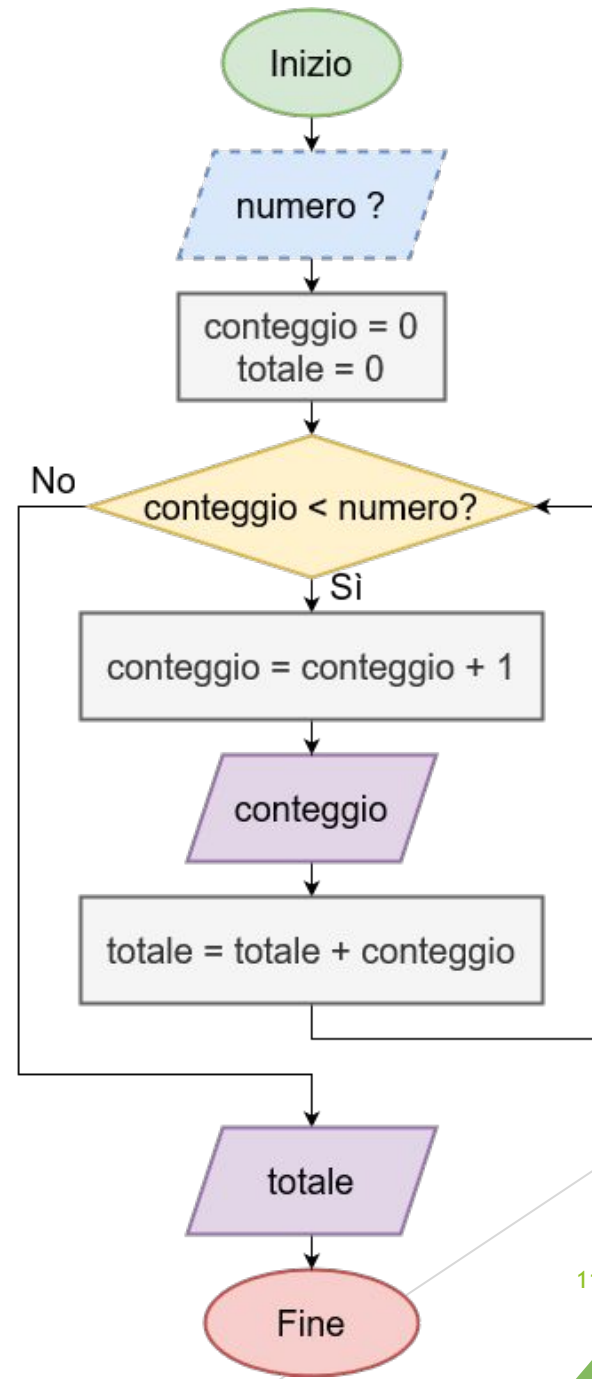
```
numero = input("Dimmi un numero: ")
numero = int(numero)

conteggio = 0
totale = 0

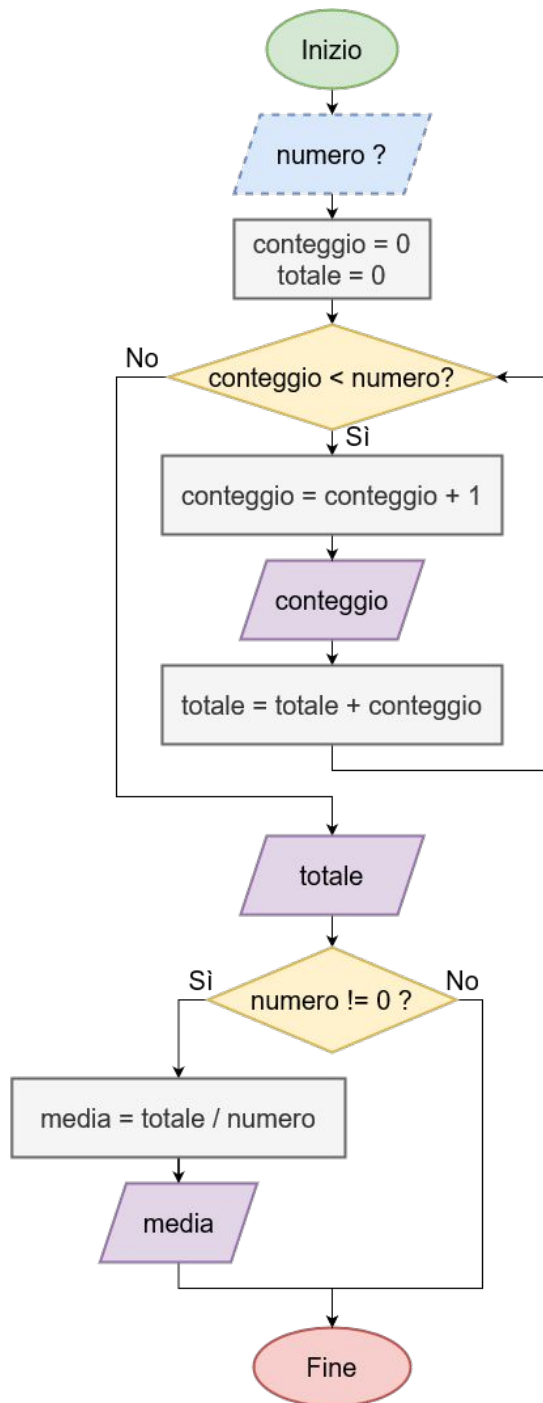
while conteggio < numero:
    conteggio = conteggio + 1
    print(conteggio)

    totale = totale + conteggio

print("La somma dei primi", numero, "numeri è", totale)
```



# Contare, fare la somma, fare la media...



```
numero = input("Dimmi un numero: ")
numero = int(numero)

conteggio = 0
totale = 0

while conteggio < numero:
    conteggio = conteggio + 1
    print(conteggio)
    totale = totale + conteggio

print("La somma dei primi", numero, "numeri è", totale)

if numero != 0:
    media = totale / numero
    print("La media dei primi", numero, "numeri è", media)
```

# Contare, fare la somma, fare la media... con un for

```
numero = input("Dimmi un numero: ")
numero = int(numero)

totale = 0

for conteggio in range(1, numero + 1):
    print(conteggio)
    totale = totale + conteggio

print("La somma dei primi", numero, "numeri è", totale)

if numero != 0:
    media = totale / numero
    print("La media dei primi", numero, "numeri è", media)
```

# Contiamo i segni zodiacali...

```
conteggio = 0
for segno in segni_zodiacali:
    conteggio = conteggio + 1

print("Ci sono", conteggio, "segni zodiacali!")
```

# ... e poi scopriamo che era molto più semplice!

- ▶ Possiamo sempre sapere la **lunghezza** di un elenco
  - ▶ Usando la funzione `len`
- ▶ Gli elenchi in realtà si chiamano **liste**
  - ▶ Sono **strutture dati** molto utili...

```
>>> len(segni_zodiacali)
12
```

# Esercizi



Esempi ed esercizi - Facciamo un po' di pratica



# The Bridge of Death

- ▶ Porre tre domande all'utente:
  - ▶ "What is your name?"
  - ▶ "What is your quest?"
  - ▶ "What is your favorite color?"
- ▶ Se le risposte sono "Lancelot", "Holy Grail" e "Blue", stampare:
  - ▶ "Right. Off you go."
- ▶ Altrimenti, stampare:
  - ▶ "Down into the Gorge of Eternal Peril!"

Provate voi!



# Il cerchio

Provate voi!

- ▶ Chiedere all'utente il valore del raggio  $r$  di un cerchio
  - ▶  $r$  reale compreso tra 0 e 200
- ▶ Se  $r$  è valido
  - ▶ Visualizzare il cerchio, al centro del canvas
  - ▶ Mostrare il valore dell'area e della circonferenza
- ▶ Se invece  $r$  è fuori range
  - ▶ Mostrare un messaggio d'errore



# Il più piccolo e il più grande

- ▶ Generare e stampare tre numeri interi casuali:  $a$ ,  $b$ ,  $c$
- ▶ Ciascuno compreso tra 1 e 6
- ▶ Determinare e mostrare qual è il minore dei tre
- ▶ Determinare e mostrare qual è il maggiore dei tre

Provate voi!



# Calcolo dell'età

- ▶ Chiedere all'utente la data di nascita
  - ▶ Anno, mese e giorno
- ▶ Comunicare l'età esatta attuale
  - ▶ Numero di compleanni già compiuti
- ▶ Usando la data di oggi:

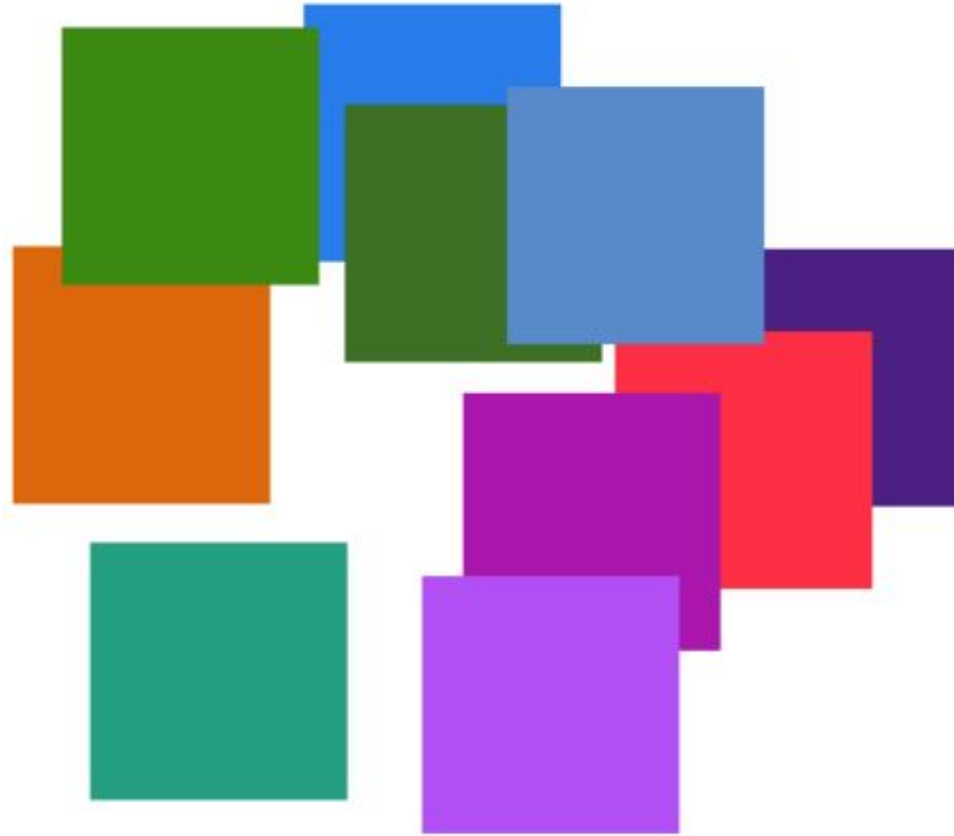
Provate voi!

```
from datetime import date  
  
now = date.today()  
  
print(now.year, now.month, now.day)
```

Provate voi!

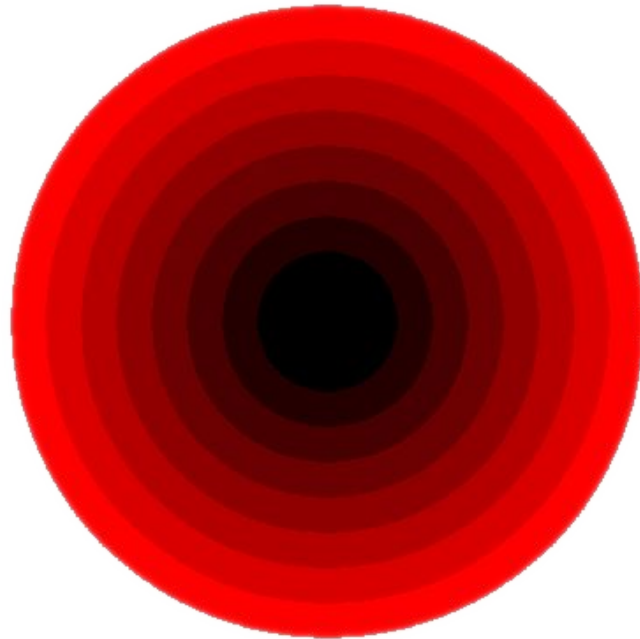
# Quadrati casuali

- ▶ Chiedere all'utente un numero  $n$
- ▶ Disegnare  $n$  quadrati
  - ▶ Tutti con lato di 100 pixel
  - ▶ Ciascuno in posizione casuale
  - ▶ Ciascuno con un colore casuale



# Cerchi concentrici

- ▶ Chiedere all'utente il numero di cerchi da disegnare
- ▶ Disegnare i cerchi con raggio gradualmente decrescente, ma tutti con lo stesso centro
- ▶ Far variare il colore dei cerchi
  - ▶ Creando un effetto gradiente!
- ▶ Prima del ciclo, determinare di quanto cambiare raggio e colore ad ogni passo



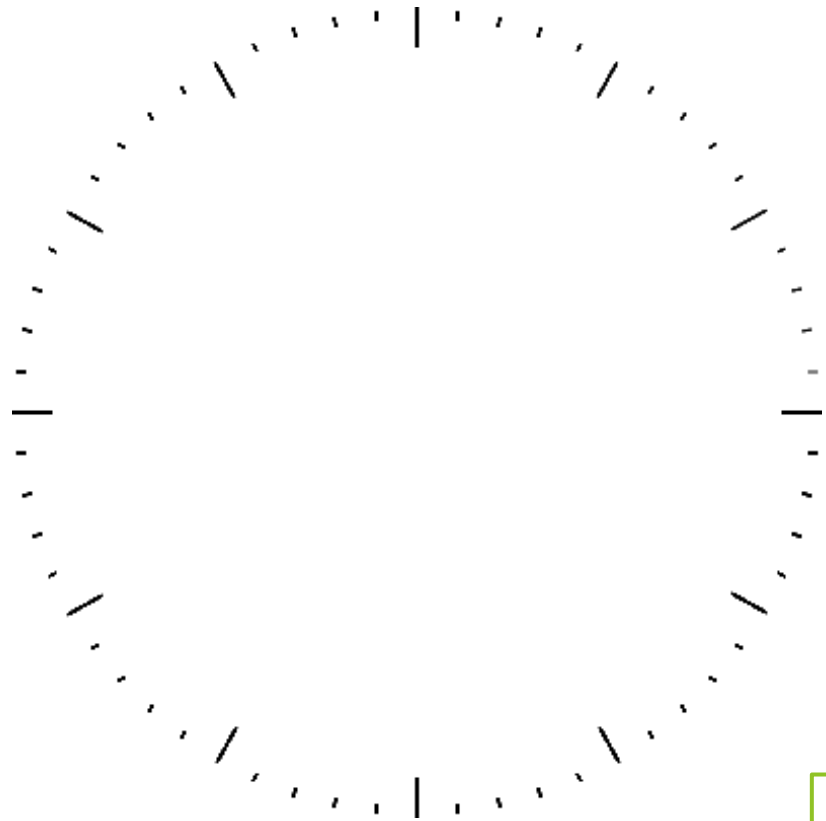
Provate voi!

# Il gioco delle 3 carte



- ▶ All'inizio l'utente ha 10 monete
- ▶ Ad ogni turno:
  - ▶ Viene estratto a sorte un numero segreto tra 1 e 3
  - ▶ L'utente sceglie quante monete puntare e su quale numero
  - ▶ Se indovina, gli viene sommato l'importo puntato
  - ▶ Altrimenti gli viene sottratto lo stesso importo
- ▶ Il gioco termina quando l'utente perde tutto o si ritira

Provate voi!

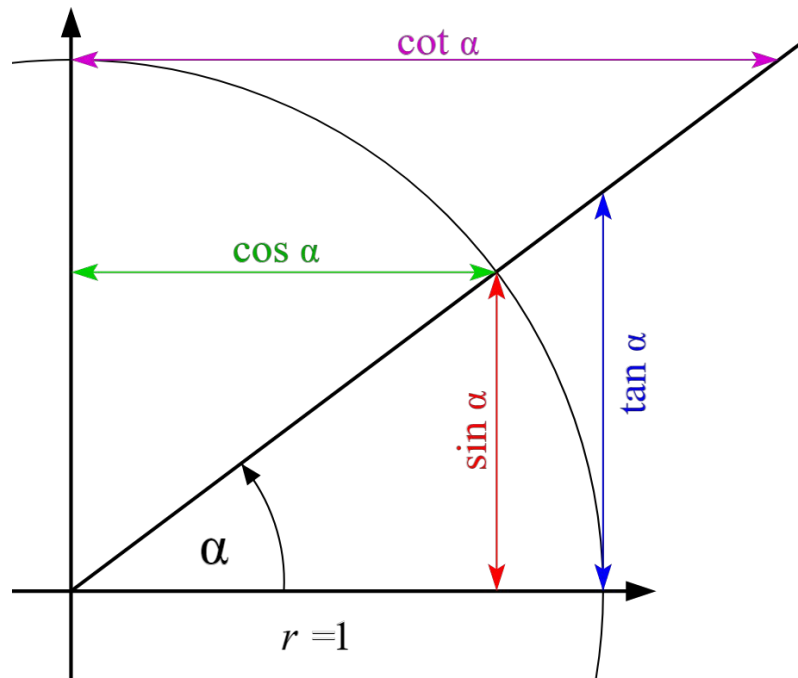


# L'orologio analogico

- ▶ Disegnare 12 tacche a raggiera, come in un orologio classico
- ▶ Miglioramento: disegnare anche le tacche dei minuti, più piccole

Suggerimento: Usare `math.sin` e `math.cos` per determinare le posizioni in cui disegnare





# L'orologio analogico

- ▶ Disegnare 12 tacche a raggiera, come in un orologio classico
- ▶ Miglioramento: disegnare anche le tacche dei minuti, più piccole

Suggerimento: Usare `math.sin` e `math.cos` per determinare le posizioni in cui disegnare