CICLI WHILE
ESEMPIO: I NUMERI PRIMI
CICLI FOR

INFORMATICA

OBIETTIVO DI QUESTA LEZIONE

SCRIVERE UN PROGRAMMA
CHE VERIFICA SE UN NUMERO È PRIMO

IDEA PER RISOLVERE IL PROBLEMA

- Consideriamo il numero *n* da testare
- Sia *m* il numero di divisori non banali di *n* trovati.
  All'inizio *m* è zero
- Per ogni numero *k* da *2* a *n-1* controlliamo
  - Se il resto della divisione di *n* per *k* è zero
    - Incrementiamo il numero di divisori non banali
- Se *m* è zero il numero è primo altrimenti è composto

RIPETERE CODICE

FINCHE X È MINORE DI 10 FAI QUESTO…

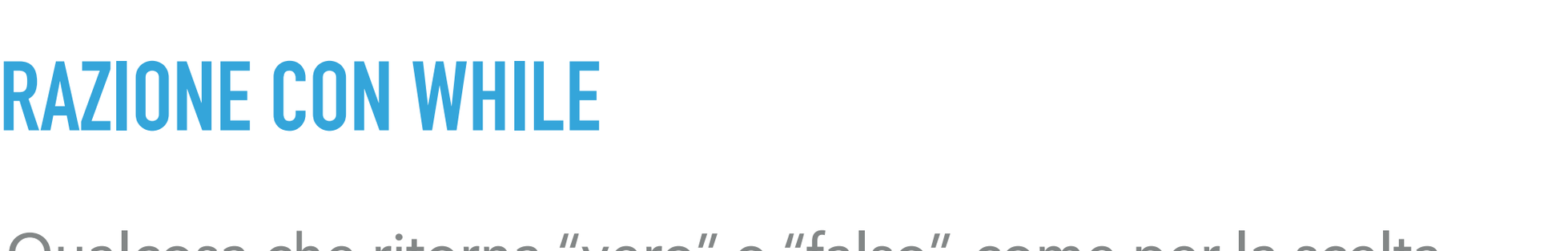
FINCHE NON HAI TROVATO UN DIVISORE…

- A volte vogliamo ripetere lo stesso codice più volte
- In generale vogliamo una istruzione che dica “ripeti finché questa condizione è vera”
- Qualcosa simile ad avere “infinite” istruzioni “if” identiche

ITERAZIONE CON WHILE

Qualcosa che ritorna “vero” o “falso”, come per la scelta

```
while condizione:
    codice da eseguire se la condizione è vera.
    Il codice viene eseguito di nuovo se la condizione
    rimane vera
```



```
if condizione:
    codice da eseguire se la condizione è vera.
    Torna ad eseguire nel punto indicato dalla freccia.
```

ESEMPI DI WHILE

```
x = 0
y = 0
n = 0
while x < 10:
  y = 0
  while y < 10:
    y = y + 1
    n = n + 1
  x = x + 1
```

Il ciclo esterno viene eseguito 10 volte

Il ciclo interno viene eseguito 10 volte...
... ad ogni esecuzione del ciclo esterno

Quindi n viene incrementato 100 volte!

QUIZ SU "WHILE"

```
x = 0
y = 1
while x < 4:
  x = x + 1
  y = 2 * y
```

Quali sono i valori di x e y dopo l'esecuzione di questo codice?

1. x è 5
   y è 8
2. Ciclo
   infinito
3. x è 4
   y è 16
4. x è "Il capitano Kirk"
   y è "gli ABBA"

I CICLI FOR

STRUTTURA COMUNE DELL'ITERAZIONE

Variabile "indice"

Dichiariamo ed inizializziamo una variabile che conta
il numero di volte che abbiamo eseguito il ciclo

Condizione di uscita

Controlliamo se proseguire il ciclo

```
i = 0
while i < 10:
    # qui mettiamo il nostro codice
    i = i + 1
```

Incremento della variabile indice

Aumentiamo il valore della variabile indice

IDEA: NON RIPETERE INUTILMENTE

- La precedente struttura di iterazione è molto comune
- Ha senso riscriverla ogni volta rischiando di sbagliare?
- **NO**
- Per questo il linguaggio Python fornisce un costrutto pensato apposta per questo: il **ciclo for**

STRUTTURA DEL CICLO FOR

Variabile “indice”

La definiamo all’inizio del ciclo for, assumere tutti i valori della “collezione” specificata dopo “in”.

Collezione

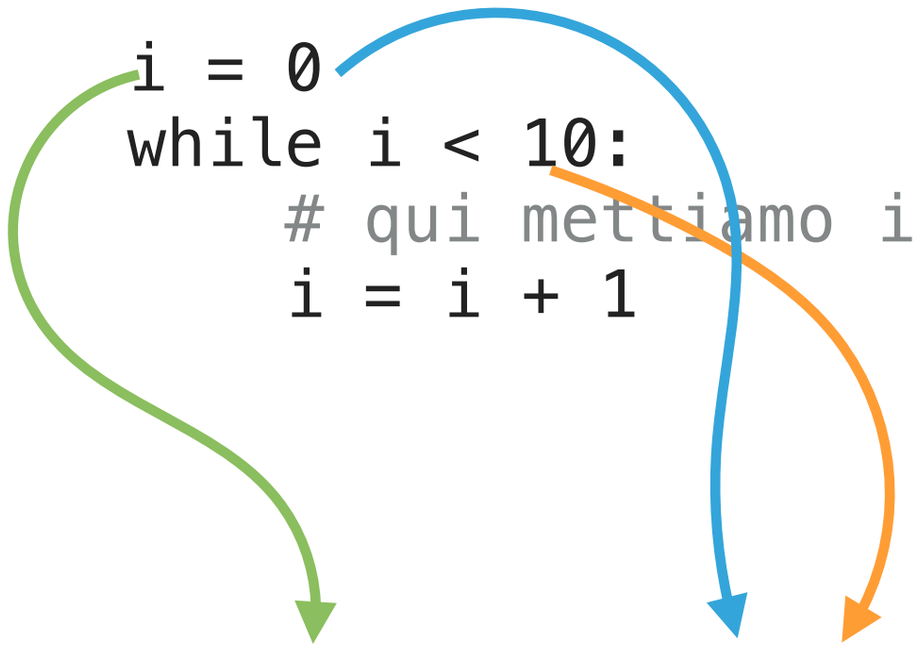
Una sequenza di valori da assegnare, uno per ciclo, alla variabile indice

```
for i in range(0,10):
    # qui mettiamo il nostro codice
```

Cosa è “Range”?

Range(inizio,fine) ci permette di generare una sequenza di numeri nell’intervallo [inizio, fine) a distanza 1 l’uno dall’altro

EQUIVALENZA



```
i = 0
while i < 10:
    # qui mettiamo il nostro codice
    i = i + 1
```

```
for i in range(0,10):
    # qui mettiamo il nostro codice
```

Questo funziona perché range crea una collezione di numeri che partono da 0 e sono incrementi di 1 fino a che sono minori di 10

ESEMPIO: CICLI ANNIDATI

Ciclo esterno, viene eseguito 3 volte

```
for i in range(1,4):
  for j in range(1,4):
    prodotto = i * j
    print(str(i) + " * " + str(j)
          + " = " + str(prodotto))
```

Ciclo interno, viene eseguito 3 volte per ogni esecuzione del ciclo esterno, quindi 9 volte in totale

Output:

```
1 * 1 = 1
1 * 2 = 2
1 * 3 = 3
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
```

QUIZ SU “FOR”

```
x = 5
for i in range(0,5):
    print(x * i)
```

Quale è l’output di questo codice?

1. 0 1 2 3 4
2. 5 5 5 5 5
3. 0
4. 0 5 10 15 20

QUIZ SU "FOR"

```
z = 10
for x in range(0,z):
    for y in range (0,z):
        print("x =" + str(x))
```

Quale è l'output di questo codice?

1. x = 0
   x = 1
   ...

2. x = 1
   x = 2
   ...

3. x = 0
   x = 0
   ...

4. Never gonna give you up
   Never gonna let you down
   ...