

# Strutture di controllo

## UDA

Terzo anno scuola  
secondaria di secondo  
grado

Presentata da:

*Simone Degiacomi*

*Carlo Fanciulli*

*Rupert Gobber*

*Francesco Penasa*



# Assunzioni/requisiti

Come eseguire programma Python

Variabili e operazioni matematiche (con modulo)

Input/print da terminale

Stringhe

Casting



# Sezioni dell'unità didattica

Ogni sezione, riguardante una struttura di controllo diversa, si svolgerà in modo interattivo e sarà composta da: nozioni, ragionamenti, esempi ed esercizi

I

**If**

Condizioni,  
espressioni  
booleane,  
sintassi,  
introduzione a  
Turtle

II

**While/do-while**

Ripetere le  
operazioni in base  
ad una  
condizione, while,  
sintassi, esercizi  
con Turtle

III

**For**

Perchè il for,  
differenza tra  
while, sintassi,  
esercizi con Turtle

# Indice presentazione

1

Metodologie e strumenti scelti

2

Condizioni, espressioni booleane e if

3

Ciclo while e do-while

4

Ciclo for

5

Esempi con Python turtle

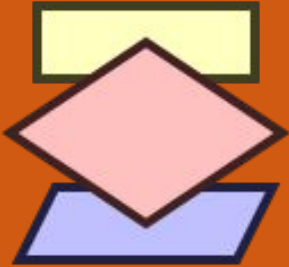


# Pedagogical Content Knowledge

- 01** Perché insegniamo le strutture di controllo e turtle?
- 02** Cosa insegniamo delle strutture di controllo e turtle?
- 03** Come insegniamo i concetti?
- 04** Quali strumenti didattici utilizziamo?



# Strumenti scelti



Flowchart



Python



Python Turtle

# Flowchart

**01** Permette di esprimere la logica dell'algoritmo senza preoccuparsi della sintassi

**02** Chiara visualizzazione del flusso del programma

**03** Facile da disegnare in aula (sulla carta o sulla lavagna)

**04** Semplice corrispondenza tra il diagramma e la struttura del codice

**05** Presenza di tool per eseguire un diagramma di flusso



# Python

## IDE

- Pycharm Edu : Code Completion, Local History, Code Inspection, Debugging intuitivo, gratis.
- Alternative: TigerJython, Spyder3, IDLE

01

User-friendly, simile al linguaggio naturale.

02

Lo studente si concentra più sull'aspetto semantico piuttosto che su quello sintattico.

03

Programmi di piccolissime dimensioni.

04

L'indentazione obbligatoria diminuisce le cattive abitudini.

05

Grande community di supporto e un enorme numero di librerie.





# Turtle Graphics

## Moltissimi esempi

- Doing Math with Python
- Python for kids
- Teach your kid to code
- IDLE turtle demo

**01** Visualizzare la sequenza delle istruzioni.

**02** Debugging più' semplice tramite l'uso della grafica.

**03** Introduzione alle librerie.

**04** Uso del piano cartesiano.



# Pair Programming

2 studenti – 1 computer

## Driver

- Scrive il codice
- Controlla mouse/tastiera
- Si occupa dei dettagli

## Navigator

- Pensa a più alto livello
- Osserva errori di battitura ed errori logici

01

Aumenta la condivisione della conoscenza

02

Aumenta l'attenzione

03

Migliora la capacità di comunicazione

04

Diminuisce le cattive abitudini normalmente presenti in programmazione

05

Meno bug e miglior codice



# Ostacoli all'apprendimento

## Reinforcement

Skinner ha identificato 5 ostacoli all'apprendimento



01

Timore di fallire

02

Compito troppo lungo e complicato

03

Compito troppo generico e senza istruzioni/ direzioni

04

Istruzioni poco chiare o confuse

05

Ambiente non adatto/ poco o nessun rinforzo positivo



L'apprendimento di un comportamento da parte di un soggetto non è solo frutto del caso ma più spesso di un agente di rinforzo



# Esercizi con error analysis

## Sbagliando si impara

- Approccio costruttivista
- Inizialmente gli studenti pensano alla soluzione
- Poi la si discute
- Si portano in primo piano le misconceptions
- Si analizzano esempi sbagliati, si capisce dove sono sbagliati e perché

01

Stimola il problem solving

02

Focalizza l'attenzione degli studenti sulla ricerca problema

03

Questo tipo di esercizi aiuta la memorizzazione della sintassi e a prendere familiarità con la struttura di un programma

04

Fa emergere le misconception e favorisce la discussione su eventuali soluzioni alternative

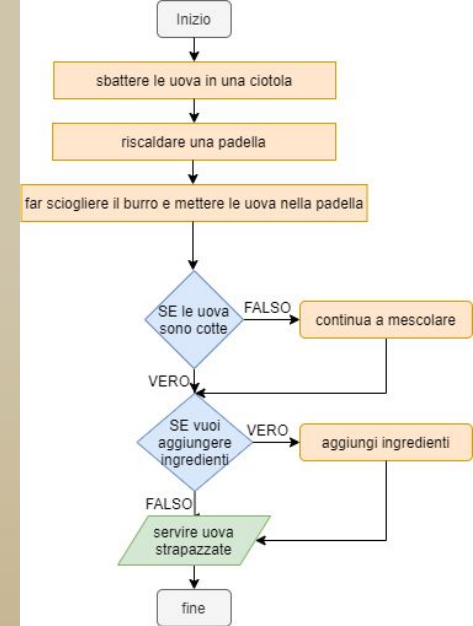
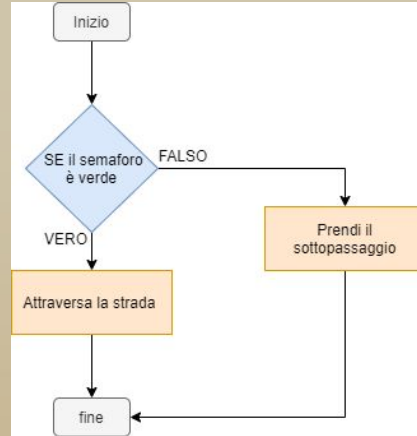




If



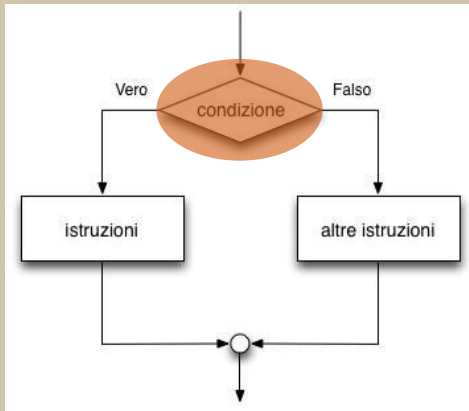
**Se** il semaforo è verde attraversa la strada, **altrimenti** (il semaforo è rosso) prendi il sottopassaggio



Altri esempi?

# Condizione

può essere formata da:



`a=5 b=7`

`a==5 and b==7 # Vero`  
`a==7 and b==7 # Falso`  
`a==7 or b==7 # Vero`  
`a==7 or b==5 # Falso`  
`not a>0 # Falso`

**Variabili booleane**

**Espressioni booleane**

**Operatori logici**

`x == y # è Vero se... x è uguale a y`  
`x != y # ... x è diverso da y`  
`x > y # ... x è maggiore di y`  
`x < y # ... x è minore di y`  
`x >= y # ... x è maggiore o uguale a y`  
`x <= y # ... x è minore o uguale a y`

`5 == (3 + 2) # Vero`  
`5 == 6 # Falso`

Altri esempi?



# Strutture

```
if x == y:  
    print x, " e ", y, "sono uguali"  
else:  
    print x, " e' maggiore o minore di ", y
```

**if..else**

```
if x == y:  
    print x, " e ", y, "sono uguali"  
else:  
    if x < y:  
        print x, " e' minore di ", y  
    else:  
        print x, " e' maggiore di ", y
```

**If annidati**

```
if x < y:  
    print x, "e' minore di", y  
elif x > y:  
    print x, "e' maggiore di", y  
else:  
    print x, "e", y, "sono uguali"
```

**if..elif..else**

Altri esempi?



# Error Analysis

## Esempio 1:

```
numero = int(input("Inserisci un numero: "))  
  
if numero %3 == 0 and num %5 == 0:  
    print ("Fizzbuzz")  
elseif numero %5 == 0:  
    print ("Buzz")  
elif numero %3 == 0:  
    print ("Fizz")  
else:  
    print ("Il numero e'", numero)
```

## Esempio 2:

```
tempo =input("Sta piovendo? \n")  
if tempo == "si":  
    vento = input("E' molto ventoso? \n")  
if vento =="si":  
    print ("E' troppo ventoso per l'ombrello")  
else:  
    print ("Prendi un ombrello")  
else:  
    print ("Buona giornata")
```

## Esempio 3:

```
eta =int(input("Quanti anni hai? \n"))  
if eta >= 3:  
    print ("Sei un neonato")  
elif eta = 4:  
    print ("Fra poco inizi la scuola!")  
elif eta != 16:  
    print ("Allora stai andando a scuola")  
else:  
    print ("Adesso puoi andare a lavorare")
```

## Versioni corrette

e con gli if annidati?



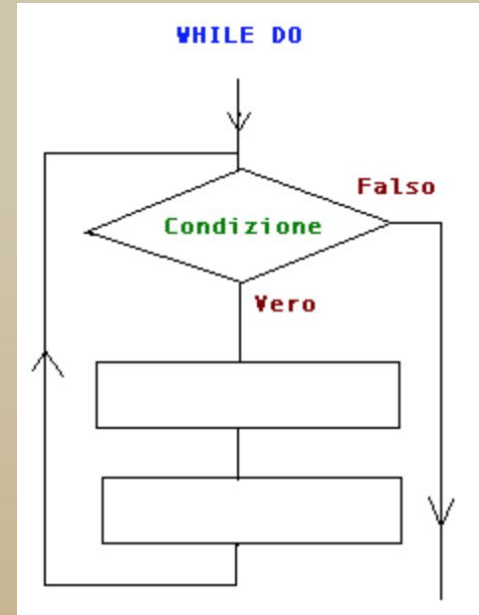
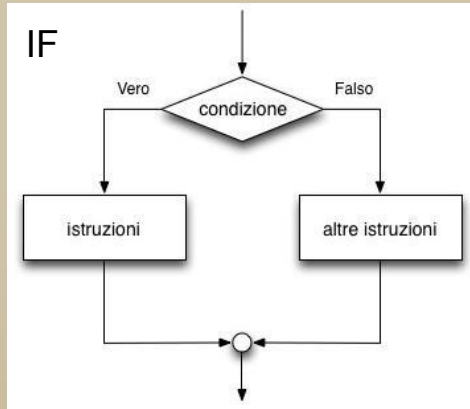
```
numero = int(input("Inserisci un numero: "))  
  
if numero %3 == 0 and numero %5 == 0:  
    print ("Fizzbuzz")  
elif numero %5 == 0:  
    print ("Buzz")  
elif numero %3 == 0:  
    print ("Fizz")  
else:  
    print ("il numero e'", numero)
```

```
tempo =input("Sta piovendo? \n")  
if tempo == "si":  
    vento = input("E' molto ventoso? \n")  
    if vento =="si":  
        print ("E' troppo ventoso per l'ombrello")  
    else:  
        print ("Prendi un ombrello")  
else:  
    print ("Buona giornata")
```

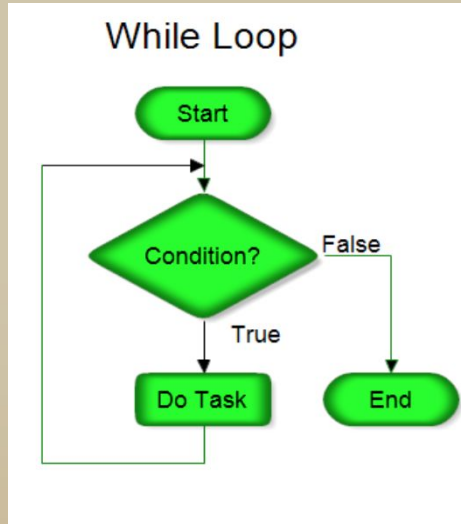
```
eta =int(input("Quanti anni hai? \n"))  
if eta <= 3:  
    print ("Sei un neonato")  
elif eta == 4:  
    print ("Fra poco inizi la scuola!")  
elif eta <=16:  
    print ("Allora stai andando a scuola")  
else:  
    print ("Adesso puoi andare a lavorare")
```



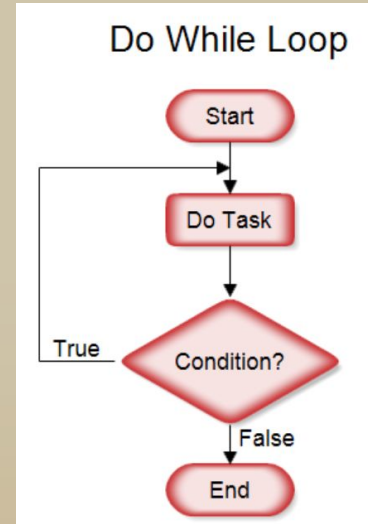
# Da if a while



# Esistono 2 tipologie di while



```
1 i = 1
2 while i < 6:
3     print(i)
4     i += 1
```



```
1 i = 1
2 while True:
3     print(i)
4     i += 1
5     if i >= 6:
6         break
```



# Istruzioni *break* e *continue*

```
1 i = 1
2 while i < 6:
3     print(i)
4     if i == 3:
5         break
6     i += 1
```

- Esce dal loop se  $i == 3$

```
1 i = 0
2 ▽ while i < 6:
3     i += 1
4     if i == 3:
5         continue
6     print(i)
```

- Continua con la prossima iterazione se  $i == 3$



# Errori più comuni nel While

## Loop infinito

```
1 i = 0
2 while i < 10:
3     print(i)
```



## Sintassi sbagliata

```
1 i = 0
2 while i < 10:
3     print i
4     i += 1
```



## Condizione sbagliata

```
1 i = 0
2 while i < 0:
3     print(i)
4     i += 1
```



```
1 i = 0
2 while i < 10:
3     print(i)
4     i += 1
```



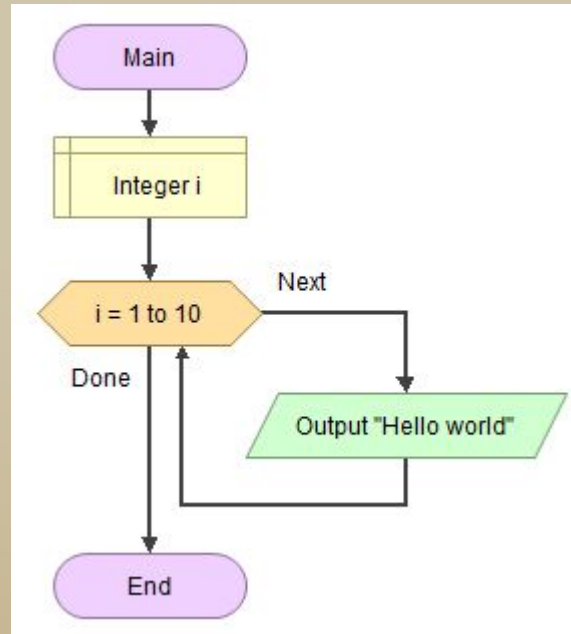


# For

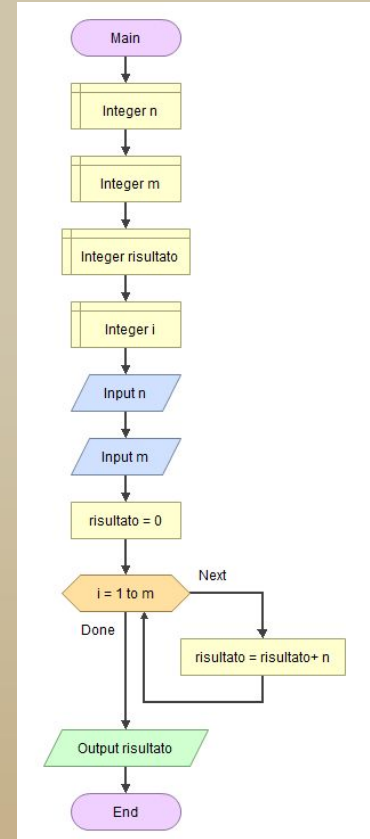
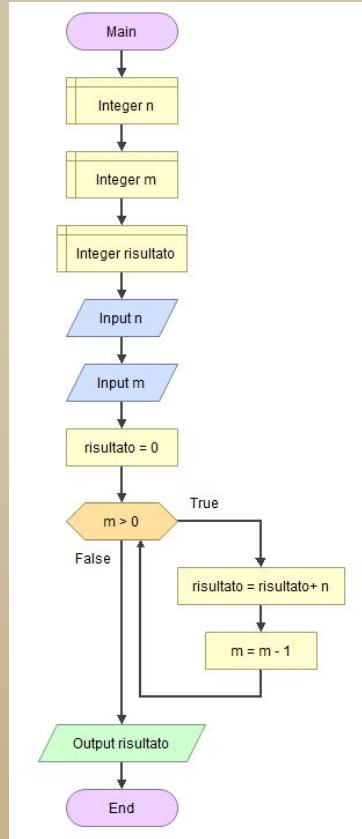
Un nuovo ciclo? a cosa ci serve?



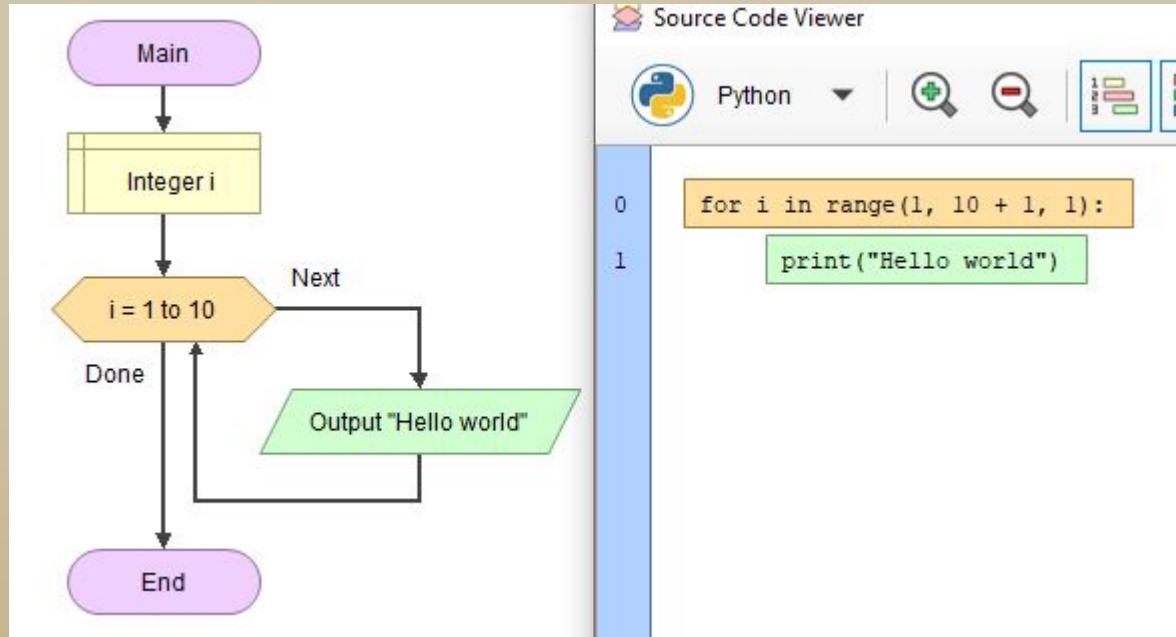
# Struttura del for



# Da while a for



# Da Flowchart a Python





```
for i in range(1, 10 + 1, 1):  
    print(i)
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

```
for i in range(1, 10, 1):  
    print(i)
```

1  
2  
3  
4  
5  
6  
7  
8  
9

```
for i in range(1, 10 + 1):  
    print(i)
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

Ultimo escluso

Step implicito

```
In [4]: for i in range(0, 10):  
        print(i)
```

0  
1  
2  
3  
4  
5  
6  
7  
8  
9

Inizia da 0

```
In [5]: for i in range(10):  
        print(i)
```

0  
1  
2  
3  
4  
5  
6  
7  
8  
9

Numero di iterazioni

# Error Analysis

```
n = int(input("Quanti valori vuoi inserire? "))
somma = 0

for i in range(n):
    valore = int(input("Inserisci il prossimo valore: "))
    somma = somma + valore
    i = i + 1

media = somma / n
print("La media è: {}".format(media))
```

```
Quanti valori vuoi inserire? 2
Inserisci il prossimo valore: 5
Inserisci il prossimo valore: 6
La media è: 5.5
```

```
n = int(input("Quanti valori vuoi inserire? "))
somma = 0

for i in range(n):
    valore = int(input("Inserisci il prossimo valore: "))
    somma = somma + valore

media = somma / n
print("La media è: {}".format(media))
```

```
Quanti valori vuoi inserire? 2
Inserisci il prossimo valore: 5
Inserisci il prossimo valore: 6
La media è: 5.5
```



# For annidati



```
foreach week {
  foreach weekday{
    wake up
    eat breakfast
    clean teeth
    goto school
    foreach period{
      goto class
      do lesson
    }
    go home
    do evening stuff
  }
  do saturday stuff
  do sunday stuff
}
```

# Note sul for

- Stiamo parlando sul funzionamento di **range**
- Approssimazione dei numeri decimali e perchè non usarli nei for





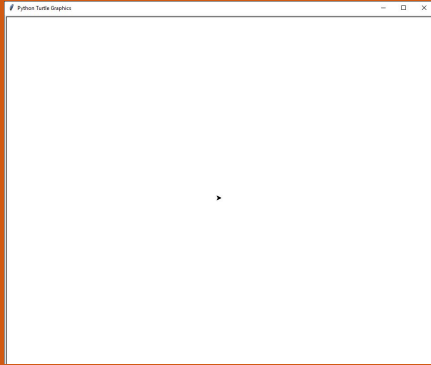
# Turtle graphics

Cos'è? A cosa ci serve?



# Struttura base di turtle

- Apriamo una shell python3 e importiamo la libreria turtle.
- Inizializziamo una finestra e una tartaruga.



Risultato

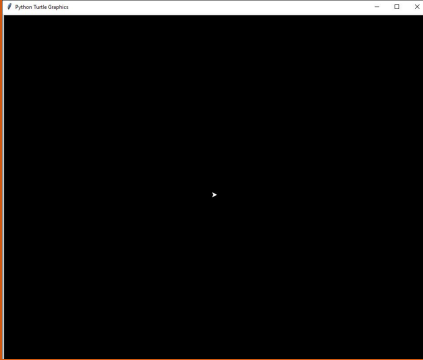


```
Python 3.7.5 Shell
File Edit Shell Debug Options Window Help
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019,
00:11:34) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()"
for more information.
>>> import turtle
>>> finestra = turtle.getscreen()
>>> tartaruga = turtle.Turtle()
>>> |
```

Codice

# Struttura base di turtle

- Modifichiamo la finestra cambiando il colore.
- Modifichiamo la tartaruga cambiando il colore.



Risultato

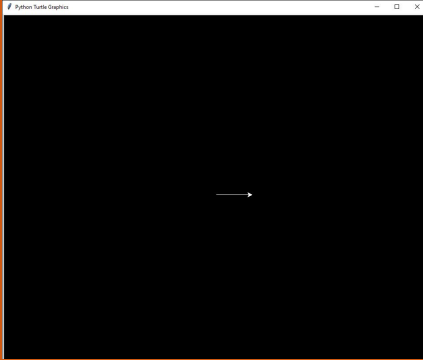


```
Python 3.7.5 Shell
File Edit Shell Debug Options Window Help
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019,
00:11:34) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()"
for more information.
>>> import turtle
>>> finestra = turtle.getscreen()
>>> tartaruga = turtle.Turtle()
>>>
>>> finestra.bgcolor("black")
>>> tartaruga.color("white")
>>> |
```

Codice

# Struttura base di turtle

- Facciamo andare avanti la tartaruga.



Risultato



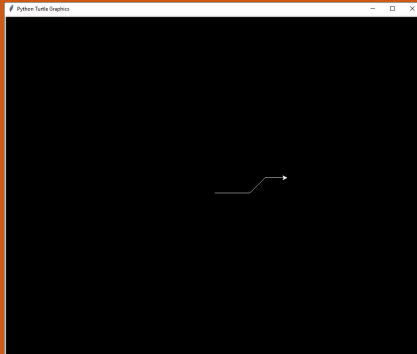
```
Python 3.7.5 Shell
File Edit Shell Debug Options Window Help
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import turtle
>>> finestra = turtle.getscreen()
>>> tartaruga = turtle.Turtle()
>>>
>>> finestra.bgcolor("black")
>>> tartaruga.color("white")
>>>
>>> tartaruga.forward(80)
>>>
```

Codice



# Struttura base di turtle

- Cambiamo direzione e andiamo avanti.



Risultato



```
Python 3.7.5 Shell
File Edit Shell Debug Options Window Help
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019,
00:11:34) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()"
for more information.
>>> import turtle
>>> finestra = turtle.getscreen()
>>> tartaruga = turtle.Turtle()
>>>
>>> finestra.bgcolor("black")
>>> tartaruga.color("white")
>>>
>>> tartaruga.forward(80)
>>>
>>> tartaruga.left(45)
>>> tartaruga.forward(50)
>>> tartaruga.right(45)
>>> tartaruga.forward(50)
>>> |
```

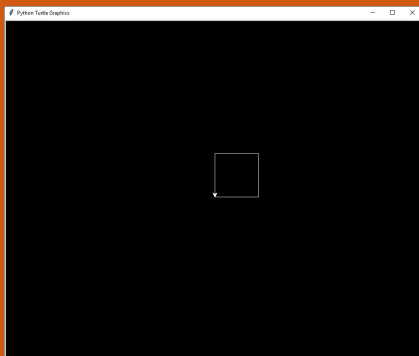
Codice

# Creiamo un quadrato

- Che istruzioni dovremmo eseguire per creare un quadrato?

Errori comuni:

- Sintassi errata (Es: `t = turtle.turtle()`)
- Argomento della funzione mancante (Es: `t.forward()`)
- Oggetto della funzione non specificato (Es: `left(90)`)



Risultato



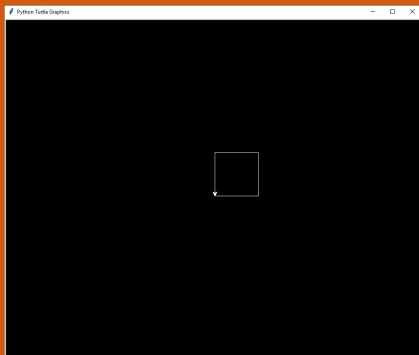
Codice

# Creiamo un quadrato

Possiamo anche inserire il codice in un file ed eseguirlo.

Documentazione online:

<https://docs.python.org/3.3/library/turtle.html?highlight=turtle#module-turtle>



Risultato



```
File Edit View Navigate Code Help  untitled - C:\Users\Francesco\PycharmProjects\untitled\venv\Lib\test.py - PyCharm
test
untitled  venv  Lib  test.py
test.py
1 import turtle
2
3 finestra = turtle.getscreen()
4 tartaruga = turtle.Turtle()
5
6 finestra.bgcolor("black")
7 tartaruga.color("white")
8
9 tartaruga.forward(100)
10 tartaruga.left(90)
11 tartaruga.forward(100)
12 tartaruga.left(90)
13 tartaruga.forward(100)
14 tartaruga.left(90)
15 tartaruga.forward(100)
16

Python Console  test
C:\Users\Francesco\PycharmProjects\untitled\venv\Scripts\python.exe D:\dev\JetBrain
import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['c:\Users\Francesco\PycharmProjects\untitled', 'c:/Users/Franc
Python Console
Process finished with exit code 0
```

Codice

# Esempi in Turtle

- Componente “comune” di un programma turtle  
{ESEMPI\_TURTLE/init.py}
- Un quadrato disegnato in turtle  
{ESEMPI\_TURTLE/quadrato.py}
- Un altro quadrato disegnato in turtle  
{ESEMPI\_TURTLE/quadrato\_inv.py}

Errori comuni:

- Sintassi errata (Es: `t = turtle.turtle()`)
- Argomento della funzione mancante (Es: `t.forward()`)
- Oggetto della funzione non specificato (Es: `speed(9)`)



Esempi: <https://drive.google.com/open?id=1g6i1vdoz9j7RGzkhFP3YfBJBymldq6S0>

# IF beginner level

Proviamo ad implementare il costrutto IF

```
{ESEMPI_TURTLE/if_quadrato_1.py}
```

```
{ESEMPI_TURTLE/if_quadrato_2.py}
```

```
{ESEMPI_TURTLE/if_quadrato_3wrong.py}
```

```
{ESEMPI_TURTLE/if_quadrato_3correct.py}
```

Errori comuni:

- Sintassi errata (Es: i due punti dopo la condizione)
- Mancata indentazione
- Mancanza di cast esplicito
- Se IF viene eseguito allora ELSE non viene eseguito



Esempi: <https://drive.google.com/open?id=1g6i1vdoz9j7RGzkhFP3YfBJBymldq6S0>

# IF intermediate level

Proviamo ad implementare il costrutto IF  
{ESEMPI\_TURTLE/pari.py}

Errori comuni:

- Usare condizioni limitate e non testare correttamente il codice.



Esempi: <https://drive.google.com/open?id=1g6i1vdoz9j7RGzkhFP3YfBJBymlDq6S0>

# IF expert level

Proviamo ad implementare il costrutto IF

{ESEMPI\_TURTLE/scf1.py}

{ESEMPI\_TURTLE/scf2.py}

{ESEMPI\_TURTLE/scf3.py}

Cosa sono questi programmi?

Errori comuni:

- Usare condizioni limitate e non testare correttamente il codice.



Esempi: <https://drive.google.com/open?id=1g6i1vdoz9j7RGzkhFP3YfBJBymldq6S0>

# WHILE beginner level

Proviamo ad implementare il costrutto WHILE

{ESEMPI\_TURTLE/quadrato\_while.py}

{ESEMPI\_TURTLE/triangolo\_while.py}

Errori comuni:

- Sintassi errata (Es: i due punti dopo il while)
- Mancata indentazione
- Quante volte il ciclo sarà eseguito
- L'argomento della funzione sbagliato.



Esempi: <https://drive.google.com/open?id=1g6i1vdoz9j7RGzkhFP3YfBJBymlDq6S0>



# WHILE intermediate level

Un esempio con while true

{ESEMPI\_TURTLE/pari\_2.py}

Una figura geometrica non banale

{ESEMPI\_TURTLE/spirale.py}

Errori comuni:

- Spaziatura piuttosto che tabulazione
- Non prevedere un'uscita dal ciclo.



Esempi: <https://drive.google.com/open?id=1g6i1vdoz9j7RGzkhFP3YfBJBymldq6S0>

# WHILE expert level

Proviamo ad implementare WHILE e IF contemporaneamente  
{ESEMPI\_TURTLE/interactive\_draw.py}

Disegniamo sfruttando dei WHILE annidati.  
{ESEMPI\_TURTLE/while\_while.py}

Errori comuni:

- Usare la variabile sbagliata come condizione per l'uscita dal ciclo
- Non prevedere un'uscita dal ciclo



Esempi: <https://drive.google.com/open?id=1g6i1vdoz9j7RGzkhFP3YfBJBymlDq6S0>

# FOR beginner level

Proviamo ad implementare il costrutto FOR

{ESEMPI\_TURTLE/quadrato\_for.py}

{ESEMPI\_TURTLE/cerchio\_for.py}

{ESEMPI\_TURTLE/square\_spiral.py}

Errori comuni:

- Sintassi errata (Es: i due punti dopo il FOR)
- Mancata indentazione
- Numero di passi



Esempi: <https://drive.google.com/open?id=1g6i1vdoz9j7RGzkhFP3YfBJBymlDq6S0>

# FOR intermediate level

Proviamo ad implementare il costrutto FOR  
{ESEMPI\_TURTLE/square\_spiral2.py}  
{ESEMPI\_TURTLE/color\_spiral.py}

Errori comuni:

- La condizione dopo il costrutto ELSE



Esempi: <https://drive.google.com/open?id=1g6i1vdoz9j7RGzkhFP3YfBJBymldq6S0>

# FOR expert level

Proviamo ad implementare il costrutto FOR  
{ESEMPI\_TURTLE/for\_annidati.py}  
{ESEMPI\_TURTLE/da\_while\_a\_for.py}

Errori comuni:

- Uso improprio del for (es. for n in 5)



Esempi: <https://drive.google.com/open?id=1g6i1vdoz9j7RGzkhFP3YfBJBymlDq6S0>

# All together now

```
{ESEMPI_TURTLE/select_figure.py}  
{ESEMPI_TURTLE/random_figure.py}  
{ESEMPI_TURTLE/random_steps.py}
```

Errori comuni:

- Non prevedere un risultato possibile



Esempi: <https://drive.google.com/open?id=1g6i1vdoz9j7RGzkhFP3YfBJBymlDq6S0>

# The sky is the limit

{ESEMPI\_TURTLE/turtle\_race.py}

Esempi presenti in IDLE turtle demo.



Esempi: <https://drive.google.com/open?id=1g6i1vdoz9j7RGzkhFP3YfBJBymldq6S0>



Grazie per l'attenzione