



Coding in Python

Gianluca Palermo – gianluca.palermo@polimi.it

Target

- To Learn the basic concepts behind computer programming
 - using Python as programming language
- To Learn how to solve simple problems using a computer
- **To develop a simple Videogame!**



POLITECNICO MILANO 1863

- 2 -

Coding, What's this?

- Use of the computer to simplify some data manipulation tasks

- Solution of the problem in a way that has been adapted to an automatic solution

- Description of the algorithm in a way that can be understood by the automatic executor (Computer)

- 3 -

POLITECNICO MILANO 1863

Computer History: From application specific machines... (one Task)



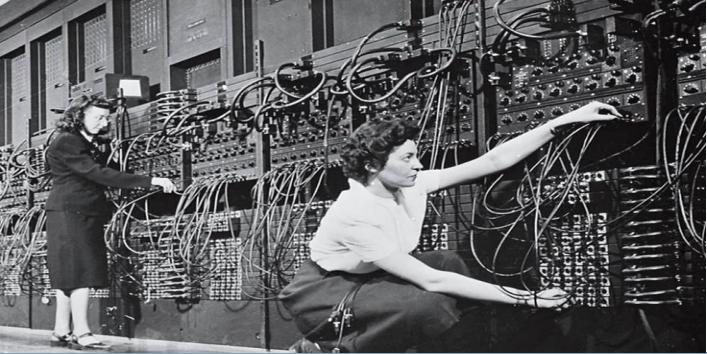
By Alessandro Nassiri - Museo della Scienza e della Tecnologia "Leonardo da Vinci".
CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=47910919>

The Enigma machine is an encryption device developed and used in the early- to mid-20th century to protect commercial, diplomatic and military communication. It was employed extensively by Nazi Germany during World War II, in all branches of the German military.

- 4 -

POLITECNICO MILANO 1863

...to programmable machines ...



- 5 -

POLITECNICO MILANO 1863

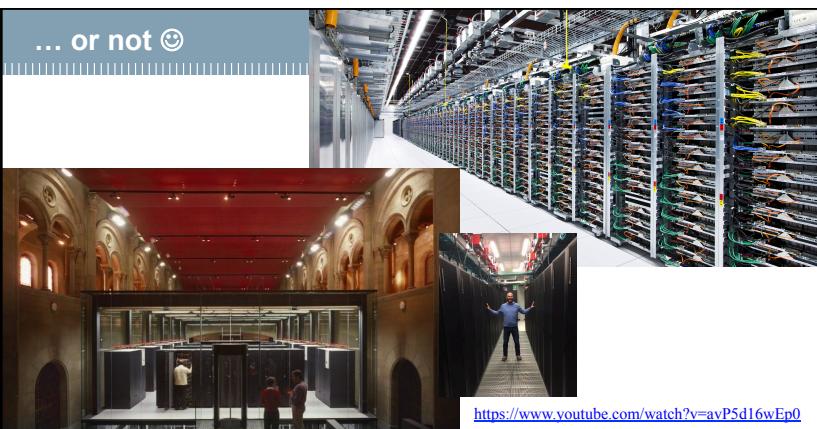
... easier to be managed ...



- 6 -

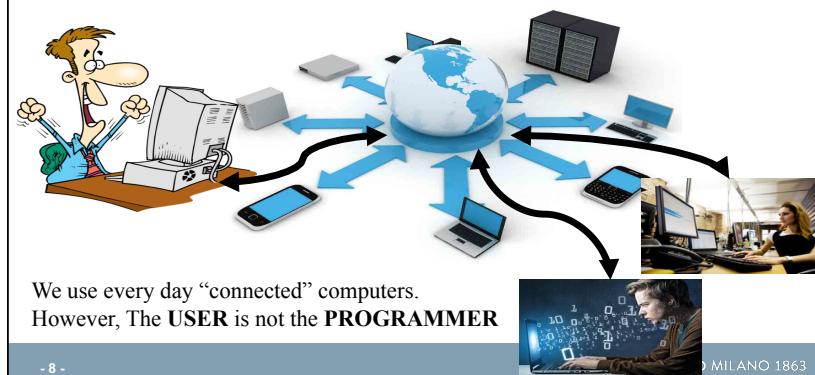
POLITECNICO MILANO 1863

... or not 😊



- 7 -

Connected and changing world...



We use every day “connected” computers.
However, The **USER** is not the **PROGRAMMER**

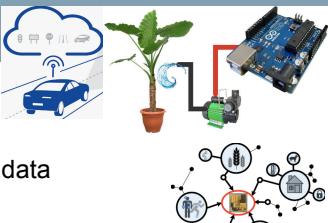
- 8 -

POLITECNICO MILANO 1863

Why be a Programmer?

To get some task done automatically

- Acting as user and programmer
- E.g. Processing collected sensor data



To produce something for others to use

- This is the task for a programmer as a job
- Fix a performance problem in a videogame
- Add a guestbook to a web site



- 9 -

POLITECNICO MILANO 1863

What is Code? Software? A Program?

A sequence of instructions

- It is a little piece of our intelligence in the computer
- We figure something out and then we encode it and then give it to someone else to save them the time and energy of figuring it out

A piece of creative art - particularly when we do a good job on user experience

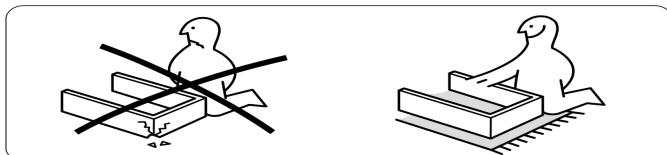
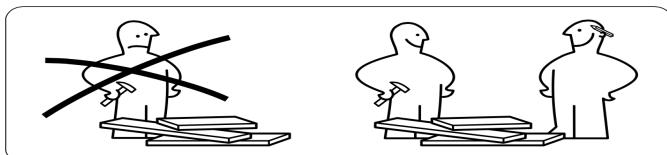
- Unfortunately the creativity of a programmer is not always exposed to the user. However, we know that it is there!



- 10 -

POLITECNICO MILANO 1863

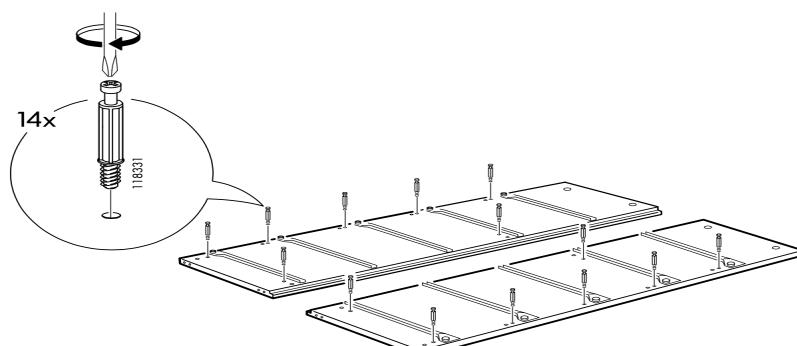
Programs for Humans



- 11 -

POLITECNICO MILANO 1863

Programs for Humans



- 12 -

POLITECNICO MILANO 1863

Programs for Humans

```
Read M1 and M2
Initialize an accumulator to zero (ACC=0)
Repeat |M2| times
    ACC=ACC+M1
If M2 is < 0
    change the sign to ACC
Write ACC
```

Multiplication by successive sums
ACC= M1*M2

Long
Polynomial
Division

Here are the steps required for Dividing by a Polynomial Containing More Than One Term (Long Division):

- Step 1: Make sure the polynomial is written in descending order. If any terms are missing, use a zero to fill in the missing term (this will help with the spacing).
- Step 2: Divide the term with the highest power inside the division symbol by the term with the highest power outside the division symbol.
- Step 3: Multiply (or distribute) the answer obtained in the previous step by the polynomial in front of the division symbol.
- Step 4: Subtract and bring down the next term.
- Step 5: Repeat Steps 2, 3, and 4 until there are no more terms to bring down.
- Step 6: Write the final answer. The term remaining after the last subtract step is the remainder and must be written as a fraction in the final answer.

- 13 -

POLITECNICO MILANO 1863

Python programs ... an example: "Is it a PANGRAM?"

```
text = input('Write a sentence: ')

Alphabet = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i',
            'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r',
            's', 't', 'u', 'v', 'w', 'x', 'y', 'z']

notAPangram = False

for letter in Alphabet:
    if letter not in text:
        notAPangram = True

if notAPangram:
    print('the string is not a pangram')
else:
    print('the string is a pangram')
```

A PANGRAM is a sentence including all the alphabet letters

- 14 -

POLITECNICO MILANO 1863

Exercise

Let's write a program/algorithm that using the following material puts together a sandwich

- 15 -

POLITECNICO MILANO 1863

Algorithms, programs, executors and programming languages

An algorithm is a sequence of **unambiguous instructions** for solving a problem in a finite amount of time

A program is the encoding of the algorithm by using a programming language with the goal of being understood by the executor (the computer)

What is the language that can be understood by the computer?

- 16 -

POLITECNICO MILANO 1863

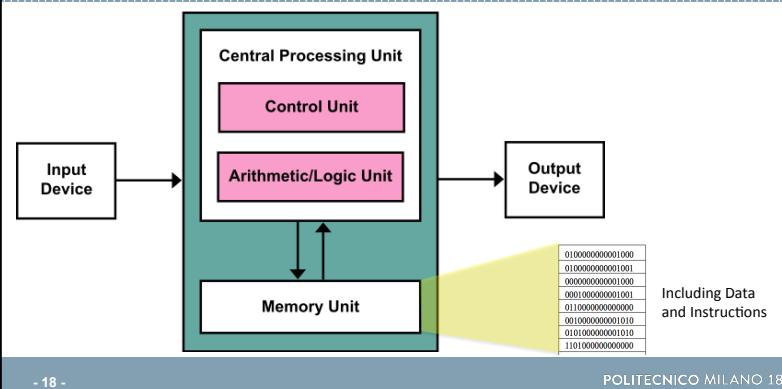
First of all, what do we mean as a computer?



- 17 -

POLITECNICO MILANO 1863

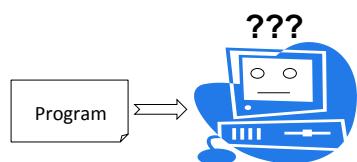
Von Neumann architecture



- 18 -

POLITECNICO MILANO 1863

Compiling and interpreting

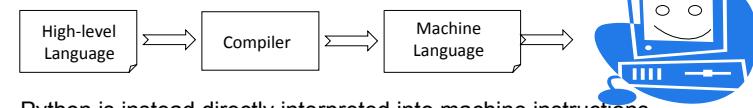


- 19 -

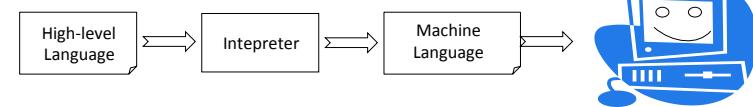
POLITECNICO MILANO 1863

Compiling and interpreting

Many languages (e.g. C/C++) require you to compile (translate) your program into a form that the machine understands.



Python is instead directly interpreted into machine instructions.



- 20 -

POLITECNICO MILANO 1863

Programming environments

An **integrated development environment (IDE)** is a software application that provides comprehensive facilities to computer programmers for software development

An IDE normally consists of:

- A source code editor to easily write code
- A compiler or interpreter, depending on the target language
- A syntax checker (including syntax highlighting)
- A debugger to control the correctness of the program execution

- 21 -

POLITECNICO MILANO 1863

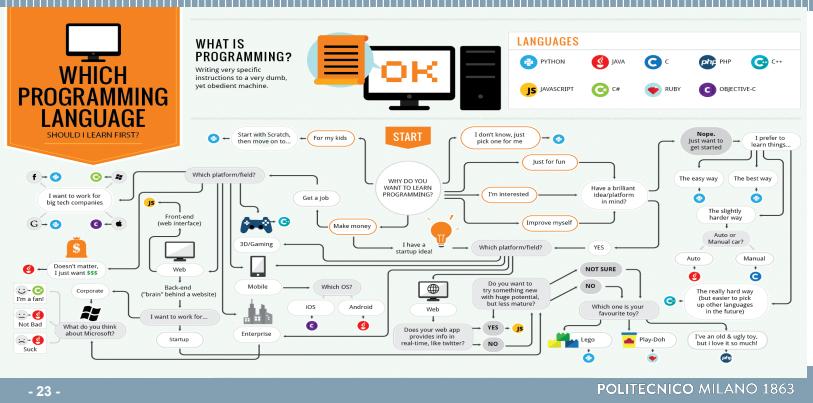
Python as Programming Language



- 22 -

POLITECNICO MILANO 1863

Languages



- 23 -

POLITECNICO MILANO 1863

Languages

THE LORD OF THE RINGS ANALOGY TO PROGRAMMING LANGUAGES

LANGUAGE	DIFFICULTY	DESCRIPTION	NOTES
Python	★★★	Helps beginers to learn programming quickly.	Very popular on all platforms, C# and Java are more difficult to learn.
Java	★★★	Very popular as the best programming language for beginners.	One of the most demanded languages in the world.
C	★★★	More power & works with everyone.	Powerful language for system and application development.
C++	★★★★	The power of C + the ease of learning.	A subset of C++ creates the file type.
JavaScript	★★★	Everyone thinks he is a good guy.	Everyone uses JavaScript, but only few people understand it.
C#	★★★★	Requires understanding of Java.	Very popular language for game development.
Ruby	★★★★	Fun & flexible.	A popular choice for enterprise and web development.
PHP	★★★	Used for websites.	Can be used to build websites with MySQL.
Objective C	★★★★	Lonely and love god.	Used for building iPhone and iPad apps.

- 24 -

POLITECNICO MILANO 1863

Why Python...

- Python is an **easy-to-learn** programming language that has some really useful features for a beginning programmer
- The code is quite **easy to read and more natural to write** when compared to other programming languages
- Python has some characteristics that make the learning process easier, giving the possibility to develop an animation or a simple **videogame in less than a week!**
- Python is the **fastest-growing programming language**



POLITECNICO MILANO 1863

- 25 -

Early Learner: Advises

We need to learn the Python language, thus

- We will “speak” gibberish like small children
- We will make lots of mistakes

When you make a mistake

- The computer says “syntax error”
- It seems like Python is cruel and unfeeling

You must remember

- You are intelligent and can learn
- The computer is simple and very fast, but “cannot learn”
 - at least for the moment ;-)



POLITECNICO MILANO 1863

- 26 -

Elements of Python

Vocabulary / Words

- Variables
- Reserved words



Sentence structure

- Valid syntax patterns

Story structure structure

- constructing a program for a purpose

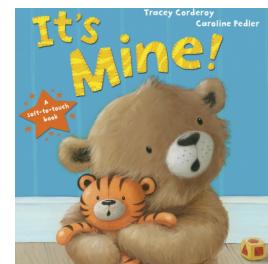
- 27 -

POLITECNICO MILANO 1863

Vocabulary: Reserved Words

Reserved words are words with a specific mean in Python

```
False  class  return  is      finally
None   if     for     lambda  continue
True   def    from    while   nonlocal
and    del    global  not    with
as     elif   try    or     yield
assert else   import  pass
break  except  in     raise
```



POLITECNICO MILANO 1863

- 28 -

Vocabulary: Variables

- A variable is a named place in the memory where a programmer can store data and later retrieve the data using the variable "name"
- Programmers get to choose the names of the variables
- You can change the contents of a variable in a later statement



x = 12.2
y = 14

x 12.2
y 14

- 29 -

POLITECNICO MILANO 1863

Vocabulary: Variables

- A variable is a named place in the memory where a programmer can store data and later retrieve the data using the variable "name"
- Programmers get to choose the names of the variables
- You can change the contents of a variable in a later statement



x = 12.2
y = 14
x = 100

x 12.2 100
y 14

- 30 -

POLITECNICO MILANO 1863

Sentences: Instructions

x = 2 ← Assignment statement
x = x + 2 ← Assignment with expression
print(x) ← Print statement

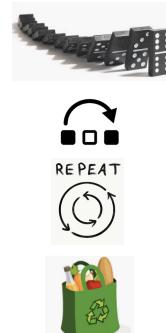
Variable Operator Constant Function

- 31 -

POLITECNICO MILANO 1863

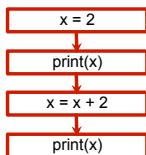
Program Steps or Program Flow

- Like a recipe or installation instructions, a program is a sequence of steps to be done in order.
- Some steps are conditional - they may be skipped.
- Sometimes a step or group of steps is to be repeated.
- Sometimes we store a set of steps to be used over and over as needed several places throughout the program (these are functions).



POLITECNICO MILANO 1863

Sequential Steps



Program:

```

x = 2
print(x)
x = x + 2
print(x)
  
```

Output:

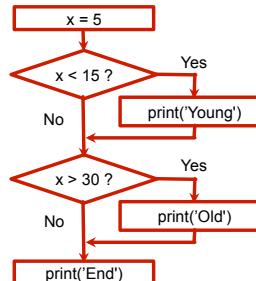
2
4

When a program is running, it flows from one step to the next. As programmers, we set up "paths" for the program to follow.

- 33 -

POLITECNICO MILANO 1863

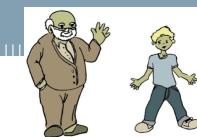
Conditional Steps



Program:

```

x = 5
if x < 10:
    print('Young')
if x > 20:
    print('Old')
print('End')
  
```



Output:
Young
End

Program:

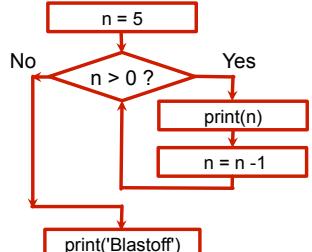
```

x = 5
if x < 10:
    print('Young')
if x > 20:
    print('Old')
print('End')
  
```

POLITECNICO MILANO 1863

- 34 -

Repeated Steps



Program:

```

n = 5
while n > 0 :
    print(n)
    n = n - 1
print('Blastoff!')
  
```

Output:

5
4
3
2
1

Blastoff!

Loops (repeated steps) have iteration variables that change each time through a loop.

- 35 -

POLITECNICO MILANO 1863

Classifying the instructions of our first example...

```

text = input('Write a sentence: ')
Alphabet = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i',
            'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r',
            's', 't', 'u', 'v', 'w', 'x', 'y', 'z']
notAPangram = False
for letter in Alphabet:
    if letter not in text:
        notAPangram = True
if notAPangram:
    print('the string is not a pangram')
else:
    print('the string is a pangram')
  
```

Sequential
Repeated
Conditional

POLITECNICO MILANO 1863

- 36 -

Words, sentences and paragraphs

```
text = input('Write a sentence: ')  
  
Alphabet = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i',  
            'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r',  
            's', 't', 'u', 'v', 'w', 'x', 'y', 'z']  
  
notAPangram = False  
  
for letter in Alphabet:  
    if letter not in text:  
        notAPangram = True  
  
if notAPangram:  
    print('the string is not a pangram')  
else:  
    print('the string is a pangram')
```

- 37 -

A short Python "Story"
about how identifying
pangrams

A word used to read
data from a user

A sentence about
considering all alphabet
letters

A paragraph about
informing the usage of
the program outcome

POLITECNICO MILANO 1863

... We have almost done!!!

CODING - TechCamp@POLIMI

Prof. Giacomo Palermo - Tutors: Davide Gachell, Andrea D'Amato

Introduction

Like humans, computers use multiple languages. We call them programming languages. A programming language is a particular way to talk to a computer, with the goal of describing the task to do. There are a lot of programming languages, and many of them are useful. However, there is no guarantee that one of them is not the best programming language. During the CODING course at the TechCamp, we are using Python.

Python is an easy-to-learn programming language that has some really useful features for a beginning programmer. The code is quite easy to read and more related to what you already know. It is also a very powerful programming language. Moreover, Python is free, so it is great for learning the basics of programming. The learning process starts giving the possibility to develop an animation or a simple videogame in less than a week (This is great for the TechCamp).

Even if you are not interested in developing something with Python, it is still a good idea to start with the basics. I personally think that when you learn the basics, then everything will be going to be more complicated and nice! Try to solve the proposed exercises, to improve your confidence with the language and its features. If you have any problem, ask the tutor or the professor. Once you understand the basics, the easier it will be to use them in complicated situations, such as videogames.

Don't think of these lessons as something boring that you have to work on. Try always to have fun in what you are doing. Programming can be fun! You can play with the language, experiment with different ways of programming as a way to create something. Be creative, and remember that your applications and the final videogames can always win the competition!

Remember the professor's rule: We are here to help you. Sometimes, it can happen that we do not have immediately the answer. However, we will find a way to find it.

Installing Python

Installing Python is generally easy, and nowadays many Linux and MacOS distributions include a recent Python version. Even some Windows computers now come with Python already installed. However, if you do not have Python installed it is enough to go to the official Python webpage <https://www.python.org/阮云>. Download the latest version of Python (at the time of writing, Python 3.6, and not Python 2.x). If you are not confident about the installation process, ask the Professor and the tutors.

38

POLITECNICO MILANO 1863



Input and Output in Python

print(SOMETHING) ... SOMETHING

print("CIAO")

CIAO

print(5)

5

```
print( 5 + 10 )  
print( 47 - 12 )  
print( 15 / 2 )  
print( 9 * 4 )
```



15
35
7.5
36

40

POLITECNICO MILANO 1863

print(SOMETHING) ... SOMETHING (2)

```
print( 2 ** 16 )
print( 37 // 3 )
print( 37 % 3 )
```



65536
12
1

```
# The following code prints the result of 3*9
print(3*9)
#print("CIAO!")
```

This is a comment



27

41

POLITECNICO MILANO 1863

Variables

Variable name: myVar
Assignment operator: =
Constant value: 100
print(myVar)



POLITECNICO MILANO 1863

42

Area of a rectangle! Area = Length * Width



How can I solve the problem by using what I just learnt in Python?

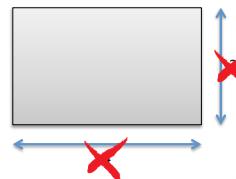


```
length = 4
width = 3
area = length * width
print(area)
```

43

POLITECNICO MILANO 1863

Area of a rectangle! Area = Length * Width (2)



How can I solve the problem by using what I just learnt in Python?



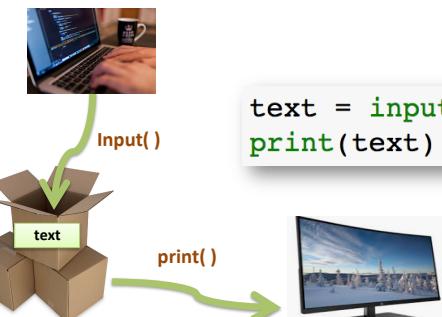
```
length = 4
width = 3
area = length * width
print(area)
```

How can I get those data as input?

44

POLITECNICO MILANO 1863

Obviously... with input()!!!



45

POLITECNICO MILANO 1863

Input(SOMETHING)

```
name = input("What is your name? ")  
print("Ciao " + name + "! Welcome to the CODING course!")
```

"+" when used with
STRINGS of characters
means concatenation

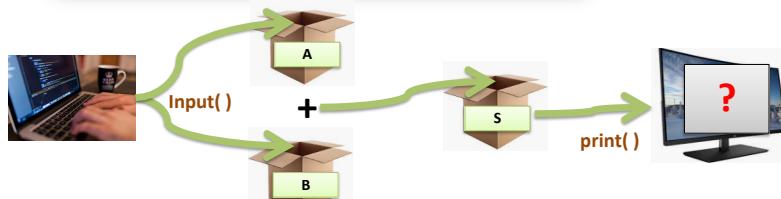
"Gian" + "luca" => "Gianluca"

```
subject = input("What is your preferred school subject? ")  
print(subject + " is nice. However, CODING is much much better")
```

POLITECNICO MILANO 1863

Sum A+B

```
a = input("Give me the value for 'a': ")  
b = input("Give me the value for 'b': ")  
s = a + b  
print(s)
```

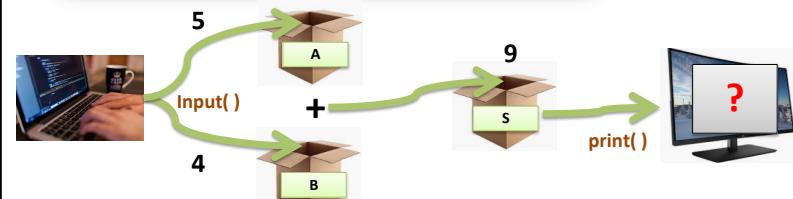


47

POLITECNICO MILANO 1863

Sum A+B

```
a = input("Give me the value for 'a': ")  
b = input("Give me the value for 'b': ")  
s = a + b  
print(s)
```

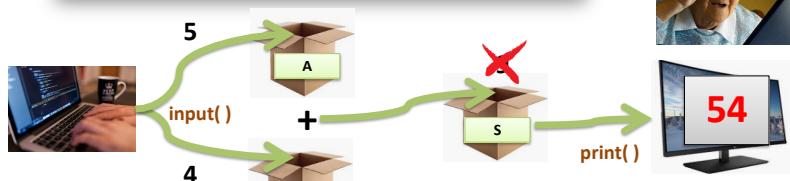


48

POLITECNICO MILANO 1863

Sum A+B

```
a = input("Give me the value for 'a': ")
b = input("Give me the value for 'b': ")
s = a + b
print(s)
```

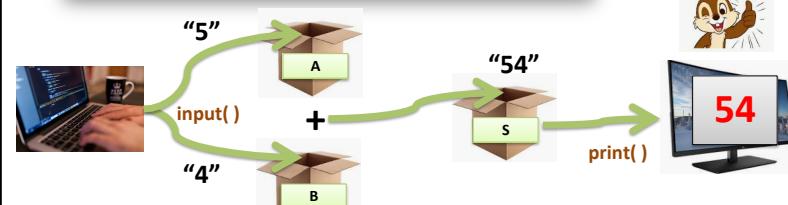


49

POLITECNICO MILANO 1863

Input() reads STRINGS of characters!

```
a = input("Give me the value for 'a': ")
b = input("Give me the value for 'b': ")
s = a + b
print(s)
```

50
Try using s = a*b
POLITECNICO MILANO 1863

Data conversion functions: cast operations

"5" $\xrightarrow{\text{int(...)}}$ 5 5 $\xrightarrow{\text{str(...)}}$ "5"

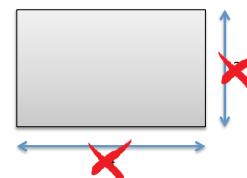
```
a = input("Give me the value for 'a': ")
b = input("Give me the value for 'b': ")
s = int(a) + int(b)
print("The result is: " + str(s))
```

Give me the value for 'a': 5
Give me the value for 'b': 4
The result is: 9

51

POLITECNICO MILANO 1863

Area of a rectangle! Area = Length * Width (3)



How can I solve the problem by using what I just learnt in Python?



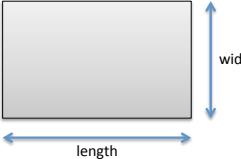
length = \times
width = \times
area = length * width
print(area)

How can I get those data as input?

52

POLITECNICO MILANO 1863

Area of a rectangle! Area = Length * Width (Final)



How can I solve the problem by using what I just learnt in Python?



```
print(" *** This program calculates the area of a rectangle ***")
length = int(input("Rectangle lenght: "))
width = int(input("Rectangle width: "))
area = length * width
print("The area of the rectangle is: " + str(area))
```

53 POLITECNICO MILANO 1863



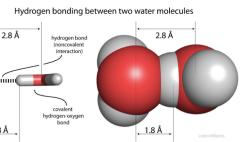
Floating Point Numbers and Modules

POLITECNICO MILANO 1863

Variable types



Distance in meter



Hydrogen bonding between two water molecules
2.8 Å hydrogen bond (noncovalent interaction)
1.8 Å covalent hydrogen bond
1.8 Å covalent bond

1.2345 = 12345×10^{-4}
exponent
mantissa

distance

Can't be simply an integer number

56 POLITECNICO MILANO 1863

Real numbers – float()

Python can also manipulate real numbers.

- Examples: 6.022 -15.9997 42.0 2.143e17

Decimal Separator

```
print(17 / 2) → 8.5
print(2 ** -2) → 0.25
```

54.36

x = float(input())
print("x = " + str(x))

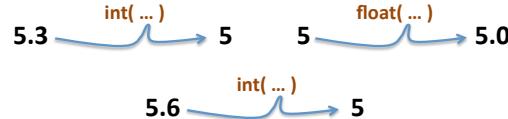
input() → “54.36”

float() → 54.36

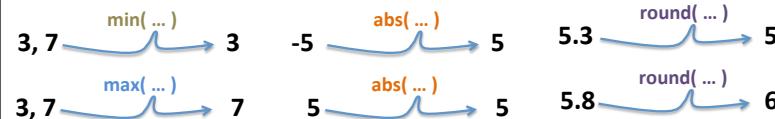
“54.36” ← str(x)

56 POLITECNICO MILANO 1863

Some useful built-in functions



In addition to conversion and I/O functions there are other useful built-in functions in Python

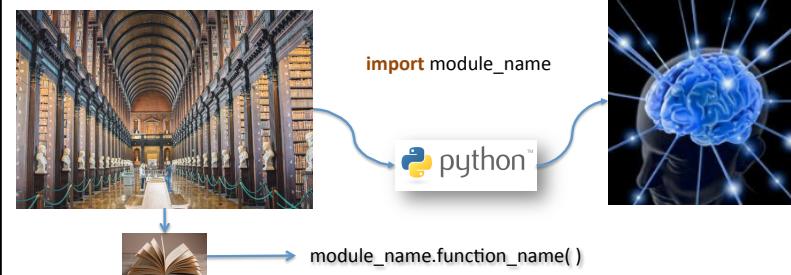


57

POLITECNICO MILANO 1863

Python Modules

Not all the functions are built-in...



58

POLITECNICO MILANO 1863

Useful Math functions

Command name	Description
ceil(value)	rounds up
floor(value)	rounds down
log(value)	logarithm, base e
log10(value)	logarithm, base 10
sin(value)	sine, in radians
cos(value)	cosine, in radians
sqrt(value)	square root
[...]	

```
import math
average_grade=5.6
good_mood = math.ceil(average_grade)
bad_mood = math.floor(average_grade)
print("If good mood: " + str(good_mood))
print("If bad mood: " + str(bad_mood))
```

If good mood: 6
If bad mood: 5

59

POLITECNICO MILANO 1863

Useful “Random” functions



```
import random
print(random.randint(1,100))
print(random.random())
print(random.uniform(-1,1))
```

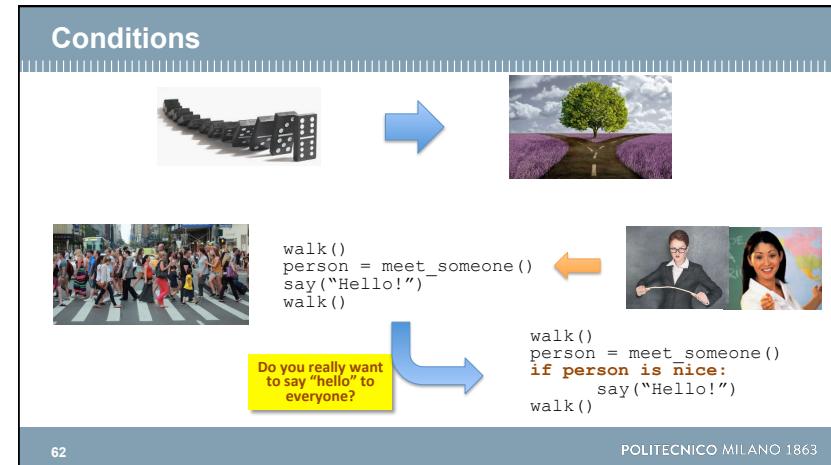
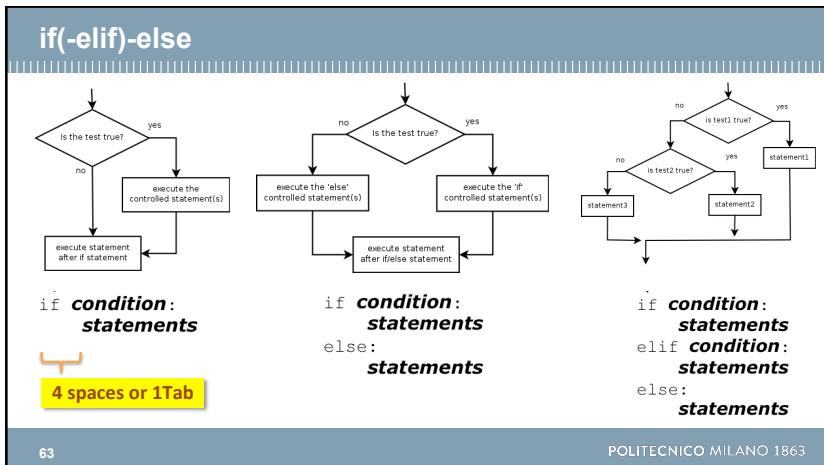
58
0.2543450307434296
-0.6826947939469641

60

POLITECNICO MILANO 1863

Conditional Statements

POLITECNICO
MILANO 1863



Conditions

Many logical expressions use relational operators:

Operator	Meaning	Example	Result
<code>==</code>	equals	<code>1 + 1 == 2</code>	True
<code>!=</code>	does not equal	<code>3.2 != 2.5</code>	True
<code><</code>	less than	<code>10 < 5</code>	False
<code>></code>	greater than	<code>10 > 5</code>	True
<code><=</code>	less than or equal to	<code>126 <= 100</code>	False
<code>>=</code>	greater than or equal to	<code>5.0 >= 5.0</code>	True

Logical expressions can be combined with logical operators:

Operator	Meaning	Example	Result
<code>and</code>	True if both True, False otherwise	<code>9 != 6 and 2 < 3</code>	True
<code>or</code>	True if at least one True, False otherwise	<code>2 == 3 or -1 < 5</code>	True
<code>not</code>	True if False, False otherwise	<code>not 7 > 0</code>	False

64

POLITECNICO MILANO 1863

Loops

POLITECNICO MILANO 1863

Need of iterative constructs

How many persons or for how much time?

```
[...]
walk()
person = meet_someone()
say("Hello!")
walk()
person = meet_someone()
say("Hello!")
walk()
person = meet_someone()
say("Hello!")
walk()
[...]
```

REPEAT

```
while not arrived:
    walk()
    person = meet_someone()
    say("Hello!")

for persons on the way:
    walk()
    person = meet_someone()
    say("Hello!")
```

POLITECNICO MILANO 1863

For loop

Repeats a set of statements over a group of values

```
for variableName in groupOfValues:
    statements
```

4 spaces or 1Tab

variableName = A
statements
variableName = B
statements
variableName = C
statements

67

POLITECNICO MILANO 1863

Range()

- The `range` function specifies a range of integers:
 - `range(stop)`
 - the integers between **0** (inclusive) and **stop** (exclusive)
 - `range(start, stop)`
 - the integers between **start** (inclusive) and **stop** (exclusive)
 - `range(start, stop, step)`
 - the integers between **start** (inclusive) and **stop** (exclusive) by **step**

```
for i in range(3):
    print(i)
print('The end!')
```

0	5 25	10
1	6 36	8
2	7 49	6

```
for i in range(10, 0, -2):
    print(i)
print('The end!')
```

10	4
8	2

68

POLITECNICO MILANO 1863

Exercise:

Write a program that computes the factorial ($n!$) of an integer

69

POLITECNICO MILANO 1863

Example and Exercise:

```
i = 0

while (i<3):
    print("[" + str(i)+"] I'm still inside the loop")
    i = i+1

print("I'm out of the loop, i=" + str(i))

[0] I'm still inside the loop
[1] I'm still inside the loop
[2] I'm still inside the loop
I'm out of the loop, i=3
```

Exercise: Write a program that reads numbers from the input until the input is an even number.

71

POLITECNICO MILANO 1863

While loop

Executes a group of statements as long as a condition is True.

- good for indefinite loops (repeat for an unknown number of times)

```
while (condition):
    Loop-Body
    4 spaces or 1Tab
```

Same as for IF

```
If condition == True:
    Loop-Body
If condition == True:
    Loop-Body
If condition == True:
    Loop-Body
...
[...]
```

70

POLITECNICO MILANO 1863



Lists & Maps

Lists

Useful to store multiple data (typically of the same type).

Months

```
January
February
March
April
May
June
July
August
September
October
November
December
```

Print all the months

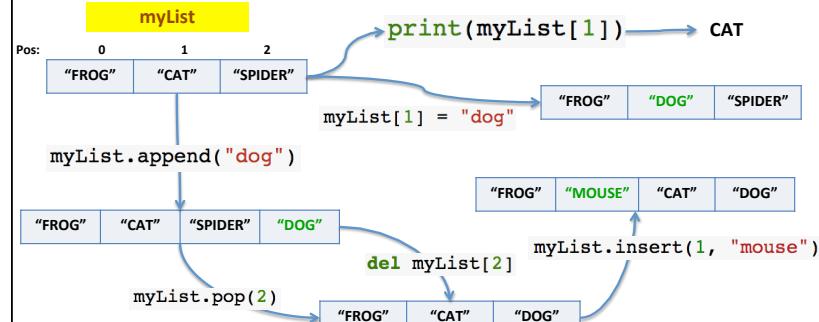
```
month1 = "January"
month2 = "February"
month3 = "March"
[...]
print(month1)
print(month2)
print(month3)
[...]
```

months = ["January",
"February", "March", ...]
print(months)
or
for x in months:
 print(x)

73

POLITECNICO MILANO 1863

Lists manipulation



74

POLITECNICO MILANO 1863

Other useful stuff on lists

```
myList1 = ['a', 'b', 'c']
myList2 = ['e', 'f', 'g']
myList3 = myList1 + myList2
print(myList3)
myList4 = myList1*3
print(myList4)
```

- `in` and `not in` operators

```
myList = [56, 76, 33, 56]
print(56 in myList)
print(14 in myList)
print(14 not in myList)
```

True
False
True

Others...

```
myList = [56, 76, 33, 56]
#Duplicate the list
newList=myList
#Sort of all the elements in the list
myList.sort()
#Reverse the elements in the list
myList.reverse()
#Count how many time there is the element 56
count56 = myList.count(56)
#Remove the first instance of 76 from the list
myList.remove(76)
```

`len(myList)` returns the number of elements of the list

75

POLITECNICO MILANO 1863

Something “more” on Strings

- Strings can be seen as a *sequence of characters*
- Characters in a string are *numbered as in lists*
- To access an individual character of a string:
 - `variableName [index]`

name	index	0	1	2	3	4	5	6	7
character	G	I	A	N	L	U	C	A	

```
name = "Gianluca"
print(name[0])
print(name[5])
```

G u

```
for myCharacter in 'MILANO':
    print(myCharacter)
```

M
I
L
A
N
O
The end!

76

POLITECNICO MILANO 1863



Modularity

POLITECNICO
MILANO 1863

Functions

Let's suppose that I would like to say "hello" to every person I meet in my school...

```

myName = input("What is your name? ")
print("Ciao " + myName + ". Good morning!")
print("This is my Python program!")
cmName = input("What is the name of one of your class mate? ")
print("Ciao " + cmName + ". Good morning!")
print("This is my Python program!")
mpName = input("What is the name of your Math professor? ")
print("Ciao " + mpName + ". Good morning!")
print("This is my Python program!")

```

```

def greetings(name):
    print("Ciao " + name + ". Good morning!")
    print("This is my Python program!")
    return

myName = input("What is your name? ")
greetings(myName)
cmName = input("What is the name of one of your class mate? ")
greetings(cmName)
mpName = input("What is the name of your Math professor? ")
greetings(mpName)

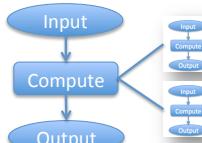
```

79

POLITECNICO MILANO 1863

Functions

- Functions help breaking our program into smaller and modular chunks
 - more organized and manageable
- Functions avoid repetition and makes code reusable



```

def function_name(parameters):
    body-of-the-function
    return value

```

78

POLITECNICO MILANO 1863



PyGame!!!

POLITECNICO
MILANO 1863

The graphic window

The basic step for a videogame is to create its graphic window

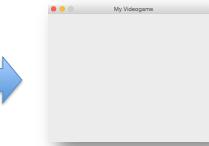
```
import pygame
pygame.init()           ← Module inclusion and initialization

sizeX = 400
sizeY = 300             ← Display settings declaration and customization

myScreen = pygame.display.set_mode((sizeX, sizeY))
pygame.display.set_caption("My Videogame")

pygame.quit()            ← Closure of the graphic module at the very end of the program
```

81



POLITECNICO MILANO 1863

Main game loop

```
gameActive = True
while (gameActive==True):
    #Clear screen
    myScreen.fill(pygame.Color("black"))

    #event detection
    for event in pygame.event.get():
        #print(event)
        if event.type==pygame.QUIT:
            gameActive = False

    #Display Update
    pygame.display.update()
```

- Fill the background of the screen
- Detect the event that are happening during the "game" life
- Update the screen with the new graphic elements (if any) ☺



POLITECNICO MILANO 1863

Graphical elements

Command	Description
<code>pygame.draw.line(surface,color,(x1,y1),(x2,y2),thickness)</code>	Draws a line between (x1, y1) and (x2, y2) with a certain thickness
<code>pygame.draw.circle(surface,color,(x,y),radius,thickness)</code>	Draws an circle centered in (x,y) , with a certain radius and thickness . If thickness is missing, the circle will be filled.
<code>pygame.draw.rect(surface,color,[x,y,width,height],thickness)</code>	Draws a rectangle with a top-left corner at (x,y) , and a certain width , height and thickness . If thickness is missing, the rectangle will be filled.
[...] more at https://www.pygame.org/docs/ref/draw.html	

```
pygame.draw.line(myScreen, pygame.Color("yellow"), (10,10), (350, 200), 1)
pygame.draw.circle(myScreen, pygame.Color("red"), (240,60), 30)
pygame.draw.rect(myScreen, pygame.Color("green"), [100, 150, 50, 35], 2)
```

83



Animation:

It is enough to draw the same item at different x/y positions within the game loop

```
xCircle = 240
gameActive = True
clock = pygame.time.Clock()

while (gameActive==True):
    #clear screen
    myScreen.fill(pygame.Color("black"))

    #geometric shapes
    pygame.draw.circle(myScreen, pygame.Color("red"), (xCircle,60), 30)
    pygame.draw.line(myScreen, pygame.Color("yellow"), (10,10), (350, 200), 1)
    pygame.draw.rect(myScreen, pygame.Color("green"), [100, 150, 50, 35], 2)
    #Position and dimension update
    xCircle = xCircle + 20

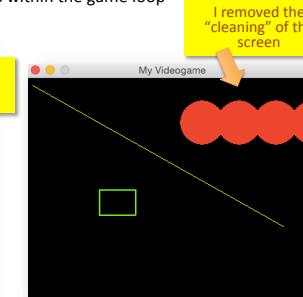
    #event detection
    for event in pygame.event.get():
        if event.type==pygame.QUIT:
            gameActive = False

    #Display Update
    pygame.display.update()

    #maximum frame rate
    clock.tick(50)

pygame.quit()
```

- We can use a variable and update its value
- It is used to limit the speed of the animation (FrameRate)



POLITECNICO MILANO 1863

More on animations:

You can easily make the circle bouncing from left to right, or in any direction

```
# Hint
# Using updateX as the value of the position update
# and xCircle as the x coordinate of the circle...
#
# xCircle= xCircle + updateX
# if (xCircle >= SizeX) or (xCircle <= 0):
#     updateX = -1*updateX
#
```

Or you can make disappearing the rectangle when hit by the ball

```
# Hint: define a variable that is used to decide
# if the rectangle should be drawn or not
#
# If hit == False:
#     pygame.draw.rect(...)
```

85

POLITECNICO MILANO 1863

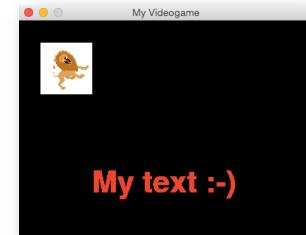
Text and Images

```
fontSize = 60
font = pygame.font.Font(None, fontSize)
textColor = pygame.Color("red")
text = font.render("My text", True, textColor)
#to get width and height of a text
#text.get_width() text.get_height()
myScreen.blit(text, (textXpos, textYpos))

image = pygame.image.load("lion.bmp")
#to get X and Y dimensions
#imageXdim = image.get_size()[0]
#imageYdim = image.get_size()[1]

# to scale the dimensions to 50x50
image = pygame.transform.scale(image,(50,50))
# to rotate the image by 45°
image = pygame.transform.rotate(image, 45)

# include the image in the game window
myScreen.blit(image, (imgXpos, imgYpos))
```



POLITECNICO MILANO 1863

86

Other possible interactions

```
for event in pygame.event.get():
    # ...
    if (event.type == pygame.MOUSEBUTTONDOWN):
        mouse_pos = pygame.mouse.get_pos()
        mouse_xPos=mouse_pos[0]
        mouse_yPos=mouse_pos[1]

    if event.type == pygame.KEYDOWN:
        if event.key == pygame.K_LEFT:
            xPosition = -5
        if event.key == pygame.K_RIGHT:
            xPosition = 5

        More event.key can be found at
        https://www.pygame.org/docs/ref/key.html
```

Mouse Pressed

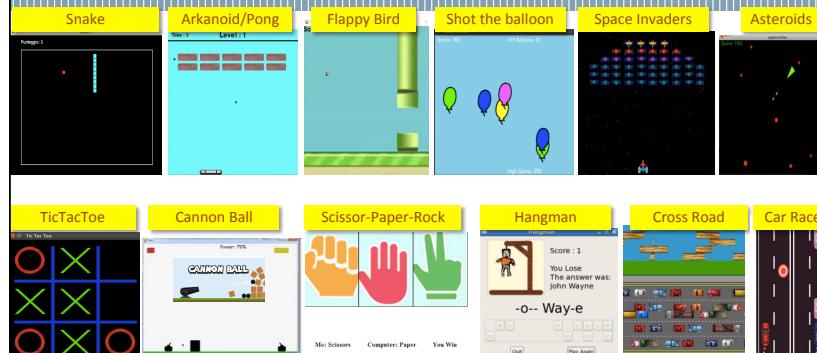
Detects the position of the cursor in the game window. It can be used also outside the event detection loop

Detects when a button of the keyboard has been pushed and which one

87

POLITECNICO MILANO 1863

Its your turn!!!



88 Other possibilities: 4in a Row, simplified version of Candy Crush, temple run ... POLITECNICO MILANO 1863