

“if statement”

Figure 1: “if statement”

## Python: an introduction

- program structure: constants, acquisition, computation, visualisation
- variables and constants
- `input()` and `print()`
- `import` statement to access existing python libraries, for instance `import math`: to be put at the beginning of our program
- if statement

### Details

- `input()` always returns a string, a sequence of character, therefore if that data represents a value (integer or real), it needs to be converted, by means of `int()` or `float()`.
- given a value (integer or real) it is possible to covert and obtain a string consisting of the numerical characters (and the "." separator) representing the value into a string, by means of the `str()` function.
- if – if-else – if-elif

### Exercises in class

#### Exe. #1:

Write a Python program that receives in input a positive integer value (it is surely so) representing the side of a square and computes the area. Finally, it shows the result.

#### Exe. #2:

Write a Python program that receives in input a positive integer value (it is surely so) representing the radius of a circle and it computes the length of the circumference and its area. Finally, it shows the results.

Initial version:

**Note:** for established values it is possible to *think* that there is a library available. This is the case of the `math` library.

**Note:** avoid putting constants directly in the code. It reduces readability and flexibility.

#### Exe. #3:

Write a python program that asks for a positive integer, representing a time expressed in seconds; the program converts it into days, hours, minutes, and seconds, and finally shows the result.

### Proposed exercises

#### Proposed #1:

Write a python program that receives a positive integer as input, representing an amount in euros. The program should calculate and display the number of €5 banknotes, €2 coins, and €1 coins needed to make up that amount, minimising the total number of pieces used.

#### Proposed #2:

Write a python program that, after receiving an integer value, calculates and displays 1 if the value is positive, 0 otherwise.

#### Proposed #3:

Write a python program that, after receiving an integer value, calculates and displays 1 if the value is odd, or 0 otherwise. Make a variation that does not use the remainder (modulus) operator.

#### Proposed #4:

Write a python program that takes three positive values and determines whether they form a Pythagorean triple. If they do, display 1; otherwise, display 0.

#### Proposed #5:

Write a python program that takes an integer value and displays + if the value is positive, - if it is negative, and a space character if it is zero.

**Proposed #6:**

Write a python program that takes an integer value in the range [1, 12], representing a month of the year (1 = January, etc.) and computes and displays the number of days in that month.

**Proposed #7:**

Write a python program that takes two integer values: the first is the month, expected to be in the interval [1, 12], the second one is the year: the program computes the number of days in that month, taking into account also the fact the year is a leap year or not.