

# Python – DI2006(P), DI2006(F)

Pablo Picazo-Sanchez

September 2022

## 1 Simple Data Types

**P1.1** Write a program that prints the string "Hello World!".

**P1.2** Write a program that stores the string "Hello World!" into a variable and later prints it.

**P1.3** Write a program that prints the string "Hello <name>!", where <name> is the name introduced by the user.

**P1.4** Write a program that outputs the result of the following operation:  
$$\left(\frac{3+2}{2.5}\right)^2$$

**P1.5** Write a program that asks the user for the number of working hours (a week) and the cost per hour. The program outputs the monthly salary.

**P1.6** Write a program that given  $n$ , a positive number ( $\mathbb{N}$ ), outputs the sum of all the numbers until  $n$ . The general formula is:  $\frac{n(n+1)}{2}$

**P1.7** Write a program that given the weight (kg) and height (meters), calculates and stores the Body Mass Index (BMI) into a variable. Later, it outputs the string: "Your BMI is <bmi>", where <bmi> is the BMI previously computed with exactly 2 decimals.

**P1.8** Write a program that given two numbers  $n, m \in \mathbb{N}$ , outputs the following string: "<n> divided by <m> gives a quotient of <q> with a remainder of <r>" where "<n>" and "<m>" are the numbers input by the user and "<q>" and "<r>" are the quotient and the remainder of the division, respectively.

**P1.9** Write a program to compute the future value of a specified principal amount, rate of interest, and a number of years.

**P1.10** An online toy shop is becoming popular because of two main items: dolls and cards. To deliver the items, the delivery company charges the toy shop based on the weight of the items (10SEK/gram). Knowing that a doll weights 112g and cards 75g, write a program that computes the total amount before the toy shop sends the package.

**P1.11** Imagine our bank offers us a rate of interest of 4%/year. These interests are not added to our account until next year. Write a program that given an amount (savings), computes the interests after the first, second, and third year. Truncate the output to 2 decimals.

**P1.12** A bakery sells cakes at 140 SEK. When a cake is not sold, the next day that cake has a 60% discount. Write a program that given the an amount of not-sold cakes it outputs the usual price and the final price (with the discount for not being sold).

## 2 Strings

**P2.1** Write a program that asks for the user name and a number  $n \in \mathbb{N}$  and prints that username  $n$  times (in different lines).

**P2.2** Write a program that given the first, middle and last name, it outputs the name (in different lines): 1) in lowercase; 2) in uppercase, and; 3) capitalized.

**P2.3** Write a program that given a name, it outputs: `<NAME> has <n> letters` where `<NAME>` is the input name (uppercase) and `<n>` the number of letters the name has.

**P2.4** Phone numbers usually follow *prefix-number-extension* pattern, where prefix is the country code (e.g., +34, +46), the number has a variable amount of digits, and the extension is the last two numbers. Write a program that given a number with that format, it outputs the phone number without the prefix nor the extension.

**P2.5** Write a program that given a sentence, it outputs the inverted sentence.

**P2.6** Write a program that given a sentence and a vowel, it replaces that vowel in the sentence by the uppercase.

**P2.7** Write a program that given an email, it replaces the domain (string after @) by hh.se.

**P2.8** Write a program that given an amount (in €) with two decimals, it outputs the amount of € and cents.

**P2.9** Given a date with the *dd/mm/aaaa* format, write a program that splits and outputs the day, month and year of the date.

**P2.10** Write a program that given a list of items split by the comma character (,), it outputs every item in different lines (No loops!).

**P2.11** Given an item, the price (SEK) and the number of items, it outputs the name of the product, its price (9 digits and 2 decimals), number of items, and the total cost (8 digits and 2 decimals).

### 3 Conditions

**P3.1** Write a program that given an age, it outputs whether it is an adult or not.

**P3.2** Write a program that given a string already stored, it outputs whether the user guessed it correctly or not (not case sensitive).

**P3.3** Write a program that given two numbers  $n, m \in \mathbb{N}$  it outputs  $\frac{n}{m}$  (if  $m \neq 0$  then it outputs an error).

**P3.4** Write a program that given a number  $n \in \mathbb{N}$  it outputs whether it is odd or even.

**P3.5** To declare taxes, you need to be 16 and your salary should be higher than 10,000SEK. Write a program that given the age and the salary, it outputs whether the user has to declare taxes or not.

**P3.6** In a particular course, teachers have split students into two groups A and B according to the sex and the name. Group A is composed of women whose name is before M and men whose name is after N. Group B is composed of all the rest. Write a program that given a name and the sex, it outputs which group the user belongs to.

**P3.7** A school has following rules for grading system:

Below 25:	F
25 to 45:	E
45 to 50:	D
50 to 60:	C
60 to 80:	B
Above 80:	A

Ask user to enter marks and print the corresponding grade.

**P3.8** Write a program to print the absolute value of a number entered by user (e.g., INPUT: 1 OUTPUT: 1; INPUT: -1 OUTPUT: 1)

**P3.9** The prices of a restaurant are calculated according to the age of the customers. Children under 4 do not pay. Children from 4 to 18, they pay 150SEK, while adults pay 180SEK. Write a program that given the age, it outputs the price.

**P3.10** An Italian restaurant has 5 pizzas classified into vegetarian and not vegetarian pizzas. The vegetarian pizzas are made of peppers or artichokes. For not vegetarian pizzas, they have pepperoni, ham, and salmon. Write a program that asks users for the pizza they want.

**P3.11** Fix the following code:

```
pass = input('Pass: ")
if pass in ['password']:
    print('OK')
else
    print('ERROR')
```

## 4 Loops

**P4.1** Write a program that given a word, it prints 10 times that word.

**P4.2** Convert the following loop into a for loop:

```
x = 4
while (x<=10):
    print(x)
    x += 2
```

**P4.3** Write a program that given a number, it outputs all the odd numbers from 1 to that number split by commas (,).

**P4.4** Write a countdown from a given number until 0 split by commas (,).

**P4.5** Given an amount, number of years, and rate interest, write a program that outputs the total amount every year.

**P4.6** Write a program that given a number, it outputs a triangle like the following (n=5):

```
*
**
***
****
*****
```

**P4.7** Write a program that prints the multiplication table from 1 to 10.

**P4.8** Write a program that prints numbers from 1 to 20 except multiple of 2 and 3.

**P4.9** Write a program that given a string already stored, it outputs whether the user guessed it correctly or not (not case sensitive) until the user guesses it.

**P4.10** Write a program that checks whether a number is prime or not.

**P4.11** Write a program that given a word, it reverses the word.

**P4.12** Given a sentence and a word, write a program that counts the number of times that word appears in the sentence.

**P4.13** Write a program that echoes what the user inputs until the string "end" is found.

## 5 List & Tuples

**P5.1** Write a program that prints a list.

**P5.2** Write a program that prints every element of a list in different lines.

**P5.3** Write a program that given 4 courses, it asks the user for the marks on every course and finally prints the course together with the mark.

**P5.4** Write a program that given a list of numbers split by commas (,) it sorts (ascending order) and outputs that list.

**P5.5** Write a program that prints a list of numbers from 1..10 in reverse order split by commas (,).

**P5.6** We say that a course is passed if the mark is at least 5 (up to 10). Write a program that given 4 courses, it asks the user for the marks on every course and removes the ones the student passed.

**P5.7** Given a list with all the letters of the alphabet, write a program that removes those letters whose position is multiple of 3.

**P5.8** Write a program that checks whether a word is palindrome or not (no loops).

**P5.9** Write a program that given a sentence, counts the number of appearance of every vowel in the sentence.

**P5.10** Given a list of numbers, write a program that prints the lowest and the largest numbers.

**P5.11** Given two vectors  $a=(1,2,3)$  and  $b=(-1,0,2)$  write a number that outputs the scalar product (dot product).

**P5.12** Write a program that generates a Fibonacci sequence.

**P5.13** Given a list of numbers, write a program that prints the mean ( $\mu = \frac{1}{n} \sum_{i=1}^n x_i$ ), the variance ( $\sigma^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2$ ), and the standard deviation ( $\sigma = \sqrt{\sigma^2}$ ).

## 6 Dictionaries

**P6.1** Write a program that given a sentence, counts the number of appearance of every vowel in the sentence and stores it in a dictionary.

**P6.2** Write a program that stores the name, age, address, and phone number of a user and stores it into a dictionary. Later it has to print "`<name> is <age> years old, lives in <address> and the phone number is <phone>`".

**P6.3** Write a program that stores all the values of the table above. Later, given a fruit and the weight (kg), it outputs the price of the purchase.

Fruit	Price
Banana	1.35
Apple	0.80
Pear	0.85
Orange	0.70

**P6.4** Write a program that given a date (dd/mm/aaaa) it replaces the month by its name and shows the string: "`The <dd> of <month_name>, <aaaa>`".

**P6.5** Write a program that given a dictionary (e.g., `{'Python': 6, 'JavaScript': 4, 'Pascal': 5}`) prints every subject together with the number of credits. Finally it should also show the sum of all the credits.

**P6.6** Write a program that stores elements into a dictionary until the user types 'N'.

**P6.7** Write a program that converts two lists (same length) into a dictionary

**P6.8** Write a program that merges two dictionaries into one.

**P6.9** Given the following dictionary:

```
sampleDict = {
    "class": {
        "student": {
            "name": "Mike",
            "marks": {
                "physics": 70,
                "history": 80
            }
        }
    }
}
```



```
}
```

Write a program that prints the value of 'history' key.

**6.10** Fix the following code:

```
agenda = {'Oliver':123456789, 'Erik':987654321}
```

```
def delete(agenda, user):  
    del(agenda[user])  
    return agenda[user]
```

```
print(delete(agenda, 'Ana'))
```

## 7 Sets

**P7.1** Given a Python list, write a program to add all its elements into a given set.

**P7.2** Write a program that returns a new set of identical items from two sets `set1 = 10, 20, 30, 40, 50` and `set2 = 30, 40, 50, 60, 70`

**P7.3** Write a Python program to return a new set with unique items from `set1 = 10, 20, 30, 40, 50` and `set2 = 30, 40, 50, 60, 70` sets.

**P7.4** Given two Python sets (`set1 = 10, 20, 30, 40, 50` and `set2 = 30, 40, 50, 60, 70`), write a program to update the first set with items that exist only in the first set and not in the second set.

**P7.5** Given two Python sets (`set1 = 10, 20, 30, 40, 50` and `set2 = 30, 40, 50, 60, 70`), write a program that returns a set of elements present in `set1` or `set2`, but not both.

**P7.6** Given two Python sets (`set1 = 10, 20, 30, 40, 50` and `set2 = 30, 40, 50, 60, 70`), write a program that checks if they have any elements in common. If yes, display the common elements

## 8 Functions

**P8.1** Write a function that outputs 'Hello World!'.

**P8.2** Write a program to create a function that takes two arguments, name and age, and print their value.

**P8.3** Write a program that prints the factorial ( $n! = 1 \cdot 2 \dots \cdot (n - 1) \cdot n$ ) of a number  $n$ .

**P8.4** Write a program to create function `func1()` to accept a variable length of arguments and print their value.

**P8.5** Write a program that calculates the area of a circle ( $\pi \cdot r^2$ ) as well as the volume of a cylinder  $\pi \cdot r^2 \cdot h$ .

**P8.6** Write a function that return the mean of a list of numbers.

**P8.7** Write a function that given a list of numbers, it returns another list with the square of the numbers.

**P8.8** Write a function that given a list of numbers, it returns a dictionary with the mean, variance and standard deviation of the list of numbers.

**P8.9** Write a function that computes the Greatest Common Divisor (GCD) and another one that computes the Least Common Multiple (LCM).

**P8.10** Write a function that converts a decimal number into binary and another one that converts a binary number into decimal.

**P8.11** Write a function that given two numbers  $n, m \in \mathbb{R}$  it returns the addition and subtraction.

**P8.12** Write a function that given two inputs, the employee's name and salary, it displays both. Also, if the salary is missing in the function call then assign default value 9000 to salary.

**P8.13** Write a function that allows computing sine (`sin`), cosine (`cos`), tangent (`tan`), exponential (`exp`), and logarithmic (`log`) functions. The program will apply such function to a range of numbers, going from 1..n, being n a number given by the user.

**HINT:** `from math import sin, cos, tan, exp, log`

**P8.14** Write a function that receives a function and a list and returns a list with the elements of the input list after applying the function.

**P8.15** Write a function that receives a Boolean function and a list and returns a list with the elements of the input list that satisfies condition of the function.

**P8.16** Write a function that given a string, it returns a dictionary with the letters and the frequency.

**P8.17** Write a function that given a dictionary of courses and grades, it outputs another dictionary with the courses in capital letters and the qualifications of the passed courses based according to the following table:

Grade	Qualifications
<5	F
<6	E
<7	D
<8	C
<9	B
<=10	A

**HINT:** Use `map()` function that returns a map object(which is an iterator) of the results after applying the given function to each item of a given iterable (list, tuple etc.).

**P8.18** Write a function that computes the module of a vector. **HINT:** Use `from functools import reduce`. This function performs a rolling-computation as specified by the passed function to the neighboring elements, by taking a function and an iterable as arguments, and returns the final computed value.

**P8.19** Write a function that given a list of number, it returns a list with only even numbers. **HINT:** use `filter()` function, a function that extracts elements from an iterable (list, tuple etc.) for which a function returns True.

## 9 Files

**P9.1** Write a program that given a number  $n \in [1, \dots, 10]$ , it creates a file named `table_n.txt` with the multiplication table of  $n$ .

**P9.2** Write a program that given a number  $n \in [1, \dots, 10]$ , it reads a file named `table_n.txt` and prints the result. If the file does not exist, it should handle the error.

**P9.3** Write a program that given two numbers  $n, m \in [1, \dots, 10]$ , it reads a file named `table_n.txt` and prints the line  $m$  as a result. If the file does not exist, it should handle the error.

**P9.4** Write a function that given the URL of a text file, it shows the number of words of that file. **HINT:** use the `from urllib import request` function to open the URL which can be either a string or a Request object.

**P9.5** Write a program that opens the a file with the GDP of every country in the European Union ([https://ec.europa.eu/eurostat/estat-navtree-portlet-prod/BulkDownloadListing?file=data/sdg\\_08\\_10.tsv.gz&unzip=true](https://ec.europa.eu/eurostat/estat-navtree-portlet-prod/BulkDownloadListing?file=data/sdg_08_10.tsv.gz&unzip=true)), and given a country code, it shows the GDP of all the years available.

**P9.6** Write a program that manages the agenda (clients and phone numbers) of a company. The program should allow the company to query, create, and delete items from the agenda. The agenda should be stored in a file called `agenda.txt` where users and phones should be split by comas (,) and every entry should be in different lines of the file.

**P9.7** Convert the following dictionary: `data = {"key1": "value1", "key2": "value2"}` into JSON format.

**P9.8** Access the value of `key2` from the following JSON: `key1": "value1", "key2": "value2"sampleJson = ""`

**P9.9** PrettyPrint following JSON `sampleJson = {"key1": "value1", "key2": "value2"}` with indent level 2 and key-value separators should be `(",", " = ")`.

**P9.10** Sort JSON keys in and write them into a file.

## 10 Classes

**P10.1** Create a Vehicle class without any variables and methods

**P10.2** Write a Python program to create a Vehicle class with `max_speed` and `mileage` instance attributes.

**P10.3** Create a child class `Bus` that will inherit all of the variables and methods of the `Vehicle` class

**P10.4** Create a `Bus` class that inherits from the `Vehicle` class. Give the capacity argument of `Bus.seating_capacity()` a default value of 50.

Use the following code for your parent `Vehicle` class.

```
class Vehicle:
    def __init__(self, name, max_speed, mileage):
        self.name = name
        self.max_speed = max_speed
        self.mileage = mileage

    def seating_capacity(self, capacity):
        return f"The seating capacity of a {self.name} is {
            capacity} passengers"
```

**P10.5** Define a class attribute `color` with a default value `white`, i.e., Every `Vehicle` should be white.

```
class Vehicle:

    def __init__(self, name, max_speed, mileage):
        self.name = name
        self.max_speed = max_speed
        self.mileage = mileage

class Bus(Vehicle):
    pass

class Car(Vehicle):
    pass
```

**P10.6** Convert the following `Vehicle` Object into JSON

```
class Vehicle:
    def __init__(self, name, engine, price):
        self.name = name
        self.engine = engine
```

```
        self.price = price  
  
vehicle = Vehicle("Toyota Rav4", "2.5L", 32000)
```

**P10.7** Convert the previous JSON into a Vehicle Object

## Acknowledgements & References

The exercises of this document are collected from different sources (e.g., <https://pynative.com>, <https://aprendeconalf.es/docencia/python>, <https://python.org>, <https://www.codesdope.com>) and own material.

Some references to keep in mind:

- <https://www.pythoncheatsheet.org>
- <https://towardsdatascience.com/collecting-data-science-cheat-sheets-d2cdf092855>