Information and Communication Technologies — S6

Formula Booklet

1 Python

1.1 Python types

- `int` (integer).
- `float` (floating-point number).
- `bool` (boolean).
- `str` (string).
- `list` (array of elements; elements may be of any type inside a list, even a list).

1.2 String manipulation

- `s.strip()` removes blank characters at the beginning and the end of a string s, in particular the end of line character "\n". For example `" This is it\n".strip()` gives `"This is it"`.
- `s.split(";")` transforms a string s into an array of strings. Each string of the array is a substring of s, contained in two consecutive places where the pattern (here, ";") is found in s.

  For example, `"a+24x+30-53t+10".split("+")` is the following array of size 4:
  `["a", "24x", "30-53t", "10"]`.
- `normalize(form, s)` returns the string s normalized, according to the given form. We used it to remove diacritics, see the example.

Example of use in Listing 1.

```python
import unicodedata
# This removes diacritics: accents, diaereses, umlauts, tildes, cedillas...
def normalize(text):
    nfkd_form = unicodedata.normalize('NFKD', text)
    return "".join([c for c in nfkd_form if not unicodedata.combining(c)])

f = open("my_file", "r", encoding="iso-8859-1")
for line in f:
    values = line.normalize().strip().split(";")
    # add your code here !
f.close()
```

Listing 1: Sample Python code for string manipulation.

1.3 Dictionaries

If you want to store the name of a pupil and the IP of its computer, you can build a dictonnary where the names will be the keys and the IP addresses the values like that:

```python
names_ip = {}
names_ip["john"] = "192.168.1.10"
names_ip["jane"] = "192.168.1.11"
```

You can get the number of items in a dicionary by typing `len(names_ip)` (here returns 2).

To loop over dictionaries, you can loop over the pairs at once (`items()`), or only on the keys (`name_ip.keys()`) or on the values (`name_ip.values()`).

```python
for name,ip in names_ip.items():
    print(name, " has ip ", ip)
for n in names_ip.keys():
    print("name = ", n)
for ip in names_ip.values():
    print("IP address = ", ip)
```

2 SQL

2.1 Count

The count keyword counts (!) how many different values are in a table. For example if you want to count the number of rows, it's just:

```
SELECT COUNT(*) FROM clients;
```

If you want to count the number of different first names, it will be instead:

```
SELECT COUNT(DISTINCT first_name) FROM clients;
```

2.2 Like

The like operator allows to compare strings to a pattern, instead of comparing them to a predetermined string. For example, if you want to find all users whose first name starts with “Jean-”, you would type:

```
SELECT * FROM clients WHERE first_name LIKE 'Jean-%';
```

This would find clients whose first name is “Jean-Paul” or “Jean-Marie” but not clients whose first name is “Jean”.

2.3 Join

We have seen that if we have Tables 1 and 2, then, if we want to print the title, the author and the literary genre of all the books in our library, we can write the following query:

```
SELECT title, author, genres.genre
FROM books, genres
WHERE books.genre=genres.identifier;
```

Id	Genre
1	Poetry
2	Novel
3	Theater

Table 1: Work 13, Exercise 1: The 3 literary genres.

Id	Author	Title	Genre
1	Baudelaire	Les fleurs du mal	1
2	Steinbeck	The Grapes of Wrath	2
3	Molière	L'avare	3
4	Corneille	Le Cid	3
5	Hugo	Les misérables	2
6	Prévert	Paroles	1

Table 2: Work 13, Exercise 1: The 6 books.

2.4 Question marks

Any SQL query executed with sqlite3 can have “?” in it. This means that the value will be given outside the query, as an additional array of arguments. If there are 3 “?” in the query, you need to provide an array of size 3 (any other size will lead to an error). Some examples of use in Listing 2.

```
#Example 1
author = 'Baudelaire'
title = 'Les fleurs du mal'
id_genre = 1
cursor.execute("INSERT INTO books(author,title,genre) VALUES (?, ?, ?);", [author, title, id_genre])

#Example 2
id_hotel = 4
cursor.execute("SELECT DISTINCT rooms.id FROM rooms WHERE rooms.hotel_id=?;", [id_hotel])
```

Listing 2: sqlite3 code using “?”.