

## 7th class

Henri Vandendriessche  
henri.vandendriessche@ens.fr

2022-11-08

# Terminal cheat sheet reminder

- Bash commands to navigate directories
  - Print Working Directory. Print the path of the current directory

```
pwd
```

- List all files of the current directory

```
ls folder
```

- Moving into folder1 and subfolder2 at once.

```
cd folder1/subfolder2
```

- Moving out of a directory

```
cd ..
```

- Going back and forth in the directory tree

```
cd ../../folder1/subfolder1
```

- Going back to the root directory

```
cd ~
```

- **"Tab"** to use the auto-completion
- **Ctrl + C** to stop a program execution
- Many more bash commands to use...

# Previously on Intro to Programming (Python)

- Data types:
  - integer
  - float
  - string
  - boolean
- **If, For and While** loops:
  - syntax
  - indentation
- Data collections:
  - list
  - tuple
  - set
  - dictionary
- Python Standard library
  - Python modules
  - Python built-in functions
- Functions:
  - Parameters and arguments
  - Return values
  - Scope of variable
- Building abstraction with :
  - Recursive functions
  - High order functions
- Read and write files

# Today

- csv
- Date format
- downloading

- Comma-separated-value (CSV) is a text file that uses comma to delimit data entries.

- Comma-separated-value (CSV) is a text file that uses comma to delimit data entries.
- .csv is a pretty simple and basic format that is often use when data needs to be compatible with many different programs and applications

- Comma-separated-value (CSV) is a text file that uses comma to delimit data entries.
- .csv is a pretty simple and basic format that is often use when data needs to be compatible with many different programs and applications
- csv files can be open by any text editors, spreadsheet like Excel and many other programs.

- Comma-separated-value (CSV) is a text file that uses comma to delimit data entries.
- .csv is a pretty simple and basic format that is often use when data needs to be compatible with many different programs and applications
- csv files can be open by any text editors, spreadsheet like Excel and many other programs.
- Data bases for example, very often support csv as an export format.



- Comma-separated-value (CSV) is a text file that uses comma to delimit data entries.
- .csv is a pretty simple and basic format that is often use when data needs to be compatible with many different programs and applications
- csv files can be open by any text editors, spreadsheet like Excel and many other programs.
- Data bases for example, very often support csv as an export format.
- example:

```
Name,Email,Phone,Address  
John Doe,johndoe@example.com,123-456-7890,123 Fake Street  
Michel Dupont,michel.dupont@test.com,098-765-4321,321 Ghost Avenue
```

## Date format 1/3

- Datetime module: <https://docs.python.org/3/library/datetime.html>

## Date format 1/3

- Datetime module: <https://docs.python.org/3/library/datetime.html>
- Module that supplies classes for manipulating dates and times

## Date format 1/3

- Datetime module: <https://docs.python.org/3/library/datetime.html>
- Module that supplies classes for manipulating dates and times
- Datetime module has a lot of different objects for example:

## Date format 1/3

- Datetime module: <https://docs.python.org/3/library/datetime.html>
- Module that supplies classes for manipulating dates and times
- Datetime module has a lot of different objects for example:
  - **timedelta** object represents a duration, the difference between two dates or times.

## Date format 1/3

- Datetime module: <https://docs.python.org/3/library/datetime.html>
- Module that supplies classes for manipulating dates and times
- Datetime module has a lot of different objects for example:
  - **timedelta** object represents a duration, the difference between two dates or times.
  - **date** object represents a date (year, month and day) in a calendar

## Date format 1/3

- Datetime module: <https://docs.python.org/3/library/datetime.html>
- Module that supplies classes for manipulating dates and times
- Datetime module has a lot of different objects for example:
  - **timedelta** object represents a duration, the difference between two dates or times.
  - **date** object represents a date (year, month and day) in a calendar
  - **datetime** object is a single object containing all the information from a date object and a time object

## Date format 1/3

- Datetime module: <https://docs.python.org/3/library/datetime.html>
- Module that supplies classes for manipulating dates and times
- Datetime module has a lot of different objects for example:
  - **timedelta** object represents a duration, the difference between two dates or times.
  - **date** object represents a date (year, month and day) in a calendar
  - **datetime** object is a single object containing all the information from a date object and a time object
  - **time** object represents a (local) time of day, independent of any particular day



## Date format 1/3

- Datetime module: <https://docs.python.org/3/library/datetime.html>
- Module that supplies classes for manipulating dates and times
- Datetime module has a lot of different objects for example:
  - **timedelta** object represents a duration, the difference between two dates or times.
  - **date** object represents a date (year, month and day) in a calendar
  - **datetime** object is a single object containing all the information from a date object and a time object
  - **time** object represents a (local) time of day, independent of any particular day
- Example 1

```
from datetime import *  
date1 = date(2022,11,8)  
date2 = date(2022,12,24)  
difference = date2 - date1  
print(difference , "Before Christmas")
```

```
## 46 days, 0:00:00 Before Christmas
```

```
print(type(date1))
```

```
## <class 'datetime.date'>
```

```
print(type(difference))
```

```
## <class 'datetime.timedelta'>
```

## Date format 2/3

- Example 2

```
from datetime import *
```

```
time = datetime.now()  
print(time)
```

```
## 2022-11-08 12:05:00.249065
```

```
time2 = datetime.now()  
print(time2)
```

```
## 2022-11-08 12:05:00.257845
```

```
if time > time2:  
    print("time is posterior to time2")  
elif time2 > time:  
    print("time2 is posterior to time")
```

```
## time2 is posterior to time
```

## Date format 3/3

- Example 3

```
from datetime import *

time = datetime(2022,11,8,14,30,10)
print(time)

time2 = date(2022,11,8)
print(time2)

if time > time2:
    print("time is posterior to time2")
elif time2 > time:
    print("time2 is posterior to time")
elif time == time2:
    print("They are the same date")
```

## TypeError: can't compare datetime.datetime to datetime.date

## Exercise

- We are gonna work today on the open data regarding location of cameras / video protection in Paris (until 2018)
- From the government website <https://www.data.gouv.fr> download the .ods file here: <https://www.data.gouv.fr/fr/datasets/r/c2f25a7d-c916-45b8-ba23-f658c63bae6b>
- Save the file in .csv
- Exercise 1: Return the number of cameras in Paris
- Exercise 2: Print all information about cameras in the 5th arrondissement
- Exercise 3: Number of cameras in each arrondissement
- Exercise 4: Print arrondissement with the highest number of cameras
- Exercise 5: Date of first camera installed in Paris
- Exercise 6: Number of all the camera installed the first date