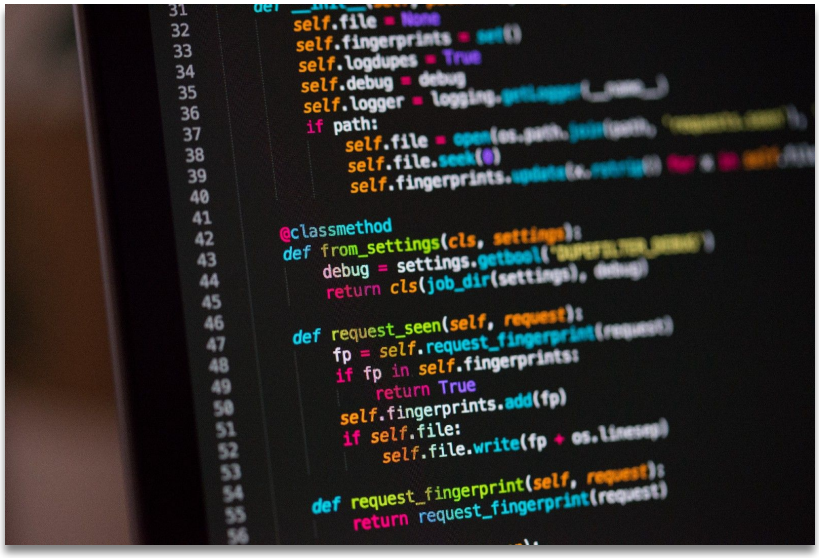
MS003 Programació bàsica



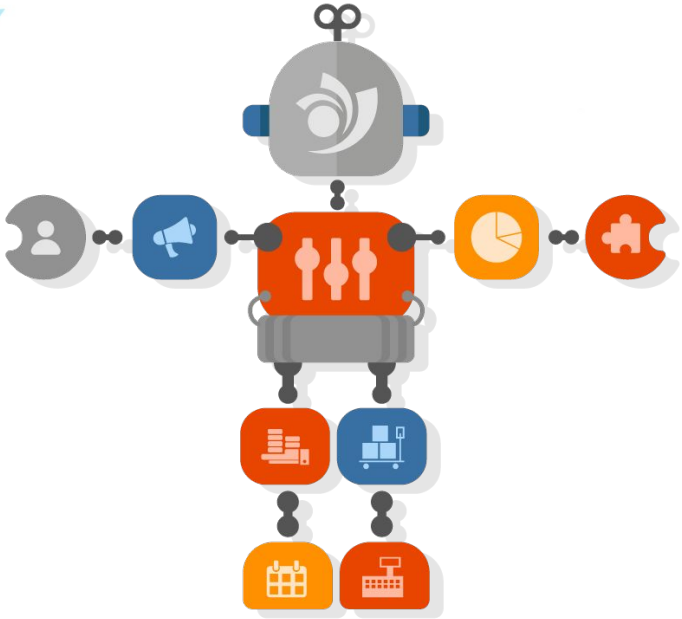
1r ASIX curs 24/25

Durada: 99 hores

5 Resultats d'Aprenentatge

institut sa palomera

Programació estructurada



RA 3. Escriu i prova programes senzills reconeixent i aplicant els fonaments de la programació modular.

Durada: 21 hores aprox.

Continguts treballats

1 Analitza els conceptes relacionats amb la programació modular.

2 Analitza els avantatges i la necessitat de la programació modular.

3 Aplica el concepte d'anàlisi descendent en l'elaboració de programes.

4 Modula correctament els programes realitzats.

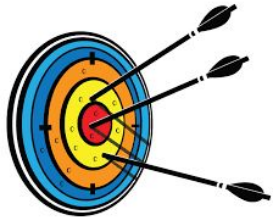
5 Realitza correctament les crides a funcions i la seva parametrització.

6 Té en compte l'àmbit de les variables en les crides a les funcions.

7 Prova, depura, comenta i documenta els programes.

8 Defineix el concepte de llibreries i la seva utilitat.

9 Utilitza llibreries en l'elaboració de programes.

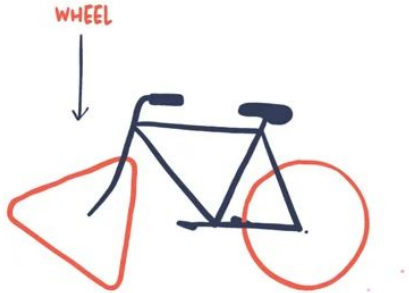


Llibreries

En qualsevol llenguatge de programació una llibreria és:

- una col·lecció de codi preescrit
- ofereix una sèrie de funcionalitats, normalment relacionades entre si
- que permeten resoldre problemes habituals sense reescriure el codi

De fet, ja n'hem utilitzat algunes.



Llibreries: tipus

Les llibreries poden ser de tres tipus:

- **integrades**: estan incloses en la instal·lació per defecte del llenguatge de programació.
  - math, random, datetime, tkinter
- **externes**: s'instal·len addicionalment, amb programes com pip
  - numpy, pandas…
- **personalitzades**: creades pel propi programador o equip de programació per reutilitzar codi

Llibreries integrades

Alguns exemples comuns de llibreries integrades a python són:

- **math** per operacions matemàtiques avançades
- **random** per generar nombres presudoaleatoris
- **datetime** per treballar amb dates i hores
- **tkinter** per fer interfícies gràfiques bàsiques

Llibreries integrades

Per utilitzar una llibreria integrada, cal utlitzar la paraula reservada **import** i el nom de la llibreria.

Després es pot cridar les funcions de la llibreria amb el nom de la mateixa seguida d'un punt

```
import math
print(math.sqrt(16))   # Arrel quadrada
print(math.pi)         # Valor de π
```

Llibreries integrades

Si només ens calen unes funcions específiques, podem reduir la quantitat de dades carregades i simplificar la crida definint exactament el que volem.

```
from math import sqrt, pi
print(sqrt(16))   # Arrel quadrada
print(pi)         # Valor de π
```

Llibreries integrades

Si es vol economitzar en nom, podem aplicar un sobrenom a la llibreria amb la paraula reservada **as**

```
import math as m
print(m.sqrt(16))   # Arrel quadrada
print(m.pi)         # Valor de π
```

Llibreries externes

Les llibreries externes funcionen de manera molt similar a les integrades, simplement cal instal·lar-les abans del seu ús. Algunes de molt comunes són

- **numpy** per treballar amb *arrays* i càlculs numèrics
- **pandas** per la manipulació i anàlisi de dades
- **matplotlib** per la generació de gràfics
- **requests** per fer peticions *http*
- **flask** o **django** per desenvolupament web

Llibreries externes

pip és el gestor de paquets oficial de Python

Per instal·lar una llibreria, només cal obrir el terminal i escriure:

- pip install <nom_de_llibreria>

Per exemple: `pip install numpy`

Per comprovar si funciona, podem fer:

```
import numpy
print(numpy.__version__)
```

Entorns virtuals

Un **entorn virtual** és una carpeta separada on es poden instal·lar llibreries de manera separada a la instal·lació global de Python. Això ens proporciona:

- **aïllament**, ja que cada projecte pot tenir versions diferents d'una llibreria concreta
- **menys conflictes**, ja que els canvis que fem a les llibreries instal·lades globalment no interfereixen en les del projecte

Entorns virtuals

Un **entorn virtual** és una carpeta separada on es poden instal·lar llibreries de manera separada a la instal·lació global de Python. Això ens proporciona:

- **aïllament**, ja que cada projecte pot tenir versions diferents d'una llibreria concreta
- **menys conflictes**, ja que els canvis que fem a les llibreries instal·lades globalment no interfereixen en les del projecte

Entorns virtuals

Per crear un entorn virtual hem d'escriure

- `python -m venv nom_entorn`

Per activar-lo:

- Windows: `<nom_entorn>\Scripts\activate`
- GNU/Linux o Mac: `source <nom_entorn>/bin/activate`

A partir d'aquest moment, ja podem instal·lar llibreries exclusives per aquest entorn amb `pip`

Per desactivar l'entorn virtual només cal escriure `deactivate`

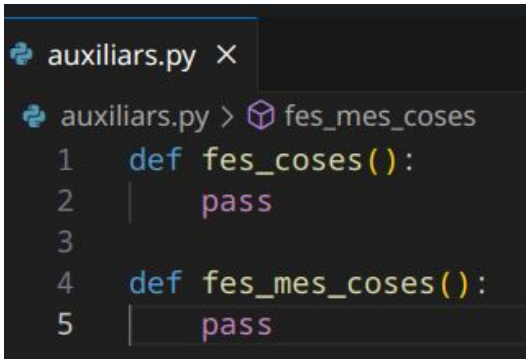
Entorns virtuals

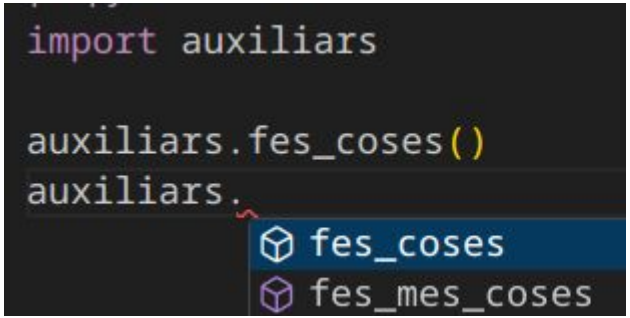
```
[jaumei@jaumei-tower ra3]$ python -m venv ra3
[jaumei@jaumei-tower ra3]$ source ra3/bin/activate
(ra3) [jaumei@jaumei-tower ra3]$ deactivate
[jaumei@jaumei-tower ra3]$
```

Llibreries personalitzades

La llibreria personalitzada més simple es crea programant una sèrie de funcions en un fitxer, que es posa al mateix directori que el programa.

Després s'importa aquest fitxer, i s'utilitzen les funcions





Exercicis

Teniu alguns exercicis d'aquest tema a l'AEA3.

Recordeu que cada setmana s'han d'entregar els obligatoris!