

Objectius: <ul style="list-style-type: none"> • Conèixer com definir rutes XPath i fer consultes amb XQuery
Instruccions: <ul style="list-style-type: none"> • Responen a l'espai de cada pregunta al codi del document relacionat. • Es valorarà la cura en la presentació del document i que segueixi l'estructura indicada.
Avaluació: <ul style="list-style-type: none"> • L'exercici 0 val 5 punts • L'exercici 1 val 2 punts • L'exercici 2 val 3 punts
Entrega: <ul style="list-style-type: none"> • Un arxiu .zip anomenat: PR21-NomCognom.zip ha de contenir: <ul style="list-style-type: none"> • Aquest document amb les respostes i en format .pdf • El projecte original de l'arxiu "base" amb l'estructura inicial i els arxius resolts segons l'enunciat

Nom i Cognom: [William Molina](#)

Materials:

Pels exercicis de XPath cal **Python** amb la llibreria lxml

Per l'exercici de XQuery cal **Java SDK** actualitzat (pot ser de OpenJDK) i l'arxiu "BaseX104.jar" que està a l'arxiu "base".

[L'arxiu "base"](#) del Moodle que conté:

- Els arxius d'exercicis (animals.py, videojocs.py, animals400.xq, ...)
- Els exemples
- Instruccions per fer anar el codi
- L'arxiu "BaseX104.jar" per fer anar BaseX

Feu servir Google per buscar els tutorials que us serveixin millor

- Exercici 0 - Emplena les funcions **tree.xpath()** de l'arxiu **“./xpath/animals.py”**, amb les rutes necessàries per resoldre els reptes que plantegen els comentaris de cada bloc de codi.

La sortida esperada és:

```
0 ['mamífers', 'peixos', 'crustacis', 'ocells']
1 ['mamífers', 'crustacis']
2 ['mamífers', 'peixos']
3 ['crustacis', 'ocells']
4 ['peixos']
5 ['gamba']
6 ['gallina']
7 ['colom']
8 ['estrella de mar', 'peix globus']
9 ['lleó', 'girafa', 'elefant']
10 ['perdiu', 'colom']
11 ['llampuga', 'salmó', 'truita', 'calamar', 'estrella de mar', 'peix globus']
12 ['langosta', 'camaró']
13 ['lleó', 'girafa']
14 ['elefant', 'gat', 'tauró', 'peix globus', 'cargol de mar', 'colom']
15 ['gos', 'peix globus', 'langosta', 'cargol de mar', 'colom']
16 ['salmó', 'truita', 'tauró', 'calamar']
17 ['lleó', 'girafa', 'gos', 'llampuga', 'salmó', 'truita', 'calamar', 'falcó', 'ànec', 'perdiu']
18 ['lleó', 'girafa', 'elefant', 'gat', 'gos', 'gallina', 'falcó', 'ànec', 'perdiu', 'colom']
19 ['lleó', 'elefant', 'gallina', 'falcó', 'colom']
```

- Exercici 1 - A la carpeta **“./xquery/”** hi ha diversos arxius **“.xq”** buits de codi però amb el comentari del què han de fer. Fes en aquests arxius, les consultes XQuery que permeten obtenir els següents resultats:

L'arxiu **“animals400.xq”** cal que mostri **“Animals de més de 400 i de color gris”**:

```
<animal><nom>elefant</nom><pes>5000</pes></animal>
<animal><nom>tauró</nom><pes>1000</pes></animal>
```

L'arxiu **“animals500.xq”** cal que mostri **“Animals de l'espècie amb id=1 que pesen menys de 500”**

```
calamar = 0.5
estrella de mar = 0.1
llampuga = 1
peix globus = 1.5
salmó = 0.1
truita = 2
```

L'arxiu **"videojocsBlanc.xq"** cal que mostri "Videoconsola amb color "blanc" i fabricant si son o no fabricats per "Nintendo":

Xbox One = No Wii = Si

L'arxiu **"videojocsUbisoft.xq"** cal que mostri "Videojocs desenvolupats per Ubisoft":

<joc><nom>Assassin's Creed III</nom><estudi>Ubisoft</estudi></joc> <joc><nom>Assassin's Creed IV</nom><estudi>Ubisoft Montreal</estudi></joc> <joc><nom>Watch Dogs</nom><estudi>Ubisoft Barcelona</estudi></joc>
--

L'arxiu **"ciutats.xq"** cal que faci un "join" entre les taules **"ciutats.xml"** i **"ciutats-estacions.xml"** per mostrar "Els habitants que té cada estació de cada ciutat":

Badalona - Badalona Estació : 0.2 Barcelona - Estació de França : 1.6 Barcelona - Estació de Sants : 1.6 Barcelona - Passeig de Gràcia : 1.6 Hospitalet - Estació de l'Hospitalet : 0.3 Montcada - Montcada - Ripollet : 0.2 Montcada - Montcada Bifurcació : 0.2 Sabadell - Sabadell Centre : 0.2 Sabadell - Sabadell Zona Hermètica : 0.2 Vic - Estació de Vic : 0.1

- Exercici 2 - Cerca a internet informació sobre XML, XPath i XQuery i respon breument a les següents preguntes:

1. Quins són els avantatges i inconvenients de fer servir XML per guardar informació, enlloc d'una base de dades relacional?

	Avantatges	Inconvenients
XML	Portabilidad Estructura jeràrquica No necesita predefinir la estructura	Poca eficiencia No hay consultas complejas Solo se almacenan datos como texto
Base de dades	Gran eficiencia para almacenar datos Se pueden realizar consultas complejas Diferentes tipos de datos	Se necesita definir la estructura de datos antes de almacenarlos No son tan portables como el XML Mantenimiento y configuración más complejo que XML

2. Perquè creus que és útil obtenir informació de manera parcial, dels documents XML?

A diferencia de las bases de datos, XML permite acceder a partes específicas de los datos sin tener que cargar todo el documento, evitando el uso de un gran volumen de datos.

Con esta carga parcial de información de los documentos XML, podemos ser más eficientes, optimizar recursos y adaptarnos a las necesidades específicas de cada situación dada.

3. A part de Python, quines altres eines permeten treballar amb arxius XML?

Algunas de las herramientas destacadas para trabajar con XML son:

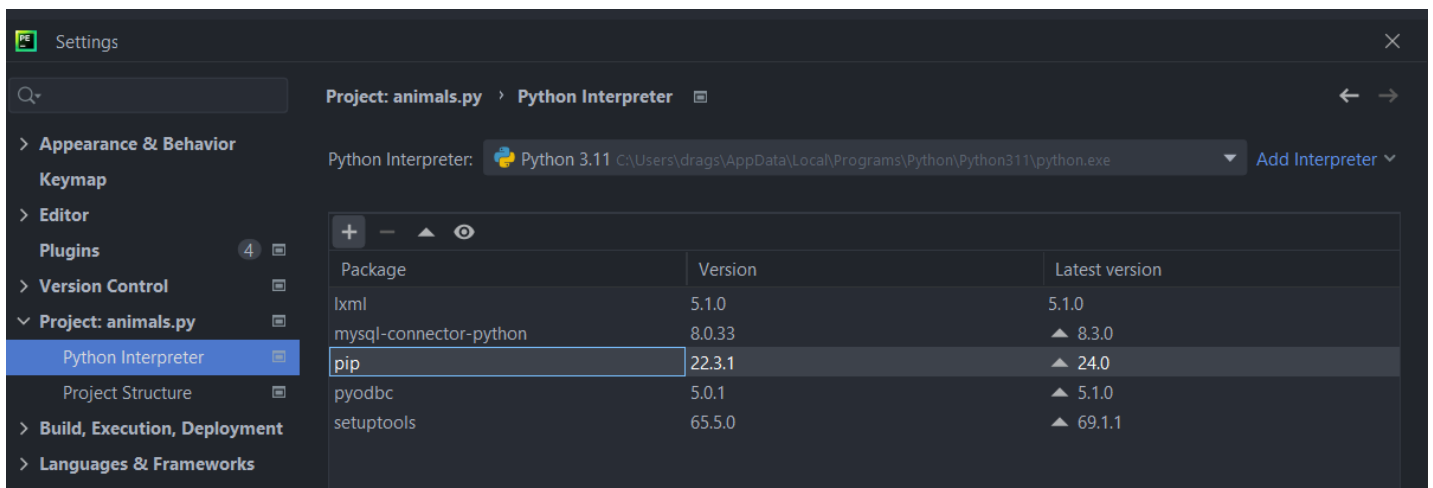
- Oxygen XML
- Notepad ++
- XML Explorer
- Emacs
- BaseX

4. Explica amb captures de pantalla els passos per executar una comanda XPath amb Python (què cal instal·lar, ...)

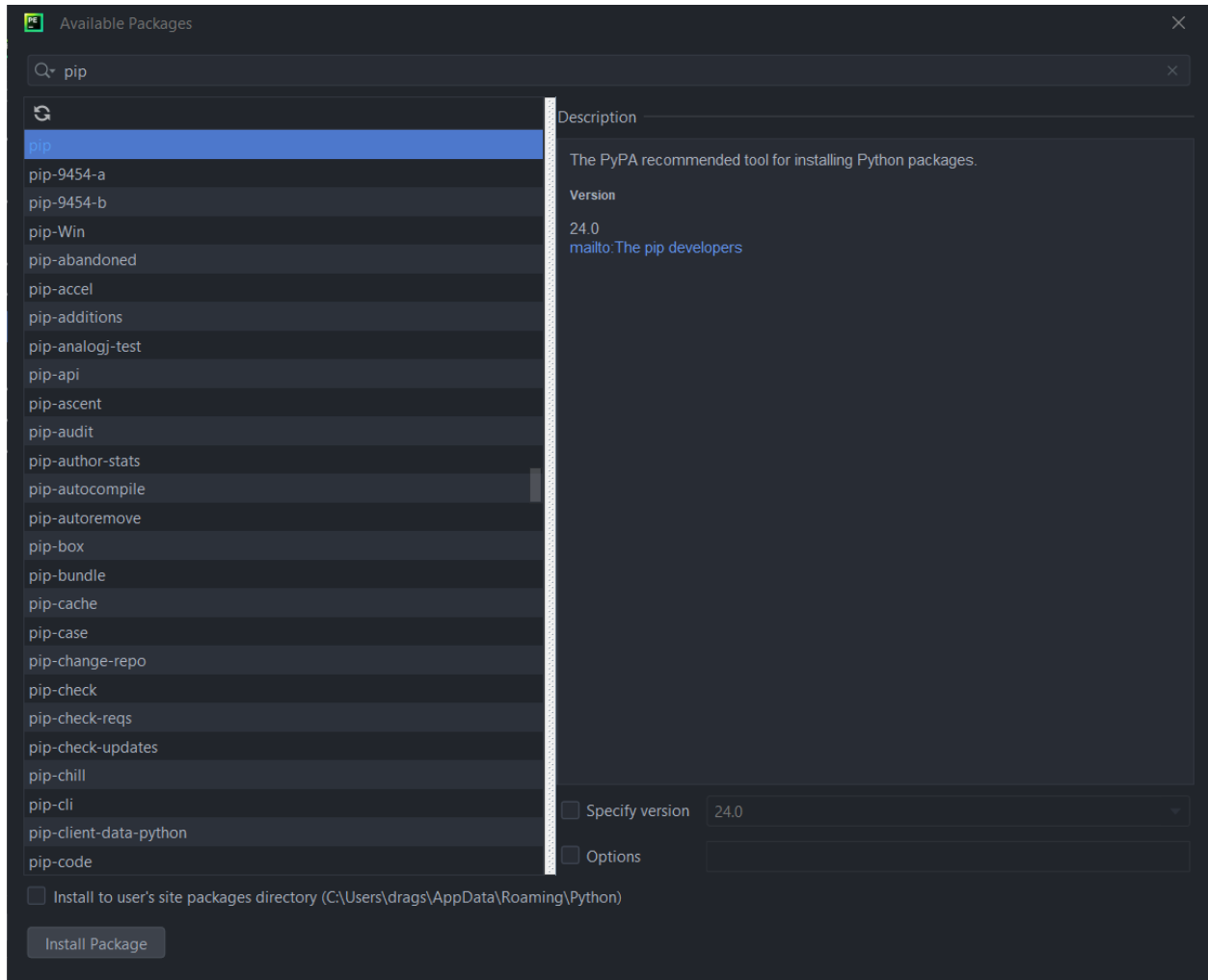
(Editor de código usado: Pycharm)

- El primer paso es instalar la librería necesaria: pip. Esta nos permitirá descargar la librería que estamos buscando: lxml.

- Para ello accedemos a File – Settings y vamos a este apartado (yo ya lo tengo instalado):



- Una vez pulsado el +, se nos abrirá un desplegable de Packacges donde tenemos que buscar pip. Lo seleccionamos y pulsamos el botón inferior de install:



- Cuando tengamos el paquete pip instalado, abrimos una terminal (cmd) y ejecutamos el siguiente comando: **pip install lxml**

```
Símbolo del sistema
Microsoft Windows [Versión 10.0.19045.4123]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\drags>pip install lxml
```

- Una vez instalada, abrimos un archivo .py e importamos la librería:

```
from lxml import etree
```

- Acto seguido, también necesitamos importar el xml que vamos a usar:
(se recomienda NO usar rutas absolutas)

```
xml = read_xml('ruta_al_teu_document.xml')  
tree = etree.XML(xml)
```

- Finalmente nos queda ejecutar una consulta Xpath:
(para ello se recomienda almacenar el resultado en una variable)

```
resultado = tree.xpath('ruta/de/elementos/xml')  
print(resultado)
```

- Cabe mencionar que se puede iterar sobre los resultados dados por una consulta de Xpath, ya sea para darle formato o para otros fines:
(ejemplo con un **for** para recorrer todos los resultados obtenidos)

```
for element in resultado:  
    print(element.text)
```

webgrafia:

<http://di002.edv.uniovi.es/~labra/cursos/ver06/pres/XMLBD.pdf>

<https://fp.uoc.fje.edu/blog/diferencias-entre-una-base-de-datos-xml-y-una-base-de-datos-relacional/><https://ayudaleyprotecciondatos.es/bases-de-datos/xml/>

<https://geekflare.com/best-xml-editors/>

- **Exercici opcional d'ampliació i reforç** - Emplena les funcions **tree.xpath()** de l'arxiu **“./xpath/videojocs.py”**, amb les rutes necessàries per resoldre els reptes que plantegen els comentaris de cada bloc de codi.

La sortida esperada és:

```
0 ['Playstation 4', 'Xbox One', 'Wii U', 'Playstation 3', 'Wii']
1 ['Sony', 'Microsoft', 'Nintendo']
2 ['Playstation 4', 'Playstation 3']
3 ['Wii U', 'Wii']
4 ['Playstation 3', 'Wii']
5 ['Grand Theft Auto', 'Gran Turismo 5', 'God of War III']
6 ['Super Mario World', 'Super Mario Kart', 'Super Mario Party', 'Super Mario Galaxy', 'Mario
  Tennis']
7 ['Call of Duty: Warzone', 'Assassin's Creed III', 'Need for Speed: Most Wanted', 'Grand
  Theft Auto']
8 ['Assassin's Creed IV', 'Watch Dogs', 'Assassin's Creed III']
9 ['Assassin's Creed IV', 'Watch Dogs', 'Assassin's Creed III']
10 ['Call of Duty: Warzone', 'Need for Speed: Most Wanted', 'Super Mario World', 'The Last
  Of Us']
11 ['Pokemon']
12 ['Little Big Planet', 'Uncharted 3', 'Gran Turismo 5']
13 ['Need for Speed: Most Wanted', 'Grand Theft Auto']
14 ['Call of Duty: Warzone', 'Assassin's Creed III', 'Grand Theft Auto']
15 ['Assassin's Creed III', 'Need for Speed: Most Wanted', 'Grand Theft Auto']
16 ['Need for Speed: Most Wanted']
17 ['Call of Duty: Warzone', 'Need for Speed: Most Wanted']
18 ['Need for Speed: Most Wanted']
19 ['Call of Duty: Warzone', 'Super Mario World', 'The Last Of Us']
20 ['Call of Duty: Warzone', 'The Last Of Us']
21 ['Super Mario World', 'Little Big Planet', 'Super Mario Galaxy']
22 ['Super Mario World', 'Super Mario Party', 'Super Mario Galaxy', 'Mario Tennis']
23 ['60', '55', '60']
24 ['Xbox One', 'Wii']
25 ['Wii', 'Wii U']
```