

Exercicis Python

Exercicis 1 - Variables, expressions i sentències

1.1. Escriu un programa que faci servir input per demanar-li a l'usuari el seu nom i després donar-li la benvinguda.

```
/home/arnaulisto/IdeaProjects/Exercicis.  
Quin és el teu nom?  
Arnau  
Hola Arnau  
  
Process finished with exit code 0
```

1.2. Escriu un programa per demanar-li a l'usuari el nombre d'hores i la tarifa per hora per calcular el salari brut.

/home/arnaulisto/IdeaProjects/Exercicis. Escriu el teu salari brut per hora 5 Escriu les hores treballades 10 El teu salari brut per hora és 50 Process finished with exit code 0	/home/arnaulisto/IdeaProjects/ExercicisA Escriu el teu salari brut per hora 2.5 Escriu les hores treballades 10.5 El teu salari brut per hora és 26 Process finished with exit code 0
---	---

1.3. Assumint que executem les següents sentències d'ASSIGNACIÓ:

```
ample = 17  
alt = 12.0
```

Per a cadascuna de les expressions següents, escriu el valor de l'expressió i el tipus (del valor de l'expressió).

1. ample / 2
2. ample / 2.0
3. alt / 3
4. 1 + 2 * 5

Utilitza l'intèrpret de Python per comprovar les teues respostes.

>>> ample/2	>>> ample/2.0	>>> alt/3	>>> 1 + 2 * 5
8.5	8.5	4.0	11

1.4. Escriu un programa que li demane a l'usuari una temperatura en graus Celsius, la converteix a graus Fahrenheit i imprimeix per pantalla la temperatura convertida.

```
Escriu la temperatura en Celcius:  
100  
La temperatura en Farenheit és: 212.0  
  
Process finished with exit code 0
```

Exercicis 2 - Execució condicional

2.1. Reescriu el programa del càlcul del salari per donar-li a l'empleat 1.5 vegades la tarifa horària per a totes les hores treballades que excedeixin de 40.

```
Escriu el teu salari brut per hora  
10  
Escriu les hores treballades  
45  
El teu salari brut per hora és 475.0  
  
Process finished with exit code 0
```

2.2. Reescriu el programa del salari usant try i except, de manera que el programa sigui capaç de gestionar entrades no numèriques en elegància, mostrant un missatge i sortint del programa. A continuació es mostren dos execucions del programa:

/home/arnaulisto/IdeaProjects/ExercicisA Escriu el teu salari brut per hora nom Error: Escriviu un nombre Process finished with exit code 1	Escriu el teu salari brut per hora 40 Escriu les hores treballades nom Error: Escriviu un nombre Process finished with exit code 1
---	---

2.3. Escriu un programa que sol·liciti una puntuació entre 0.0 i 1.0. Si la puntuació està fora d'este rang, mostra un missatge d'error. Si la puntuació està entre 0.0 i 1.0, mostra la qualificació usant la taula següent:

```
Escriu una puntuació entre 0.0 i 1.0
No vull
Error: Has d'introduir un número vàlid.

Process finished with exit code 0
```

```
Escriu una puntuació entre 0.0 i 1.0
0.3
insuficient

Process finished with exit code 0
```

```
Escriu una puntuació entre 0.0 i 1.0
0.65
suficient

Process finished with exit code 0
```

```
Escriu una puntuació entre 0.0 i 1.0
0.73
bé

Process finished with exit code 0
```

```
Escriu una puntuació entre 0.0 i 1.0
0.86
notable

Process finished with exit code 0
```

```
Escriu una puntuació entre 0.0 i 1.0
0.99
excel·lent

Process finished with exit code 0
```

Exercicis 3 - Bucles

3.1. Escriu un programa que llig repetidament números fins que l'usuari introduix "fi". Un cop s'hagi introduït "fi", mostra per pantalla el total, la quantitat de números i la mitjana. Si l'usuari introduceix qualsevol altra cosa que no sigui un número, detecta el seu error fent servir try i except, mostra un missatge d'error i passa al número següent.

```
/home/arnaulisto/IdeaProjects/Exercicis3> ./exercici3  
Introdueix un nombre: 4  
Introdueix un nombre: 5  
Introdueix un nombre: marrameu  
Error: dada errònia  
Introdueix un nombre: 7  
Introdueix un nombre: fi  
16 3 5.333333333333333  
  
Process finished with exit code 0
```

3.2. Escriu un altre programa que demane una llista de números com l'anterior i al final mostre per pantalla el màxim i mínim dels números, en comptes de la mitjana.

```
Introdueix una llista de nombres (de un en un):  
1  
Nombre gran = 1  
Nombre petit = 1  
0  
Nombre gran = 1  
Nombre petit = 0  
4  
Nombre gran = 4  
Nombre petit = 0  
fi  
Fi del programa
```

3.3. Dissenyeu un algorisme que calcule una aproximació al número π mitjançant el producte de Wallis:

```
/home/arnaulisto/IdeaProject  
3.0050125313283207  
  
Process finished with exit code 0
```

3.4. Dissenyeu un algorisme que determini si un número natural és de l'apocalipsi. Un número és de l'apocalipsi si conté tres dígits consecutius de la forma 666.

```
/home/arnaulisto/IdeaProjects/Exercici  
66677687  
66677687 és un nombre del apocalipsi  
  
Process finished with exit code 0
```

```
/home/arnaulisto/IdeaProjects/ExercicisAcces  
66166166166  
66166166166 NO és un nombre del apocalipsi  
  
Process finished with exit code 0
```

3.5. A la botiga dels germans Pastafullada és tradició presentar les llaunes de conserva apilades triangularment. Per exemple, sis llaunes es posen com:

* * *

Els germans tenen grans problemes per fer les comandes de llaunes. Fixeu-vos que no tot número de llaunes pot apilar-se triangularment, per exemple 8. Es demana que dissenyeu un algorisme que, donat un número natural, determini si representa un número adequat per muntar piles triangulars.

```
1
1 és un nombre vàlid
2
2 No és un nombre vàlid
3
3 és un nombre vàlid
6
6 és un nombre vàlid
21
21 és un nombre vàlid
28
28 és un nombre vàlid
36
36 és un nombre vàlid
61
61 No és un nombre vàlid
100
100 No és un nombre vàlid
|
```

3.6. Els germans Pastafullada estan enamorats del seu algorisme. Ara volen perfeccionar-lo per saber quins són els números adequats per fer piles triangulars i quadrades indistintament. El 36, per exemple, n'és un : Es demana que dissenyeu un algorisme que, donat un número natural, indiqui si és vàlid per fer piles triangulars i quadrades indistintament, o només d'un dels dos tipus o de cap.

3.7. En una pastisseria hi ha m tipus diferents de pastissos. De quantes maneres es poden escollir n pastissos? Són les combinacions en repetició d' m elements agafats d' n en n , és a dir:

$$\frac{(m + n - 1)!}{n! * (m - 1)!}$$

Se us demana dissenyar un algorisme que permeti calcular-ho.

OJITO: igual que passava amb els problemes del Joel, el primer nombre és el nombre de casos.

```
/home/arnaulisto/IdeaProjects/Exercic  
3  
2  
3  
6  
2  
3  
6  
4  
5  
70  
  
Process finished with exit code 0
```

3.8. Es demana que dissenyeu un algorisme que simule el llançament simultani d'un dau i d'una moneda 100 vegades, i informe de quantes vegades ha sortit simultàniament un número senar en el dau i cara a la moneda.

```
/home/arnaulisto/IdeaProjects/Exercici:  
Cara ha sortit 61 vegades  
Un nombre senar ha sortit 43 vegades
```

3.9. Dissenyeu un algorisme que determine si un número natural és perfecte. Un número natural s'anomena perfecte quan és igual a la suma de tots els seus divisors, llevat d'ell mateix. Per exemple, 28 és perfecte perquè $28 = 1 + 2 + 4 + 7 + 14$.

```
Introduceix un nombre: 5  
496 és un nombre perfecte  
  
Process finished with exit code 0
```

```
Introduceix un nombre: 7  
8128 és un nombre perfecte  
  
Process finished with exit code 0
```

3.10. Dissenyeu un algorisme per determinar si un número natural és primer. Un número $n > 1$ és primer si els seus únics divisors (factors) són 1 i ell mateix.

```
Introdueix un nombre: (0 per sortir): 2
Es primer
Introdueix un nombre: (0 per sortir): 4
No es primer
Introdueix un nombre: (0 per sortir): 6
No es primer
Introdueix un nombre: (0 per sortir): 7
Es primer
Introdueix un nombre: (0 per sortir): 8

Process finished with exit code 0
```

Exercicis 4 - Cadenes (i funcions)

4.1. Investigueu què fan els següents mètodes de les cadenes, vos seran útils per fer els següents exercicis:

'capitalize', 'center', 'count', 'decode', 'encode', 'endswith', 'expandtabs', 'find', 'format', 'index', 'isalnum', 'isalpha', 'isdigit', 'islower', 'isspace', 'istitle', 'isupper', 'join', 'ljust', 'lower', 'lstrip', 'partition', 'replace', 'rfind', 'rindex', 'rjust', 'rpartition', 'rsplit', 'rstrip', 'split', 'splitlines', 'startswith', 'strip', 'swapcase', 'title', 'translate', 'upper', 'zfill'

Ho fico en un document apart perquè ocupa casi 2 pàgines. És, bàsicament, una taula feta per ChatGPT que utilitzaré per fer els exercicis.

https://docs.google.com/document/d/10TWCJjoh4wV2lacgloVJXJpd5j5L6MpTOZd_3SVHBC4/edit?tab=t.0

2. Escriu un bucle **while** que comence en l'últim caràcter d'una cadena introduïda per l'usuari i faigui el seu recorregut cap enrere fins al primer caràcter de la mateixa, mostrant cada lletra en una línia separada.

```
Introdueix una cadena de caracters: hola test
t
s
e
t
a
l
o
h

Process finished with exit code 0
```

4.3. Atès que **fruita** és una variable de tipo cadena, què vol dir **fruita [:]**? Mostra-ho a un programa d'exemple.

fruita [] fa referència a tots els caràcters de la variable, si volguem fer un slicing hauríem de fer fruita [inici:final] per indicar d'on comencem a tallar-la i fins on. Si no fiquem res, els agafariem tots.

Per exemple:

```
/home/arnaulisto/IdeaProjects/Exerc  
Hello, World!  
Hello, World!  
Hello,  
  
Process finished with exit code 0
```

4.4. Fes un programa en una funció anomenada **comptador**, que accepte una cadena i una lletra com a arguments i compte/retorne el número de vegades que apareix la lletra a la cadena.

```
al habla el capitan  
a  
5  
  
Process finished with exit code 0
```

```
al habal el capitan  
u  
0  
  
Process finished with exit code 0
```

4.5. Hi ha un mètode de cadena anomenat **count**. Escriu un programa que el cride i compte el nombre de vegades que apareix la lletra "a" a 'banana'.

```
al habla el capitan (ara en count)  
a  
7  
  
Process finished with exit code 0
```

4.6. Pren el codi en Python següent, que emmagatzema una cadena: '

```
cad = 'X-DSPAM-Confidence: 0.8475 Km'
```

Utilitza **find** i llescat de cadenes (slicing) per extreure la porció de la cadena corresponent al número, i després fes servir la funció float per convertir la cadena extreta en un nombre en punt flotant.

```
0.8475, <class 'float'>  
  
Process finished with exit  
1
```

4.7. Escriu 2 funcions per, donada una cadena de caràcters:

- a) Retornar la cadena extreent cada dos caràcters del paràmetre. Ex.: 'recta' hauria de retornar 'rca'

```
recta
rca
Process finished.
```

- b) Retornar la cadena concatenant el paràmetre en un sentit i en sentit invers. Ex: 'reflex' torna 'reflexxelfer'.

```
reflex
reflexxelfer
Process finished.
```

4.8. Escriu 4 funcions, que donada una cadena i un caràcter:

- a) Inserix el caràcter entre cada lletra de la cadena. Ej: 'separar' i ',' --> 's,e,p,a,r,a,r'

```
separar
,
s,e,p,a,r,a,r
```

- b) Substituix tots els espais pel caràcter. Ex: 'el meu fitxer de text.txt' i '_' hauria de tornar 'el_meu_fitxer_de_text.txt'

```
Introduceix una cadena de caracters:el meu fitxer de text.txt
Introduceix un caracter:_ 
el_meu_fitxer_de_text.txt
```

- c) Substituix tots els díigits a la cadena pel caràcter. Ex: 'la seua clau és: 1540' i 'X' hauria de tornar 'la seua clau és: XXXX'

```
Ea generarà una clau aleatòria
la seva clau és XXXX
la clau era: 4948
```

- d) Inserix el caràcter cada 3 díigits en la cadena. Ex. '2552552550' i '.' hauria de tornar '255.255.255.0'

```
Introduir cadena: 2552552550
Introduir caracter: .
255.255.255.0
```

4.9. Escriu 4 funcions que, donada una cadena de caràcters:

- a) Retorne només les lletres consonants. Per exemple, si rep 'algoritmes' o 'logaritmes' ha de tornar 'lgrtms'.

```
Introduir cadena: algoritmes
lgrtms
Process finished with exit code 0
```

- b) Retorne només les lletres vocals. Per exemple, si rep 'sense consonants' ha de tornar 'eeooa'.

```
Introduir cadena: sense consonants
eeooa
Process finished with exit code 0
```

- c) Reemplaça cada vocal per la següent vocal. Per exemple, si rep 'vestuari' ha de tornar 'vistaero'.

```
Introduceix cadena: vestuari
vistaero
```

- d) Indique si es tracta d'un palíndrom. Per exemple, 'A flacs ell escalfa' és un palíndrom (es llegix igual d'esquerra a dreta que de dreta a esquerra).

```
Introduceix una cadena de caracters: A flacs ell escalfa
Es palindrom
```

```
Introduceix una cadena de caracters: hola
No es palindrom
```

4.10. Escriure 2 funcions que, donades dues cadenes de caràcters:

- a) Indique si la segona cadena és una subcadena de la primera. Per exemple, 'cadena' és una subcadena de 'subcadena'.

```
Introduceix la cadena principal: subcadena
Introduceix la subcadena: cadena
És una subcadena
```

- b) Retorne la que sigue anterior en ordre alfabètic. Per exemple, si rep 'kde' i 'gnome' ha de tornar 'gnome'.

```
Arnau és un crack
Non tens ni idea, Arnau
Arnau és un crack

Process finished with exit code 0
```

Exercicis 5 - Llistes

5.1. Escriu una funció anomenada **retalla**, que prengue una llista, la modifique eliminant els elements primer i últim, i retorna None. Mostra què ha passat en la llista paràmetre després de la crida a la funció per saber si s'ha modificat o no.

```
[2, 3]
None

Process finished with exit code 0
```

5.2. Escriu una funció anomenada **centre**, que prengue una llista i retorna una altra que contingue tots els elements de l'original, menys el primer i l'últim.

```
[2, 3]

Process finished with exit code 0
```

5.3. Reescriu el programa que demana a l'usuari una llista de números i imprimeix en pantalla el màxim i mínim dels nombres introduïts, quan l'usuari introduceix "fi". Escriu ara el programa de manera que emmagatzeme els números que l'usuari introduceixi en una llista i usa les funcions max() i min() per calcular els nombres màxim i mínim després que el bucle acabe.

```
Introduceix un número: 1
Introduceix un número: 2
Introduceix un número: 7
Introduceix un número: 1111
Introduceix un número: -78
Introduceix un número: fi
Màxim: 1111.0
Mínim: -78.0
```

5.4. Escriu una funció que rep dos llistes i retorna el seu producte escalar.

Mira el codi per entendre-ho (Els inputs son: a = [1,3,5,7], b = [2,4,6,8])

El resultat és: $1 \cdot 2 + 3 \cdot 4 + 5 \cdot 6 + 7 \cdot 8 = 100$

```
100.0

Process finished with exit code 0
```

5.5. Donada una llista de números enters, escriu una funció que retorna una llista en tots els que siguen primers. Inputs: **llista** = [1,2,3,4,5,6,7,8,9,10,11]

```
[2, 3, 5, 7, 11]

Process finished with exit code 0
```

5.6. Donada una llista de números enters i un enter k, escriu una funció que: retorna tres llistes, una amb els menors, una altra amb els majors i una altra en els iguals a k.

Inputs: **llista** = [1,2,3,4,5,6,7,8,9,10], k = 5

```
('menors = [1, 2, 3, 4]', 'majors = [6, 7, 8, 9, 10]', 'iguals = [5]')
```

```
Process finished with exit code 0
```

5.7. Inversió de llistes. Realitzar una funció que, donada una llista, retorna una nova llista en el mateix contingut que l'original però invertida. ['Digues', 'bon', 'dia', 'al', 'papa'], haurà de tornar ['papa', 'al', 'dia', 'bon', 'Digues'].

return llista[::-1]

```
['Papa', 'Al', 'Dia', 'Bon', 'Digues']
```

5.8. Realitzar una altra funció que inverteix la llista, però en lloc de tornar una de nova, modifique la llista donada per invertir-la, sense usar llistes auxiliars.

```
[10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
```

```
Process finished with exit code 0
```

5.9. Investigueu i expliqueu què fan els següents mètodes de les llistes, i mostrar exemples del seu ús:

append(x) → Afegeix un element al final de la llista.

Exemple: si tens [1, 2] i fas append(3), la llista queda [1, 2, 3].

clear() → Elimina tots els elements de la llista, deixant-la buida [].

copy() → Crea una còpia superficial de la llista (una nova llista amb els mateixos valors).

count(x) → Retorna quantes vegades apareix un valor dins la llista.

Exemple: [1, 2, 2, 3].count(2) → 2.

extend(iterable) → Afegeix tots els elements d'un altre iterable (com una altra llista) al final.

Exemple: [1, 2].extend([3, 4]) → [1, 2, 3, 4].

index(x) → Retorna la posició (índex) del primer element igual a x. Si no hi és, dona error.

insert(i, x) → Insereix un element x a la posició i.

Exemple: [1, 3] amb insert(1, 2) → [1, 2, 3].

pop([i]) → Elimina i retorna l'element a la posició i. Si no especifiques i, treu l'últim.

remove(x) → Elimina la primera aparició del valor x dins la llista.

reverse() → Inverteix l'ordre dels elements dins la mateixa llista.

sort() → Ordena els elements de la llista (per defecte, de menor a major). Pots indicar opcions com reverse=True per invertir l'ordre, o key= per ordenar segons un criteri.

Exercicis 6 - Arxius

6.1. Escriu un programa que utilitze les següents dos funcions:

- una per obrir un arxiu, dins la qual ens té que demanar el seu nom, informant en un missatge d'error si no es pot obrir (no existix) i en este cas ens el torne a demanar,
- una altra, on passat per paràmetre el nom de l'arxiu introduït abans, ens digue el número de línies que conté.

```
Nom de l'arxiu: practica6
No es pot obrir (no existix o no és un fitxer). Torna-ho a provar.
Nom de l'arxiu: practica6.txt
5

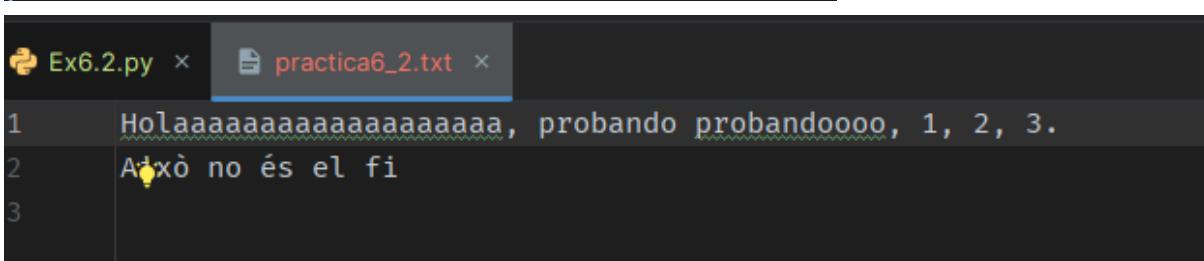
Process finished with exit code 0
```

6.2. Crea un programa que demane a l'usuari que ens digue el nom d'un arxiu a crear i un text per incloure a l'arxiu. El text a introduir ha de finalitzar quan l'usuari escrigue 'fi'. Per implementar-lo heu d'usar dos funcions:

- Per demanar el nom de l'arxiu modifiqueu la de l'exercici anterior,
- La segona rebrà com a paràmetre el nom de l'arxiu i l'omplirà de text.

```
Nom de l'arxiu a crear: practica6_2.txt
Escriu línies. Escrui 'fi' per acabar.
Holaaaaaaaaaaaaaaaaaa, probando probandooooo, 1, 2, 3.
Això no és el fi
fi

Process finished with exit code 0
```



```
Ex6.2.py * praktica6_2.txt *
1 Holaaaaaaaaaaaaaaaaaa, probando probandooooo, 1, 2, 3.
2 Això no és el fi
3
```

6.3. Volem saber quantes línies té el nou arxiu creat a l'anterior exercici. Modifica la funció del primer exercici per tal de que ens digue el total de línies i la llargada de cadascuna.

```
Nom de l'arxiu: practica6_2.txt
(2, [54, 16])

Process finished with exit code 0
```

6.4. Fes un programa que demane a l'usuari un nom de directori del sistema d'arxius local i mostre el seu contingut. Si passem un directori inexistent mostrerà un missatge i en demanarà un altre. Per implementar-ho fes 2 funcions diferents, però semblants, una que use la llibreria **os** i l'altra la **pathlib**.

```
Directori: \Users\Arnau\Documents
[os.listdir] Contingut de \Users\Arnau\Documents:
- Assassin's Creed Syndicate
- Call of Duty
- Call Of Duty Black Ops Cold War
- CV i coses important
- desktop.ini
- DuckGame
- dying light 2
- dying light the beast
- DyingLight
- Flutter SDK
- League of Legends
```

```
[pathlib] Contingut de \Users\Arnau\Documents:
- Assassin's Creed Syndicate
- Call of Duty
- Call Of Duty Black Ops Cold War
- CV i coses important
- desktop.ini
- DuckGame
- dying light 2
- dying light the beast
- DyingLight
- Flutter SDK
```

6.5. Amplieu l'exercici anterior, fent que a més es demane si volem llistar arxius, directoris, o ambdós. A la llista mostreu la màxima informació possible dels elements a mostrar (useu el mètode **stat()**, i mireu/imprimiu el que conté, mostrant les dates de forma amigable, i indicant en cada cas quin informació mostra).

```
Directori: \Users\arnau\Documents
Vols veure fitxers ("files") o directoris ("dirs")? files
Arxiu      402 bytes  mtime=2025-01-04 08:50:36.610896  -> desktop.ini
Arxiu    320883 bytes  mtime=2024-08-24 18:54:16.820968  -> Pito DATABASE.pdf
Vols veure fitxers ("files") o directoris ("dirs")? dirs
Directori      0 bytes  mtime=2025-10-28 22:46:44.008650  -> Assassin's Creed Syndicate
Directori      0 bytes  mtime=2024-09-16 15:42:49.931697  -> Call of Duty
Directori      0 bytes  mtime=2025-08-19 16:48:36.706528  -> Call Of Duty Black Ops Cold War
```

6.6. Useu el mètode **`walk()`** per recórrer el contingut d'un directori existent, indicat per l'usuari. Feu que hi haigue l'opció de vore tot el contingut del directori, o de parar si/quan es trobe cert directori/arxiu indicat per l'usuari.

```
Directori: \users\arnau\documents
Indica el directory o fitxer a trobar: Call of Duty
Trobat: Call of Duty a \users\arnau\documents, Aturant...
Directori: hola
Directori inexistent. Torna-ho a provar.
Directori: \users\arnau\documents
Indica el directory o fitxer a trobar: cod
No s'ha trobat el directori o fitxer "cod"

Process finished with exit code 0
```

6.7. Milloreu el programa anterior de manera que permètigue no només buscar arxius o directoris pel seu nom, sinó que també accepte expressions regulars (“shell-style wildcards” del mètode **`glob()`**). Podeu trobar informació de la sintaxi de les expressions regulars a internet, o als següents [apunts de Santi](#).

```
Directori: \Users\arnau\Desktop\clase\ProjectesIdea\ExercicisAccesDades\Exercicis
Quin patró vols seguir? (Ex: *.py): Ex1./*
\Users\arnau\Desktop\clase\ProjectesIdea\ExercicisAccesDades\Exercicis\Ex1.0.py
\Users\arnau\Desktop\clase\ProjectesIdea\ExercicisAccesDades\Exercicis\Ex1.2.py
\Users\arnau\Desktop\clase\ProjectesIdea\ExercicisAccesDades\Exercicis\Ex1.4.py
Directori: \Users\arnau\Desktop\clase\ProjectesIdea\ExercicisAccesDades\Exercicis
Quin patró vols seguir? (Ex: *.py): *.txt
\Users\arnau\Desktop\clase\ProjectesIdea\ExercicisAccesDades\Exercicis\practica6.txt
\Users\arnau\Desktop\clase\ProjectesIdea\ExercicisAccesDades\Exercicis\practica6_2.txt
```

6.8. Executeu el següent procediment en ordre:

1. Obriu el directori **`/tmp`** a l'explorador d'arxius del sistema, i mireu quants fitxers/directoris conté. Dixeу-lo obert.
2. Copieu el següent tros de codi al PyCharm:

```
from tempfile import NamedTemporaryFile

# Create a temporary file and write some data to it

while True:
    fp = NamedTemporaryFile('w+t')
    tfName = fp.name
    fp.write('Hello universe!')
    print(tfName)
```

- Dividiu la pantalla de manera que es veigu en tant l'explorador com el PyCharm, i executeu el codi durant màxim 5 segons. L'haureu de parar manualment ja que conté un bucle infinit.
 - Fixeu-vos en el que passa a l'explorador. Quants fitxers s'han creat al directori temporal?. Se borren automàticament al tancar el programa?. Quan ha tardat en borrar-los tots?. S'han borrat tots o s'ha quedat algun?. Intenteu mostrar captures que recolzen les vostres respostes.

No estic segur, però crec que el programa crea i elimina el fitxer pràcticament a l'instant abans d'acabar, les comandes del terminal s'han fet **MENTRE** s'executaba el programa i només han aparegut entre 2 i 3 fitxers durant el procés. segons després he tornat a comprovar-ho i n'hi havia encara menys (segurament algun del propi sistema o una altra app). També he intentat fer un grep per veure si per casualitat el fitxer en qüestió seguia estant. però no he trobat res.

6.9. Practiqueu els següents processos de creació i destrucció de fitxers i directoris, en ordre. És important que primer creeu arxius i després borreu estos mateixos arxius, així no borrarem res important:

1. Creeu un directori que n'incloue d'intermediaris utilitzant **`os.makedirs()`**.
 2. Creeu un arxiu de text dins del directori (podeu usar el codi d'algun exercici anterior).
 3. Borreu l'arxiu en algun dels mètodes indicats. Mireu que passa si intentem borrar un directori en estos mètodes i feu que mostre un missatge enllloc d'error.
 4. Borreu ara el directori (no useu el mètode **`shutil.rmtree`**). Mireu que passa si intenteu borrar un directori no buit, i tracteu l'error.
 5. Useu ara el mètode **`shutil.rmtree`** per borrar un arbre sense de directoris no buits.

```
Directori: \Users\arnau\Desktop\clase\ProjectesIdea\ExercicisAccesDades\Exercicis
Creat directori: \Users\arnau\Desktop\clase\ProjectesIdea\ExercicisAccesDades\Exercicis\nivell\nivell2
Creat fitxer: \Users\arnau\Desktop\clase\ProjectesIdea\ExercicisAccesDades\Exercicis\nivell\nivell2\prova.txt
Esborrat fitxer: \Users\arnau\Desktop\clase\ProjectesIdea\ExercicisAccesDades\Exercicis\nivell\nivell2\prova.txt
No s'ha pogut esborrar: \Users\arnau\Desktop\clase\ProjectesIdea\ExercicisAccesDades\Exercicis\nivell - [WinError 2] El sistema no puede encontrar el archivo especificado: '\\Users\\arnau\\Desktop\\clase\\ProjectesIdea\\ExercicisAccesDades\\Exercicis\\nivell'
Esborrat directori: \Users\arnau\Desktop\clase\ProjectesIdea\ExercicisAccesDades\Exercicis\nivell
No s'ha pogut esborrar: \Users\arnau\Desktop\clase\ProjectesIdea\ExercicisAccesDades\Exercicis - [WinError 32] El proceso no tiene acceso al archivo porque está siendo utilizado por otro proceso:
Creat arbre: \Users\arnau\Desktop\clase\ProjectesIdea\ExercicisAccesDades\Exercicis\arbre_demo
Esborrat arbre amb rmmtree: \Users\arnau\Desktop\clase\ProjectesIdea\ExercicisAccesDades\Exercicis\arbre_demo

Process finished with exit code 0
```

6.10. Proveu com se comprimixen i descomprimixen fitxers usant els mètodes explcats a l'apartat *An Easier Way of Creating Archives*.

```
Directori: \Users\arnau\Desktop\clase\ProjectesIdea\ExercicisAccesDades\Exercicis
Creat ZIP: C:\Users\arnau\Desktop\clase\ProjectesIdea\ExercicisAccesDades\Exercicis\arxiu.zip
Descomprimit a: \Users\arnau\Desktop\clase\ProjectesIdea\ExercicisAccesDades\Exercicis\unpack

Process finished with exit code 0
```

