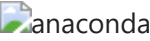


Nivell 1

Exercici 1

Instal·la el programa Anaconda amb Python 3, i Jupyter Notebook.

Per instal·lar Anaconda + Python + Jupyter Notebook he descarregat l'executable per Windows de la pàgina d'Anaconda i he seguit el procés d'instal·lació



Exercici 2 + Exercici 3

Utilitzant Jupyter Notebook executa alguns càlculs senzills, a la vegada que et familiaritzes amb el llenguatge Markdown.

Prova de crear títols, llistes, canviar l'estil de la lletra o afegir imatges dins del Notebook.

Creació de variables

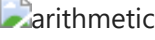
- Una variable es crea en el mateix moment que li assignes un valor.
- No és necessari declarar les variables d'un tipus determinat. Fins i tot poden canviar de tipus després de ser creades
- La variable es crea utilitzant la sintaxis: **nom del variable = valor**

Exemples de creació de variable:

x = 5

nom = "Albert"

Operacions matemàtiques



Font: https://www.w3schools.com/python/python_operators.asp

In []:

```
# Exemples d'operacions matemàtiques

# Declaració de variables
num1 = 10
num2 = 5

# suma
suma = num1 + num2

# resta
resta = num1 - num2

# multiplicació
multiplicacio = num1 * num2

# divisió
divisio = num1 / num2

resultats = "Suma: {}\nResta: {}\nMultiplicació: {}\nDivisió: {}"
print(resultats.format(suma, resta, multiplicacio, divisio))
```

Transformar amb casting

El *casting* s'utilitza quan volem canviar el tipus d'una variable. Per exemple, si una variable és de tipus **int** podem transforma-la en **str** utilitzant la tècnica de *casting* amb la funció **str()**



Font: https://www.w3schools.com/python/python_operators.asp

In []:

```
# Exemples de casting

# Declarem una variable de tipus int
integer = 5

# Transformar a float
float_cast = float(5)

# Transformar a string
string_cast = str(5)

resultat = "El tipus de la variable integer es: {} y el seu valor: {} \
\nEl tipus de la variable float_cast es: {} y el seu valor: {} \
\nEl tipus de la variable string_cast es: {} y el seu valor: '{}'"

print(resultat.format(type(integer), integer, type(float_cast), float_cast, type(string_cast), string_cast))
```

Tècniques amb strings

Els *strings* son cadenes de text. Al declarar-les, es poden utilitzar cometes dobles (") o simples (')

Algunes funcions bàsiques que es poden aplicar als strings son:

- len()** - per obtenir la longitud, el nombre de caràcters d'un string
- print()** - per imprimir un string

In []:

```
# Declaració duna variable string
nom = "Albert" # també seria vàlid nom = 'Albert'

# Longitud
longitud = len(nom)

# Imprimir string i longitud
print(nom)
print(longitud)
```

Slicing Strings

Per obtenir una part d'un string s'utilitza l'slicing. Per fer-ho, s'utilitzen els símbols [] i al seu interior, separats per dos punts (:) l'índex del primer caràcter que volem i el de l'últim + 1.

S'ha de tenir en compte que **l'índex del primer caràcter es 0** i que el **a l'index de l'últim caràcter se li ha de sumar 1** perquè no està inclòs a la selecció

In []:

```
# Obténir els 2 primers caràcters del nom Albert
print("Els 2 primers caràcters son: " + nom[0:2])

# Obténir els 4 darrers caràcters del nom Albert
print("Els 4 darrers caràcters son: " + nom[2:6])
```

Si volem obtenir un slice a partir del primer caràcter o fins a l'últim caràcter, aquest es poden deixar buits.

A continuació tenim els mateixos exemples d'abans sense indicar l'índex del primer o l'últim caràcter:

In []:

```
# Obténir els 2 primers caràcters del nom Albert
print("Els 2 primers caràcters son: " + nom[:2])

# Obténir els 4 darrers caràcters del nom Albert
print("Els 4 darrers caràcters son: " + nom[2:])
```

Mètodes String

Els strings tenen una serie de mètodes que permeten manipular-los o obtenir informació sobre ells. N'hi ha molts, aquí només anomenaré alguns:

- lower()** - passa tots els caràcters a minúscula
- upper()** - passa tots els caràcters a majúscula
- replace()** - substitueix caràcters
- count()** - obtenir el nombre de vegades que apareix un caràcter

In []:

```
print("Text a minúscula: " + nom.lower())
print("Text a majúscula: " + nom.upper())
print("Modificar string: " + nom.replace("Al", "Ro"))
print("Nombre de vegades 'l': " + str(nom.count("l")))
```

Un altre mètode interessant és el mètode **split()**. Converteix una cadena de text en una llista, separant la cadena mitjançant un separador determinat. Si no s'especifica, el separador per defecte es l'espai en blanc

In []:

```
# Declaració de variable string
text = "Estudio Data Science a l'IT Academy"

# Imprimir el resultat del mètode split()
print(text.split())
```

Booleans

Els *booleans* representen dos valors: **True** o **False**. S'utilitzen per saber si una expressió és verdadera o falsa. Per això s'utilitzen juntament amb els *operadors condicionals*:



Font: https://www.w3schools.com/python/python_operators.asp

In []:

```
# Exemples de Booleans

print("5 i 4 Són iguals?")
print(5 == 4)
print("\n5 és més gran que 4?")
print(5 > 4)
print("\n5 és més petit que 4?")
print(5 < 4)
print("\n5 és diferent de 4")
print(5 != 4)
```