Lliçó 1 **1**
La primera funció: print() 1
Tipus de dades 2
Comentaris del programa 2
Arguments en print() 3
Variables 4
L'operador aritmètic + 4
Començar una cadena amb la lletra f a la funció print 5
Lliçó 2 **6**
La segona funció: input() 7
La tercera funció: int() 7
La quarta funció: float() 8
El nostre primer mòdul: random 9
Per acabar la lliçó: operadors matemàtics 10
Lliçó 3 **11**
Booleans: quan la veritat sembla mentida 11
Operacions booleanes: àlgebra de boole 13
Lliçó 4 **15**
El primer condicional: if 15
El segon condicional: else 16
El tercer condicional: elif 16
Lliçó 5 **17**
Voltes i més voltes: els bucles a Python 17
El primer bucle: for 18
El bucle condicional: while 19
El segon bucle condicional: do while 22
Lliçó 6 **23**
Definir funcions: omplir la bossa màgica 23
Exercici final 25

Lliçó 1

Editor online de Python

La primera funció: print()

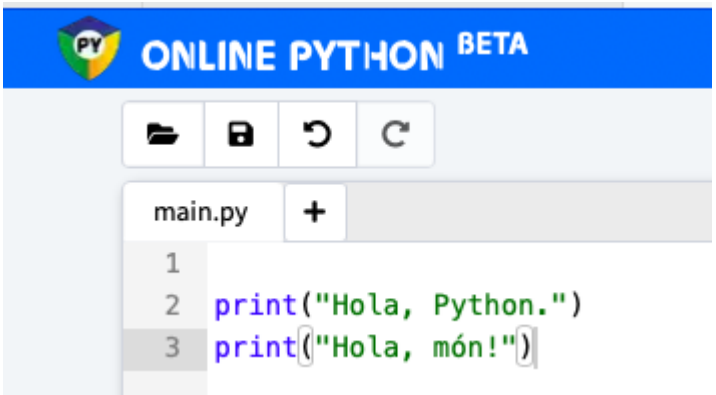
print() és una funció a Python.

Una funció és com un encanteri que quan el crides obtindràs una tasca específica de l'ordinador.
Per exemple, si vols que l'ordinador et digui "Hola!" per la pantalla, simplement la crides i ella fa això.

La funció print() es fa servir per imprimir informació a la pantalla o a la sortida estàndard del programa.

Quan es crida la funció print(), els arguments que s'especifiquen dins dels parèntesis es mostren com a text a la consola.

Exercici 1



Heu de ser molt curosos amb el què escribiu. Si oblideu tancar unes cometes o un parèntesi, donarà error.

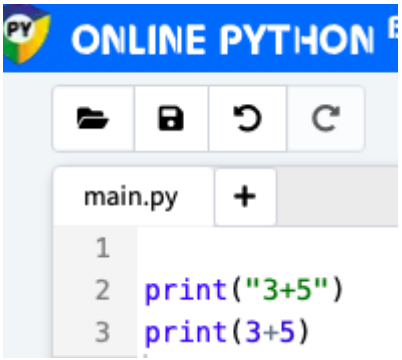
está licenciado bajo CC BY-NC 4.0

Tipus de dades

Hi ha tres tipus de dades:

- Cadenes alfanumèriques. Es posen entre cometes “”
- Números
- Lògics: Yes/No True/False

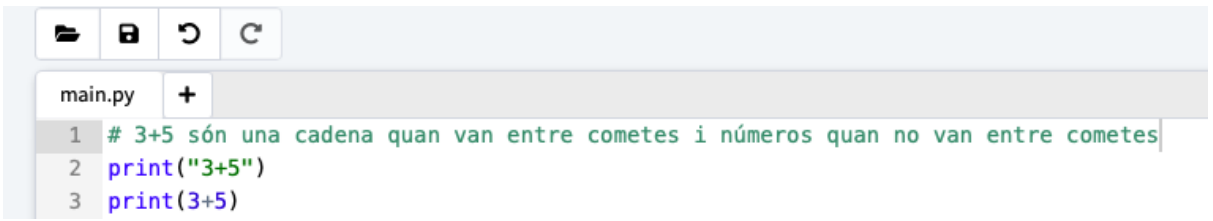
Exercici 2



Comentaris del programa

Els comentaris a Python són com notes que escrivim en el nostre codi per explicar el que fem.
La seva funció principal és fer el codi més fàcil de llegir i entendre. Quan altres persones o fins i tot tu mateix revises el codi en el futur, els comentaris t'ajuden a recordar què fa cada part del codi i per què la vas escriure d'aquella manera.
També són útils per explicar el que fa una funció o una peça de codi en particular, especialment si és una mica complicada.

Exercici 3



está licenciado bajo CC BY-NC 4.0

Arguments en print()

Si volem juntar diverses dades en una sola línia, usarem una sola funció print() i separem els arguments per una coma.

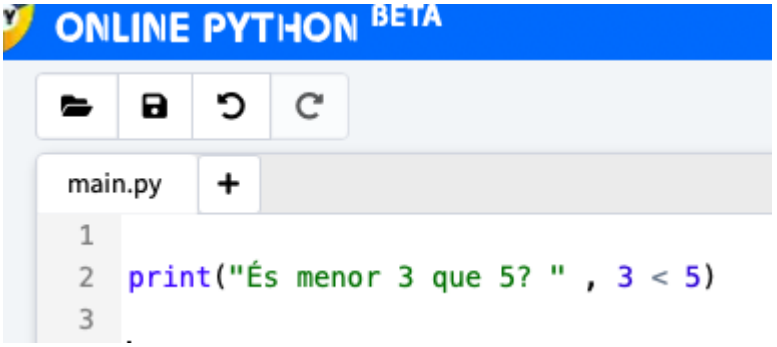
Exercici 4

£

Exercici 5

Podem usar operacions aritmètiques com 3 + 5 o operacions lògiques com 3 < 5. Et proposem un codi d'operació lògica i mira de posar aquestes operacions (a la captura de pantalla només posem la primera)

3 < 5
3 > 5
3 == 5
5 == 5



Variables

Les variables a Python són com contenidors on guardem informació, com números, lletres o llistes.

Pensa-hi com a caixes on posem coses diferents.

está licenciado bajo CC BY-NC 4.0

La funció principal de les variables és emmagatzemar dades que volem utilitzar més tard al nostre programa.

Per exemple, si volem guardar el nom d'una persona, podem crear una variable anomenada "nom" i guardar dins la cadena de text del nom d'aquella persona.

Exercici 6

```
1
2  alumnat = 30
3  malalts = 4
4  assistents = alumnat - malalts
5  print("Avui han vingut", assistents , "persones a classe")
6
```

L'operador aritmètic +

Hem vist com funciona el signe + amb dos número amb la funció print a l'exercici 3.

Què passa si sumem dues cadenes de text (alfanumèriques)? No seria una suma aritmètica. Us enrecordeu d'aquella frase que us deien quan apreníeu a llegir?

—La ema amb l'a: MA!

Quan sumem dues cadenes es fa la concatenació, és a dir: fer una única cadena amb les dues originals, una darrera de l'altra

Exercici 7

main.py +

```
1
2  lletra1 = "P"
3  lletra2 = "i"
4  lletra3 = "z"
5  lletra4 = "z"
6  lletra5 = "a"
7  lletra6 = "P"
8  lletra7 = "e"
9  lletra8 = "l"
10 lletra9 = "i"
11
12 # Primer correu el programa tal qual. Després mireu de treura la coma del final a veure què passa
13
14 print(lletra1 + lletra2 + lletra3 + lletra4 + lletra5 , ' ')
15 print(lletra6 + lletra7 + lletra8 + lletra9)
```

está licenciado bajo CC BY-NC 4.0

Començar una cadena amb la lletra f a la funció print

Quan col·loquem la lletra f al començament d'una cadena (fora de les cometes), estem dient a print que haurà de concatenar variables entremig d'aquella cadena.

Això ens permet simplificar el codi i no haver d'obrir i tancar cometes cada cop que posem variables a dins.

Exercici 8

```
1
2  meu_nom = "Scarlett Johansson"
3  meva_edat = 39 #Faig 40 al novembre
4  meva_alçada = 160 #Centímetres
5  meu_pes = 58 #Quilos
6  meus_ulls = "verds"
7  meu_cabell = "castany"
8
9  print(f"Parlem de l'actriu {meu_nom} .")
10 print(f"No és molt alta, fa {meva_alçada} centímetres d'alçada.")
11 print(f"Pesa {meu_pes} quilos.")
12 print(f"Té els cabells de color {meu_cabell} .")
13 print(f"Els seus ulls son de color {meus_ulls} .")
14 print("Vols saber la meva edat?")
15
16 total = meva_edat + meva_alçada + meu_pes
17 print(f"Si sumes la seva alçada, el seu pes i la seva edat, tens {total} .")
18
```

está licenciado bajo CC BY-NC 4.0

Lliçó 2

Editor online de Python

La segona funció: input()

input() és una funció a Python.

Aquesta és una funció que permet que l'usuari introdueixi dades.

Aquestes dades que entrem gràcies a input es poden guardar en una variable i ser utilitzades pel programa per a realitzar diverses accions.

La funció input és molt útil per a interactuar amb l'usuari i fer que els programes siguin més dinàmics i personalitzats.

Exercici 9

main.py

```
1
2 nom_usuari = input("Com et dius?")
3 print(f"M'han dit que ets molt intel·ligent {nom_usuari} i per això saps programar python.")
4
```

La tercera funció: int()

Us plantejem una qüestió:

> *Si entrem un 5 amb la instrucció input, com pot saber l'ordinador si es tracta d'un nombre o d'una cadena de text (alfanumèrica)?*

La resposta és que no ho pot saber, així que TOT el que entra amb input és, en principi una cadena de text.

Per comprovar-ho et recomano que facis els dos exercicis següents on es mostra una funció nova: int(), de la qual parlarem després.

está licenciado bajo CC BY-NC 4.0

Exercici 10

Nota: aquest exercici produeix error, es fa junt al següent per donar sentit al codi.

main.py +

```
1
2 numero = input("Entra un número de l'1 al 9")
3 print("10 més el teu número és" , 10 + numero)
4
```

Exercici 11 (continuació de l'anterior)

Transformeu el codi anterior incloent la funció int()

main.py +

```
1
2 numero = int(input("Entra un número de l'1 al 9"))
3 print("10 més el teu número és" , 10 + numero)
4
```

La funció `int()` en Python s'utilitza per convertir un valor a un nombre sencer (número sense decimals).

Això significa que si tens una cadena de text que representa un nombre o una fracció i vols treballar amb aquest valor com a nombre sencer, pots utilitzar la funció `int()` per convertir-lo.

Per exemple, si tens la cadena `"5"` i vols tractar-la com el nombre sencer 5, pots fer-ho utilitzant `int("5")`.

Si la cadena no pot ser convertida a un nombre enter, la funció `int()` provocarà un error.

La quarta funció: float()

I si el número que volem introduir té decimals? la funció int() transforma en nombre sencer i en el millor dels casos treurà els decimals.

La funció float() és similar a int(), però serveix per a nombres no sencers (decimals).

está licenciado bajo CC BY-NC 4.0

Proveu com funciona amb el següent exercici.

Nota: Es fa servir el punt com a separador dels decimals a python.
Si useu coma es produirà error.

Exercici 12

main.py +

```
1
2 numero = float(input("Entra un número amb decimals"))
3 print("10 més el teu número és" , 10 + numero)
4
```

El nostre primer mòdul: random

Podem generar números aleatoris (a l'atzar) amb python?

Sí, a Python pots generar números aleatoris utilitzant el mòdul `random`.

Què és un mòdul a python? Si una funció era la paraula màgica per a fer una tasca específica, el mòdul és el capítul del llibre de màgia que conté les diferents funcions (encanteris) que fan coses semblants.

Un mòdul conté un conjunt de funcions i eines relacionades que poden ser utilitzades en els teus programes. En lloc de tenir totes les funcions en un sol lloc, es poden agrupar en mòduls segons la seva funcionalitat.

Al proper exercici tens un exemple senzill de com generar un nombre enter aleatori entre 1 i 100.

Exercici 13

main.py +

```
1
2 import random
3
4 numero_aleatori = random.randint(1, 100)
5 print("El número aleatori és:", numero_aleatori)
```

está licenciado bajo CC BY-NC 4.0

Què fa el codi?

Aquest codi importa el mòdul `random` i utilitza la funció `randint()` per generar un nombre aleatori.

> *Importar a Python és com obrir el capítol d'encanteris del mateix tipus d'un llibre, de la llibreria màgica de Python. Quan escrius "import" en Python, estàs dient-li al programa que vols utilitzar les eines que estan guardades en un mòdul específic.*

La funció `randint()` accepta dos paràmetres, que són el valor mínim (1 en aquest cas) i el valor màxim (100 en aquest cas) del rang dels nombres que vols generar.

Així que en aquest exemple, el valor de `numero_aleatori` serà un nombre aleatori entre 1 i 100, inclosos.

Per acabar la lliçó: operadors matemàtics

Els operadors matemàtics en Python són com les eines que utilitzes per fer càlculs matemàtics en els teus programes. Com les eines en una caixa d'eines, cada operador té una funció específica.

- Suma (+). Suma dos valors. Per exemple, 3 + 5 donaria com a resultat 8.
- Resta (-): Restaun valor d'un altre. Per exemple, 7 - 4 donaria com a resultat 3.
- Multiplicació (*): Multiplica dos valors. Per exemple, 2 * 6 donaria com a resultat 12.
- Divisió (/): Aquest operador s'utilitza per dividir un valor pel seu divisor. Per exemple, 10 / 2 donaria com a resultat 5.0.

Altres operadors no tan habituals:

- Divisió sencera (//): Aquest operador s'utilitza per realitzar una divisió sencera, que retorna només la part sencera del resultat de la divisió i no inclou els decimals. Per exemple, 7 // 3 donaria com a resultat 2.
- Mòdul (%) : Aquest operador retorna el residu de la divisió entre dos números. Per exemple, 7 % 3 donaria com a resultat 1, ja que 7 dividit per 3 dóna 2 amb un residu de 1.

está licenciado bajo CC BY-NC 4.0

Lliçó 3

Editor online de Python

Booleans: quan la veritat sembla mentida

Imagina que un mag té dos pots idèntics. En un hi ha aigua i a l'altre un terrible verí inodor, i incolor. Si no es pot diferenciar l'aigua del verí, millor que llenci els pots i s'oblid d'usar el verí amb un enemig

I si et dic que tens una vareta màgica que li fas una pregunta i et respon?

Aquesta vareta són els booleans, un instrument màgic per saber si la cosa que volem saber és veritat o mentida: *true* o *false*.

Els booleans són un tipus de dades en programació que només poden tenir dos valors possibles: cert o fals.

El concepte de booleans es deriva de la lògica booleana, que tracta amb operacions lògiques i expressions que com a resultat tenen veritat o fals.

Imagina que el mag utilitza la vareta màgica (el booleà) per avaluar si el pot triat conté verí. Si la vareta diu "cert" podrà tilitzar-lo per defensar-se del monstre sense cap risc.

D'altra banda, si el booleà retorna "fals", significa que el pot conté aigua i no farà res al monstre fatal.

Apunts de PythonVirgilio Gonzalo Edesaestá licenciado bajo CC BY-NC 4.0

El mag (programa) pren decisions segons el que diu la vareta màgica (eina dels booleans).

Els valors dels booleans són essencials per avaluar condicions dins del codi i decidir quines parts del programa s'executen o se salten basant-se en aquesta avaluació.

Per exemple, si volem que una part del nostre programa (donar pot al monstre) s'executi només si una certa condició és veritable (pot conté verí), utilitzaríem un booleà per expressar aquesta condició que podria ser:

pot_triat = = verí

i controlar el flux d'execució del programa:

- True. Donarem el contingut del pot al monstre
- False. Donarem el contingut de l'altre pot al monstre

Segons el que indica la eina dels booleans, el programa fa una cosa o una altre.Aquest comportament del programa és coneix amb el nom de condicionals.

La funció input és molt útil per a interactuar amb l'usuari i fer que els programes siguin més dinàmics i personalitzats.

Exercici 14

Respon per escrit al teu quadern el resultat d'aquests booleans:

CORRAL2 > CORRAL3
CORRAL1 = = CORRAL2
CORRAL1 < CORRAL3
CORRAL1 != CORRAL4

está licenciado bajo CC BY-NC 4.0

Exercici 15

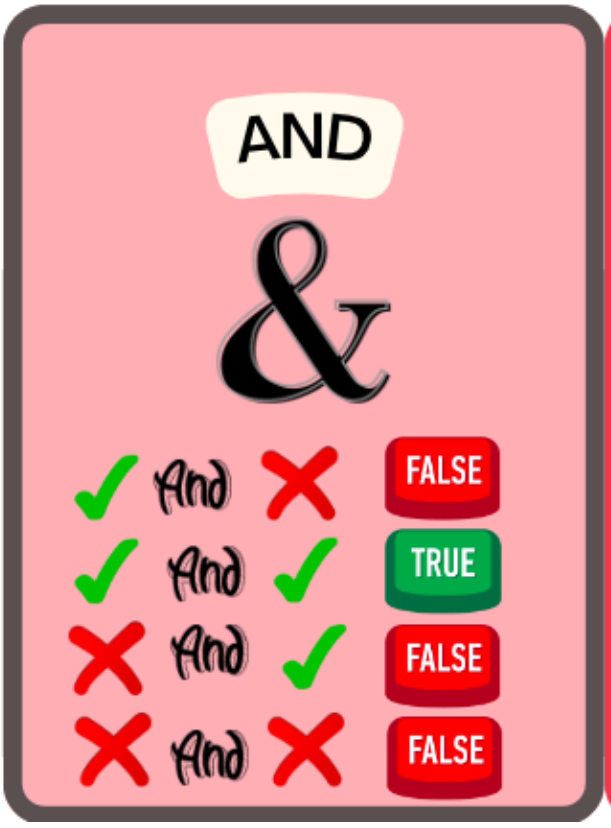
```
2  pot_1 = "verí"
3  pot_2 = "aigua"
4
5  print("El pot 1 té el verí?", pot_1 == "verí")
6  print("El pot 2 té el verí?", pot_2 == "verí")
7
```

Operacions booleanes: àlgebra de boole

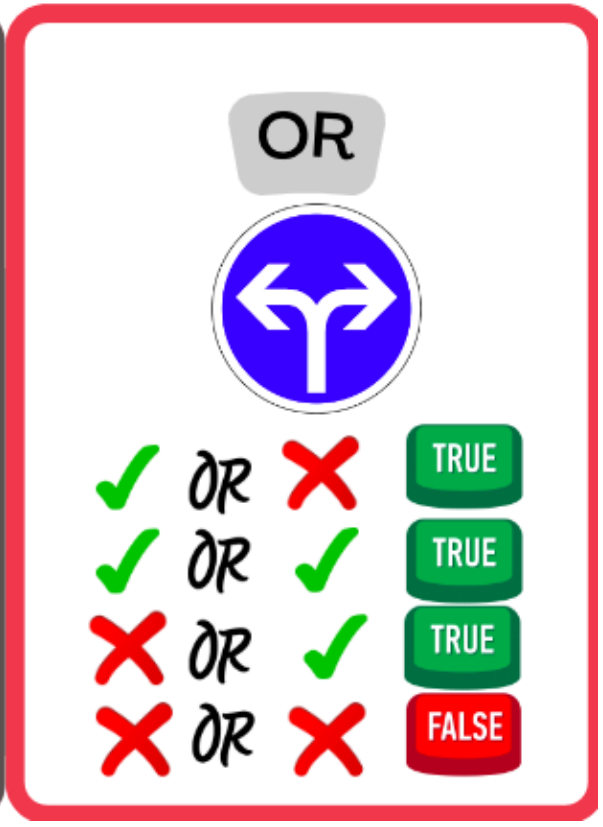
L'àlgebra de Boole és una part de les matemàtiques que es basa en operacions lògiques com "i", "o" i "no", que s'apliquen a valors que només poden ser veritat o fals.

En altres paraules, el resultat de les operacions només pot ser només "certe" o "fals". Aquest àlgebra és útil per a la lògica de computadores, ja que ajuda a manipular i entendre com les computadores prenen decisions basades en certes condicions.

AND: és l'operació "i"

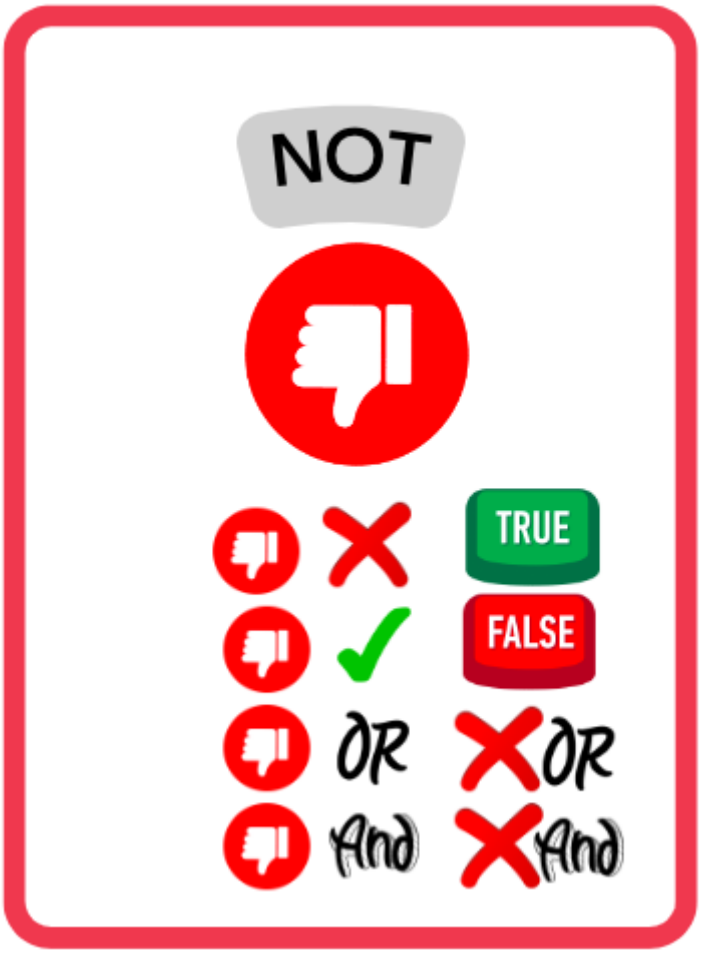


OR: és l'operació "o"



está licenciado bajo CC BY-NC 4.0

NOT és l'operació "no"



Exercici 16

Obre l'arxiu BOOLEANS EASY i mira de fer-ho a la llibreta.

Veuràs que en color blau estan les dades dels problemes i en color taronja es plantejen els exercicis de booleans. Has de dir el resultat de cada exercici.

Són tres pàgines i a cada pàgina canvien les dades per resoldre'ls.

está licenciado bajo CC BY-NC 4.0

Lliçó 4

Editor online de Python

El primer condicional: if

Les sentències condicionals a Python són les paraules màgiques per usar les varetes màgiques (booleans).

La paraula màgica (sentència condicional) "if" sempre està seguida d'una condició que s'avalua com a certa o falsa (operador booleà).

Si la condició és veritable, s'executarà el bloc de codi associat, un petit encanteri associat.

Si la condició és falsa, el bloc de codi no s'executarà i no es produirà el petit encanteri que hem preparat per la condició.

Exercici 17

Posem que estàs en un bosc encantat on viuen moltes i diferents criatures màgiques. Per a cada criatura li tens preparada una feina, per exemple saber si els follets mesuren més de 5 centímetres

```
2  mida_follet = 10
3
4  if mida_follet > 5:
5      print("El follet és major que 5 centímetres")
6
```

FEs una petita explicació de què és if i parla de la importància dels dos punts i la identació.

está licenciado bajo CC BY-NC 4.0

El segon condicional: else

Després de la sentència “if” podem posar altres sentències condicionals. La primera seria “else”.

La sentència"else" és una altra paraula clau que s'utilitza per especificar un bloc de codi que s'executarà si cap de les condicions anteriors (la condició de l'if prèvia) és veritable.

En altres paraules, "else" captura totes les altres possibilitats que no s'han cobert amb les condicions anteriors.

Exercici 18

```
2  mida_follet = 4
3
4  if mida_follet > 5:
5      print("El follet és major que 5 centímetres")
6  else:
7      print("El follet és menor que 5 centímetres")
```

El tercer condicional: elif

Quan necessitem avaluar diverses condicions no repetim “if”, sinó que a continuació del primer “if” usem sentències “elif”.

Quan la condició de l'if del principi no és veritable, a continuació el codi avaluarà la condició de "elif" si està present.

Exercici 19

```
2  mida_follet = 5
3
4  if mida_follet > 5:
5      print("El follet és major que 5 centímetres")
6  elif mida_follet == 5:
7      print("El follet fa 5 centímetres")
8  else:
9      print("El follet és menor que 5 centímetres")
```

está licenciado bajo CC BY-NC 4.0

Lliçó 5

[Editor online de Python]

Voltes i més voltes: els bucles a Python

A la pel·lícula d'animació "The sword in the Stone" que va fer Disney a 1963, el famós mag Merlí ajuda al jovenet Artús a fer les feines de casa. Hi ha piles i piles de plats bruts per netejar!

Encara que amb la seva vareta fa un encanteri per aconseguir que el raspall funcioni tot sol i que els plats vagin al cub d'aigua, està clar que l'encanteri és per rentar un únic plat.

Així que el mag ha d'agitar la vareta per fer que aquell encateri es repetiexi plat a plat fins acabar de netejar-ho tot. [Pots veure la seqüència completa aqui.]

Sempre que hem de fer una tasca (encara que sigui no fer res) de forma repetitiva, haurem d'usar bucles.

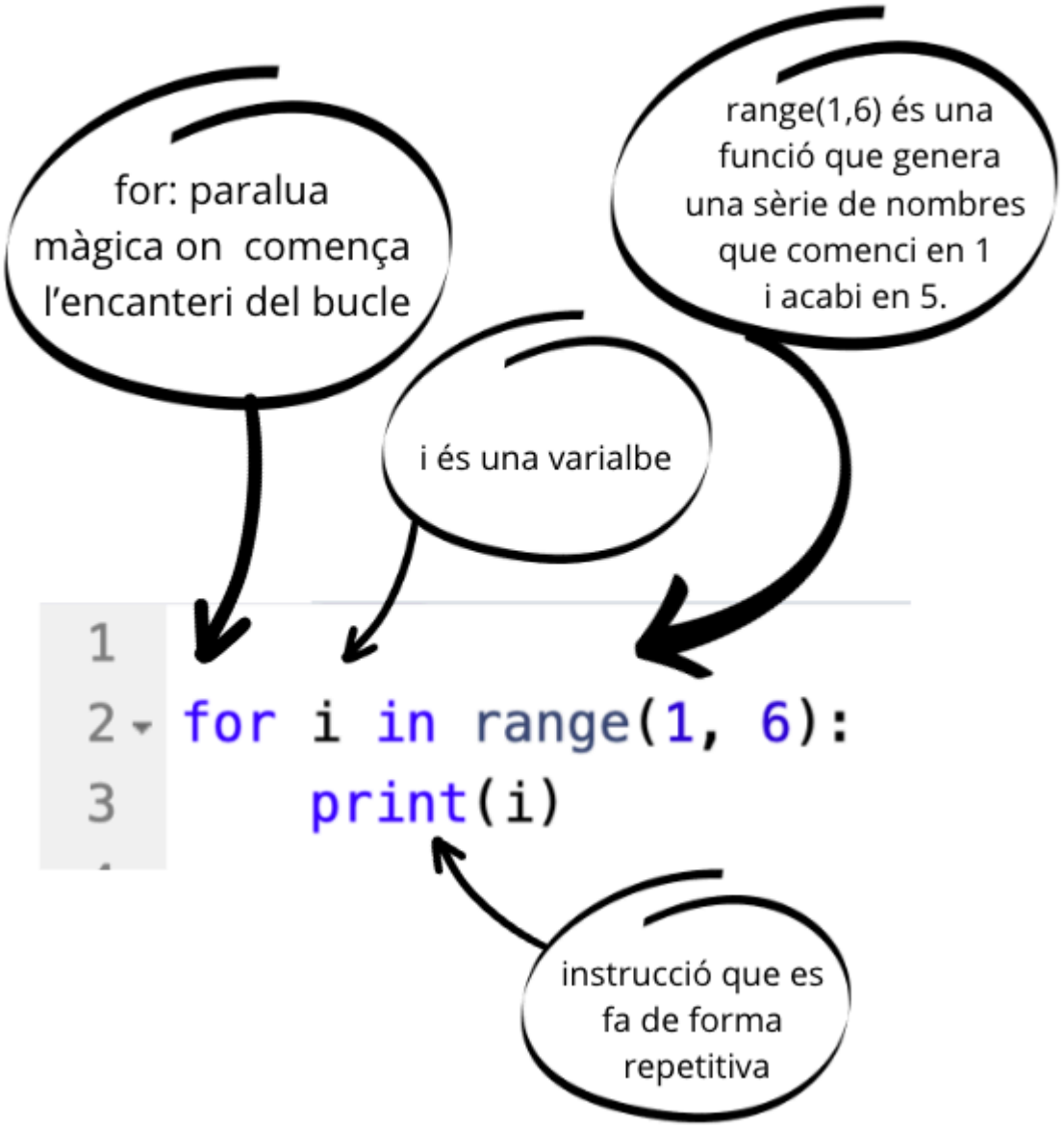
Els bucles en Python són estructures que permeten executar un conjunt d'instruccions repetidament fins que es compleix una condició específica. Aquestes estructures són fonamentals per automatitzar tasques i processar dades de manera eficient.

Apunts de PythonVirgilio Gonzalo Edesaestá licenciado bajo CC BY-NC 4.0

Amb els bucles, podem recórrer llistes d'elements, processar fitxers, interactuar amb bases de dades i molt més.

El primer bucle: for

El bucle “for” és el más senzill d’entendre, s’utilitza per repetir una acció un nombre fixat de vegades.



Exercici 20

Posem que tenim cinc plats. Primer rentarem el número 1, després el número 2 i així fins rentar els cinc plats. El bucle comptará del número 1 fins el 6.

está licenciado bajo CC BY-NC 4.0

Perquè comptar fins a 6 si tenim 5 plats?

Fàcil, perquè en arribar a l'últim número (6), el codi tanca el bucle sense fer res.

```
2 for i in range(1, 6):
3     print("Rento el plat", i)
```

Prova a variar els nombres de dins del parèntesi de la línia 2 per veure quin efecte causa en executar el codi.

Exercici 21

La variable "i" serveix per anar comptant. Es pot posar el nom de variable que més ens convingui.

Els nombres del parèntesi dins de "in range()" poden estar determinats per variables.

```
2 fi = int(input("Quants plats vols rentar? Escriu un nombre del 2 al 10"))
3
4 for i in range(1, fi+1):
5     print("Rento el plat", i)
```

Nota: l'usuari decideix fins on es vol comptar, pero quan posem el valor al parèntesi de "in range" hem de sumar 1 pel motiu que s'ha explicat a l'exercici anterior.

El bucle condicional: while

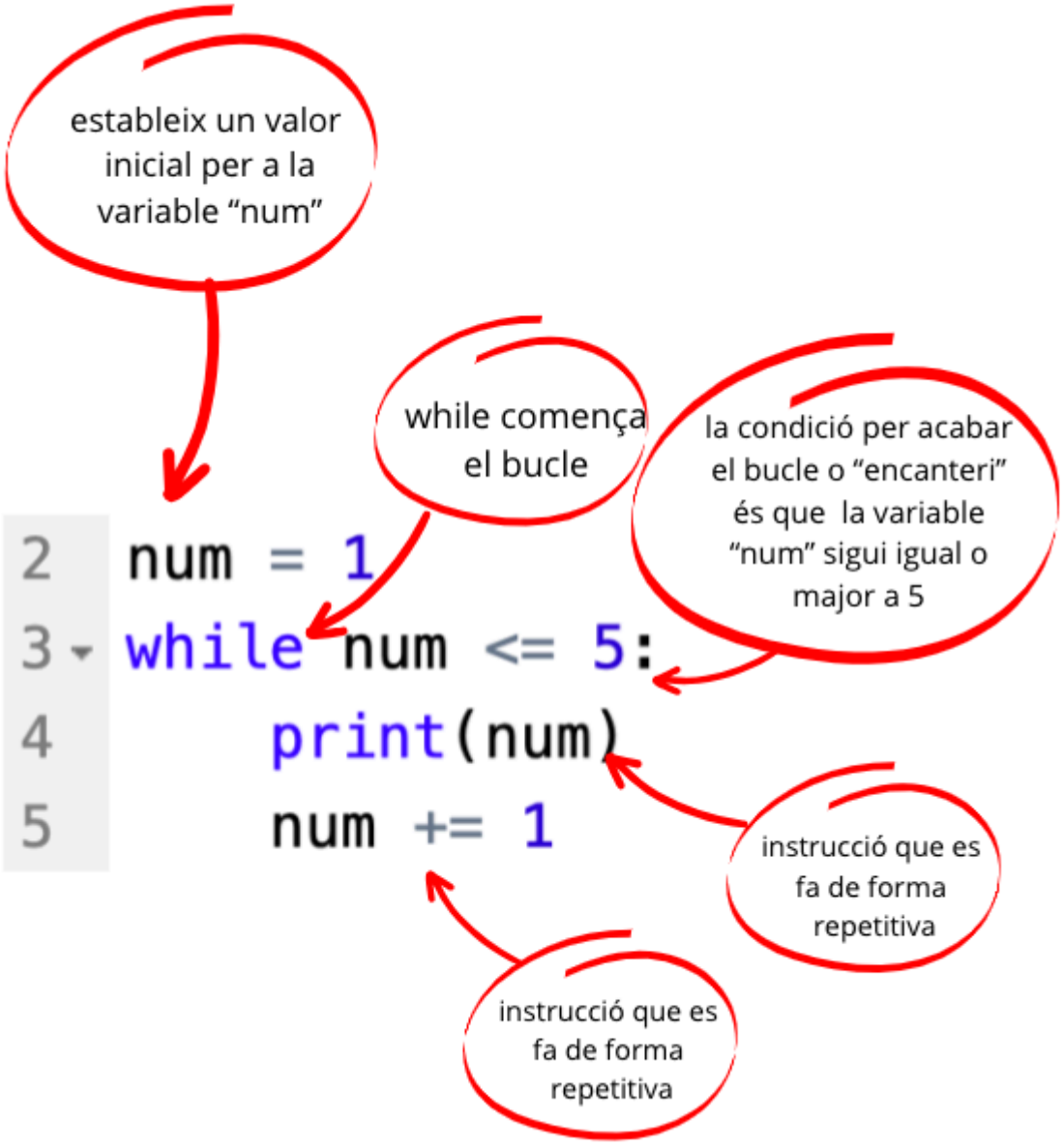
Els encanteris no han de ser permanents sempre.

L'encanteri "while" et permet repetir una acció màgica mentre alguna cosa sigui certa. Exemples:

- A l'encanteri de la Malèfica, la Bella Dorment és posada en un somni profund fins que un príncep la rescata. Quan s'acompleix la condició (recat pel príncep), l'encanteri i el seu somni finalitzen. És com si el somni continués "mentre" l'encanteri estigui actiu.
- A la Bella i la Bèstia, el príncep és qui ha rebut l'encateri que el transforma en Bèstia. El príncepi roman atrapat en la seva forma animal fins que algú l'estimi sinceramente.

está licenciado bajo CC BY-NC 4.0

- La fada madrina fa un encanteri que vesteix de forma elegant a la Ventafocs sempre que la hora sigui menor a la mitjanit. Quan al rellotge toquen les dotze de la nit, la pobre Ventafocs perd l'encanteri i les seves robes tornen a la forma inicial.



Exercici 22

Com seria l'encanteri de la fada madrina de la ventafocs?

```
2  hora = 19  # Hora d'inici del ball
3
4  while hora <= 12:
5      print(f"A les {hora} Ventafocs disfruta del ball")
6      hora += 1
```

está licenciado bajo CC BY-NC 4.0

Exercici 23

Demana a l'usuari que introdueixi un nombre positiu i suma tots els nombres des de l'1 fins a aquest nombre.

```
total = 0
nombre = int(input("Introdueix un nombre positiu: "))

while nombre > 0:
    total += nombre
    nombre -= 1

print("La suma de tots els nombres des de 1 fins a", nombre, "és:", total)
```

Exercici 24

Què passa quan obrim el bucle del tipus "while" amb "while true:"

```
numero = 5
while True:
    print("El bucle està funcionant")
```

está licenciado bajo CC BY-NC 4.0

El segon bucle condicional: do while

El bucle “do while” és una estructura de control similar al bucle “while”, però amb una diferència clau al bucle “do while” la condició es comprova després de cada iteració o volta, garantint que el bloc de codi s'executi com a mínim una vegada.

Aquest bucle amb aquestes paraules no està disponible a Python però sabem prou Python per simular-ho.

Exercici 25

Aquest exercici mostra com es pot usar el que has aprés de Python per simular una instrucció “do while”.

```
numero = 5
while True:
    print("El bucle està funcionant")
    numero += 1
    if numero > 10:
        break

print("Fi del bucle")
```

está licenciado bajo CC BY-NC 4.0

Lliçó 6

[Editor online de Python]

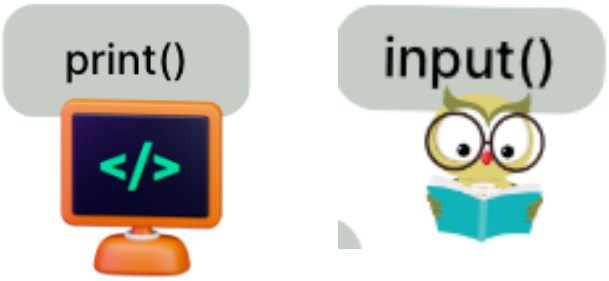
Definir funcions: omplir la bossa màgica

Hem vist que les funcions són com encanteris màgics que realitzen tasques específiques quan són cridats.

Cada encanteri té un nom únic.

Cada funció pot prendre arguments (ingredients) per realitzar la seva màgia.

Els arguments van entre parèntesi.



Quan volem un nou encanteri per usar segons quan i que podrien ser la combinació dels altres, llavors hem de crear la seva fórmula màgica.

Per crear una nova fòrmula màgica o sortilegi, hem de fer servir la definició de funcions.

Si omplim una bossa amb aquests sortilegis o funcions definides, podrem fer-las servir més endavant.

Al cridar el sortilegi o la funció definida. podrem fer moltes coses sense necessitat de dir el sortilegi cada cop o d'escriure el codi cada vegada que el volem utilitzar.

També podem dir que definir una funció és com escriure la fórmula d'un encanteri, i cridar una funció és com recitar la fórmula per invocar la seva màgia.

está licenciado bajo CC BY-NC 4.0

Exercici 26

Farem el primer sortilegi per a omplir la nostra bossa. La funció serveix per sumar dos nombres,

Primer has de definir la fórmula de l'encanteri (definir una funció), com ara:


```
def encanteri_suma(num1, num2):
    resultat = num1 + num2
    return resultat
```

Si fas run, veuràs que no passar res.

Normal! Sols has posat un encanteri a la teva bossa.

Per que el sortilegi funcioni, hauràs de cridar-lo més endavant.

Defineix la funció

```
3  def encanteri_suma(num1, num2):
4      resultat = num1 + num2
5      return resultat
6
7  encanteri_suma(num1, num2)
```

Cridar la funció

está licenciado bajo CC BY-NC 4.0

Exercici final

Escriu aquest codi i mira el funcionament a l'apartat A.5. Tutor de Python del teu Classroom.

```python
#introduim un valor a la variable quad
quad = int( input("Escriu un número:"))
#definim funció el quadrat d'un número
def quadrat(num):
    resultat = num ** 2
    return resultat
# Cridem la funció
resultat_quadrat = quadrat(quad)
# Ara hem de mostrar "El quadrat de 4 és: 16"
print("El quadrat de" , quad, " és:", resultat_quadrat)
```

Qüestions exercici final

Què és "quad després" d'executar la primera línia de codi?

Com es diu la funció que està definida a la segona part del codi?

Què fa aquesta funció?

Si l'usuari introdueix 5 com a entrada, quin resultat s'imprimirà al final?

Si l'usuari introdueix -3 com a entrada, quin resultat s'imprimirà al final?

Què passaria si l'usuari introdueix una cadena de text en lloc d'un número a la primera entrada?

está licenciado bajo CC BY-NC 4.0