09 – LISTAS DE BÁSICOS

Jorge Muñoz

IIC1103 – Introducción a la Programación

Se usa casi tanto como el if

Pero es muy fácil por que ya lo hemos visto casi todo

CREAR LISTAS

```
#Lista vacia
p = []

#Lista de enteros
q = [5, 7, -1]

#Lista de floats
r = [5.5, 7.0, 4.0, 2.4]

#Lista de strings
s = ["hola", "adios"]

#Lista de bools
t = [True,True,False,True]

#Listas de varios tipos
u = [True,"hola",7.0,"adios",-1]
```

ITERAR ELEMENTO A ELEMENTO

```
cursos = ["IIC1103","IIC3757","IIC2322"]

for e in cursos:
    print("Curso", e)
```

```
>>>
Curso IIC1103
Curso IIC3757
Curso IIC2322
>>>
```

memesintroalaprogr • Following

58 likes

OCTOBER 13, 2019

Add a comment... Post

APROBADOS

INTERESES

“cosas cuotidianas de la universidad”

```
def num_aprobados(notas):
```

```
n = [6.6, 2.3, 4.7, 3.3, 6.1]
a = num_aprobados(n)
print("Aprobados:",a)
```

```
>>>
Aprobados: 3
>>>
```


I'LL WAIT
FOR YOU HERE

```
def num_aprobados(notas):
```

```
n = [6.6, 2.3, 4.7, 3.3, 6.1]
a = num_aprobados(n)
print("Aprobados:",a)
```

```
>>>
Aprobados: 3
>>>
```

```
def num_aprobados(notas):
    r = 0
    for e in notas:
        if e >= 4.0:
            r += 1
    return r


n = [6.6, 2.3, 4.7, 3.3, 6.1]
a = num_aprobados(n)
print("Aprobados:",a)
```

```
>>>
Aprobados: 3
>>>
```

LONGITUD

0

```
#Lista vacia
p = []
lp = len(p)
print(lp)
```

3

```
#Lista de enteros
q = [5, 7, -1]
lq = len(q)
print(lq)
```

4

```
#Lista de floats
r = [5.5, 7.0, 4.0, 2.4]
lr = len(r)
print(lr)
```

2

```
#Lista de strings
s = ["hola", "adios"]
ls = len(s)
print(ls)
```

4

```
#Lista de bools
t = [True,True,False,True]
lt = len(t)
print(lt)
```

5

```
#Listas de varios tipos
u = [True,"hola",7.0,"adios",-1]
lu = len(u)
print(lu)
```

memesintroalaprogr • Following

58 likes

OCTOBER 13, 2019

Add a comment... Post

ACCEDER AL ELEMENTO EN UNA POSICIÓN

```
p = ["arbol","barco","casa","dado"]

a = p[0]
print(a)

b = p[1]
print(b)

ult = len(p)-1
c = p[ult]
print(c)
```

arbol

barco

dado

memesintroalaprogr • Following

58 likes

OCTOBER 13, 2019

Add a comment... Post

APROBADOS (FOR IN RANGE / WHILE)

INTERESES

“cosas cuotidianas de la universidad”

```
def num_aprobados(notas):
    r = 0
    for e in notas:
        if e >= 4.0:
            r += 1
    return r


n = [6.6, 2.3, 4.7, 3.3, 6.1]
a = num_aprobados(n)
print("Aprobados:",a)
```

```
>>>
Aprobados: 3
>>>
```

```
def num_aprobados_fr(notas):
```

```
n = [6.6, 2.3, 4.7, 3.3, 6.1]
a = num_aprobados(n)
print("Aprobados:",a)
```

```
>>>
Aprobados: 3
>>>
```


I'LL WAIT
FOR YOU HERE

```
def num_aprobados_fr(notas):
```

```
n = [6.6, 2.3, 4.7, 3.3, 6.1]
a = num_aprobados(n)
print("Aprobados:",a)
```

```
>>>
Aprobados: 3
>>>
```

```
def num_aprobados_fr(notas):
    r = 0
    for i in range(0,len(notas)):
        if notas[i] >= 4.0:
            r += 1
    return r


n = [6.6, 2.3, 4.7, 3.3, 6.1]
a = num_aprobados(n)
print("Aprobados:",a)
```

```
>>>
Aprobados: 3
>>>
```

```
def num_aprobados_wh(notas):
```

```
n = [6.6, 2.3, 4.7, 3.3, 6.1]
a = num_aprobados(n)
print("Aprobados:",a)
```

```
>>>
Aprobados: 3
>>>
```


I'LL WAIT
FOR YOU HERE

```
def num_aprobados_wh(notas):
```

```
n = [6.6, 2.3, 4.7, 3.3, 6.1]
a = num_aprobados(n)
print("Aprobados:",a)
```

```
>>>
Aprobados: 3
>>>
```

```
def num_aprobados_wh(notas):
    r = 0
    i = 0
    while i < len(notas):
        if notas[i] >= 4.0:
            r += 1
        i += 1
    return r


n = [6.6, 2.3, 4.7, 3.3, 6.1]
a = num_aprobados(n)
print("Aprobados:",a)
```

```
>>>
Aprobados: 3
>>>
```

```python
def num_aprobados(notas):
    r = 0
    for e in notas:
        if e >= 4.0:
            r += 1
    return r
```

```python
def num_aprobados_fr(notas):
    r = 0
    for i in range(0,len(notas)):
        if notas[i] >= 4.0:
            r += 1
    return r
```

```python
def num_aprobados_wh(notas):
    r = 0
    i = 0
    while i < len(notas):
        if notas[i] >= 4.0:
            r += 1
        i += 1
    return r
```

AGREGAR

```
v = ["arbol","barco","casa"]
x = "zebra"

#Agregar (al final) a una lista
v.append(x)
print(v)
```

```
['arbol', 'barco', 'casa', 'zebra']
```

SEMANA DE AJUSTE

INTERESES

“cosas cuotidianas de la universidad”


```
Numero de estudiantes? 5
Estudiante 0
Ajuste 1-Si 2-No? 1
Estudiante 1
Ajuste 1-Si 2-No? 2
Estudiante 2
Ajuste 1-Si 2-No? 1
Estudiante 3
Ajuste 1-Si 2-No? 1
Estudiante 4
Ajuste 1-Si 2-No? 2
SI: 3 NO: 2
SE APRUEBA
```

```
Numero de estudiantes? 3
Estudiante 0
Ajuste 1-Si 2-No? 2
Estudiante 1
Ajuste 1-Si 2-No? 1
Estudiante 2
Ajuste 1-Si 2-No? 2
SI: 1 NO: 2
SE RECHAZA
```

```
Numero de estudiantes? 2
Estudiante 0
Ajuste 1-Si 2-No? 1
Estudiante 1
Ajuste 1-Si 2-No? 2
SI: 1 NO: 1
EMPATE
```


I'LL WAIT
FOR YOU HERE


```
Numero de estudiantes? 5
Estudiante 0
Ajuste 1-Si 2-No? 1
Estudiante 1
Ajuste 1-Si 2-No? 2
Estudiante 2
Ajuste 1-Si 2-No? 1
Estudiante 3
Ajuste 1-Si 2-No? 1
Estudiante 4
Ajuste 1-Si 2-No? 2
SI: 3 NO: 2
SE APRUEBA
```

```
Numero de estudiantes? 3
Estudiante 0
Ajuste 1-Si 2-No? 2
Estudiante 1
Ajuste 1-Si 2-No? 1
Estudiante 2
Ajuste 1-Si 2-No? 2
SI: 1 NO: 2
SE RECHAZA
```

```
Numero de estudiantes? 2
Estudiante 0
Ajuste 1-Si 2-No? 1
Estudiante 1
Ajuste 1-Si 2-No? 2
SI: 1 NO: 1
EMPATE
```

```
vs = []

n = int(input("Numero de estudiantes? "))

#Pedir y recopilar los votos
for i in range(0,n):
    print("Estudiante",i)
    v = int(input("Ajuste 1-Si 2-No? "))
    vs.append(v)
```

```
vs = []

n = int(input("Numero de estudiantes? "))

#Pedir y recopilar los votos
for i in range(0,n):
    print("Estudiante",i)
    v = int(input("Ajuste 1-Si 2-No? "))
    vs.append(v)

#Contar los votos
numsi = 0
numno = 0
for e in vs:
    if e == 1:
        numsi += 1
    elif e == 2:
        numno += 1
```

...

```
vs = []

n = int(input("Numero de estudiantes? "))

#Pedir y recopilar los votos
for i in range(0,n):
    print("Estudiante",i)
    v = int(input("Ajuste 1-Si 2-No? "))
    vs.append(v)

#Contar los votos
numsi = 0
numno = 0
for e in vs:
    if e == 1:
        numsi += 1
    elif e == 2:
        numno += 1

#Se imprime el resultado
print("SI:",numsi,"NO:",numno)
if numsi > numno:
    print("SE APRUEBA")
elif numsi < numno:
    print("SE RECHAZA")
else:
    print("EMPATE")
```

CONTENIDO

```
p = ["aa","bb","cc","dd","ee","22"]

x = "aa" in p
print(x)

y = "zz" in p
print(y)

z = "a" in p
print(z)

v = 22 in p
print(v)

w = "22" in p
print(w)
```

True

False

False

False

True

memesintroalaprogr • Following

58 likes

OCTOBER 13, 2019

Add a comment... Post

JUEGO DE LOS REPETIDOS

INTERESES

“Me gustaría ver ejemplos en juegos, cualquier tipo de juegos, ya sean de consolas, computadoras, juegos de mesa o los que jugábamos de pequeños (escondidas, pillarse, etc)”

```
Letra: A
Palabra: alemania
Bien!
Palabra: holanda
Bien!
Palabra: Alemania
Has repetido ALEMANIA
Juego acabado
```

```
Letra: i
Palabra: islandia
Bien!
Palabra: irlanda
Bien!
Palabra: holanda
La palabra NO tiene I
Juego acabado
```


I'LL WAIT
FOR YOU HERE

```
Letra: A
Palabra: alemania
Bien!
Palabra: holanda
Bien!
Palabra: Alemania
Has repetido ALEMANIA
Juego acabado
```

```
Letra: i
Palabra: islandia
Bien!
Palabra: irlanda
Bien!
Palabra: holanda
La palabra NO tiene I
Juego acabado
```

```
#Lista de palabras (inicialmente vacia)
ps = []

#Pedir letra
let = input("Letra: ")
let = let.upper()

continuar = True
while continuar:

    #Pedir palabra
    p = input("Palabra: ")

    # En mayusculas, para estandarizar
    p = p.upper()

    if not (let in p):
        print("La palabra NO tiene", let)
        continuar = False

    elif p in ps:
        print("Has repetido",p)
        continuar = False

    else:
        print("Bien!")
        ps.append(p)

print("Juego acabado")
```

TEMAS

Países, ciudades, deportes, actores, películas, libros, …

CUMPLEAÑOS

INTERESES

Situación Real

“cosas curiosas”

“ los juegos pequeños de probabilidad”

“Casinos y apuestas”

“acertijos”

“Matemáticas”

¡APUESTA!

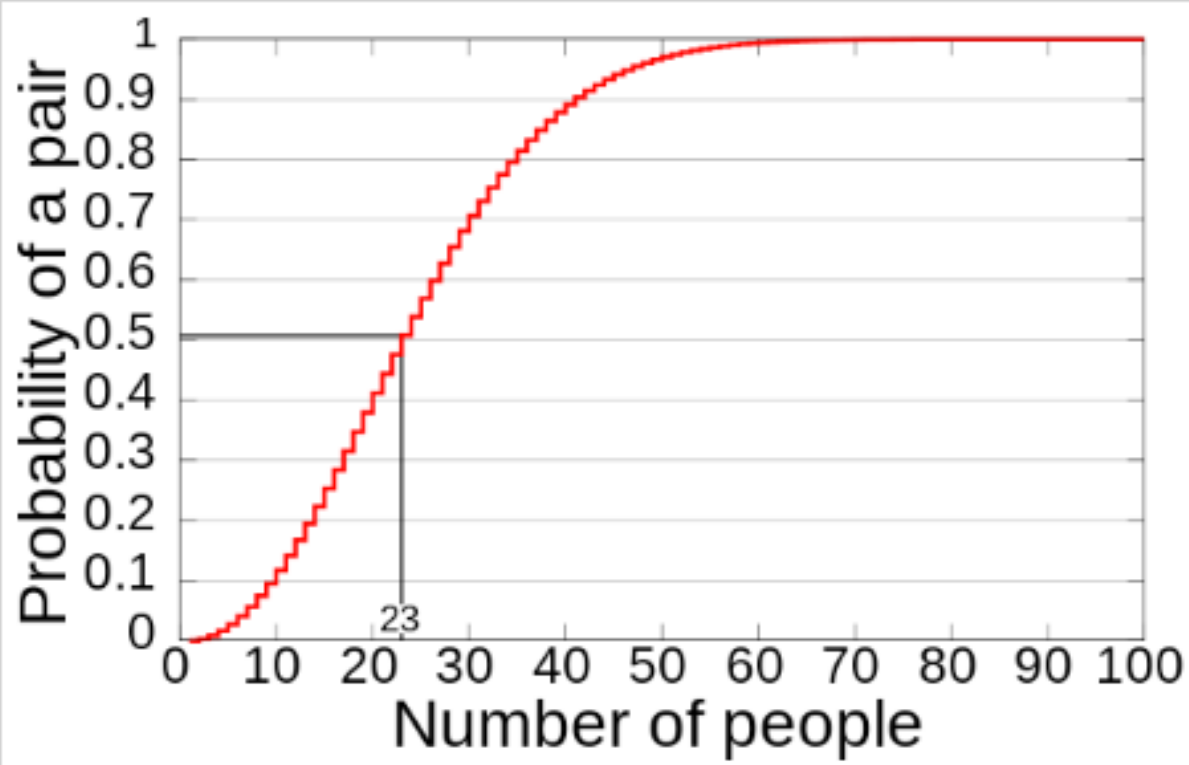
Solo en los primeros **50** estudiantes:

- Si dos estudiantes comparten cumpleaños, doble de problemas de Hackerrank esta semana.
- ¡Si no, **puntos ADA gratis** para todos los de clase!

Recordar: hay **365 días** en 1 año

“

```
>>>
Nombre: Alice
Dia: 30
Mes: 4
Nombre: Bob
Dia: 23
Mes: 4
Nombre: Charlie
Dia: 30
Mes: 4
COINCIDENCIA
Charlie y Alice tienen el cumple el 30 del 4
Has necesitado 3 personas
>>>
```



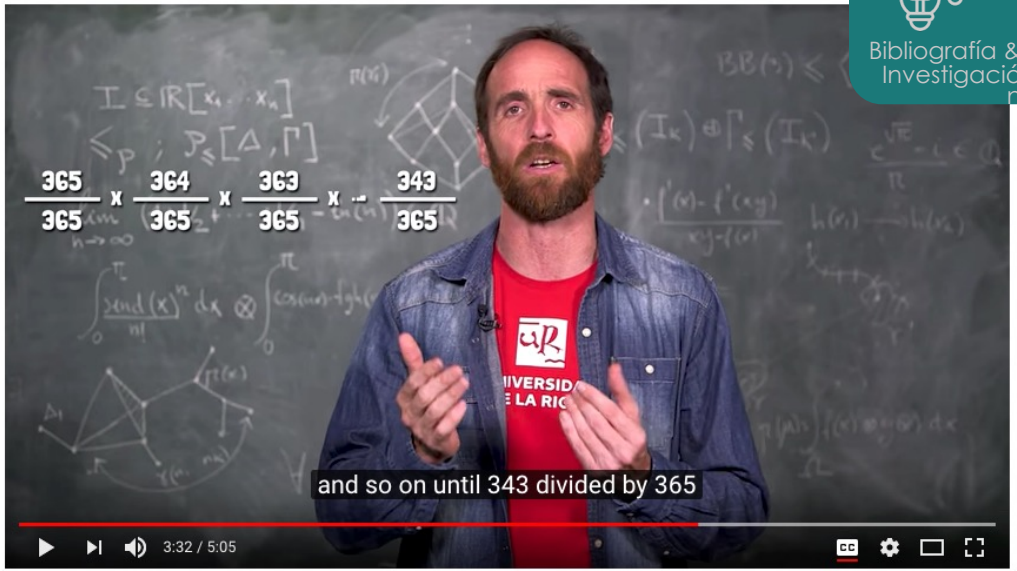
A graph showing the computed probability of at least two people sharing a birthday amongst a certain number of people.


I'LL WAIT
FOR YOU HERE

```
>>>
Nombre: Alice
Dia: 30
Mes: 4
Nombre: Bob
Dia: 23
Mes: 4
Nombre: Charlie
Dia: 30
Mes: 4
COINCIDENCIA
Charlie y Alice tienen el cumple el 30 del 4
Has necesitado 3 personas
>>>
```

```
ns = []
ds = []
ms = []

continuar = True
while continuar:

    n = input("Nombre: ")
    d = input("Dia: ")
    m = input("Mes: ")

    #Mirar si hay coincidencia
    esta = False
    for i in range(0,len(ns)):
        if (ds[i] == d) and (ms[i] == m):
            print("COINCIDENCIA")
            print(n,"y",ns[i],"tienen el cumple el",d,"del",m)
            continuar = False
            esta = True

    if not esta:
        ns.append(n)
        ds.append(d)
        ms.append(m)

print("Has necesitado", len(ns)+1, "personas")
```

- Birthday Paradox
  - https://en.wikipedia.org/wiki/Birthday_problem
  - https://www.youtube.com/watch?v=7uzx6D_0V7M



¡¡Cumples años el mismo día que yo!! ¿Casualidad? | PARADOJA DEL CUMPLEAÑOS

145,508 views 9K 101 SHARE

Derivando
Published on Aug 9, 2017

SUBSCRIBE 274K

¿Conoces a dos personas que nacieron el mismo día? ¿Si? ¡¡Pero qué casualidad!! ¿O no...? La respuesta a esto, está en la llamada "Paradoja del cumpleaños": ¿Cuál es la probabilidad de que, en un determinado grupo de personas, haya dos que cumplan años el mismo día? ¡Sorprendente!

SHOW MORE

PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE

FACULTAD DE MATEMÁTICAS

- Facultad
- Personas
- Departamento de Estadística
- Departamento de Matemática
- Instituto de Ingeniería Matemática y Computacional
- Proyectos Institucionales
- Alumni
- Oportunidades de Trabajo
- Facultad en Cifras
- Información Para Visitantes

EDUARDO SÁENZ DE CABEZÓN REALIZARÁ CHARLA EN LA FACULTAD DE MATEMÁTICAS UC

2019-10-02

- **Eduardo Sáenz de Cabezón, matemático y reconocido divulgador de la ciencia a nivel mundial, dictará una charla abierta en la Facultad de Matemáticas de la UC, en el marco de su visita a Chile con motivo del Festival Exacta 2019.**
- **La actividad es gratuita y abierta, previo registro.**

SLICING Y CONCATENAR

```
p = ["des","pa","ci","to","luis","fonsi"]

a = p[1:3]
print(a)

b = p[:3]
print(b)

c = p[3:]
print(c)

d = p[:]
print(d)
```

```
>>>
['pa', 'ci']
['des', 'pa', 'ci']
['to', 'luis', 'fonsi']
['des', 'pa', 'ci', 'to', 'luis', 'fonsi']
>>>
```

```
x = ["en","un","lugar"]
y = ["de","la","mancha"]
z = x + y
print(z)
```

```
['en', 'un', 'lugar', 'de', 'la', 'mancha']
```

memesintroalaprogr • Following

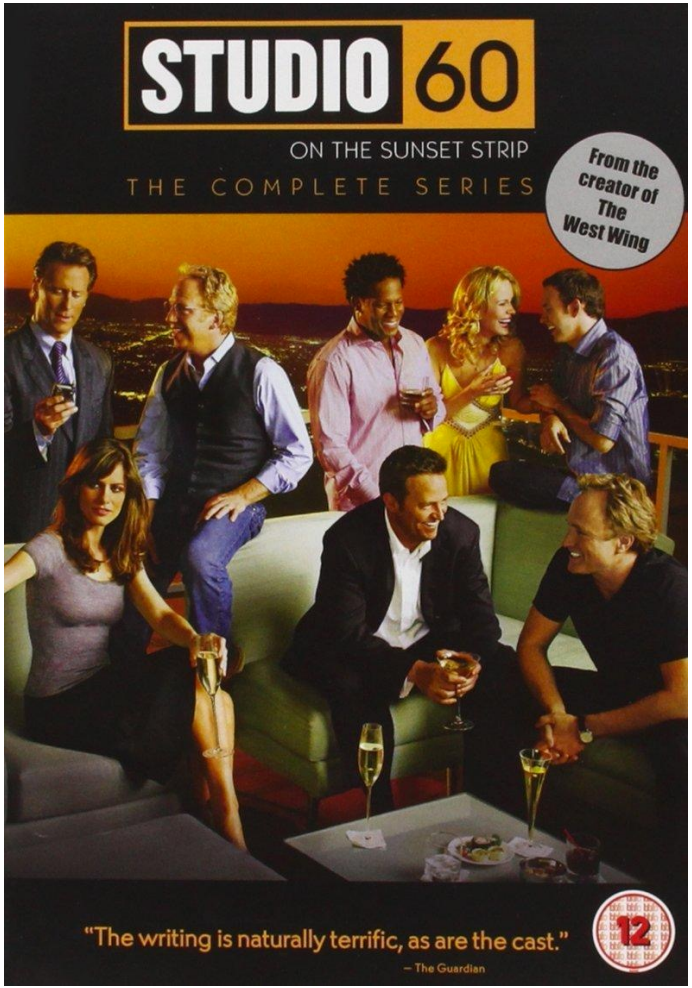
58 likes

OCTOBER 13, 2019

Add a comment... Post

AARON SORKIN


Blu-ray Disc
"MAY BE THE MOST ENTERTAINING SHOW ON TV..." VANITY FAIR
FROM THE CREATOR OF THE WEST WING & WRITER OF THE SOCIAL NETWORK
THE NEWSROOM
THE COMPLETE SERIES


STUDIO 60
ON THE SUNSET STRIP
THE COMPLETE SERIES
From the creator of The West Wing
"The writing is naturally terrific, as are the cast."
– The Guardian
12

```
#'o' list de video original
#'ini' int del inicio del corte
#'fin' int del fin del corte
#'r' list de relleno

def fake(o,ini,fin,r):
    [redacted]

#CODIGO DE TEST
a = ["Estoy","en","contra","del","confinamiento"]
i = 1
f = 2
r = ["a","favor"]
m = fake(a,i,f,r)

print(a)
print(m)
```

```
['Estoy', 'en', 'contra', 'del', 'confinamiento']
['Estoy', 'a', 'favor', 'del', 'confinamiento']
```


I'LL WAIT
FOR YOU HERE

```
#'o' list de video original
#'ini' int del inicio del corte
#'fin' int del fin del corte
#'r' list de relleno

def fake(o,ini,fin,r):
    [redacted]

#CODIGO DE TEST
a = ["Estoy","en","contra","del","confinamiento"]
i = 1
f = 2
r = ["a","favor"]
m = fake(a,i,f,r)

print(a)
print(m)
```

```
['Estoy', 'en', 'contra', 'del', 'confinamiento']
['Estoy', 'a', 'favor', 'del', 'confinamiento']
```

Resolución Problema

```
#'o' list de video original
#'ini' int del inicio del corte
#'fin' int del fin del corte
#'r' list de relleno

def fake(o,ini,fin,r):
    first = o[:ini]
    last = o[fin+1:]
    m = first+r+last
    return m


#CODIGO DE TEST
a = ["Estoy","en","contra","del","confinamiento"]
i = 1
f = 2
r = ["a","favor"]
m = fake(a,i,f,r)

print(a)
print(m)
```

```
['Estoy', 'en', 'contra', 'del', 'confinamiento']
['Estoy', 'a', 'favor', 'del', 'confinamiento']
```

- The Newsroom
  - https://www.imdb.com/title/tt1870479/
- Studio 60 on the Sunset Strip
  - https://www.imdb.com/title/tt0485842

MUTABILIDAD

Conceptos

```
a = "Barça"
a[0] = "F"
a[3] = "$"
print(a)
```

¿Que imprime?

```
Traceback (most recent call last):
  File "/Users/jmunoz/Dropbox/Lectures/IIC1103/IIC1103-6 2016-2 Jorge/material/S6
-Strings/farsa.py", line 8, in <module>
    a[0] = "F"
TypeError: 'str' object does not support item assignment
```

memesintroalaprogr • Following

likes

OCTOBER 13, 2019

Add a comment... Post

```
verdad = ["En","el","Barça","son","unos","mercenarios"]
verdad[2] = "Far$a"
print(verdad)
```

```
['En', 'el', 'Far$a', 'son', 'unos', 'mercenarios']
```

```
ADAs = [43,54,10,33]
ADAs[2] += 6
print(ADAs)
```

```
[43, 54, 16, 33]
```

QUINOA

INTERESES

“Agronomia”

“aplicaciones sobre temas agronómicos, por ejemplo algún robot que deba cultivar y cosechar diversas plantas en tales épocas del año, etc...”

“cuidado del medio ambiente y recursos hídricos”

Situación Real


```
HELLO
1-Nueva 2-Eliminar 3-Hojas 4-Mostrar 0-Salir: 4
X X X X X
0 0 0 0 0
1-Nueva 2-Eliminar 3-Hojas 4-Mostrar 0-Salir: 1
Maceta? 2
Tipo: (B-Blanca N-Negra R-Rojas)? B
1-Nueva 2-Eliminar 3-Hojas 4-Mostrar 0-Salir: 4
X X B X X
0 0 0 0 0
1-Nueva 2-Eliminar 3-Hojas 4-Mostrar 0-Salir: 3
Maceta? 3
ERROR: No hay planta
1-Nueva 2-Eliminar 3-Hojas 4-Mostrar 0-Salir: 3
Maceta? 2
Hojas? 7
1-Nueva 2-Eliminar 3-Hojas 4-Mostrar 0-Salir: 4
X X B X X
0 0 7 0 0
1-Nueva 2-Eliminar 3-Hojas 4-Mostrar 0-Salir: 1
Maceta? 1
Tipo: (B-Blanca N-Negra R-Rojas)? N
1-Nueva 2-Eliminar 3-Hojas 4-Mostrar 0-Salir: 4
X N B X X
0 0 7 0 0
```

```
1-Nueva 2-Eliminar 3-Hojas 4-Mostrar 0-Salir: 3
Maceta? 1
Hojas? 3
1-Nueva 2-Eliminar 3-Hojas 4-Mostrar 0-Salir: 4
X N B X X
0 3 7 0 0
1-Nueva 2-Eliminar 3-Hojas 4-Mostrar 0-Salir: 2
Maceta? 1
1-Nueva 2-Eliminar 3-Hojas 4-Mostrar 0-Salir: 4
X X B X X
0 0 7 0 0
1-Nueva 2-Eliminar 3-Hojas 4-Mostrar 0-Salir: 1
Maceta? 1
Tipo: (B-Blanca N-Negra R-Rojas)? R
1-Nueva 2-Eliminar 3-Hojas 4-Mostrar 0-Salir: 3
Maceta? 1
Hojas? 5
1-Nueva 2-Eliminar 3-Hojas 4-Mostrar 0-Salir: 4
X R B X X
0 5 7 0 0
1-Nueva 2-Eliminar 3-Hojas 4-Mostrar 0-Salir: 0
CIAO
```

- 5 Macetas (0,1,2,3,4)
- Registrar que he plantado una planta, la he eliminado, o cuantas hojas he contado
- Mostrar que tipo y cuantas hojas hay actualmente en las macetas
- Salir

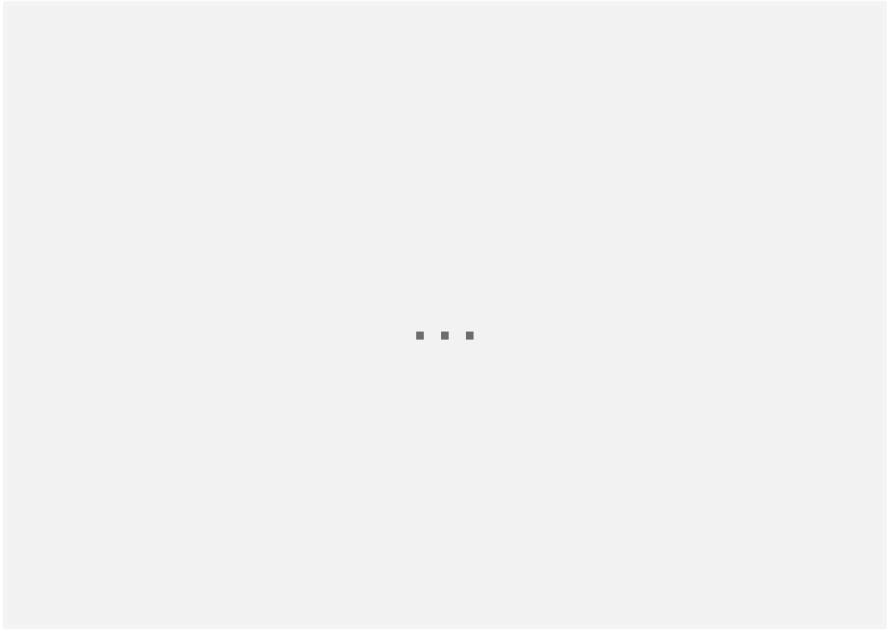

I'LL WAIT
FOR YOU HERE


```
HELLO
1-Nueva 2-Eliminar 3-Hojas 4-Mostrar 0-Salir: 4
X X X X X
0 0 0 0 0
1-Nueva 2-Eliminar 3-Hojas 4-Mostrar 0-Salir: 1
Maceta? 2
Tipo: (B-Blanca N-Negra R-Rojas)? B
1-Nueva 2-Eliminar 3-Hojas 4-Mostrar 0-Salir: 4
X X B X X
0 0 0 0 0
1-Nueva 2-Eliminar 3-Hojas 4-Mostrar 0-Salir: 3
Maceta? 3
ERROR: No hay planta
1-Nueva 2-Eliminar 3-Hojas 4-Mostrar 0-Salir: 3
Maceta? 2
Hojas? 7
1-Nueva 2-Eliminar 3-Hojas 4-Mostrar 0-Salir: 4
X X B X X
0 0 7 0 0
1-Nueva 2-Eliminar 3-Hojas 4-Mostrar 0-Salir: 1
Maceta? 1
Tipo: (B-Blanca N-Negra R-Rojas)? N
1-Nueva 2-Eliminar 3-Hojas 4-Mostrar 0-Salir: 4
X N B X X
0 0 7 0 0
```

```
1-Nueva 2-Eliminar 3-Hojas 4-Mostrar 0-Salir: 3
Maceta? 1
Hojas? 3
1-Nueva 2-Eliminar 3-Hojas 4-Mostrar 0-Salir: 4
X N B X X
0 3 7 0 0
1-Nueva 2-Eliminar 3-Hojas 4-Mostrar 0-Salir: 2
Maceta? 1
1-Nueva 2-Eliminar 3-Hojas 4-Mostrar 0-Salir: 4
X X B X X
0 0 7 0 0
1-Nueva 2-Eliminar 3-Hojas 4-Mostrar 0-Salir: 1
Maceta? 1
Tipo: (B-Blanca N-Negra R-Rojas)? R
1-Nueva 2-Eliminar 3-Hojas 4-Mostrar 0-Salir: 3
Maceta? 1
Hojas? 5
1-Nueva 2-Eliminar 3-Hojas 4-Mostrar 0-Salir: 4
X R B X X
0 5 7 0 0
1-Nueva 2-Eliminar 3-Hojas 4-Mostrar 0-Salir: 0
CIAO
```

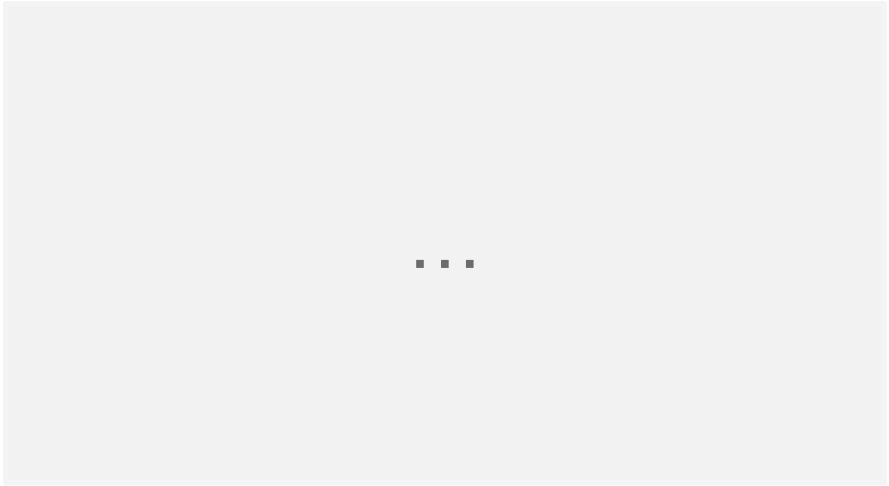
- 5 Macetas (0,1,2,3,4)
- Registrar que he plantado una planta, la he eliminado, o cuantas hojas he contado
- Mostrar que tipo y cuantas hojas hay actualmente en las macetas
- Salir

```
ns = ["X","X","X","X","X"]
hs = [0 ,0 ,0 ,0 ,0]

print("HELLO")

continuar = True
while continuar:
    op = int(input("1-Nueva 2-Eliminar 3-Hojas 4-Mostrar 0-Salir: "))

    if op == 0:
        continuar = False
```



```
print("CIAO")
```

```
ns = ["X","X","X","X","X"]
hs = [0 ,0 ,0 ,0 ,0]

print("HELLO")

continuar = True
while continuar:
    op = int(input("1-Nueva 2-Eliminar 3-Hojas 4-Mostrar 0-Salir: "))

    if op == 0:
        continuar = False

    elif op == 1:
        s = int(input("Maceta? "))
        n = input("Tipo: (B-Blanca N-Negra R-Rojas)? ")
        ns[s] = n
```



```
print("CIAO")
```

```
ns = ["X","X","X","X","X"]
hs = [0 ,0 ,0 ,0 ,0]

print("HELLO")

continuar = True
while continuar:
    op = int(input("1-Nueva 2-Eliminar 3-Hojas 4-Mostrar 0-Salir: "))

    if op == 0:
        continuar = False

    elif op == 1:
        s = int(input("Maceta? "))
        n = input("Tipo: (B-Blanca N-Negra R-Rojas)? ")
        ns[s] = n

    elif op == 2:
        s = int(input("Maceta? "))
        ns[s] = "X"
        hs[s] = 0
```

```
print("CIAO")
```

```
ns = ["X","X","X","X","X"]
hs = [0 ,0 ,0 ,0 ,0]

print("HELLO")

continuar = True
while continuar:
    op = int(input("1-Nueva 2-Eliminar 3-Hojas 4-Mostrar 0-Salir: "))

    if op == 0:
        continuar = False

    elif op == 1:
        s = int(input("Maceta? "))
        n = input("Tipo: (B-Blanca N-Negra R-Rojas)? ")
        ns[s] = n

    elif op == 2:
        s = int(input("Maceta? "))
        ns[s] = "X"
        hs[s] = 0

    elif op == 3:
        s = int(input("Maceta? "))
        if ns[s] == "X":
            print("ERROR: No hay planta")
        else:
            h = int(input("Hojas? "))
            hs[s] = h

                    ...

print("CIAO")
```

```
ns = ["X","X","X","X","X"]
hs = [0 ,0 ,0 ,0 ,0]

print("HELLO")

continuar = True
while continuar:
    op = int(input("1-Nueva 2-Eliminar 3-Hojas 4-Mostrar 0-Salir: "))

    if op == 0:
        continuar = False

    elif op == 1:
        s = int(input("Maceta? "))
        n = input("Tipo: (B-Blanca N-Negra R-Rojas)? ")
        ns[s] = n

    elif op == 2:
        s = int(input("Maceta? "))
        ns[s] = "X"
        hs[s] = 0

    elif op == 3:
        s = int(input("Maceta? "))
        if ns[s] == "X":
            print("ERROR: No hay planta")
        else:
            h = int(input("Hojas? "))
            hs[s] = h

    elif op == 4:
        print(ns[0],ns[1],ns[2],ns[3],ns[4])
        print(hs[0],hs[1],hs[2],hs[3],hs[4])


print("CIAO")
```

COPIANDO LISTAS

El concepto no entra en el temario,

pero puede ser que os encontréis su efecto cuando programéis por vuestra cuenta

```
a = 10
b = a
b = 5
print(a)
print(b)
```

¿print?

```
10
5
```

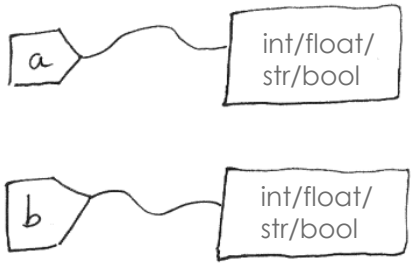
```
a = ["arbol","barco","casa"]
b = a
a.append("dado")
print(a)
print(b)
```

¿print?

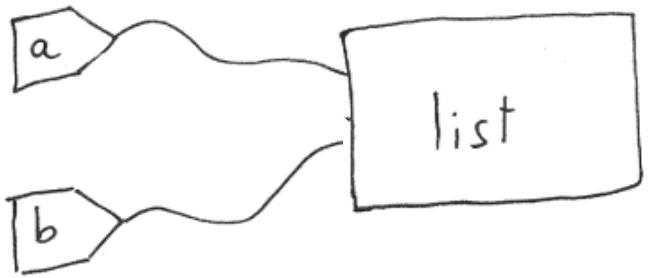
```
['arbol', 'barco', 'casa', 'dado']
['arbol', 'barco', 'casa', 'dado']
```

```
a = 10
b = a
b = 5
print(a)
print(b)
```

```
10
5
```

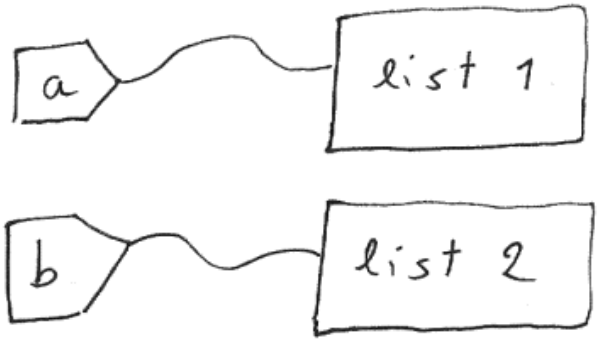


```
a = ["arbol","barco","casa"]
b = a
a.append("dado")
print(a)
print(b)
```




```
['arbol', 'barco', 'casa', 'dado']
['arbol', 'barco', 'casa', 'dado']
```

```
a = ["arbol","barco","casa"]
b = a[:] #Hace una copia
a.append("dado")
print(a)
print(b)
```




```
['arbol', 'barco', 'casa', 'dado']
['arbol', 'barco', 'casa']
```

MÉTODOS

```
#Agregar al final
p = ["arbol","barco","casa","dado"]
p.append("estufa")
print(p)

#Sacar el elemento de la posicion tal
q = ["arbol","barco","casa","dado"]
q.pop(2)
print(q)

#Pop no solo lo saca, sino que lo retorna
#asi que lo podemos guardar en una variable
#si nos interesa
r = ["arbol","barco","casa","dado"]
x = r.pop(2)
print(x)
print(r)

#Si queremos agregar un elemento en una posicion concreta
#lo podemos hacer con slicing y concatenar
s = ["arbol","barco","dado","estufa"]
t = s[0:2]+["caracol"]+s[2:]
print(t)
```

TEMARIO SOLO ESTOS 2:
append y **pop**

```
['arbol', 'barco', 'casa', 'dado', 'estufa']
['arbol', 'barco', 'dado']
casa
['arbol', 'barco', 'dado']
['arbol', 'barco', 'caracol', 'dado', 'estufa']
```

ELIMINAR LOS 6

100% real (no fake)

Primer semestre de Intro

```
def eliminar6(x):
```

```
p = [5,2,5,7,6,3,6]
q = eliminar6(p)
print(q)
```

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


I'LL WAIT
FOR YOU HERE

```
def eliminar6(x):
```

```
p = [5,2,5,7,6,3,6]
q = eliminar6(p)
print(q)
```

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

```
def eliminar6(x):
    i = 0
    while i < len(x):
        if x[i] == 6:
            x.pop(i)
        i += 1
    return x

p = [5,2,5,7,6,3,6]
q = eliminar6(p)
print(q)
```

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

```
def eliminar6(x):
    i = 0
    while i < len(x):
        if x[i] == 6:
            x.pop(i)
        i += 1
    return x
```

```
[5,6,6,6,6,3,6]

[5, 6, 6, 3]
```

CONSEJO

Mejor **agregar** a una lista las cosas que queréis

que **intentar** eliminar las cosas que no queréis

```
def eliminar6buena(x):
    r = []
    for e in x:
        if e != 6:
            r.append(e)
    return r
```

```
[5,6,6,6,6,3,6]
```

```
[5, 3]
```

Se puede eliminando si se tiene cuidado.

Pero os recomiendo mucho mi consejo

```
def eliminar6cuidado(x):
    i = 0
    while i < len(x):
        if x[i] == 6:
            x.pop(i)
        else:
            i += 1
    return x
```

```
[5,6,6,6,6,3,6]

[5, 3]
```

100 PRISIONEROS

Situación Real

INTERESES

“probabilidades en juegos, tales como, black jack 21”

“Como resolver un problema matemático”

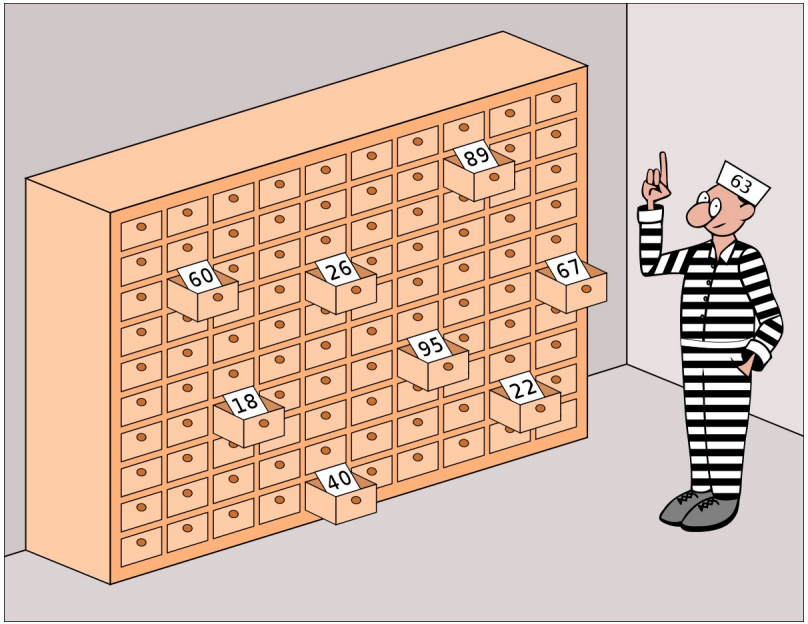
“Matematicas”

2 Partes: CONTEXTO y PROBLEMA

Contexto puede ser un poco complejo

No se necesita entender el contexto para resolver el problema

CONTEXTO


89
63
60
26
67
95
22
18
40


DOESN'T FEEL LIKE HOME ANY-MORE!
PFFF! AMATEURS!
I HATE PINKERTON! I'M GONNA DESTROY PINKERTON!!

¿Y si abrimos todos las 50 primeras cajas?

0

(todos los que tienen su numero en las 50 segundas cajas no lo encontrarán)

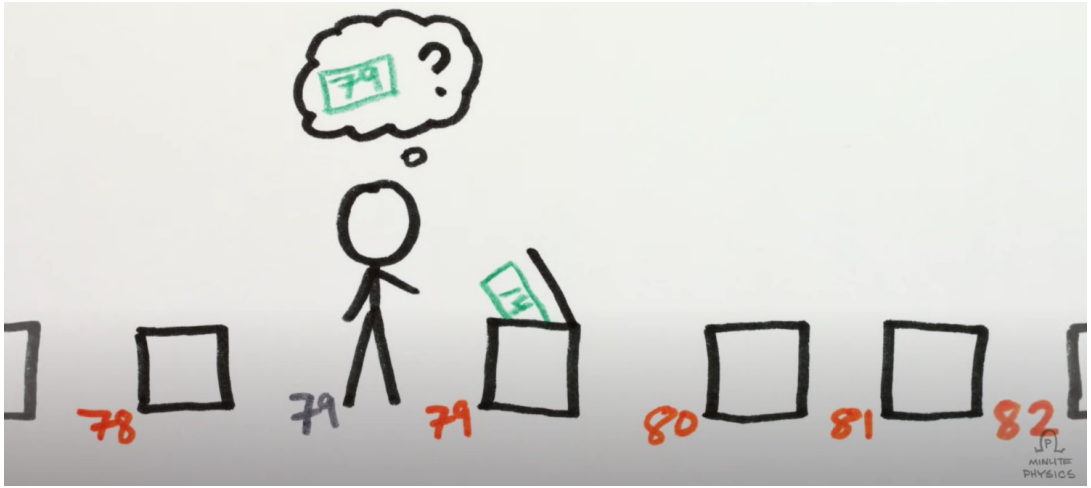
¿Y si cada uno abre 50 cajas al azar?

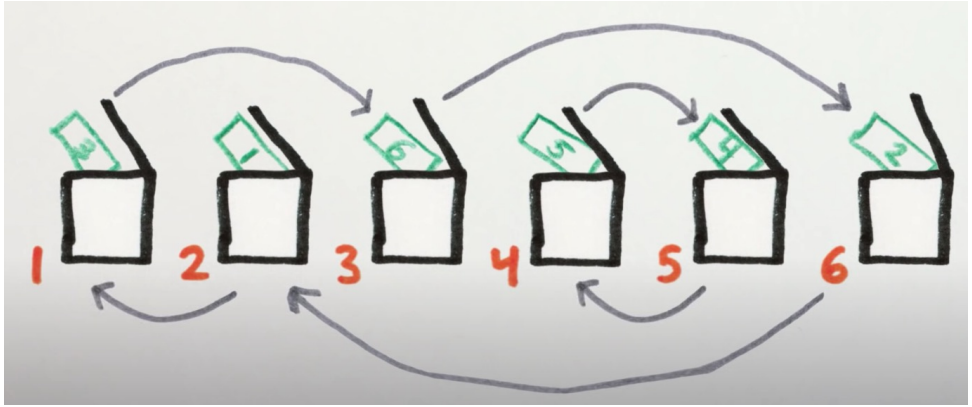
$$\underbrace{\tfrac{1}{2} \times \tfrac{1}{2} \times \tfrac{1}{2} \times \tfrac{1}{2} \times \cdots \times \tfrac{1}{2}}_{100 \text{ times}}$$

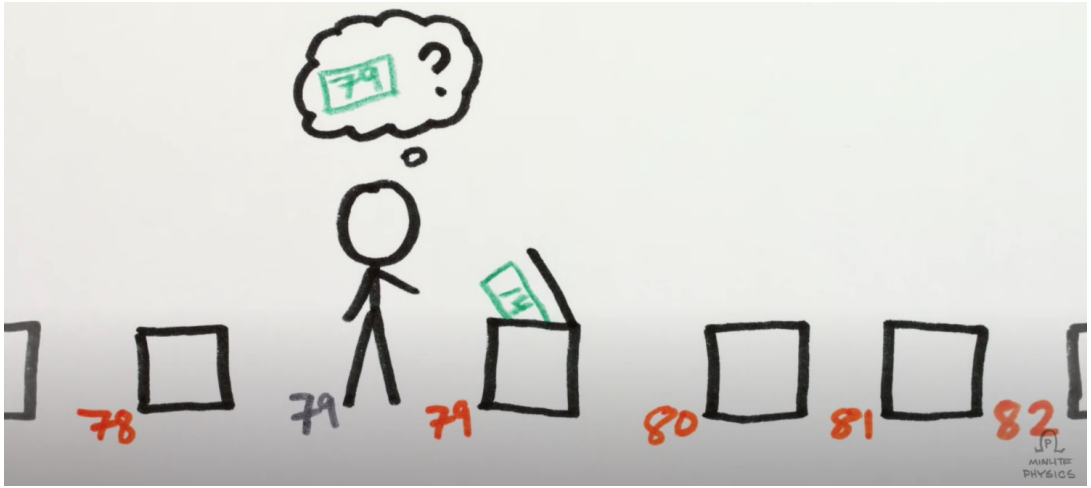
$$\left(\frac{1}{2}\right)^{100} = 0.000000000000000000000000000000008$$

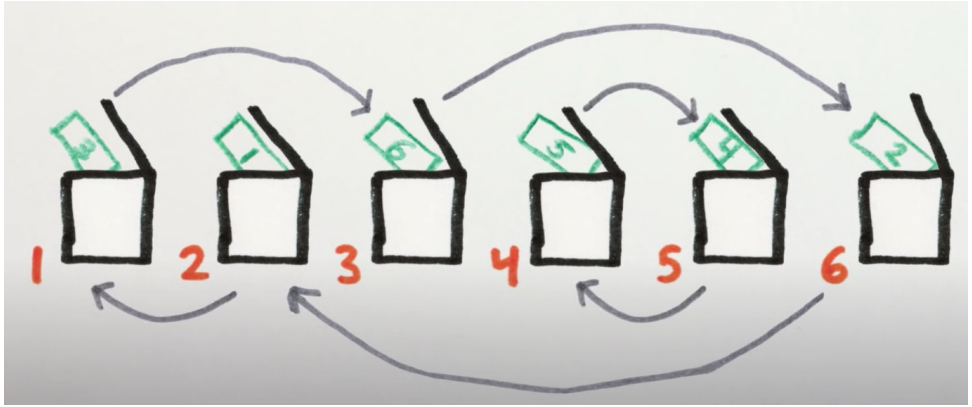
$$\approx 0$$

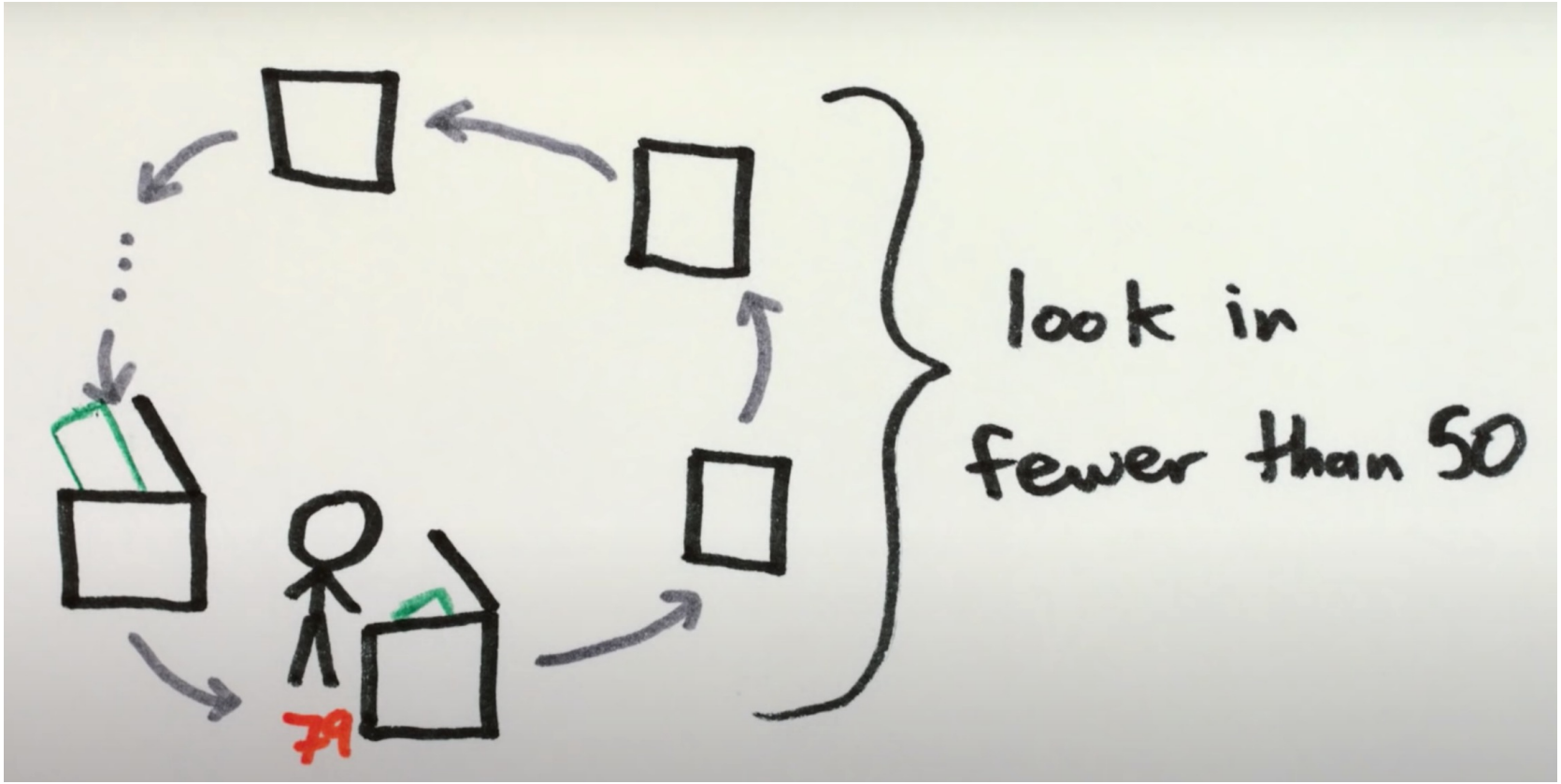
¿Y si os digo que hay una estrategia con un 31% de posibilidades de ganar?

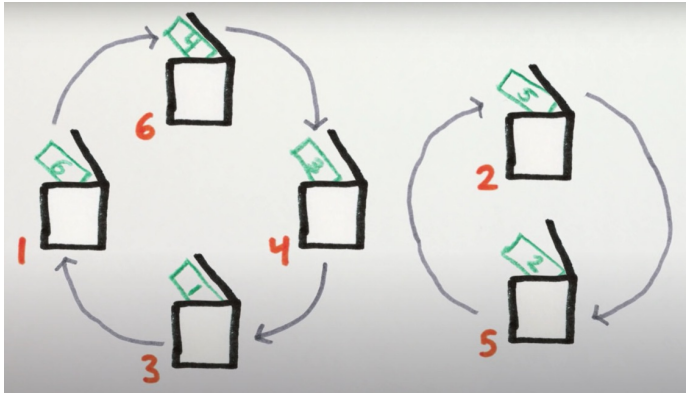

79
?
78
79
79
80
81
82
MINUTE PHYSICS

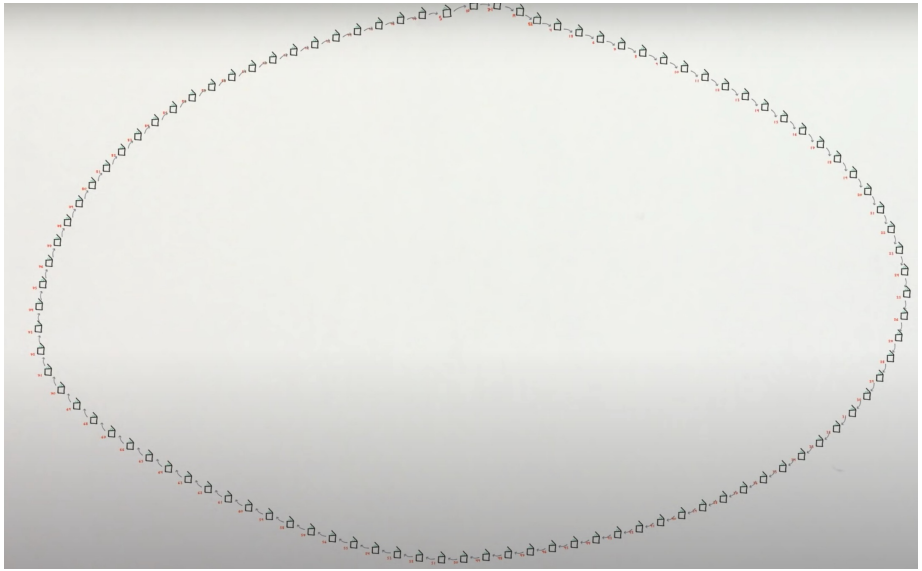

1
2
3
4
5
6

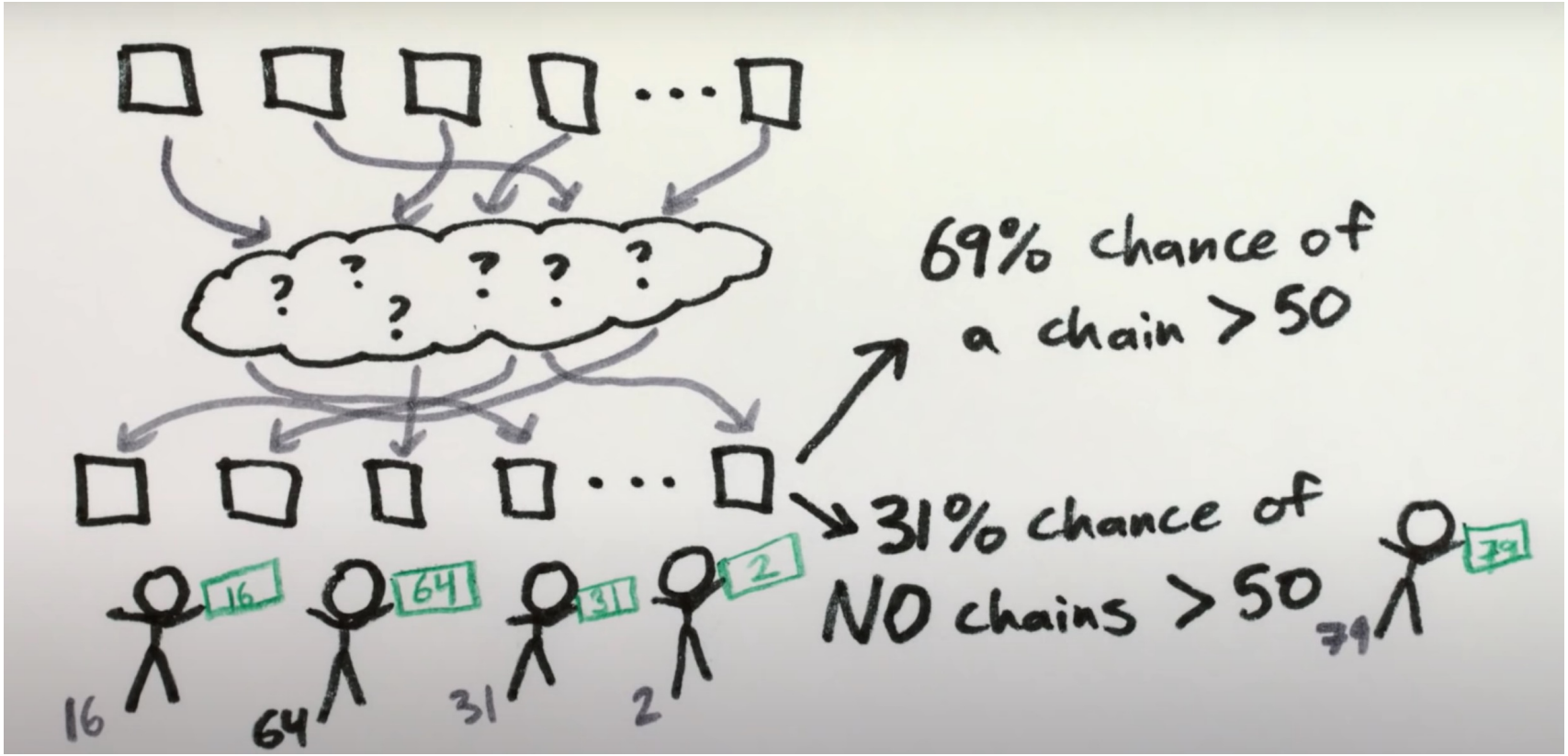

79
?
78
79
79
80
81
82
MINUTE PHYSICS


1
2
3
4
5
6


look in
fewer than 50
79


6
1
4
3
2
5




69% chance of a chain > 50
31% chance of NO chains > 50
16
64
31
2
79

A permutation of the numbers of the numbers from 1 to 100 can only contain at most one cycle of length $l > 50$, where l is the length of the longest cycle. We want to calculate how many different permutations would have a cycle of length l, in order to calculate the probability of having a cycle of that length.

There are $\binom{100}{l}$ ways to select which numbers are in the cycle. Within the cycle, there are $(l-1)!$ ways of organizing the numbers because of the cyclic symmetry. Lastly, the remaining numbers can be arranged in $(100-l)!$ ways. By consequence, the numbers of permutations of the numbers from 1 to 100 with a cycle of length $l > 50$ is

$$\binom{100}{l}(l-1)!(100-l)! = \frac{100!}{l}$$

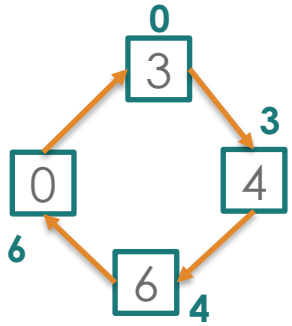
Since there are a total of 100! possible permutations, then the probability of success of this strategy is

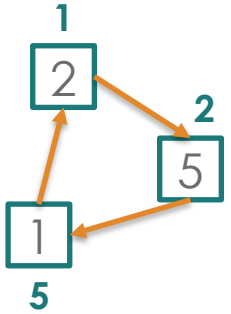
$$1 - \frac{1}{100!}\left(\frac{100!}{51} + \ldots + \frac{100!}{100}\right) = 1 - \left(\frac{1}{51} + \ldots + \frac{1}{100}\right) \approx 0.311$$

Even if we increase the number of prisoners to $2n$ and each prisoners can open

PROBLEMA

```
def num_abiertas(cs,x):
```



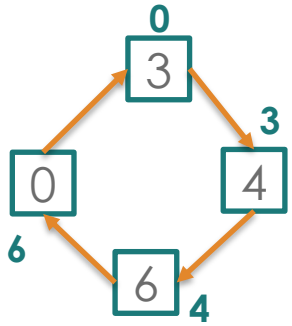


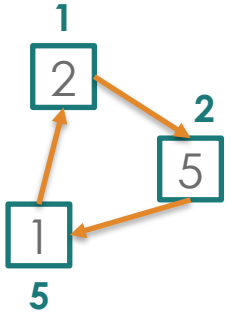

```
#TEST
cs = [3,2,5,4,6,1,0]
print("Prisionero",3,"Abiertas",num_abiertas(cs,3))
print("Prisionero",2,"Abiertas",num_abiertas(cs,2))
```

```
Prisionero 3 Abiertas 4
Prisionero 2 Abiertas 3
```


I'LL WAIT
FOR YOU HERE

```
def num_abiertas(cs,x):
```

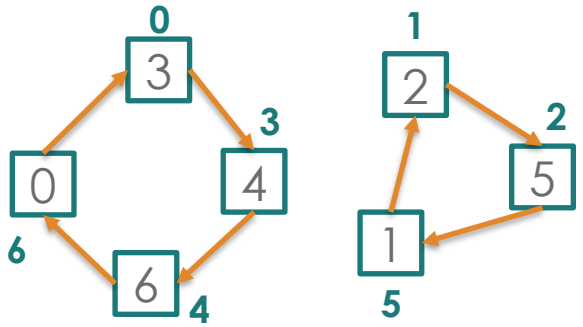





```
#TEST
cs = [3,2,5,4,6,1,0]
print("Prisionero",3,"Abiertas",num_abiertas(cs,3))
print("Prisionero",2,"Abiertas",num_abiertas(cs,2))
```

```
Prisionero 3 Abiertas 4
Prisionero 2 Abiertas 3
```

```
def num_abiertas(cs,x):
    num = 0
    a = x
    encontrado = False

    while not encontrado:
        v = cs[a]
        num += 1
        if v == x:
            encontrado = True
        else:
            a = v

    return num
```



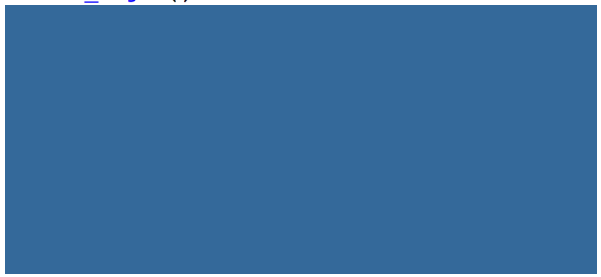

```
#TEST
cs = [3,2,5,4,6,1,0]
print("Prisionero",3,"Abiertas",num_abiertas(cs,3))
print("Prisionero",2,"Abiertas",num_abiertas(cs,2))
```

```
Prisionero 3 Abiertas 4
Prisionero 2 Abiertas 3
```

```
Veces LIBRES: 3134
Veces MUERTOS: 6866
```

Situación Real

```python
import random

def num_abiertas(cs,x):
    ...

def crear_cajas():
```

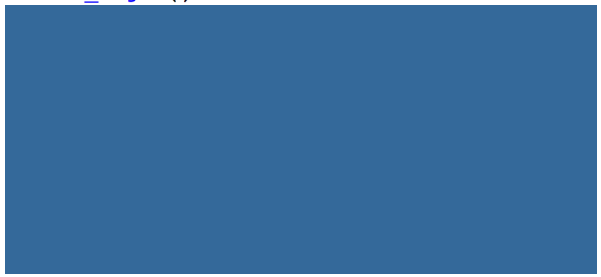


```python

#CODIGO PRINCIPAL
numlibres = 0
nummuertos = 0

#Numero de carceles donde jugar al juego
carceles = 10000
for c in range(0,carceles):

    #Crear cajas al azar
    cs = crear_cajas()

    #Cada prisionero se pone a abrir las cajas
    libres = True
    for i in range(0,100):
        if num_abiertas(cs,i) > 50:
            libres = False

    if libres:
        numlibres += 1
    else:
        nummuertos += 1

#Imprimir resultados
print("Veces LIBRES:",numlibres)
print("Veces MUERTOS:", nummuertos)
```


I'LL WAIT
FOR YOU HERE

Situación Real

```python
import random

def num_abiertas(cs,x):
    ...

def crear_cajas():
```



```python

#CODIGO PRINCIPAL
numlibres = 0
nummuertos = 0

#Numero de carceles donde jugar al juego
carceles = 10000
for c in range(0,carceles):

    #Crear cajas al azar
    cs = crear_cajas()

    #Cada prisionero se pone a abrir las cajas
    libres = True
    for i in range(0,100):
        if num_abiertas(cs,i) > 50:
            libres = False

    if libres:
        numlibres += 1
    else:
        nummuertos += 1

#Imprimir resultados
print("Veces LIBRES:",numlibres)
print("Veces MUERTOS:", nummuertos)
```

Resolución Problema

```
import random

def num_abiertas(cs,x):
    ...

def crear_cajas():

    ordenadas = []
    for i in range(0,100):
        ordenadas.append(i)

    desordenadas = []
    for i in range(0,100):
        x = random.randint(0,len(ordenadas)-1)
        a = ordenadas.pop(x)
        desordenadas.append(a)

    return desordenadas


#CODIGO PRINCIPAL
numlibres = 0
nummuertos = 0

#Numero de carceles donde jugar al juego
carceles = 10000
for c in range(0,carceles):

    #Crear cajas al azar
    cs = crear_cajas()

    #Cada prisionero se pone a abrir las cajas
    libres = True
    for i in range(0,100):
        if num_abiertas(cs,i) > 50:
            libres = False

    if libres:
        numlibres += 1
    else:
        nummuertos += 1

#Imprimir resultados
print("Veces LIBRES:",numlibres)
print("Veces MUERTOS:", nummuertos)
```

- 100 Prisoners Problem
  - https://en.wikipedia.org/wiki/100_prisoners_problem
- Demostración
  - http://www-math.mit.edu/~apost/courses/18.204_2018/Timothee_Schoen_paper.pdf
- Explicación (Muy buena)
  - https://www.youtube.com/watch?v=C5-l0bAuEUE

SPLIT

"arbol,barco,casa,dado,espejo"

['arbol', 'barco', 'casa', 'dado', 'espejo']

"esto es una frase"

['esto', 'es', 'una', 'frase']

```
x = "arbol,barco,casa,dado,espejo"
p = x.split(",")
print(p)
```

```
['arbol', 'barco', 'casa', 'dado', 'espejo']
```

```
y = "esto es una frase"
q = y.split(" ")
print(q)
```

```
['esto', 'es', 'una', 'frase']
```

```
a = "holaUWUhermosoUWUmundo"
s = a.split("UWU")
print(s)
```

```
['hola', 'hermoso', 'mundo']
```

Split siempre devuelve una lista de **STRINGS**

```
z = "1;3;23;8"
r = z.split(";")
print(r)
```

```
['1', '3', '23', '8']
```

¿Jorge, hay algún parámetro de Split o algún comando que nos devuelva una lista de ints?

¡PERO POR QUE OS COMPLICAIS TANTO LA VIDA!

```
a = "1;3;23;8"
p = a.split(";")

#Hazlo tu mismo trivialmente
r = []
for n in p:
    x = int(n)
    r.append(x)

print(r)
```

```
[1, 3, 23, 8]
```

ANID

Situación Real

5. Fundamente sobre el potencial e idoneidad del/de la candidato/a para proseguir investigación o estudios de postgrado en la disciplina escogida. (*)

Quedan 120 palabras.

Si, a veces, pones las cosas sin espacios, lo cuenta como una palabra. Lo mismo pasa si no pones espacios en, por ejemplo, los puntos.

Words 25 of 25

Si,a veces,pones las cosas sin espacios,lo cuenta como una palabra.Lo mismo pasa si no pones espacios en,por ejemplo,los puntos.

Words 25 of 25

Las ”,” “.” “:” y “;” separan palabras

```
Texto? Si, a veces, pones las cosas sin espacios, lo cuenta como una palabra. Lo
 mismo pasa si no pones espacios en, por ejemplo, los puntos.
25 Palabras
```

```
Texto? Si,a veces,pones las cosas sin espacios,lo cuenta como una palabra.Lo mis
mo pasa si no pones espacios en,por ejemplo,los puntos.
25 Palabras
```


I'LL WAIT
FOR YOU HERE

```
Texto? Si, a veces, pones las cosas sin espacios, lo cuenta como una palabra. Lo
 mismo pasa si no pones espacios en, por ejemplo, los puntos.
25 Palabras
```

```
Texto? Si,a veces,pones las cosas sin espacios,lo cuenta como una palabra.Lo mis
mo pasa si no pones espacios en,por ejemplo,los puntos.
25 Palabras
```

```
t = input("Texto? ")

#Lista de las palabras entre espacios
pe = t.split(" ")

np = len(pe)

...

print(np,"Palabras")
```

```
t = input("Texto? ")

#Lista de las palabras entre espacios
pe = t.split(" ")

np = len(pe)

#Si hay puntuacion, incrementar palabras
for p in pe:
    num = 0
    for i in range(0,len(p)):
        #Todos los signos que no son el ultimo caracter
        #esconden una palabra extra
        if p[i] in ".,:;" and i != len(p)-1:
            num += 1
    np += num

print(np,"Palabras")
```

- Magister y Doctorado Ingeniería UC
  - https://www.ing.uc.cl/programas-de-estudio/postgrado/
- Becas de Doctorado en Chile ANID
  - https://www.conicyt.cl/becasconicyt/category/fichas-concursos/becas-conicyt-para-estudios-en-chile/becas-de-postgrado/becas-de-doctorado/
- Becas de Doctorado en el Extranjero ANID
  - https://www.conicyt.cl/becasconicyt/category/fichas-concursos/becas-en-el-extranjero/becas-de-postgrado-becas-chile/becas-de-doctorado-convocatoria-2012/

MAX (RECURSIVO)

*Jorge, me encanta lo de las **funciones recursivas**..*
¿Crees que podrías hacer un ejemplo de función recursiva con listas?

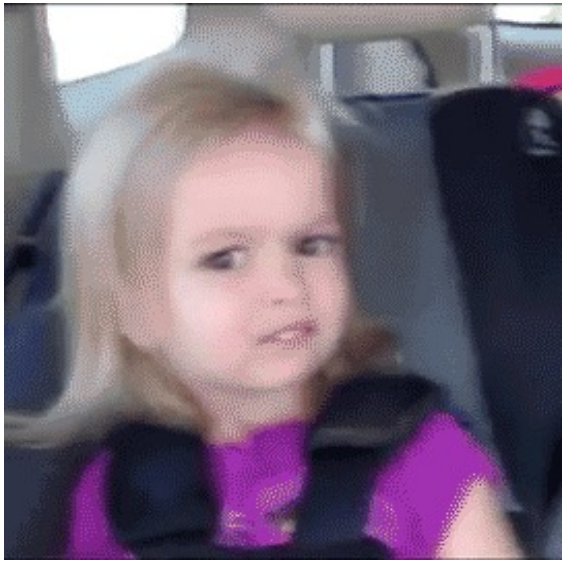
SÍ CLARO

```
a = [8,23,-1,7,30,6]
maxi = max_rec(a)
print(maxi)
```

30

```
a = [8,23,-1,7,30,6]
maxi = max_rec(a)
print(maxi)
```

```
30
```



… sin iteraciones ☺

1. Entiendo que tarea tengo que hacer **yo**

2. Pienso en el caso más pequeño, más trivial, más cercano al final => **Caso Base**

3. Me creo que conozco a alguien que hace exactamente **la misma tarea** que yo, pero **más simple**

4. Llama a esa persona para que haga esa tarea pero un poco más simple. **NO te preocupes de cómo lo va a hacer**. No es tu problema

5. Usa el resultado que te ha devuelto para hacer la tarea que te han pedido


I'LL WAIT
FOR YOU HERE

1. Entiendo que tarea tengo que hacer **yo**

2. Pienso en el caso más pequeño, más trivial, más cercano al final => **Caso Base**

3. Me creo que conozco a alguien que hace exactamente **la misma tarea** que yo, pero **más simple**

4. Llama a esa persona para que haga esa tarea pero un poco más simple. **NO te preocupes de cómo lo va a hacer**. No es tu problema

5. Usa el resultado que te ha devuelto para hacer la tarea que te han pedido

1. Entiendo que tarea tengo que hacer **yo**

   Retornar el max int de una lista. E.g., 30 de [8,23,-1,7,30,6]

2. Pienso en el caso más pequeño, más trivial, más cercano al final => **Caso Base**

   Lista de un número=> el numero es el máximo ☺

3. Me creo que conozco a alguien que hace exactamente **la misma tarea** que yo, pero **más simple**

   Alguien que sabe retornar el int max de una lista, pero solo con listas más cortass (si mi lista tiene 6 elementos, el sabe hacerlo con 5 o 4 …)

4. Llama a esa persona para que haga esa tarea pero un poco más simple. **NO te preocupes de cómo lo va a hacer**. No es tu problema

   Si mi lista es [8,23,-1,7,30,6] (6 elementos) … ¡Eh tu, dame el inverso de [23,-1,7,30,6] (5 elementos) ! ¡Buscate la vida!

5. Usa el resultado que te ha devuelto para hacer la tarea que te han pedido

   Si tengo el max de [23,-1,7,30,6] (que le he pedido a mi colega), para saber cual es el máximo de lo que me han pedido a mi ([8,23,-1,7,30,6]) es simplemente mirar si el 8 es mayor que lo que me ha dado mi colega


```
def max_rec(q):

    #CASO BASE
                                ...

    #CASO RECURSIVO
                                ...
```

```
def max_rec(q):

    #CASO BASE
    if len(q) == 1:      #Un num
        return q[0]      #El max es el num

    #CASO RECURSIVO
```

...

```
def max_rec(q):

    #CASO BASE
    if len(q) == 1:      #Un num
        return q[0]      #El max es el num

    #CASO RECURSIVO
    else:
        h = q[0]         #Primer num
        b = q[1:]        #Resto de nums

        mb = max_rec(b) #Et tu, dame el max del resto

        if h > mb:
            res = h
        else:
            res = mb
        return res
```