

Creado por:

Isabel Maniega

In [23]: `import pandas as pd`

## Ejercicio 1

Cada número es la suma de los 2 anteriores:

0-1-1-2-3-5-8-13-21-34...

Se pide programar esa secuencia con Python.

Nota:

Apendiza elementos hasta tener 10 primeros resultados.

(los 10 números indicados desde 0 hasta 34)

**Si sabes, hazlo de varias formas diferentes**

In [1]: `# OPCIÓN 1`

```
In [2]: import numpy as np

# 0 - 1 - 1 - 2 - 3 - 5 - 8 - 13 - 21 - 34

listado_fibo = []

p,q = 0,1    # p = 0, q = 1

listado_fibo.append(p)
listado_fibo.append(q)

print("cuantos numeros quieres imprimir? - 10 por ejemplo")
print("\n")

while(len(listado_fibo)<10):
    p = p+q
    listado_fibo.append(p)
    q = p+q
    listado_fibo.append(q)

print(listado_fibo)
```

cuantos numeros quieres imprimir? - 10 por ejemplo

[0, 1, 1, 2, 3, 5, 8, 13, 21, 34]

In [3]: `# OPCIÓN 2`

```
In [4]: L = [0, 1]
while (len(L) < 10):
    L.append(L[-1] + L[-2])

L
```

```
Out[4]: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
```

```
In [5]: # MÁS POSIBLES SOLUCIONES...
```

```
In [6]: # la siguiente solución se muestra para ver un posible caso, donde se nos
# 1000-5000 por ejemplo
```

```
In [7]: L = [0, 1]
while (len(L) < 20):
    L.append(L[-1] + L[-2])

print(np.array(L))
```

```
[  0   1   1   2   3   5   8  13  21  34  55  89 144 233
 377 610 987 1597 2584 4181]
```

## Ejercicio 2

Cada número es la suma de los 2 anteriores:

0-1-1-2-3-5-8-13-21-34...

Se pide programar para los números de fibonacci mayores de 1000

```
In [8]: # Primero muestra los valores de 0 hasta 10000000, crea una lista
# con ese listado crea una segunda lista con los mayores de 1000
```

```
In [9]: # OPCIÓN 1
```

```
In [10]: p,q = 0,1

fibonacci = []
fibonacci.append(0)
fibonacci.append(1)

while p<10000000:
    p = p + q
    fibonacci.append(p)
    q = p + q
    fibonacci.append(q)
```

```
In [11]: print(fibonacci)
```

```
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2
584, 4181, 6765, 10946, 17711, 28657, 46368, 75025, 121393, 196418, 31781
1, 514229, 832040, 1346269, 2178309, 3524578]
```

```
In [12]: fibo_mayor_1000 = []

for numero in fibo:
    if (numero>1000) and (len(fibo_mayor_1000)<10):
        fibo_mayor_1000.append(numero)

fibo_mayor_1000
```

```
Out[12]: [1597, 2584, 4181, 6765, 10946, 17711, 28657, 46368, 75025, 121393]
```

```
In [13]: # son estos 10:
# 1597, 2584, 4181, 6765, 10946, 17711, 28657, 46368, 75025, 121393
```

```
In [14]: # OPCIÓN 2
```

```
In [15]: p, q = 0,1
sum_fibo=0
list_values = []

# aquí me doy cuenta que en 150.000 hacen justo la longitud de 10
# si son menos de 1000 los calcula
# si son mayores los appendiza a esta lista Fibonacci
while p < 150000:
    if p < 1000:
        p,q = q,p+q
    else:
        if p < 2000:
            list_values.append(p)
        p,q = q, p+q
        #print(p)
        list_values.append(p)
list_values[:10]
```

```
Out[15]: [1597, 2584, 4181, 6765, 10946, 17711, 28657, 46368, 75025, 121393]
```

```
In [16]: # OPCIÓN 3
```

```
In [17]: # Obtengo los primeros valores y se observa que apartir de los 2 últimos
fibo = [0,1]
while fibo[-1]<1000 and (fibo[-1]+fibo[-2]<1000):
    fibo.append(fibo[-1]+fibo[-2])

np.array(fibo)
```

```
Out[17]: array([ 0,  1,  1,  2,  3,  5,  8, 13, 21, 34, 55, 89, 144,
                233, 377, 610, 987])
```

```
In [18]: fibo2 = []
fibo2.append(fibo[-1]+fibo[-2])
fibo2
```

```
Out[18]: [1597]
```

```
In [19]: fibo2.append(fibo2[-1]+fibo[-1])
fibo2
```

```
Out[19]: [1597, 2584]
```

```
In [20]: print(len(fibo2))
```

2

```
In [21]: while len(fibo2)<10:
          fibo2.append(fibo2[-1]+fibo2[-2])

          fibo2
```

```
Out[21]: [1597, 2584, 4181, 6765, 10946, 17711, 28657, 46368, 75025, 121393]
```

## Ejercicio 3

Se pide crear un NUEVO dataframe para cada uno de los siguientes casos

planteados y que están relacionados con el Titanic DataSet

(antes debéis de cargar el archivo como df)

**1) Leer el archivo train.csv del titanic dataset**

```
In [24]: df = pd.read_csv("train.csv")
          df.head()
```

```
Out[24]:
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450

**2) Crear un dataframe de nombre `df_sobreviven` refiriéndose a un dataframe en el que todos los pasajeros sobreviven**

NOTA: si al principio no estás seguro del resultado, puedes usar `value_counts()` para comprobar tu resultado

```
In [25]: df_sobreviven = df[df["Survived"]==1]
df_sobreviven.head()
```

```
Out[25]:
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736

```
In [26]: len(df_sobreviven)
```

```
Out[26]: 342
```

```
In [27]: df_sobreviven.Survived.value_counts()
```

```
Out[27]: 1    342
Name: Survived, dtype: int64
```

**3) Crear un dataframe de nombre `df_no_sobreviven` refiriéndose a un dataframe en el que NINGUNO de los pasajeros sobrevive**

```
In [28]: df_no_sobreviven = df[df["Survived"]==0]
df_no_sobreviven.head()
```

Out[28]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.25
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.05
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.45
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.86
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.07

In [29]: `len(df_no_sobreviven)`

Out[29]: 549

In [30]: `df_no_sobreviven.Survived.value_counts()`

Out[30]: 0 549  
Name: Survived, dtype: int64

#### 4) DataFrame de hombres que no sobrevivieron en el títanic

In [31]: `df_hombres_no = df[(df["Sex"]=="male") & (df["Survived"]==0)]`  
`df_hombres_no.head()`

Out[31]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.25
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.05
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.45
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.86
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.07

In [32]: `len(df_hombres_no)`

Out[32]: 468

**5) DataFrame de hombres que si sobrevivieron en el titanic**

```
In [33]: df_hombres_si = df[(df["Sex"]=="male") & (df["Survived"]==1)]
df_hombres_si.head()
```

Out[33]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	
17	18	1	2	Williams, Mr. Charles Eugene	male	NaN	0	0	244373	13.0
21	22	1	2	Beesley, Mr. Lawrence	male	34.0	0	0	248698	13.0
23	24	1	1	Sloper, Mr. William Thompson	male	28.0	0	0	113788	35.0
36	37	1	3	Mamee, Mr. Hanna	male	NaN	0	0	2677	7.0
55	56	1	1	Woolner, Mr. Hugh	male	NaN	0	0	19947	35.0

**6) DataFrame de mujeres que no sobrevivieron en el titanic**

```
In [34]: df_mujeres_no = df[(df["Sex"]=="female") & (df["Survived"]==0)]
df_mujeres_no.head()
```

Out[34]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	
14	15	0	3	Vestrom, Miss. Hulda Amanda Adolfina	female	14.0	0	0	350406	7
18	19	0	3	Vander Planke, Mrs. Julius (Emelia Maria Vande...	female	31.0	1	0	345763	18
24	25	0	3	Palsson, Miss. Torborg Danira	female	8.0	3	1	349909	21
38	39	0	3	Vander Planke, Miss. Augusta Maria	female	18.0	2	0	345764	18
40	41	0	3	Ahlin, Mrs. Johan (Johanna Persdotter Larsson)	female	40.0	1	0	7546	9

In [35]: `len(df_mujeres_no)`

Out[35]: 81

7) DataFrame de mujeres que si sobrevivieron en el títanic

In [36]: `df_mujeres_si = df[(df["Sex"]=="female") & (df["Survived"]==1)]  
df_mujeres_si.head()`



Out[36]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736

In [37]: `len(df_mujeres_si)`

Out[37]: 233

In [38]: `df_mujeres_si.Survived.value_counts()`

Out[38]: 1     233  
Name: Survived, dtype: int64

In [39]: `df_mujeres_si.Sex.value_counts()`

Out[39]: female     233  
Name: Sex, dtype: int64

*Creado por:*

*Isabel Maniega*