



FrancoOlivera /  
Mi-repo



 Código  Asuntos  Solicitudes de extracción  Comportamiento  Proyectos  wiki  Seguridad  Perspectivas 


mi-repo / TP1\_Comisión\_4\_para\_entregar.ipynb 

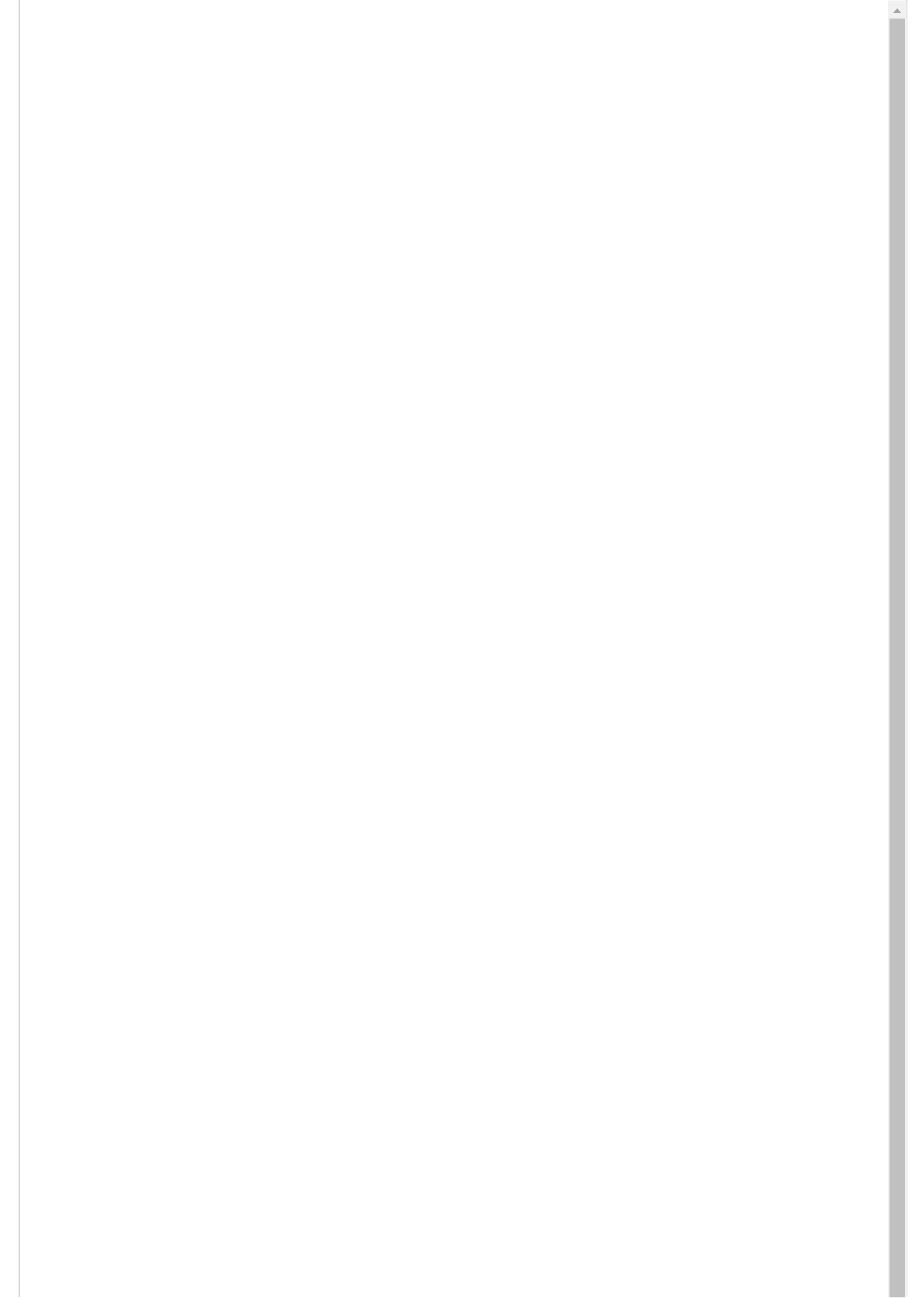


 FrancoOlivera ahora



2.43 MB

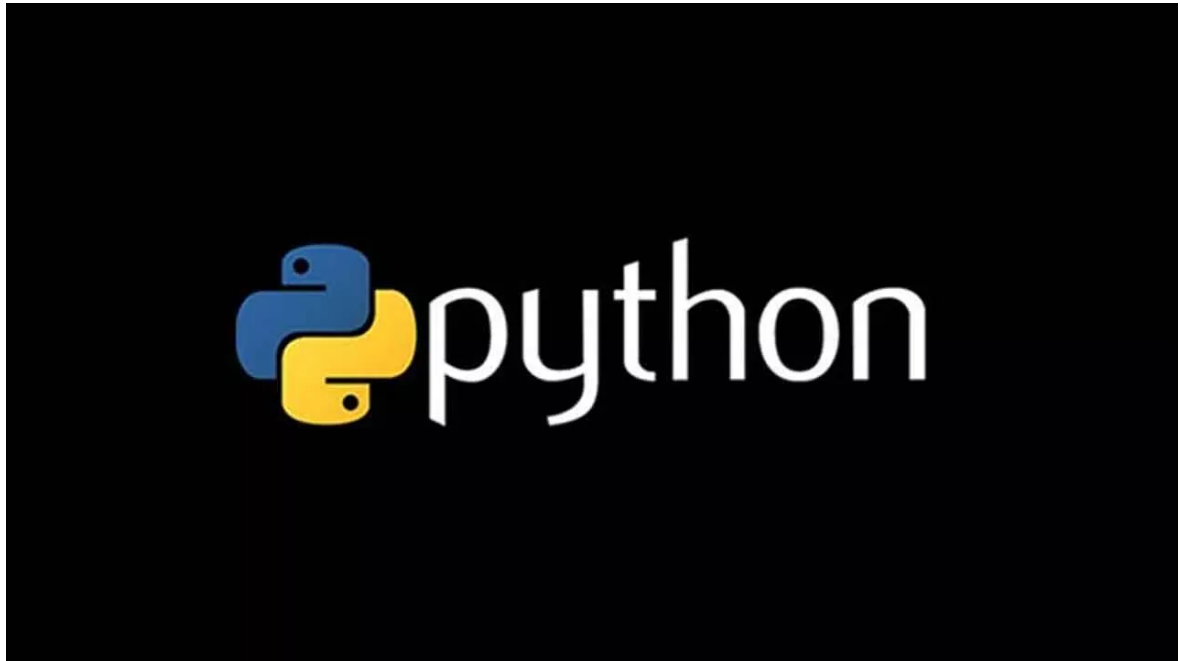




[Open in Colab](#)

## Trabajo Practico 1 Parte A



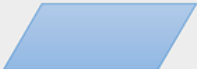



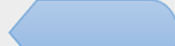
## Introducción al Desarrollo en Python


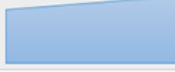




## Indice

[Ejercicio 1](#) - Operadores[Ejercicio 2](#) - Estructura de control condicional[Ejercicio 3](#) - Ciclo for[Ejercicio 4](#) - Rango[Ejercicio 5](#) - Ciclo while y funciones

## Símbolos en un Diagrama de Flujo

Nombre	Función	Símbolo
<b>Terminal</b>	Determina el inicio o final de un programa.	
<b>Proceso, acción</b>	Representa cualquier operación que se le aplique a la información guardada. Esta operación debe cambiar la posición, valor o forma de la información a la que se le aplica. Comúnmente son operaciones matemáticas.	
<b>Entrada / Salida</b>	Nos indica cualquier tipo de dato que ingrese o salga del sistema.	
<b>Decisión</b>	Representa una instrucción de comparación lógica entre dos o más valores. Si el valor de la comparación es verdadero, el programa toma una dirección, si es falsa, toma otra.	
<b>Línea de flujo</b>	Estas indican la dirección y el orden en el que se ejecuta el algoritmo.	
<b>Impresora</b>	Indica la salida de un dato o expresión, normalmente en formato visual.	
<b>Pantalla</b>	Indica la salida de un dato, normalmente información que se	

	muestra en formato visual.	
<b>Entrada</b>	Símbolo alternativo para simbolizar la entrada de datos.	
<b>Teclado, entrada manual</b>	Se utiliza para representar la entrada manual de datos por parte del usuario.	
<b>Conector misma página</b>	Se utiliza para conectar o indicar la continuación de un algoritmo en una misma página.	
<b>Conector diferente página</b>	Se utiliza para conectar o indicar la continuación de un algoritmo en una página distinta.	

Fuente: [https://solucioningenieril.com/imagenes/ asignaturas/programacion\\_en\\_c/tema\\_4/1.png](https://solucioningenieril.com/imagenes/ asignaturas/programacion_en_c/tema_4/1.png)



## EJERCICIO 1 ESTRUCTURA DE CONTROL CONDICIONAL

[Ir al índice](#)

Determinar si un numero ingresado por el usuario es PAR

### Solución 1.1

Usando el operador módulo (%) que devuelve el resto de la división de dos números:

- Si el resto es 0, el número es par.
- Si el resto es 1, el número es impar.

```
In [2]: #Por ejemplo
        resto=5%2
        print(resto)
```

1

```
In [3]: # Porque al dividir 5 entre 2, el resto es 1.
```

```
In [4]: # Paso 1: Pido un número al usuario y lo convierto a número entero
        numero=input("Ingrese un número: ")
        print(f"Lo que el usuario ingresa siempre es una cadena: {numero}")
        # numero_entero=int(numero)
```

```
# print(f"Convertido a número entero: {numero_entero}")
```

Ingrese un número: 4

Lo que el usuario ingresa siempre es una cadena: 4

```
In [5]: # Paso 2: Verifico si lo que el usuario ingresa es un número entero

cadena='a'
if cadena.isnumeric():
    print(f'El usuario ingresó un número: {cadena}')
else:
    print(f'{cadena} no es un número')
```

a no es un número

```
In [6]: """
cadena=1
if cadena.isnumeric():
    print(f'El usuario ingresó un número: {cadena}')
else:
    print(f'{cadena} no es un número')

"""
# AttributeError: 'int' object has no attribute 'isnumeric'

# Como cadena es un objeto 'int' no tiene un atributo 'isnumeric'
```

```
Out[6]: "\ncadena=1\nif cadena.isnumeric():\n    print(f'El usuario ingresó un número: {cadena}')\nelse:\n    print(f'{cadena} no es un número')\n\n"
```

Usa la función **str()** para convertir la cadena a un objeto str antes de usar el método **isnumeric()**

De esta forma, nos aseguramos que el método **isnumeric()** se aplica sobre una cadena de texto, y no sobre un número. Así evitamos el error anterior.

```
In [7]: cadena=1
if str(cadena).isnumeric():
    print(f'El usuario ingresó un número entero: {cadena}')
else:
    print(f'{cadena} no es un número entero')
```

El usuario ingresó un número entero: 1

```
In [8]: cadena=1.5
if str(cadena).isnumeric():
    print(f'El usuario ingresó un número entero: {cadena}')
else:
    print(f'{cadena} no es un número entero')
```

1.5 no es un número entero

Usa la función **isdecimal()**:

- Devuelve **True** si la cadena contiene solo dígitos (caracteres del 0 al 9)
- Devuelve **False** en caso contrario.

```
In [9]: cadena=1.5

print('Es un número entero? ',str(cadena).isnumeric())
```

Es un número entero? False

```
In [10]: cadena=1.5

print('Es un número decimal?', str(cadena).isdecimal())
```

Es un número decimal? False

```
In [10]:
```

Usa la función **float()** para intentar convertir la cadena a un número de punto flotante.

- Si la conversión tiene éxito, significa que la cadena es un número decimal válido.
- Si la conversión falla, significa que la cadena no es un número decimal válido.

```
In [11]: cadena = "1.5"
try:
```

```

numero = float(cadena)
print(f"El usuario ingresó un número decimal: {cadena}")
except:
    print(f"{cadena} no es un número decimal")

# try significa intentar ejecutar algo
# except se ejecuta en caso de fallar el intento

```

El usuario ingresó un número decimal: 1.5

Usa la función **int()** convierte un valor especificado en un número entero.

- Devuelve un objeto entero construido a partir de un número o una cadena x
- Devuelve 0 si no se dan argumentos.

```

In [12]: # Convertir un número de coma flotante en un entero
print(int(3.14))

```

3

```

In [13]: # Convertir una cadena en un entero
print(int("42"))

```

42

```

In [14]: # Convertir una cadena hexadecimal en un entero
print(int("0x2a", 16))

```

42

```

In [15]: # Devolver 0 si no se dan argumentos
print(int())

```

0

```

In [31]: # Convertir una cadena no válida en un entero
# print(int("abc"))

# ValueError: invalid literal for int() with base 10: 'abc'
# Dice que 'abc' no es un número entero

```

```

In [16]: cadena = "abc"
try:
    numero = int(cadena)
    print(f"El usuario ingresó: {cadena}")
    print(f"Convertido a número entero es: {numero}")
except:
    print(f"{cadena} no es un número entero")

# try significa intentar ejecutar algo
# except se ejecuta en caso de fallar el intento

```

abc no es un número entero

# 1 - GANÁ MILLAS PARA EL DESTINO PYTHON

¿Cómo vamos a validar si lo que ingresa el usuario por teclado es un número?

Responde "A es mejor" o "B es mejor": "A es mejor"

Tus comentarios (opcional): "A es mejor" ya que trata de ejecutar y si falla por no tratarse de un numero entero, imprimirá que no lo es sin dar lugar a errores del sistema.

```

In [ ]: """
OPCION A

cadena=input("Ingrese un número entero:")
try:
    numero = int(cadena)
    print(f"El usuario ingresó: {cadena}")
    print(f"Convertido a número entero es: {numero}")
except:
    print(f"{cadena} no es un número entero")
"""

```

```

In [18]: """
OPCION B

```

```
numero=int(input("Ingrese un número entero:"))
"""
```

Out[18]: '\nOPCION B\n\nnumero=int(input("Ingrese un número entero:"))\n\n'

### Solución 1.2

Determinamos si un número dado es par

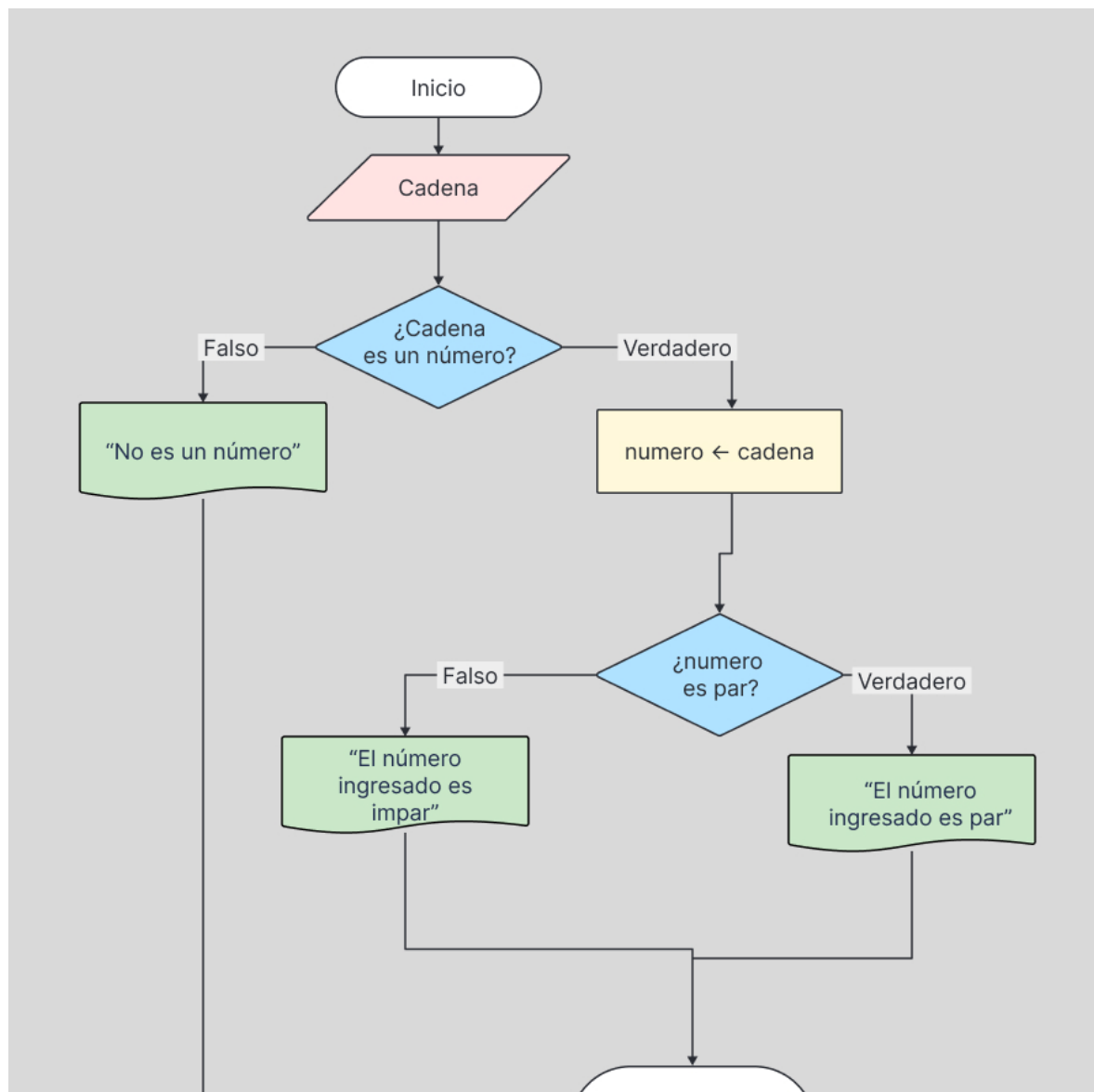
```
In [19]: # Pedimos un número al usuario
cadena=input("Ingrese un número entero: ")

try:
    numero = int(cadena)
    print(f"Lo que ingresa por teclado: {cadena}")
    print(f"Convertido a número entero es: {numero}")

    # Vemos si el número es par o impar usando el operador %
    if numero % 2 == 0:
        print("El número", numero, "es par.")
    else:
        print("El número", numero, "es impar.")

except:
    print(f"{cadena} no es un número entero")
```

Ingrese un número entero: 2  
 Lo que ingresa por teclado: 2  
 Convertido a número entero es: 2  
 El número 2 es par.







Fin



## EJERCICIO 2

[Ir al índice](#)

Determinar si un numero ingresado por el usuario esta DENTRO DEL RANGO [1.10] y es par. Si el número no está dentro de ese rango.



Determinar si un número ingresado por el usuario está dentro del rango [1,10] y si por el contrario no está dentro de ese rango, se muestra el mensaje 'NÚMERO FUERA DEL RANGO'.

In [20]:

```
# El rango

rango=range(1,11)

print(rango)
```

range(1, 11)

In [21]:

```
# Veamos los números individuales del rango

for numero in rango:
    print(numero)
```

```
1
2
3
4
5
6
7
8
9
10
```

In [22]:

```
# Convertimos el rango en una lista

lista=list(rango)

print(lista)
```

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

In [23]:

```
# Veamos si un número está en el rango del 1 al 10
# Y además, si el número es par.

numero=5

if numero in range(1, 11) and numero % 2 == 0:
    print("El número", numero, "está dentro del rango [1, 10] y es par.")
else:
    print("Número fuera del rango o es número impar.")
```

Número fuera del rango o es número impar.

In [24]:

```
# Veamos si un número ni está en el rango ni es impar
# Nos preguntamos si es impar, mostramos "es impar".

numero = 5

if numero>=1 and numero<=10 and (numero)%2==0:
    print("El número está dentro del rango y es par.")
else:
    if (numero)%2==1:
        if numero>=1 and numero<=10:
            print("Número dentro del rango pero impar")
        else:
            print("Número fuera del rango y es impar")
    else:
        print("El numero es par pero está fuera del rango")
```

Número dentro del rango pero impar

## 2 - GANÁ MILLAS PARA TU DESTINO PYTHON

¿Es cierto que ambos algoritmos hacen lo mismo?

Responde "VERDADERO, ambas opciones hacen lo mismo" o "FALSO, no hacen lo mismo": "VERDADERO, ambas opciones hacen lo mismo"

Tus comentarios (opcional): Son algoritmos que ingresan la misma información y generan mismas decisiones. Elif genera evaluar múltiples condiciones a la vez.

In [25]:

```
"""
OPCION A

cadena=input("Ingrese un número entero:")
try:
    numero=int(cadena)
```

```

print(f"El usuario ingresó: {cadena}")
print(f"Convertido a número entero es: {numero}")

if numero>=1 and numero<=10 and (numero)%2==0:
    print("El número está dentro del rango y es par.")
else:
    if (numero)%2==1:
        if numero>=1 and numero<=10:
            print("Número dentro del rango pero impar")
        else:
            print("Número fuera del rango y es impar")
    else:
        print("El numero es par pero está fuera del rango")
except:
    print(f"{cadena} no es un número entero")

"""

```

```

Out[25]: '\nOPCION A\nncadena=input("Ingrese un número entero:")\ntry:\n numero=int(cadena)\n print(f"El usuario ingresó: {cad
ena}")\n print(f"Convertido a número entero es: {numero}")\n\n if numero>=1 and numero<=10 and (numero)%2==0:\n pri
nt("El número está dentro del rango y es par.")\n else:\n     if (numero)%2==1:\n         if numero>=1 and numero<=10:\n
print("Número dentro del rango pero impar")\n         else:\n             print("Número fuera del rango y es impar")\n     els
e:\n         print("El numero es par pero está fuera del rango")\nexcept:\n print(f"{cadena} no es un número entero")\n\n'

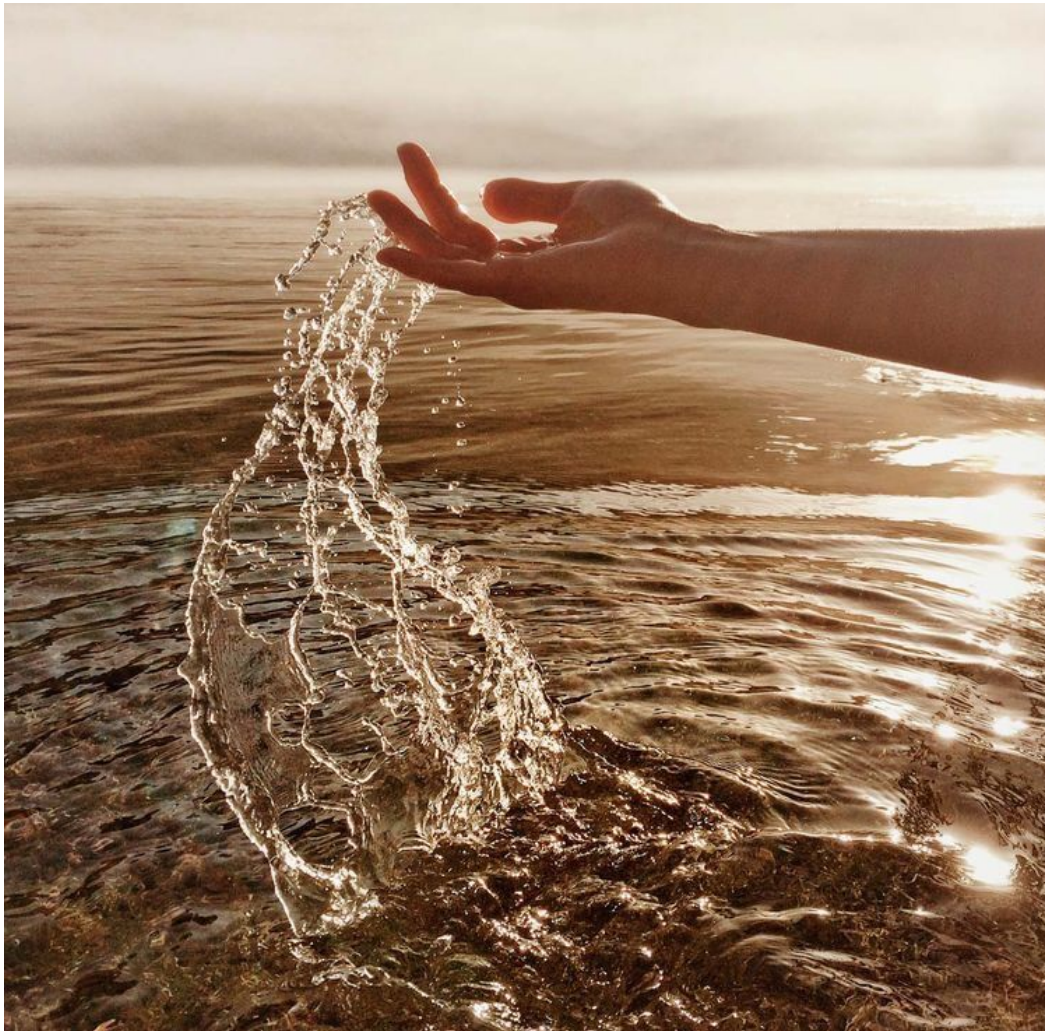
```

```

In [ ]: """
OPCION B

cadena=input("Ingrese un número entero:")
try:
    numero=int(cadena)
    print(f"El usuario ingresó: {cadena}")
    print(f"Convertido a número entero es: {numero}")
    if numero in range(1, 11) and numero % 2 == 0:
        print("El número está dentro del rango y es par.")
    elif numero in range(1, 11) and numero % 2 == 1:
        print("Número dentro del rango pero impar")
    elif numero % 2 == 0:
        print("El número es par pero está fuera del rango")
    else:
        print("Número fuera del rango y es impar")
except:
    print(f"{cadena} no es un número entero")
"""

```



**EJERCICIO 3 CICLO FOR**[Ir al índice](#)

## 3 - Ganá millas para tu destino Python

Este algoritmo solicita al usuario que ingrese 10 números enteros y calcula la suma de todos los números enteros ingresados.

Responde cuál es la opción correcta: Es la C

Tus comentarios (opcional): Es la C porque es la unica opcion que permite ingresar 10 numeros y suma a todos ellos. La A solo permitía 9 numeros. La B no los sumaba a todos.

Ganás millas opcionales extra, si te animás a hacer el diagrama de flujo, con la opción que consideras correcta.

```
In [ ]: """
OPCION A

suma=0
for i in range(1,10):
    cadena = input(f"{i} - Ingresá un número entero: ")
    try:
        numero = int(cadena)
        print(f"El usuario ingresó: {cadena}")
        print(f"Convertido a número entero es: {numero}")
        suma+=numero
    except:
        print(f"{cadena} no es un número entero")
print(f"La suma de todos los números ingresados es :{suma}")

"""
```

```
In [ ]: """
OPCION B

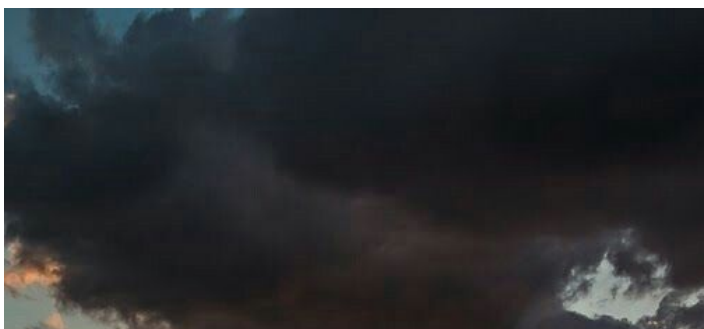
for i in range(10):
    suma=0
    cadena = input(f"{i} - Ingresá un número entero: ")
    try:
        numero = int(cadena)
        print(f"El usuario ingresó: {cadena}")
        print(f"Convertido a número entero es: {numero}")
        suma+=numero
    except:
        print(f"{cadena} no es un número entero")
print(f"La suma total es: {suma}")

"""
```

```
In [ ]: """
OPCION C

suma=0
for i in range(10):
    cadena = input(f"{i} - Ingresá un número entero: ")
    try:
        numero = int(cadena)
        print(f"El usuario ingresó: {cadena}")
        print(f"Convertido a número entero es: {numero}")
        suma+=numero
    except:
        print(f"{cadena} no es un número entero")
print(f"La suma de todos los números ingresados es :{suma}")

"""
```





#### EJERCICIOS 4

[Ir al índice](#)

```
In [87]: # Paso 1: Mostrar todos Los números enteros en el rango del 56 al 1230.  
  
for numero in range(56, 1231):  
    print(numero)
```

56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
...

93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185



186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277

278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369

370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462

462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554

555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646



647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738

739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831

831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923

924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015

1016  
1017  
1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107



1108  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137  
1138  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1190  
1191  
1192  
1193  
1194  
1195  
1196  
1197  
1198  
1199  
1200

```
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
```

```
In [63]: # Paso 2: Pedir al usuario que ingrese el valor a.

a=int(input("Ingrese el valor a:"))
```

Ingrese el valor a:45

```
In [65]: # Paso 3: Pedir al usuario que ingrese el valor b.

b=int(input("Ingrese el valor b:"))
```

Ingrese el valor b:60

```
In [88]: # Paso 4: Mostrar todos Los números enteros en el rango desde a hasta b.
a=int(input("Ingrese el valor a:"))
b=int(input("Ingrese el valor b:"))
for numero in range(a,b+1):
    print(numero)
```

Ingrese el valor a:12

Ingrese el valor b:67

```
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
```

48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67

## 4 - Ganá millas para tu destino Python

Completá el código para el paso 2, paso 3 y paso 4.

Tus comentarios (opcional):



EJERCICIO 5 CICLO WHILE

[Ir al índice](#)

Solicitar al usuario que ingrese un número, si es negativo o nulo, solicitar reiteradamente hasta que ingrese un número positivo.

## 5 - Ganá millas para tu destino Python

Mirá este desarrollo paso a paso, y completá en cada parte del código donde dice completar, reemplazá la palabra "pass" por lo que consideres adecuado.

Tus comentarios (opcional):

```
In [72]: # Paso 1: Dado un número entero, veamos si es positivo, en caso contrario, es negativo o cero.

numero=2
if numero > 0:
    print(f'El número {numero} es positivo.')
else:
    print(f'El número {numero} es negativo o cero.')
```

El número 2 es positivo.

```
In [80]: # Paso 2: Definimos una función positivo() que recibe un número entero
#         Devuelve 1 si la entrada es positivo.
#         Devuelve 0 si la entrada es un número negativo o cero.

def positivo(numero_entrada):
    if numero_entrada > 0:
        salida=1
    else:
        salida=0
    return salida

print(positivo(-5))
```

0

```
In [166... # Paso 3: Modificamos la función anterior para que en vez de un número,
#         recibe una cadena como parámetro de entrada.
#         Intentamos ver si esa cadena es un número, antes de ver si es positivo.
#         Si el número es positivo, la función devuelve 1.
#         Si no es un número, o bien, es negativo o es cero,
#         La función devuelve 0.

def positivo(cadena):
    try:
        numero_entrada=int(cadena) # Completar aquí con una función que convierta cadena a número entero.
        if numero_entrada > 0:
            salida=1
        else:
            salida=0
    except:
        salida=0
    return salida

print(positivo("a"))
```

0

```
In [181... # Paso 4: Solicitamos al usuario que ingrese un número entero
#         Llamamos a la función positivo para que nos devuelva
#         1 si es un número entero positivo
#         0 si es negativo o cero o no es un número.
#         Recuerda que la función que quedó definida en el paso anterior,
#         No hace falta volver a definirla, la podemos usar en este paso.

algo=input("Ingrese un número entero: ")

espositivo=positivo(algo) # Completar aquí con la llamada a la función

if espositivo:
    print(f"El usuario ingresó un número positivo: {algo}")
else:
    print(f"El usuario ingresó algo que no es un número positivo.")
```

Ingrese un número entero: 3

El usuario ingresó un número positivo: 3

In [242]

In [183]...

# Paso 5: Veamos cómo funciona el ciclo while

```

numero=3

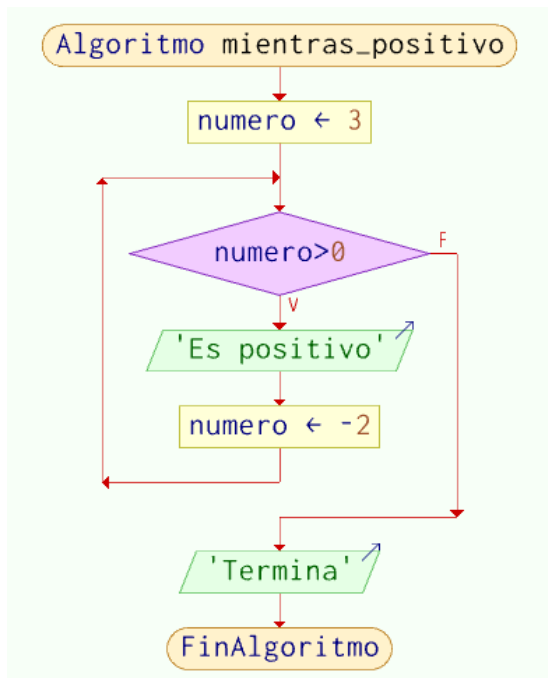
while numero > 0:
    print("es positivo")
    numero=-2

print("El ciclo while termina porque el segundo número es negativo.")

```

es positivo

El ciclo while termina porque el segundo número es negativo.



In [183]...

# Paso 6: Pasamos en limpio lo que desarrollamos hasta ahora

# Definimos la función positivo

```

def positivo(cadena):
    try:
        numero_entrada=int(cadena) # Completar aquí con una función que convierta cadena a número entero.
        if numero_entrada > 0:
            salida=1
        else:
            salida=0
    except:
        salida=0
    return salida

# Llamamos la función positivo

algo=input("Ingrese un número entero: ")

```