

Lección 2

datatypes01.py

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + v [icon] [icon] ... ^ x
PS C:\Users\aline\Desktop\repositorio_de_GIT\pcd3am1_2024\parcial1> & C:/Users/aline/AppData/Local/Programs/Python/Python311/python.exe c:/Users/aline/Desktop/repositorio_de_GIT/pcd3am1_2024/parcial1/leccion02/datatypes01.py
Esta es una cadena: Hola Mundo!
Primer caracter de la cadena: H
Ultimo caracter de la cadena: !
PS C:\Users\aline\Desktop\repositorio_de_GIT\pcd3am1_2024\parcial1>
```

Datatypes02.py

```
PS C:\Users\aline\Desktop\repositorio_de_GIT\pcd3am1_2024\parcial1> & C:/Users/aline/AppData/Local/Programs/Python/Python311/python.exe c:/Users/aline/Desktop/repositorio_de_GIT/pcd3am1_2024/parcial1/leccion02/datatypes02.py
Esta es una cadena: Hola Mundo!
Posicion 0: H
Posicion 1: o
Posicion 2: l
Posicion 3: a
Posicion -1: !
Posicion -2: o
```

Datatypes03.py

```
Esta es una cadena: Hola Mundo!
Silice: Hola
Silice: Mundo
Silice: Mundo!
Silice: Hola
Silice: !
Silice: a Mund
PS C:\Users\aline\Desktop\repositorio_de_GIT\pcd3am1_2024\parcial1>
```

Datatypes04.py

```
PS C:\Users\aline\Desktop\repositorio_de_GIT\pcd3am1_2024\parcial1> & C:/Users/aline/AppData/Local/Programs/Python/Python311/python.exe c:/Users/aline/Desktop/repositorio_de_GIT/pcd3am1_2024/parcial1/leccion02/datatypes04.py
Esta es una cadena: Hola Mundo!
Invertir: !odnuM aloH
PS C:\Users\aline\Desktop\repositorio_de_GIT\pcd3am1_2024\parcial1>
```

Datatypes05.py

```
line/Desktop/repositorio_de_GIT/pcd3am1_2024/parcial1/leccion02/datatypes05.py
Esta es una cadena: Hola Mundo!
Posicion pares: Hl ud!
```

Datatypes06.py

```
line/Desktop/repositorio_de_GIT/pcd3am1_2024/parcial1/leccion02/datatypes06.py
Esta es una cadena: Hola Mundo!
Puedes adivinar que esta imprimiendo?: !n o
PS C:\Users\aline\Desktop\repositorio_de_GIT\pcd3am1_2024\parcial1>
```

Datatypes07.py

```
line/Desktop/repositorio_de_GIT/pcd3am1_2024/parcial1/leccion02/datatypes07.py
Esta es una cadena: Hola Mundo!
Invertir 2da opcion: !odnuM aloH
PS C:\Users\aline\Desktop\repositorio_de_GIT\pcd3am1_2024\parcial1>
```

Datatypes08.py

```

line/Desktop/repositorio_de_GIT/pcd3am1_2024/parcial1/leccion02/datatypes08.py
Esta es una cadena: Hola Mundo!
Traceback (most recent call last):
  File "c:\Users\aline\Desktop\repositorio_de_GIT\pcd3am1_2024\parcial1\leccion02\datatypes08.py", line 5, in <module>
    cadena[4] = '*'
    ~~~~~^
TypeError: 'str' object does not support item assignment
PS C:\Users\aline\Desktop\repositorio_de_GIT\pcd3am1_2024\parcial1>

```

Datatypes09.py

```

line/Desktop/repositorio_de_GIT/pcd3am1_2024/parcial1/leccion02/datatypes09.py
Esta es una cadena: Hola Mundo!
Se modifiko la cadena?: Hola*Mundo!

```

Datatypes10.py

```

File "c:\Users\aline\Desktop\repositorio_de_GIT\pcd3am1_2024\parcial1\leccion02\datatypes10.py", line 5
del cadena(4)
    ~~~~~^
SyntaxError: cannot delete function call

```

Datatypes11.py

```

Esta es una cadena: Hola Mundo!
Se eliminaron varios caracteres?: Hola!

```

Datatypes12.py

```

Esta es una cadena: Hola Mundo!
Cadena eliminada...
Traceback (most recent call last):
  File "c:\Users\aline\Desktop\repositorio_de_GIT\pcd3am1_2024\parcial1\leccion02\datatypes12.py", line 7, in <module>
    print("Se elimino la cadena?: ", cadena)
    ~~~~~^
NameError: name 'cadena' is not defined

```

Datatypes13.py

```

Esta es una cadena con formato: Hola Mundo !
Esta es una cadena con formato desordenado: Mundo ! Hola
Esta es una cadena con formato: Hola Mundo !

```

Datatypes14.py

```

Binario del 256: 1101
Formato exponencial: 1.234345e+03
Flotante truncado a 4 digitos: 3.1416

```

Datatypes15.py

```

Alineacion de cadenas:
|Izq      |  Cen  |      Der|

>> Mario  << soy yo, y tengo >>>43 << años!

```

Datatypes16.py

```
5
0.6666666666666666
5.666666666666667
7
-1
12
0.75
0
3
```

Datatypes17.py

```
Suma: (3+8j)
Resta: (-1+8j)
Division: (1.307692307692308+0.5384615384615384j)
Multiplicacion: (-13+13j)
```

Datatypes18.py

```
<class 'bool'>
a=True
b=False
34 == 34 : True
23 == 24: False
34 != 34 : False
23 != 24: True
x < y < z
x > y o y < z
Negacion
```

Datatypes19.py

```
Lista Vacía: []
Tamaño Lista Vacía: 0
Lista Numeros: [1, 2, 3, 4, 5, 6]
Tamaño Lista Numeros: 6
Lista Letras: ['a', 'b', 'c']
Tamaño Lista Letras: 3
Lista Objetos: [1, '*', 0, True, 3.141592, False, 'Hola Mundo!', ['a', 'b', 'c']]
Tamaño Lista Objetos: 8
```

Datatypes20.py

```
Matriz completa: [[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]]
Segundo Renglon: [5, 6, 7, 8]
Tercer Renglon, segunda Columna: 10
```

Datatypes21.py

```
Ultima posicion 1: e
Ultima posicion 2: e
Ultima posicion 3: e
Penultima posicion d
```

Datatypes22.py

```
Line/Desktop/repositorio_de_GIT/pcd3ami_2024/parcial11/leccion02/datatypes22.py
Separada por espacios en una lista: ['Hola', 'Mundo!', 'Aqui', 'estoy', '...']
Otro separador: ['1', '2', '3', '4']
```

Datatypes23.py

```
Lista vacia: []
Universidades: ['IPN', 'UNAM', 'TEC', 'IBERO']
Lista con universidades y numeros: ['IPN', 'UNAM', 'TEC', 'IBERO', 2, 3, 4]
Insertando un objeto: ['IPN', 'UNAM', 'ANAHUAC', 'TEC', 'IBERO', 2, 3, 4]
```

Datatypes24.py

```
Lista original: ['IPN', 'UNAM', 'ANAHUAC', 'TEC', 'IBERO', 2, 3, 4]
Lista invertida: [4, 3, 2, 'IBERO', 'TEC', 'ANAHUAC', 'UNAM', 'IPN']
Lista original nuevamente: ['IPN', 'UNAM', 'ANAHUAC', 'TEC', 'IBERO', 2, 3, 4]
Volvemos a invertirla: ['IPN', 'UNAM', 'ANAHUAC', 'TEC', 'IBERO', 2, 3, 4]
```

Datatypes25.py

```
Lista original: [1, 2, 3, 4, 2, 2, 5, 6, 2, 7]
Se elimino un elemento: [3, 4, 2, 2, 5, 2, 7]
Se extrajo el ultimo elemento: 7 de la lista: [3, 4, 2, 2, 5, 2]
```

Datatypes26.py

```
Lista original: [100, 200, 300, 400, 500, 600, 700, 800, 900, 1000]
Slice de 0 a 9: [100, 200, 300, 400, 500, 600, 700, 800, 900, 1000]
Slice de 0 a 2: [100, 200, 300]
Slice de Inicio a 2: [100, 200, 300]
Slice de 3 a 9: [400, 500, 600, 700, 800, 900, 1000]
Slice de 3 al final: [400, 500, 600, 700, 800, 900, 1000]
Slice de 4 a 6: [500, 600, 700]
Slice de Inicio a Fin: [100, 200, 300, 400, 500, 600, 700, 800, 900, 1000]
Slice posiciones pares: [100, 300, 500, 700, 900]
Slice posiciones impares: [200, 400, 600, 800, 1000]
Slice invertido impares: [1000, 800, 600, 400, 200]
Slice invertido pares: [900, 700, 500, 300, 100]
```