



## Lección 02

### Datatypes01

```
datatypes01.py X
parcial1 > leccion02 > datatypes01.py > ...
augustoramirez, last week | 1 author (augustoramirez)
1 # Cadenas de caracteres
2 cadena = "Hola Mundo!"
3 print("Esta es una cadena: ", cadena)
4
5 print("Primer caracter de la cadena: ", cadena[0])
6 print("Ultimo caracter de la cadena: ", cadena[-1])
7

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02> py datatypes01.py
Esta es una cadena: Hola Mundo!
Primer caracter de la cadena: H
Ultimo caracter de la cadena: !
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02>
```

### Datatypes02

```
datatypes02.py X
parcial1 > leccion02 > datatypes02.py > ...
augustoramirez, last week | 1 author (augustoramirez)
1 # Otras posiciones
2 cadena = "Hola Mundo!"
3 print("Esta es una cadena: ", cadena)
4
5 print("Posicion 0: ", cadena[0])
6 print("Posicion 1: ", cadena[1])
7 print("Posicion 2: ", cadena[2])
8 print("Posicion 3: ", cadena[3])
9 print("Posicion -1: ", cadena[-1])
10 print("Posicion -2: ", cadena[-2])
11

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02> py 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
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02>
```



## Datatypes03

```
datatypes03.py X
parcial1 > leccion02 > datatypes03.py > ...
augustoramirez, last week | 1 author (augustoramirez)
1 # Slicing augustoramirez, last week • Primera parte leccion02
2 cadena = "Hola Mundo!"
3 print("Esta es una cadena: ", cadena)
4
5 print("Silice: ", cadena[0:4])
6 print("Silice: ", cadena[5:10])
7 print("Silice: ", cadena[5:])
8 print("Silice: ", cadena[:4])
9 print("Silice: ", cadena[-1:])
10 print("Silice: ", cadena[3:-2])
11
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02> py datatypes03.py
Esta es una cadena: Hola Mundo!
Silice: Hola
Silice: Mundo
Silice: Mundo!
Silice: Hola
Silice: !
Silice: a Mund
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02> █
```

## Datatypes04

```
datatypes04.py M X
parcial1 > leccion02 > datatypes04.py > ...
You, 1 second ago | 2 authors (augustoramirez and one other)
1 # Invertir cadena
2 cadena = "Hola Mundo!"
3 print("Esta es una cadena: ", cadena)
4
5 print("Invertir: ", cadena[::-1])
6 print("Invertir: ", ''.join(reversed(cadena))) # devuelve el iterador (list)
7
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02> py datatypes04.py
Esta es una cadena: Hola Mundo!
Invertir: !odnuM aloH
Invertir: !odnuM aloH
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02> █
```



## Datatypes05

```
pcd3am1_2024

datatypes05.py X
parcial1 > leccion02 > datatypes05.py > ...
  augustoramirez, last week | 1 author (augustoramirez)
1  # Posiciones pares
2  cadena = "Hola Mundo!"
3  print("Esta es una cadena: ", cadena)
4
5  print("Posicion pares: ", cadena[::2])
6

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS

PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02> py datatypes05
Esta es una cadena:  Hola Mundo!
Posicion pares:  Hl ud!
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02>
```

## Datatypes06

```
datatypes06.py X
parcial1 > leccion02 > datatypes06.py > ...
  augustoramirez, last week | 1 author (augustoramirez)
1  # Que imprime?
2  cadena = "Hola Mundo!"          augustoramirez, last week • Primera parte lección
3  print("Esta es una cadena: ", cadena)
4
5  print("Puedes adivinar que esta imprimiendo?: ", cadena[-1::-3])
6

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS

PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02> py datatypes06
Esta es una cadena:  Hola Mundo!
Puedes adivinar que esta imprimiendo?:  !n o
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02>
```



## Datatypes07

```
datatypes07.py X
parcial1 > leccion02 > datatypes07.py > ...
  augustoramirez, last week | 1 author (augustoramirez)
1  # Otra forma de invertir una cadena
2  cadena = "Hola Mundo!"
3  print("Esta es una cadena: ", cadena)
4
5  cadenaInvertida = "".join(reversed(cadena))
6  print("Invertir 2da opcion: ", cadenaInvertida)
7

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS

PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02> py datatypes07.py
Esta es una cadena:  Hola Mundo!
Invertir 2da opcion:  !odnuM aloH
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02>
```

## Datatypes08

```
datatypes08.py M X
parcial1 > leccion02 > datatypes08.py > ...
  You, 1 second ago | 2 authors (You and one other)
1  # Que esta pasando en este ejercicio?
2  cadena = "Hola Mundo!"
3  print("Esta es una cadena: ", cadena)
4
5  #cadena[4] = '*'
6  cadena = cadena[:4] + '*' + cadena[5:]
7  print("Porque no puedo imprimir?: ", cadena)
8
9  # LAS CADENAS EN PYTHON SON INMUTABLES, NO SOPORTAN
10 # ASIGNACION A LOS CARACTERES

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS

PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02> py datatypes08.py
Esta es una cadena:  Hola Mundo!
Traceback (most recent call last):
  File "C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02\datatypes08.py", line
    cadena[4] = '*'
    ~~~~~^~
TypeError: 'str' object does not support item assignment
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02> py datatypes08.py
Esta es una cadena:  Hola Mundo!
Porque no puedo imprimir?:  Hola*Mundo!
```



## Datatypes09

```
datatypes09.py X
parcial1 > leccion02 > datatypes09.py > ...
augustoramirez, last week | 1 author (augustoramirez)
1 # Para modificar una cadena podemos cambiarla completa
2 cadena = "Hola Mundo!"
3 print("Esta es una cadena: ", cadena)
4
5 cadena = "Hola*Mundo!"
6 print("Se modifiko la cadena?: ", cadena)
7

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02> py d
Esta es una cadena: Hola Mundo!
Se modifiko la cadena?: Hola*Mundo!
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02>
```

## Datatypes10

```
datatypes10.py M X
parcial1 > leccion02 > datatypes10.py > ...
You, 6 seconds ago | 2 authors (augustoramirez and one other)
1 # Para borrar un caracter de una cadena
2 cadena = "Hola Mundo!"
3 print("Esta es una cadena: ", cadena)
4
5 # del cadena[4] -> inmutables, tampoco se pueden borrar
6 cadena = cadena.replace(cadena[4], '')
7 print("Se elimino el caracter?: ", cadena)
8

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02> py datatypes10.py
Esta es una cadena: Hola Mundo!
Traceback (most recent call last):
  File "C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02\datatypes10.py",
    del cadena[4]
    ~~~~~^
TypeError: 'str' object doesn't support item deletion
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02> py datatypes10.py
Esta es una cadena: Hola Mundo!
Se elimino el caracter?: HolaMundo!
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02>
```



## Datatypes11

```
datatypes11.py X
parcial1 > leccion02 > datatypes11.py > ...
augustoramirez, last week | 1 author (augustoramirez)
1 # Para borrar un caracter de una cadena podemos hacer esto
2 cadena = "Hola Mundo!"
3 print("Esta es una cadena: ", cadena)
4 augustoramirez, last week • Primera parte leccion02
5 nuevaCadena = cadena[:4] + cadena[-1:]
6 print("Se eliminaron varios caracteres?: ", nuevaCadena)
7

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02> py data
Esta es una cadena: Hola Mundo!
Se eliminaron varios caracteres?: Hola!
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02>
```

## Datatypes12

```
datatypes12.py M X
parcial1 > leccion02 > datatypes12.py > ...
You, 4 seconds ago | 2 authors (augustoramirez and one other)
1 # Borrando una cadena completa
2 cadena = "Hola Mundo!"
3 print("Esta es una cadena: ", cadena)
4
5 del cadena
6 print("Cadena eliminada...")
7 #print("Se elimino la cadena?: ", cadena)
8 print("Se elimino la cadena?: sí, se elimino la cadena al usar del") You, 1
9

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02> py datatypes12.py
Esta es una cadena: Hola Mundo!
Cadena eliminada...
Traceback (most recent call last):
  File "C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02\datatypes12.py",
    print("Se elimino la cadena?: ", cadena)
    ~~~~~
NameError: name 'cadena' is not defined
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02> py datatypes12.py
Esta es una cadena: Hola Mundo!
Cadena eliminada...
Se elimino la cadena?: sí, se elimino la cadena al usar del
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02>
```



## Datatypes13

```
datatypes13.py X
parcial1 > leccion02 > datatypes13.py > ...
augustoramirez, last week | 1 author (augustoramirez)
1 # Formato de cadenas augustoramirez, last week • Primera parte
2 cadena1 = "{} {} {}".format('Hola', 'Mundo', '!')
3 print("Esta es una cadena con formato: ", cadena1)
4
5 cadena2 = "{1} {2} {0}".format('Hola', 'Mundo', '!')
6 print("Esta es una cadena con formato desordenado: ", cadena2)
7
8 cadena3 = "{a} {b} {c}".format(a='Hola', b='Mundo', c='!')
9 print("Esta es una cadena con formato: ", cadena3)
10

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02> py data
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 !
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02>
```

## Datatypes14

```
datatypes14.py X
parcial1 > leccion02 > datatypes14.py > ...
augustoramirez, last week | 1 author (augustoramirez)
1 # Mas Formato de cadenas con numeros augustoramirez, last week • Primer
2 cadena1 = "{0:b}".format(13)
3 print("Binario del 256: ", cadena1)
4
5 cadena2 = "{0:e}".format(1234.34534535359379)
6 print("Formato exponencial: ", cadena2)
7
8 cadena3 = "{a:.4f}".format(a=3.141592)
9 print("Flotante truncado a 4 digitos: ", cadena3)
10
11

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02> py datatypes14.
Binario del 256: 1101
Formato exponencial: 1.234345e+03
Flotante truncado a 4 digitos: 3.1416
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02>
```







## Datatypes16

```
datatypes16.py X
parcial1 > leccion02 > datatypes16.py > ...
augustoramirez, 23 hours ago | 1 author (augustoramirez)
1 # ENTERos y flotantes augustoramirez, 23 hours ago • Mas ejercicios
2 entero = 5
3 flotante = 2/3
4
5 print(type(entero))
6 print(type(flottante))
7 print(entero)
8 print(flottante)
9 print(entero + flotante)
10 print(3+4)
11 print(3-4)
12 print(3*4)
13 print(3/4)
14 print(3//4)
15 print(3%4)
16

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02> py datatypes16.py
<class 'int'>
<class 'float'>
5
0.6666666666666666
5.666666666666667
7
-1
12
0.75
0
7
```



## Datatypes17

```
datatypes17.py M X
parcial1 > leccion02 > datatypes17.py > ...
You, 7 seconds ago | 2 authors (augustoramirez and one other)
1  # Complejos
2
3  a = 1 + 5j
4  b = 2 + 3j
5
6  print("Type:", type(a))
7
8  c = a + b
9  print("Suma:",c, type(c))
10
11 d = 1 + 5j
12 e = 2 - 3j
13
14 f = d - e
15 print("Resta:",f)
16
17

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02> py datatypes17
Type: <class 'complex'>
Suma: (3+8j) <class 'complex'>
Resta: (-1+8j)
Division: (1.307692307692308+0.5384615384615384j)
Multiplicacion: (-13+13j) <class 'complex'>
```



## Datatypes18

```
datatypes18.py X
parcial1 > leccion02 > datatypes18.py > ...
augustoramirez, 24 hours ago | 1 author (augustoramirez)
1 # Booleanos augustoramirez, 24 hours ago • Mas ejercicios
2
3 a = True
4 b = False
5
6 print(type(a))
7
8 print(f'a={a}')
9
10 print(f'b={b}')
11
12 print(f'34 == 34 : {34==34}')
13
14 print(f'23 == 24: {23 == 24}')
15
16 print(f'34 != 34 : {34!=34}')
17

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02> py 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
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02>
```



## Datatypes19

```
datatypes19.py M X
parcial1 > leccion02 > datatypes19.py > ...
augustoramirez, 24 hours ago | 1 author (augustoramirez)
1 # Listas augustoramirez, 24 hours ago • Mas ejercicios
2 listaVacia = []
3 listaNumeros = [1,2,3,4,5,6]
4 listaLetras = ['a', 'b', 'c']
5 listaObjetos = [1, '*', 0, True, 3.141592, False, "Hola Mundo!", listaLetras]
6
7 print(f'Lista Vacia: {listaVacia}')
8 print(f'Tamaño Lista Vacia: {len(listaVacia)}')
9
10 print(f'Lista Numeros: {listaNumeros}')
11 print(f'Tamaño Lista Numeros: {len(listaNumeros)}')
12
13 print(f'Lista Letras: {listaLetras}')
14 print(f'Tamaño Lista Letras: {len(listaLetras)}')
15
16 print(f'Lista Objetos: {listaObjetos}')
17 print(f'Tamaño Lista Objetos: {len(listaObjetos)}')
18

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02> py datatypes19.py
Lista Vacia: []
Tamaño Lista Vacia: 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

```
datatypes20.py M X
parcial1 > leccion02 > datatypes20.py > ...
augustoramirez, 24 hours ago | 1 author (augustoramirez)
1 # Listas Bidimensionales
2
3 matriz = [[1,2,3,4], [5,6,7,8], [9,10,11,12]]
4
5 print(f'Matriz completa: {matriz}')
6 print(f'Segundo Renglon: {matriz[1]}')
7 print(f'Tercer Renglon, segunda Columna: {matriz[2][1]}')
8

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02> py 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

```
datatypes21.py X
parcial1 > leccion02 > datatypes21.py > ...
augustoramirez, 24 hours ago | 1 author (augustoramirez)
1 # Listas acceso indexado negativo
2
3 lista = ['a', 'b', 'c', 'd', 'e']
4
5 print(f'Ultima posicion 1: {lista[4]}')
6 print(f'Ultima posicion 2: {lista[len(lista)-1]}')
7 print(f'Ultima posicion 3: {lista[-1]}')
8
9 print(f'Penultima posicion {lista[-2]}')

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02> py datatypes21.py
Ultima posicion 1: e
Ultima posicion 2: e
Ultima posicion 3: e
Penultima posicion d
```



## Datatypes22

```
datatypes22.py X
parcial1 > leccion02 > datatypes22.py > ...
augustoramirez, 24 hours ago | 1 author (augustoramirez)
1 # Cadena a Lista augustoramirez, 24 hours ago • Mas ejercicios
2
3 cadena = 'Hola Mundo! Aqui estoy ...'
4 lista = cadena.split()
5
6 print(f'Separada por espacios en una lista: {lista}')
7
8 cadena2 = "1&2&3&4"
9 print(f'Otro separador: {cadena2.split('&')}')
10

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02> py datatypes22.py
Separada por espacios en una lista: ['Hola', 'Mundo!', 'Aqui', 'estoy', '...']
Otro separador: ['1', '2', '3', '4']
```



## Datatypes23

```
datatypes23.py X
parcial1 > leccion02 > datatypes23.py > ...
  augustoramirez, 24 hours ago | 1 author (augustoramirez)
1  # Agregando a Listas
2
3  # Primero es una lista vacia
4  lista = []
5  print(f'Lista vacia: {lista}')
6
7  lista.append('IPN')
8  lista.append('UNAM')
9  lista.append('TEC')
10 lista.append('IBERO')
11
12 print(f'Universidades: {lista}')
13
14 for i in range(2,5):
15     lista.append(i)
16
17 print(f'Lista con universidades y numeros: {lista}')
18
19 lista.insert(2, 'ANAHUAC')
20 print(f'Insertando un objeto: {lista}')
21

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS

PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02> py 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]
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02> |
```



## Datatypes24

```
datatypes24.py M X
parcial1 > leccion02 > datatypes24.py > ...
You, 1 second ago | 2 authors (augustoramirez and one other)
1 # Invertiendo una lista
2
3 lista = ['IPN', 'UNAM', 'ANAHUAC', 'TEC', 'IBERO', 2, 3, 4]
4
5 print(f'Lista original: {lista}')
6
7 lista.reverse()
8
9 print(f'Lista invertida: {lista}')
10
11 print(f'Lista original nuevamente: {lista[::-1]}')
12
13 print(f'Volvemos a invertirla: {list(reversed(lista))}')
14 # reversed crea una nueva vista ITERABLE del objeto lista, es decir
15 # uno nuevo, no la invierte sobre esa misma
16 # (necesario que sea convertido en una list()) You, 1 second ago • Uncommitt

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02> py 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]
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02> |
```





## Datatypes25

```
datatypes25.py M X
parcial1 > leccion02 > datatypes25.py > ...
3  lista = [1, 2, 3, 4, 2, 2, 5, 6, 2, 7]
4
5  print(f'Lista original: {lista}')
6
7  lista.remove(2)
8  lista.remove(6)
9
10 print(f'Se eleimino un elemento: {lista}')
11
12 x = lista.pop()
13 x = lista.pop()
14 x = lista.pop()
15
16 print(f'Se extrajo el ultimo elemento: {x} de la lista: {lista}')
17
```

augustoramirez, 24 hours ago • Mas ejercicios

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

```
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02> py datatypes25.py
Lista original: [1, 2, 3, 4, 2, 2, 5, 6, 2, 7]
Se eleimino un elemento: [1, 3, 4, 2, 2, 5, 2, 7]
Se extrajo el ultimo elemento: 5 de la lista: [1, 3, 4, 2, 2]
PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02>
```



## Datatypes26

```
datatypes26.py X
parcial1 > leccion02 > datatypes26.py > ...
augustoramirez, 24 hours ago | 1 author (augustoramirez)
1 # Slicing
2
3 lista = [i for i in range(100,1100, 100)]
4
5 print(f'Lista original: {lista}')
6
7 print(f'Slice de 0 a 9: {lista[0:len(lista)]}')
8
9 print(f'Slice de 0 a 2: {lista[0:3]}') augustoramirez, 24 hours ago •
10
11 print(f'Slice de Inicio a 2: {lista[:3]}')
12
13 print(f'Slice de 3 a 9: {lista[3:10]}')
14
15 print(f'Slice de 3 al final: {lista[3:]}')
16
17 print(f'Slice de 4 a 6: {lista[4:7]}')
18

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

PS C:\Users\Dell Latitude\Documents\GitHub\pcd3am1_2024\parcial1\leccion02> py datatypes26
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]
```