



INSTITUTO POLITECNICO
NACIONAL



**Escuela Superior de
Computación**

Lección 2

PROGRAMACION PARA LA CIENCIA DE DATOS

Prof. Mario Augusto Ramírez

Alumna: Santiago Trejo Edna Jania

CODIGO 1

```
C: > Users > pipo > Downloads > datatypes01.py > ...  
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
```

CODIGO COMPILADO

```
Esta es una cadena:  Hola Mundo!  
Primer caracter de la cadena:  H  
Ultimo caracter de la cadena:  !
```

CODIGO MODIFICADO

```
Esta es una cadena:  Hola Mundo!  
segundo caracter de la cadena:  o  
tercer caracter de la cadena:  l  
cuarto caracter de la cadena:  a  
quinto caracter de la cadena:  
penultimo caracter de la cadena:  o
```

CODIGO 2

C: > Users > pipo > Downloads > datatypes02.py > ...

```
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
```

CODIGO COMPILADO

```
Esta es una cadena:  Hola Mundo!
Posicion 0:  H
Posicion 1:  o
Posicion 2:  l
Posicion 3:  a
Posicion -1:  !
Posicion -2:  o
```

CODIGO 3

```
# Slicing
cadena = "Hola Mundo!"
print("Esta es una cadena: ", cadena)

print("Silice: ", cadena[0:4])
print("Silice: ", cadena[5:10])
print("Silice: ", cadena[5:])
print("Silice: ", cadena[:4])
print("Silice: ", cadena[-1:])
print("Silice: ", cadena[3:-2])
```

CODIGO COMPILADO

```
Esta es una cadena:  Hola Mundo!
Silice:  Hola
Silice:  Mundo
Silice:  Mundo!
Silice:  Hola
Silice:  !
Silice:  a Mund
```

CODIGO MODIFICADO

```
Esta es una cadena:  Hola Mundo!
Silice:  Mundo
Silice:
Silice:  Mundo!
Silice:  Hola
Silice:  Hola Mun
Silice:  a Mund
```

CODIGO 4

C: > Users > pipo > Downloads > datatypes04.py > ...

```
1  # Invertir cadena
2  cadena = "Hola Mundo!"
3  print("Esta es una cadena: ", cadena)
4
5  print("Invertir: ", cadena[::-1])
6
7
```

CODIGO COMPILADO

```
Esta es una cadena:  Hola Mundo!
Invertir:  !odnuM aloH
```

CODIGO MODIFICADO

```
Esta es una cadena:  Hola Mundo!
Invertir:  !du lH
```

CODIGO 5

```
C: > Users > pipo > Downloads > datatypes05.py > ...  
1  # Posicioes pares  
2  cadena = "Hola Mundo!"  
3  print("Esta es una cadena: ", cadena)  
4  
5  print("Posicion pares: ", cadena[::2])  
6
```

CODIGO COMPILADO

```
Esta es una cadena:  Hola Mundo!  
Posicion pares:  Hl ud!
```

CODIGO MODIFICADO

```
Esta es una cadena:  Hola Mundo!  
Posicion pares:  !du lH
```

CODIGO 6

```
# Que imprime?  
cadena = "Hola Mundo!"  
print("Esta es una cadena: ", cadena)  
  
print("Puedes adivinar que esta imprimiendo?: ", cadena[-1::-3])
```

CODIGO COMPILADO

```
Esta es una cadena:  Hola Mundo!  
Puedes adivinar que esta imprimiendo?:  !n o
```

CODIGO MODIFICADO

```
Esta es una cadena:  Hola Mundo!  
Puedes adivinar que esta imprimiendo?:  dnuM aloH
```

CODIGO 7

```
C: > Users > pipo > Downloads > datatypes07.py > ...  
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
```

CODIGO COMPILADO

```
Esta es una cadena:  Hola Mundo!  
Invertir 2da opcion:  !odnuM aloH
```


CODIGO 8

```
C: > Users > pipo > Downloads > datatypes08.py > ...  
1  # Que esta pasando en este ejercicio?  
2  cadena = "Hola Mundo!"  
3  print("Esta es una cadena: ", cadena)  
4  
5  cadena[4] = '*'  
6  print("Porque no puedo imprimir?: ", cadena)  
7
```

CODIGO COMPILADO

```
Esta es una cadena:  Hola Mundo!  
Traceback (most recent call last):  
  File "/home/main.py", line 5, in <module>  
    cadena[4] = '*'  
TypeError: 'str' object does not support item assignment
```

***la cadena es inmutable**

CODIGO 9

```
C: > Users > pipo > Downloads > datatypes09.py > ...  
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
```

CODIGO COMPILADO

```
Esta es una cadena:  Hola Mundo!  
Se modifiko la cadena?:  Hola*Mundo!
```

CODIGO 10

```
C: > Users > pipo > Downloads > datatypes10.py > ...  
1  # Para borrar un caracter de una cadena  
2  cadena = "Hola Mundo!"  
3  print("Esta es una cadena: ", cadena)  
4  
5  del cadena[4]  
6  print("Se elimino el caracter?: ", cadena)  
7
```

CODIGO COMPILADO

```
Esta es una cadena:  Hola Mundo!  
Traceback (most recent call last):  
  File "/home/main.py", line 5, in <module>  
    del cadena[4]  
TypeError: 'str' object doesn't support item deletion
```

***la cadena es inmutable**

CODIGO 11

```
C: > Users > pipo > Downloads > datatypes11.py > ...  
1  # Para borrar un caracter de una cadena podemos hacer esto  
2  cadena = "Hola Mundo!"  
3  print("Esta es una cadena: ", cadena)  
4  
5  nuevaCadena = cadena[:4] + cadena[-1:]  
6  print("Se eliminaron varios caracteres?: ", nuevaCadena)  
7
```

CODIGO COMPILADO

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

CODIGO MODIFICADO

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

CODIGO 12

```
C: > Users > pipo > Downloads > datatypes12.py > ...  
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  |
```

CODIGO COMPILADO

```
Esta es una cadena:  Hola Mundo!  
Cadena eliminada...  
Traceback (most recent call last):  
  File "/home/main.py", line 7, in <module>  
    print("Se elimino la cadena?: ", cadena)  
NameError: name 'cadena' is not defined
```

***no se puede acceder a una cadena ya eliminada**

CODIGO 13

```
> Users > pipo > Downloads > datatypes13.py > ...
1  # Formato de cadenas
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
11
```

CODIGO COMPILADO

```
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 !
```

CODIGO 14

```
C: > Users > pipo > Downloads > datatypes14.py > ...
1  # Mas Formato de cadenas con numeros
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
```

CODIGO COMPILADO

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

CODIGO 15

```
# Alineando cadenas
String1 = "|{:<10}|{: ^10}|{:>10}|".format('Izq',
                                           'Cen',
                                           'Der')

print("\nAlineacion de cadenas: ")
print(String1)

# Alineacion de espacios
String1 = "\n>>{0:^10}<< soy yo, y tengo >>{1:<4}<< años!".format("Mario",
                                                                    43)
print(String1)
```

CODIGO COMPILADO

```
Alineacion de cadenas:
|Izq      |   Cen   |           Der|

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

CODIGO MODIFICADO

```
|fecha    |edad|nombre|

Alineacion de cadenas:
|12-13-24|18|santiago|
```


CODIGO 16

```
# ENteros y flotantes
entero = 5
flotante = 2/3

print(type(entero))
print(type(flotante))
print(entero)
print(flotante)
print(entero + flotante)
print(3+4)
print(3-4)
print(3*4)
print(3/4)
print(3//4)
print(3%4)
```

CODIGO COMPILADO

```
<class 'int'>
<class 'float'>
5
0.6666666666666666
5.666666666666667
7
-1
12
0.75
0
3
```

CODIGO 17

```
a = 1 + 5j
b = 2 + 3j

c = a + b
print("Suma:",c)

d = 1 + 5j
e = 2 - 3j

f = d - e
print("Resta:",f)

g = 1 + 5j
h = 2 + 3j

i = g / h
print("Division:",i)

j = 1 + 5j
k = 2 + 3j

l = j * k
print("Multiplicacion:",l)
```

CODIGO COMPILADO

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

CODIGO 18

```
# Booleanos

a = True
b = False

print(type(a))

print(f'a={a}')

print(f'b={b}')

print(f'34 == 34 : {34==34}')

print(f'23 == 24: {23 == 24}')

print(f'34 != 34 : {34!=34}')

print(f'23 != 24: {23 != 24}')
```

```
x,y,z = 1,2,3

if x < y and y < z:
    print('x < y < z')
else:
    print('Error en el and')

if x > y or y < z:
    print('x > y o y < z')
else:
    print('Error en el or')

if not x > z:
    print('Negacion') S
```

CODIGO COMPILADO

```
File "/home/main.py", line 35
    print('Negacion') S
                        ^
SyntaxError: invalid syntax
```

CODIGO 19

```
listaVacia = []  
listaNumeros = [1,2,3,4,5,6]  
listaLetras = ['a', 'b', 'c']  
listaObjetos = [1, '*', 0, True, 3.141592, False, "Hola Mundo!", listaLetras]  
  
print(f'Lista Vacía: {listaVacia}')  
print(f'Tamaño Lista Vacía: {len(listaVacia)}')  
  
print(f'Lista Numeros: {listaNumeros}')  
print(f'Tamaño Lista Numeros: {len(listaNumeros)}')  
  
print(f'Lista Letras: {listaLetras}')  
print(f'Tamaño Lista Letras: {len(listaLetras)}')  
  
print(f'Lista Objetos: {listaObjetos}')  
print(f'Tamaño Lista Objetos: {len(listaObjetos)}')
```

CODIGO COMPILADO

```
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
```

CODIGO 20

```
matriz = [[1,2,3,4], [5,6,7,8], [9,10,11,12]]  
  
print(f'Matriz completa: {matriz}')  
print(f'Segundo Renglon: {matriz[1]}')  
print(f'Tercer Renglon, segunda Columna: {matriz[2][1]}')
```

CODIGO COMPILADO

```
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
```

CODIGO 21

```
lista = ['a', 'b', 'c', 'd', 'e']  
  
print(f'Ultima posicion 1: {lista[4]}')  
print(f'Ultima posicion 2: {lista[len(lista)-1]}')  
print(f'Ultima posicion 3: {lista[-1]}')  
  
print(f'Penultima posicion {lista[-2]}')
```

CODIGO COMPILADO

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

CODIGO 22

```
cadena = 'Hola Mundo! Aqui estoy ...'
lista = cadena.split()

print(f'Separada por espacios en una lista: {lista}')

cadena2 = "1&2&3&4"
print(f'Otro separador: {cadena2.split('&')}')
```

CODIGO COMPILADO

```
Separada por espacios en una lista: ['Hola', 'Mundo!', 'Aqui', 'estoy', '...']
Otro separador: ['1', '2', '3', '4']
```

CODIGO 23

```
# Agregando a Listas

# Primero es una lista vacia
lista = []
print(f'Lista vacia: {lista}')

lista.append('IPN')
lista.append('UNAM')
lista.append('TEC')
lista.append('IBERO')

print(f'Universidades: {lista}')

for i in range(2,5):
    lista.append(i)

print(f'Lista con universidades y numeros: {lista}')

lista.insert(2, 'ANAHUAC')
print(f'Insertando un objeto: {lista}')
|
```

CODIGO COMPILADO

```
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]
```


CODIGO 24

```
lista = ['IPN', 'UNAM', 'ANAHUAC', 'TEC', 'IBERO', 2, 3, 4]
print(f'Lista original: {lista}')
lista.reverse()
print(f'Lista invertida: {lista}')
print(f'Lista original nuevamente: {lista[::-1]}')
print(f'Volvemos a invertirla: {list(reversed(lista))}')
```

CODIGO COMPILADO

```
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]
```

CODIGO 25

```
lista = [1, 2, 3, 4, 2, 2, 5, 6, 2, 7]

print(f'Lista original: {lista}')

lista.remove(2)
lista.remove(6)

print(f'Se eleimino un elemento: {lista}')

x = lista.pop()

print(f'Se extrajo el ultimo elemento: {x} de la lista: {lista}')
```

CODIGO COMPILADO

```
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: 7 de la lista: [1, 3, 4, 2, 2, 5, 2]
```

CODIGO 26

```
lista = [i for i in range(100,1100, 100)]  
print(f'Lista original: {lista}')  
print(f'Slice de 0 a 9: {lista[0:len(lista)]}')  
print(f'Slice de 0 a 2: {lista[0:3]}')  
print(f'Slice de Inicio a 2: {lista[:3]}')  
print(f'Slice de 3 a 9: {lista[3:10]}')  
print(f'Slice de 3 al final: {lista[3:]}')  
print(f'Slice de 4 a 6: {lista[4:7]}')  
print(f'Slice de Inicio a Fin: {lista[:]}')  
print(f'Slice posiciones pares: {lista[::2]}')  
print(f'Slice posiciones impares: {lista[1::2]}')  
print(f'Slice invertido impares: {lista[-1::-2]}')  
print(f'Slice invertido pares: {lista[-2::-2]}')
```

CODIGO COMPILADO

```
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]
```