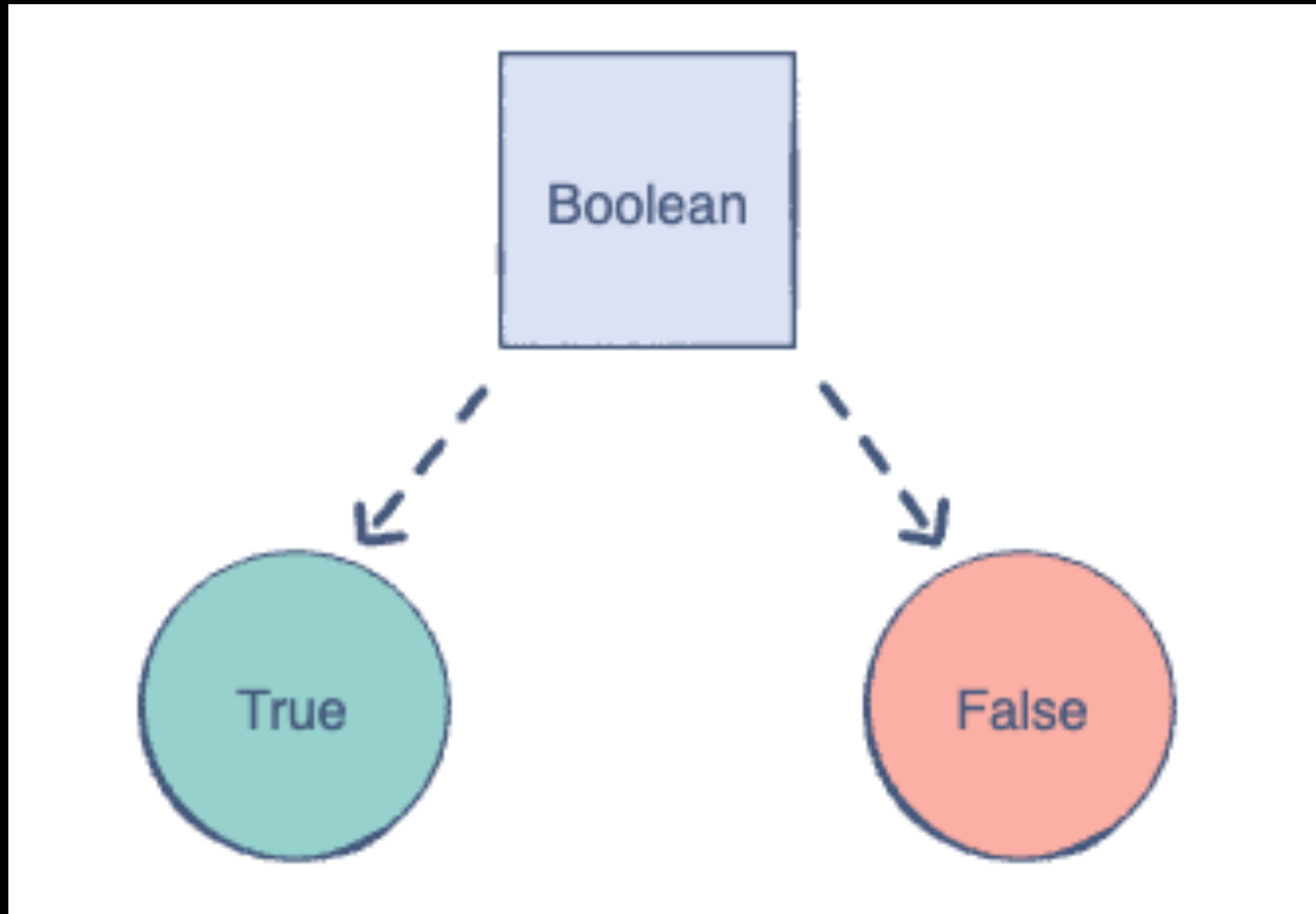
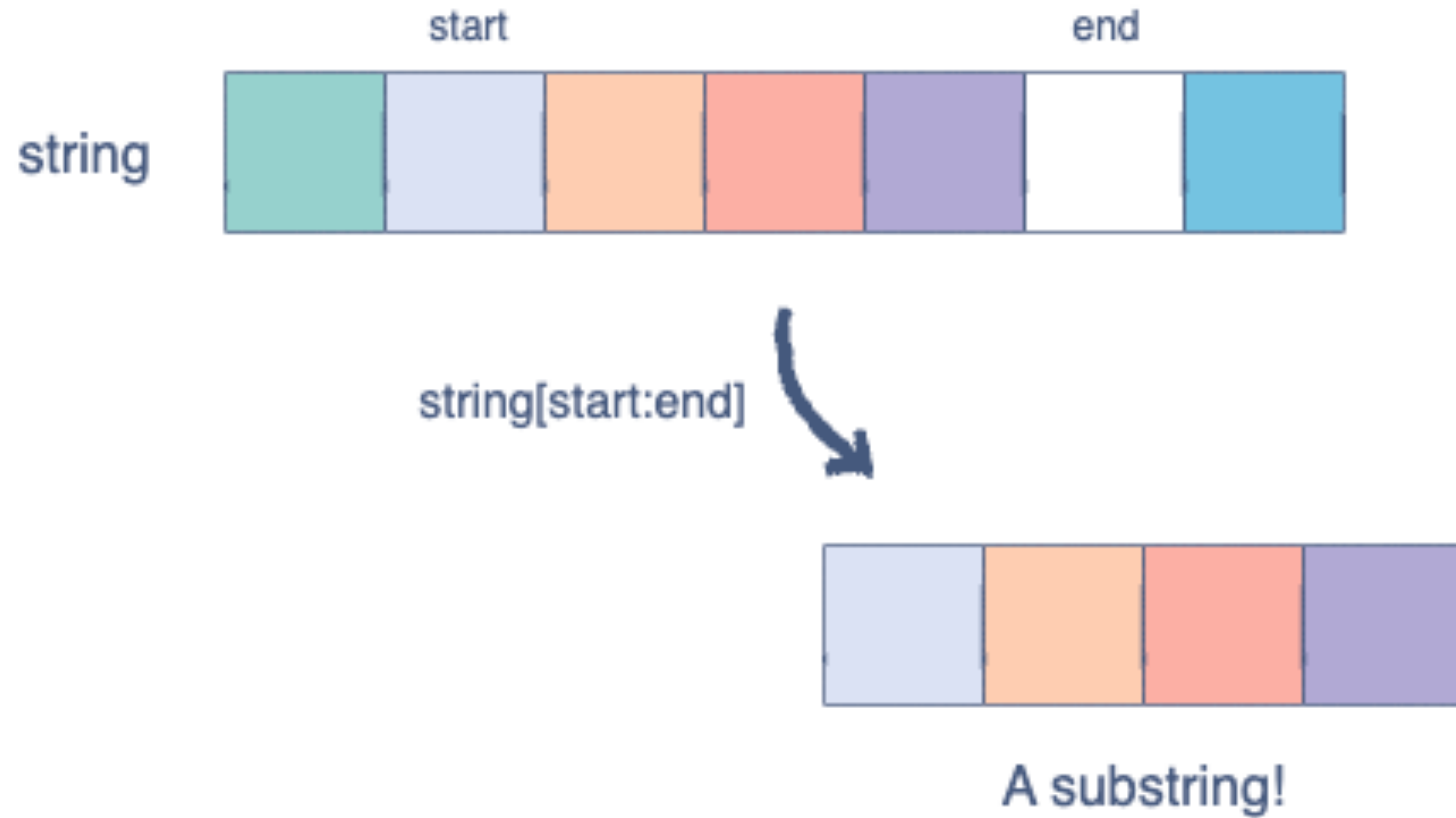


# Tipos de Datos y sentencias condicionales

# Booleans



```
string[start:end]
```



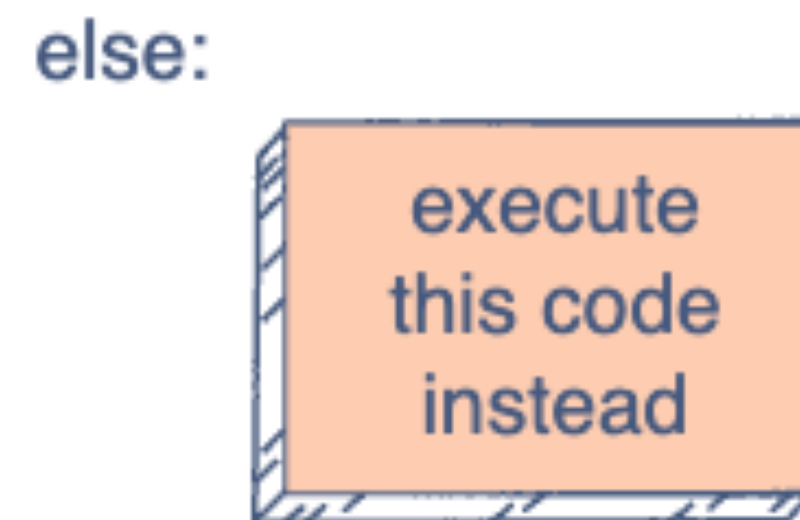
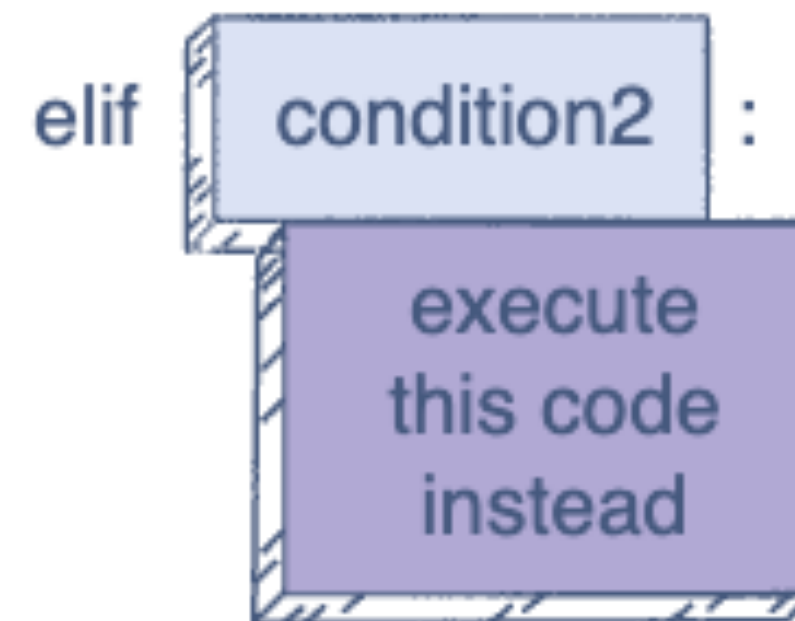
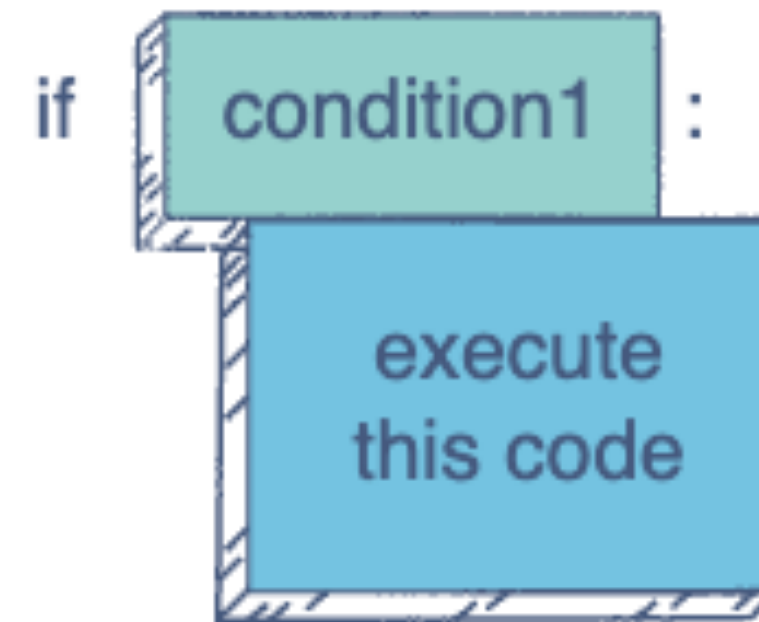
# Strings

```
24  
25 #Partial slicing  
26 my_string = "This is MY string"  
27 print(my_string[:8]) # todos los caracteres antes de la "M"  
28 print(my_string[8:]) # todos los caracteres comenzando desde la "M"  
29 print(my_string[:]) # todos los caracteres  
30 print(my_string[::-1]) # todos los caracteres en reversa con step -1  
31
```

Index:	0	1	2	3	4
	H	e	l	l	o

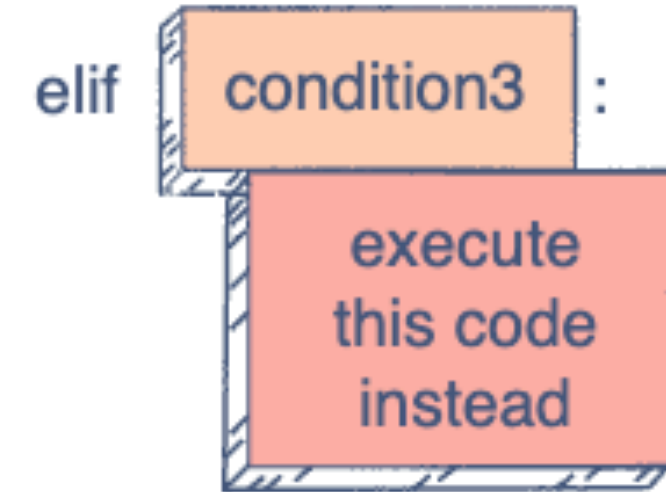
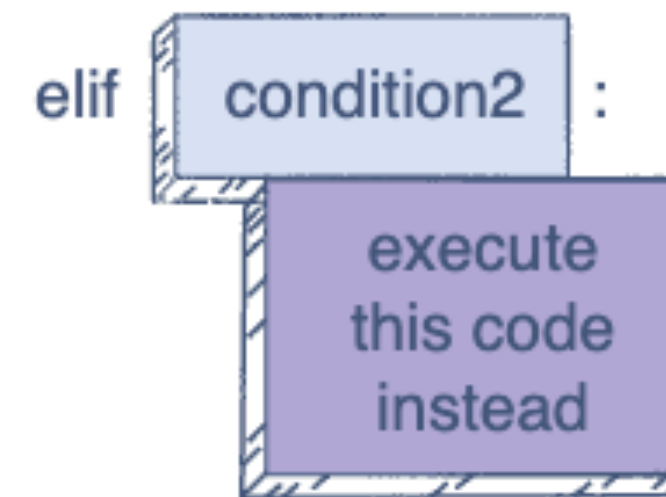
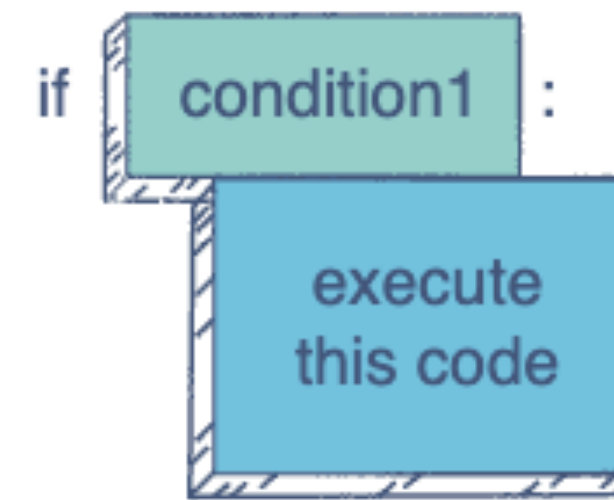
Length: 5

# If-elif-else



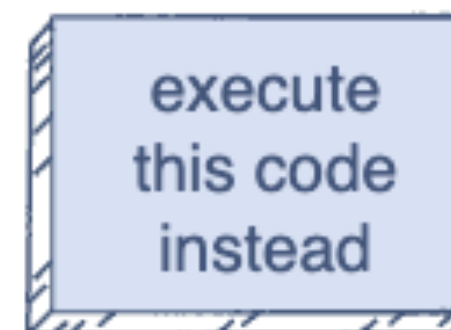
1102 / clase1 / em.py / ...

```
1 light = "Red"
2
3 if light == "Green":
4     print("Go")
5 elif light == "Yellow":
6     print("Caution")
7 elif light == "Red":
8     print("Stop")
9 else:
10    print("Incorrect light signal")
```



Multiple elif statements...

else:



# Quiz 1

## Obtener el descuento de un precio

Price	Discount
300 or greater	30%
200 - 300	20%
100 - 200	10%
0 - 100	5%
Less than 0	0



# Operadores Lógicos

And

Or

Not

```
modulo2 > clase1 > logical.py > ...  
1 my_bool = True or False  
2 print(my_bool)  
3  
4 my_bool = True and False  
5 print(my_bool)  
6  
7 my_bool = not True  
8 print(my_bool)  
9  
10 # True es considerado un 1  
11 # False es considerado un 0  
12 |  
13 print(10 * True)  
14 print(10 * False)
```

# Operadores Bitwise

&

|

^ (Xor)

~ (Not)

<< (shift bits left)

>> (shift bits right)

modulo2 > classe1 > bitwise.py > ...

```
1 num1 = 10 # 01010
2
3 num2 = 20 # 10100
4
5 print(num1 & num2) # 00000
6 print(num1 | num2) # 11110
7 print(num1 ^ num2) # 11110
8 print(~num1) # -11
9 print(num1 << 3) # 01010000 80
10 print(num2 >> 3) # 00000001 2
11
```

# Operaciones con String

Comparaciones

Concatenaciones

Search

```
ulo2 > clase1 > concatenacion.py > ...
1 first_half = "Bat"
2 second_half = "Man"
3
4 full_name = first_half + second_half
5 print(full_name)
6
7 print("ha" * 3)
8
9
10 random_string = "This is a random string"
11
12 print('of' in random_string) # verifica si 'of' existe en randomString
13 print('random' in random_string) # verifica si 'random' existe en randomString
```

# Grouping Values

Haciendo una lista

```
slicing.py  elif.py  numbers.py  if_independientes.py  quiz1.py
modulo2 > clase1 > grouping_values.py > ...
1 my_list = [1, 2.5, "A string", True]
2 print(my_list)
3
4 print(my_list[2])
5 print(len(my_list))
```