Python Programming

Session - 2

- Ajay Kumar

String Formatting

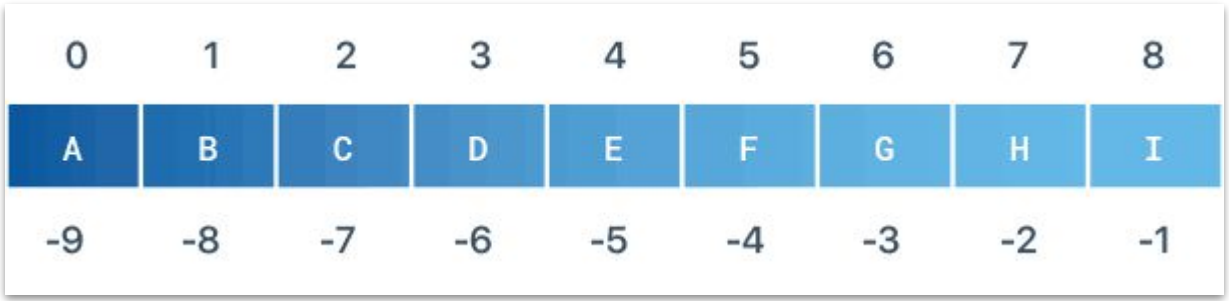
```
# printf-style % string formatting
S = '%s is %d years old.' % ('Bob', 25)
print(S)
# Prints Bob is 25 years old.
```

```
# format() Built-in Method
S = '{1} is {0} years old.'.format(25, 'Bob')
print(S)
# Prints Bob is 25 years old.
```

```
# f-String Formatter
name = 'Bob'
age = 25
S = f"{name} is {age} years old."
print(S)
# Prints Bob is 25 years old.
```

Basic String Operations

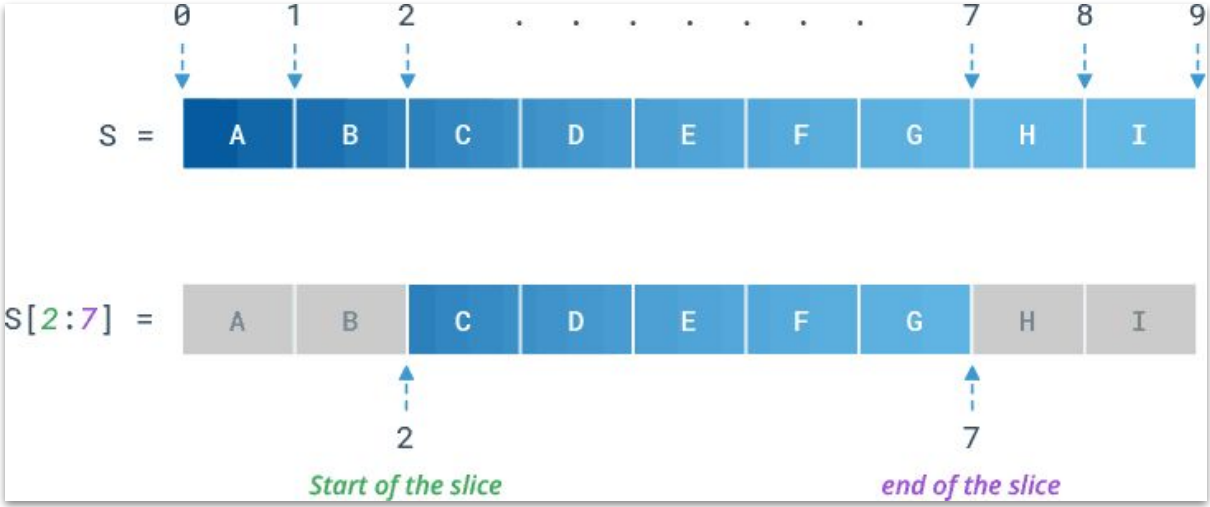
- Length, Index, Count
- Other basic string functions - upper, lower, switchcase, capitalize, startswith, endswith, split



Basic String Operations - String Slicing

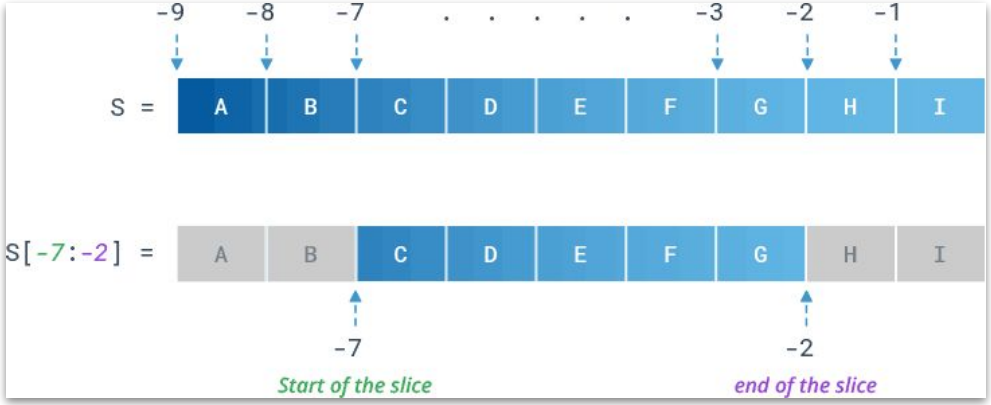
```
S[start:stop:step]
```

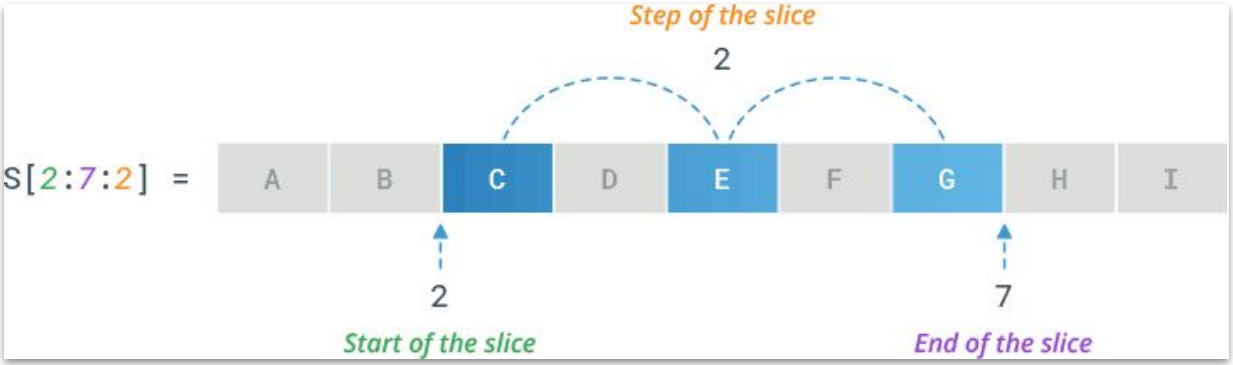
Start position *End position* *The increment*



ARINTU
Enrich Your Skills

Basic String Operations - String Slicing



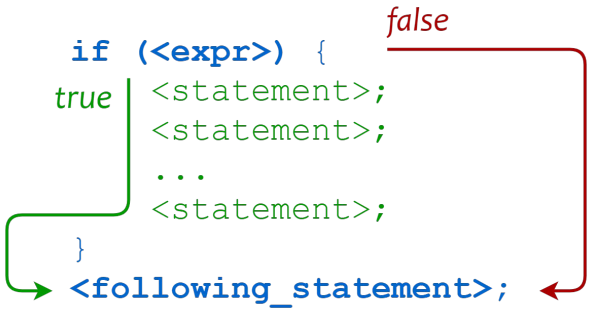


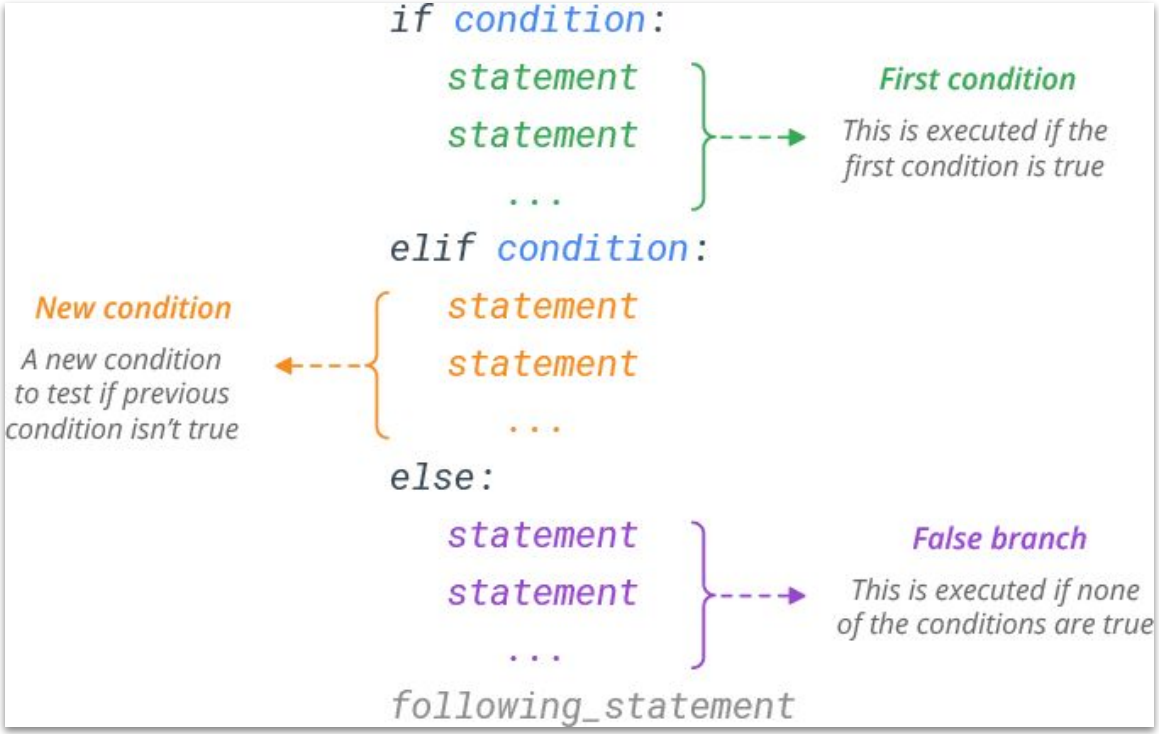
ARINTU
Enrich Your Skills

Conditions

- Python supports the usual logical conditions from mathematics:
  - Equals: a == b
  - Not Equals: a != b
  - Less than: a < b
  - Less than or equal to: a <= b
  - Greater than: a > b
  - Greater than or equal to: a >= b
- Boolean Operators - And, Or, in, is, not

Conditions - if, elif, else



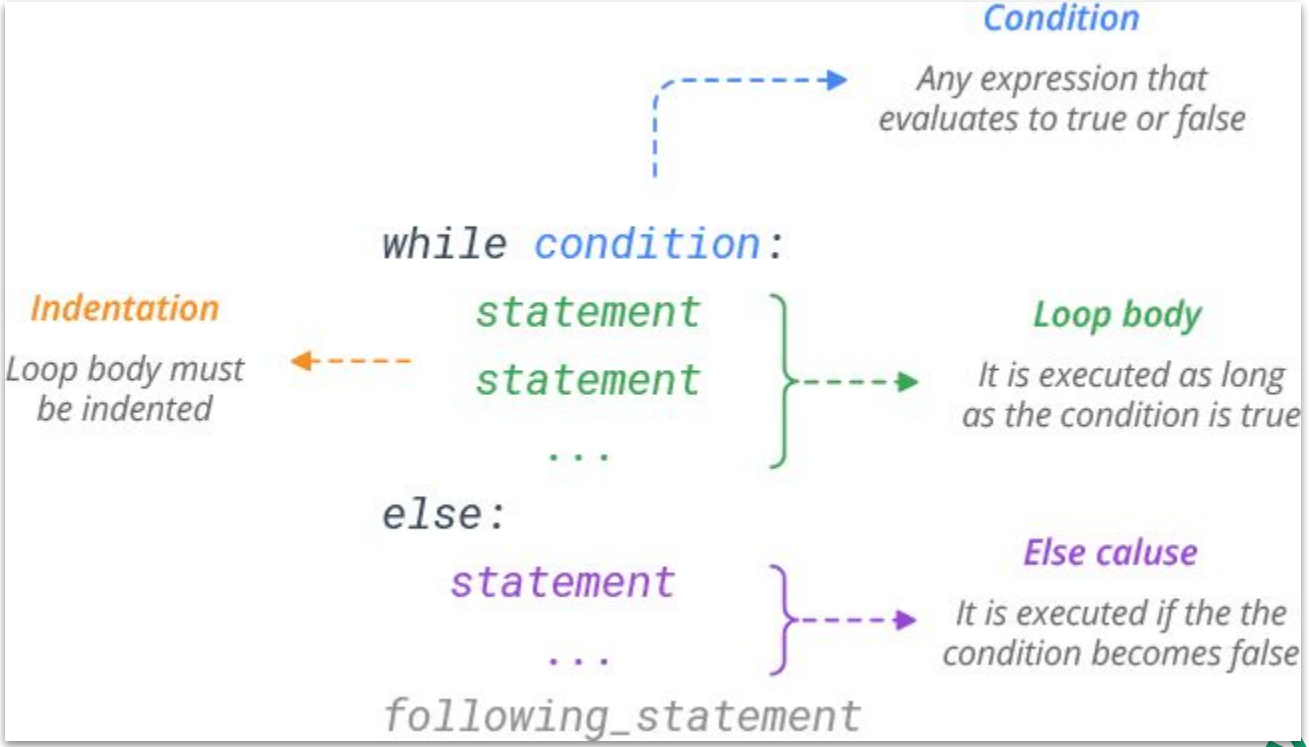


Conditions - nested if, pass

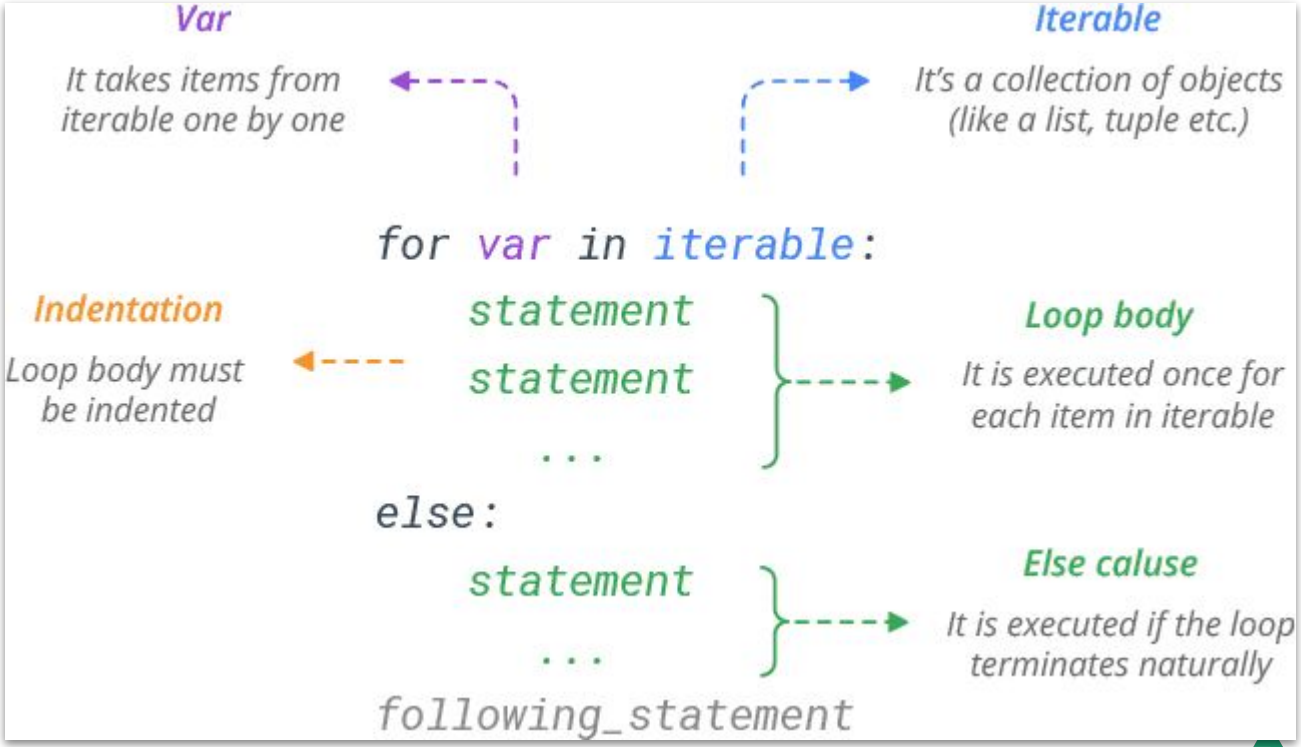
```
if Test_expression_1:
    statement(s)
    if Test_expression_1:
    statement(s)
    else:
    statement(s)
else:
    statement(s)
```

```
x = 0
if x > 0 :
    print ("x is +")
elif x < 0 :
    print ("x is -")
elif x ==0 :
    pass
else :
    print ("There is nothing to say!")
```

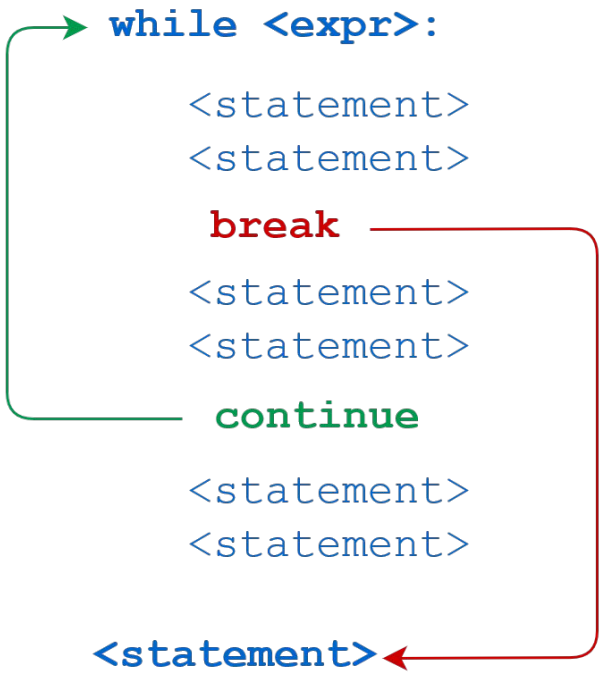
Loops - while



Loops - for



Loops - break, continue



Session Break

GitHub Link