

# Python 03



# Python Operators

Python imparte operatorii in urmatoarele grupuri

- Arithmetic operators
- Assignment operators
- Comparison operators
- Logical operators
- Identity operators
- Membership operators



# Arithmetic operators

Operatorii aritmetici sunt folositi in cazul unor operatiuni matematice

Operator	Name	Example
+	Addition	$x + y$
-	Subtraction	$x - y$
*	Multiplication	$x * y$
/	Division	$x / y$
%	Modulus	$x \% y$
**	Exponentiation	$x ** y$
//	Floor division	$x // y$

# Strings

## Concatenation

Folosind operatorul de concatenare "+", putem alatura doua sau mai multe stringuri, fie direct sau sub forma de variabila

## Multiplication

Cu ajutorul operatorului de multiplicare "\*", putem inmulti un string cu un integer

Python returneaza un nou string, continand stringul intital multiplicat de n ori

note: multiplicand un string cu un numar negativ, rezultatul va fi un string gol

# Assignment Operators

Operator	Exemplu	Ameneza
=	$x = 5$	$x = 5$
+=	$x += 3$	$x = x + 3$
-=	$x -= 3$	$x = x - 3$
*=	$x *= 3$	$x = x * 3$
/=	$x /= 3$	$x = x / 3$
%=	$x \% = 3$	$x = x \% 3$
//=	$x //= 3$	$x = x // 3$
**=	$x ** = 3$	$x = x ** 3$



# Comparison operators

Operator	Nume	Exemplu
==	Equal	$x == y$
!=	Not equal	$x != y$
>	Greater than	$x > y$
<	Less than	$x < y$
>=	Greater than or equal to	$x >= y$
<=	Less than or equal to	$x <= y$



# Logical operators

Operator	Descriere	Exemplu
and	Returneaza True daca ambele conditii sunt valide	$x < 5$ and $x < 10$
or	Returneaza True daca cel putin una dintre conditii este valida	$x < 5$ or $x < 4$
not	Returneaza inversul rezultatului. Daca rezultatul e True , ceea ce e returnat este False	not( $x < 5$ and $x < 10$ )

# Membership Operators

Operator	Descriere	Exemplu
in	Returneaza True daca variabila este prezenta in sir/lista	x in y
not in	Returneaza True daca variabila NU este prezenta in sir/lista	x not in y



# Research Operator Precedence

Ordinea operatiilor in Python

# Exercitii

- Scrie un program care sa returneze aria totala si volumul unei cutii
- Scrie un calculator BMI
  - $BMI = kg/m^2$
  - Input in kg, cm
  - Output BMI = Value
  - Printeaza toate categoriile cu true sau false daca apartinem sau nu categoriei
  - BMI Categories:
    - Underweight =  $<18.5$
    - Normal weight =  $18.5-24.9$
    - Overweight =  $25-29.9$
    - Obesity =  $30 +$



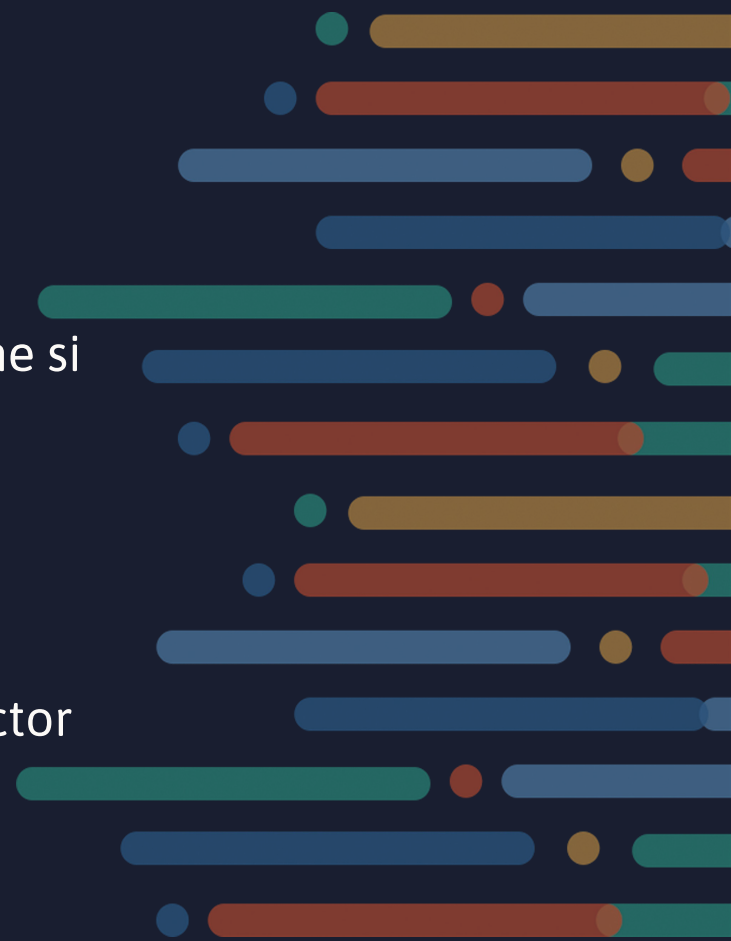
# Complex Data Types

- Lists
- Tuples
- Sets
- Dictionaries



# Lists

- Listele sunt folosite sa salveze multiple date in aceeași variabila.
- Listele sunt una dintre cele 4 modalitati de a tine și de a folosi o colectie de date in Python.
- Listele sunt create folosind paranteze drepte []  
`thislist = ["apple", "banana", "cherry"]`
- Listele se mai pot crea folosind functia constructor `list()`  
`thislist = list(("apple", "banana", "cherry"))`



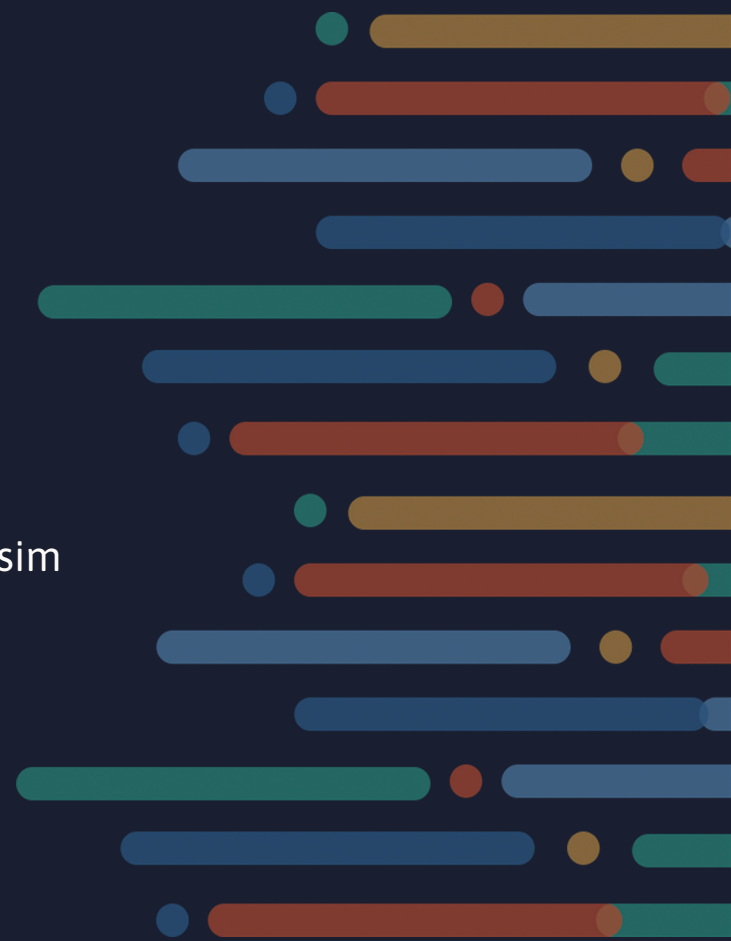
# List Items

- Elementele din lista sunt indexate
  - `thislist[0] >> "apple"`
- Elementele se pot schimba
- Pot exista valori duplicate



# List Items

- Pot contine orice datatype  
`list1 = ["apple", "banana", "cherry"]`  
`list2 = [1, 5, 7, 9, 3]`  
`list3 = [True, False, False]`
- In aceeași lista putem avea tipuri de date diferite  
`list1 = ["abc", 34, True, 40, "male"]`
- Pentru a verifica dacă avem un element în lista o să folosim operatorul de apartinență în  
`thislist = ["apple", "banana", "cherry"]`  
`print("apple" in thislist)`



# Change list items

- Pentru un element specific o sa folosim indexul  
`thislist = ["apple", "banana", "cherry"]`  
`thislist[1] = "blackcurrant"`  
`print(thislist)`
- Pentru a inlocui o parte a listei cu o alta lista o sa folosim  
sliceingul/specificam rangeul

```
thislist = ["apple", "banana", "cherry", "orange", "kiwi", "mango"]  
thislist[1:3] = ["blackcurrant", "watermelon"]  
print(thislist)
```

# Adaugarea elementelor in lista

- Append

- Adaugare element la finalul listei

```
thislist = ["apple", "banana", "cherry"]  
thislist.append("orange")  
print(thislist)
```

- Insert

- Adaugare element la indexul mentionat

```
thislist = ["apple", "banana", "cherry"]  
thislist.insert(1, "orange")  
print(thislist)
```





# Adaugarea elementelor in lista

- Extend
  - Adaugarea elementelor altei liste la finalul listei initiale

```
thislist = ["apple", "banana", "cherry"]  
tropical = ["mango", "pineapple", "papaya"]  
thislist.extend(tropical)  
print(thislist)
```

# Stergerea elementelor din lista

- Remove

- Sterge din lista elementul mentionat

```
thislist = ["apple", "banana", "cherry"]  
thislist.remove("banana")  
print(thislist)
```

- Pop

- Scoate din lista elementul de la indexul specificat
- Valoarea este returnata
- Daca nu specificam indexul, o sa fie scos ultimul element

```
thislist = ["apple", "banana", "cherry"]  
thislist.pop(1)  
print(thislist)
```



# Stergerea elementelor din lista

- Clear
  - O sa fie sterse elementele din lista, aceasta devenind goala/fara continut

```
thislist = ["apple", "banana", "cherry"]  
thislist.clear()  
print(thislist)
```



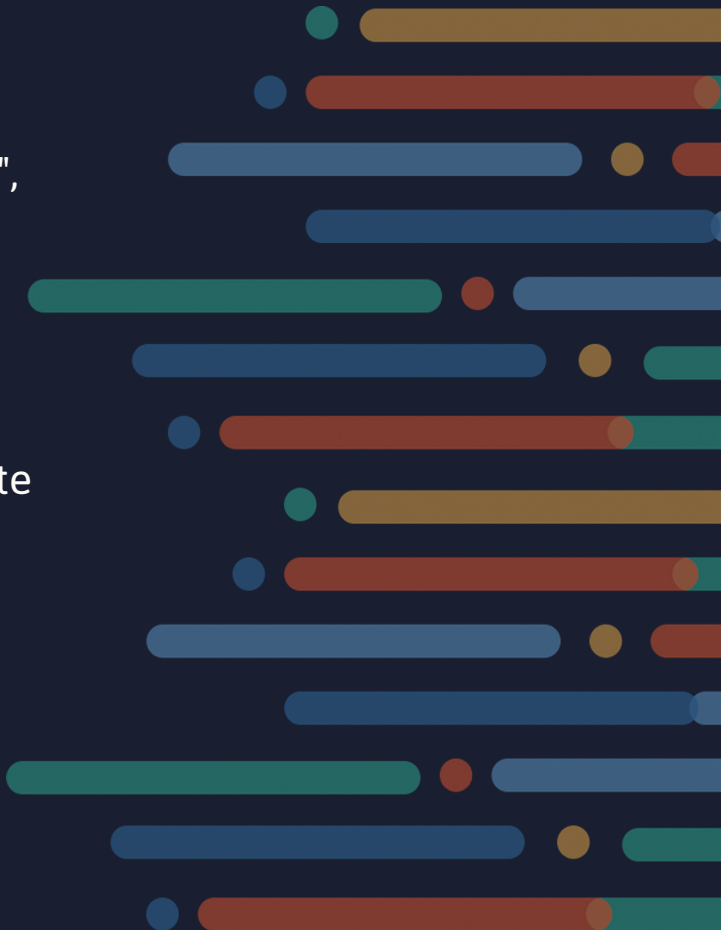
# Copiaza o lista

- Pentru a copia o lista trebuie sa folosim
  - `copy()`  
`thislist = ["apple", "banana", "cherry"]`  
`mylist = thislist.copy()`  
`print(mylist)`
  - `list()`  
`thislist = ["apple", "banana", "cherry"]`  
`mylist = list(thislist)`  
`print(mylist)`



# Exercitii

- Printeaza elementul 3 din lista fruits = ["apple", "banana", "cherry"]
- Din lista mentionata anterior schimba valoarea din "banana" in "kiwi"
- Adauga elemetele "mar","cireasa" in locul elementului "kiwi"
- Adauga lista masini= ["volvo", "dacia", "aro"] la finalul liste fruits
- Sterge elementul "volvo" din lista rezultata
- Scoate din lista elementul de pe indexul 4 si printeaza valoarea



# Homework

- Avand lista `list = [1,2,3,4,5,6,7,8,9,10]`
  - Inverseaza primul element cu ultimul
  - Lista se poate schimba (lungime, valori)
- Avand lista `list = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]`
  - Scoate elementul cu valoarea 10 din lista
  - Schimba valoarea elementului de pe indexul 2 cu patratul acestuia
  - Printeaza daca valoarea 22 se afla in lista
  - Extinde lista "list" cu lista `['salut', 'sunt', 'razvan']`
  - Scoate elementul cu indexul 10 din lista si il printeaza
  - Adauga `[99,100]` in locul elementului de la indexul 3
  - Sterge intreaga lista



# Next session

---

1 Tuples

2 Exercises