ОСНОВИ СИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ, НЕЙРОННИХ МЕРЕЖ ТА ГЛИБОКОГО НАВЧАННЯ

Модуль 6. ВИСОКОРІВНЕВА МОВА ПРОГРАМУВАННЯ PYTHON

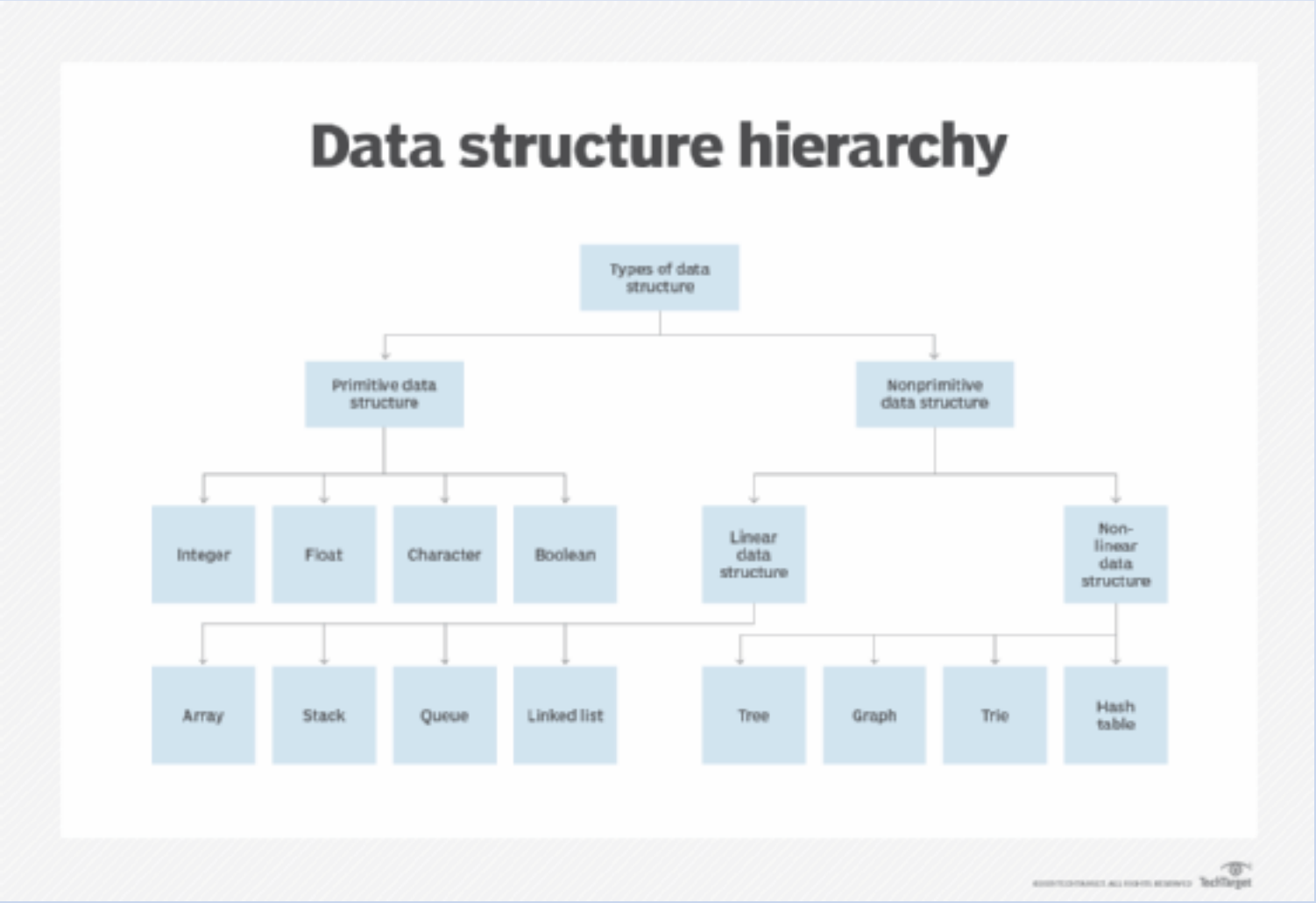
Лекція 6.2. Загальні поняття колекції та складних структур даних.

Визначення структур даних

Структура даних - це спеціалізований формат для організації, обробки, пошуку та зберігання даних. Існує кілька базових і розширених типів структур даних, кожна з яких призначена для впорядкування даних відповідно до певної мети. Структури даних полегшують користувачам доступ до потрібних їм даних і роботу з ними відповідним чином.

Кожна структура даних містить інформацію про значення даних, зв'язки між даними і - в деяких випадках - функції, які можна застосувати до даних.

Визначення структур даних



Визначення структур даних

Існують такі непримітивні структури даних:

Масив. Масив зберігає набір елементів у суміжних комірках пам'яті. Елементи одного типу зберігаються разом, тому позицію кожного елемента можна легко обчислити або знайти за індексом. Масиви можуть бути фіксованої або гнучкої довжини.

Стек. Стек зберігає колекцію елементів у лінійному порядку, в якому застосовуються операції. Цей порядок може бути останнім прийшов, першим пішов (LIFO) або першим прийшов, першим пішов (FIFO).

Черга. Черга зберігає колекцію елементів подібно до стеку, але порядок виконання операцій може бути тільки першим прийшов - першим пішов.

Зв'язаний список. Зв'язаний список зберігає колекцію елементів у лінійному порядку. Кожен елемент, або вузол, зв'язаного списку містить елемент даних, а також посилання на наступний елемент у списку.

Визначення структур даних

Дерево. Дерево зберігає колекцію елементів в абстрактний, ієрархічний спосіб. Кожен вузол пов'язаний з ключовим значенням, а батьківські вузли пов'язані з дочірніми вузлами - або підвузлами.

Купа. Купа - це деревоподібна структура, в якій асоційоване значення ключа кожного батьківського вузла більше або дорівнює значенню ключа будь-якого з його дочірніх вузлів.

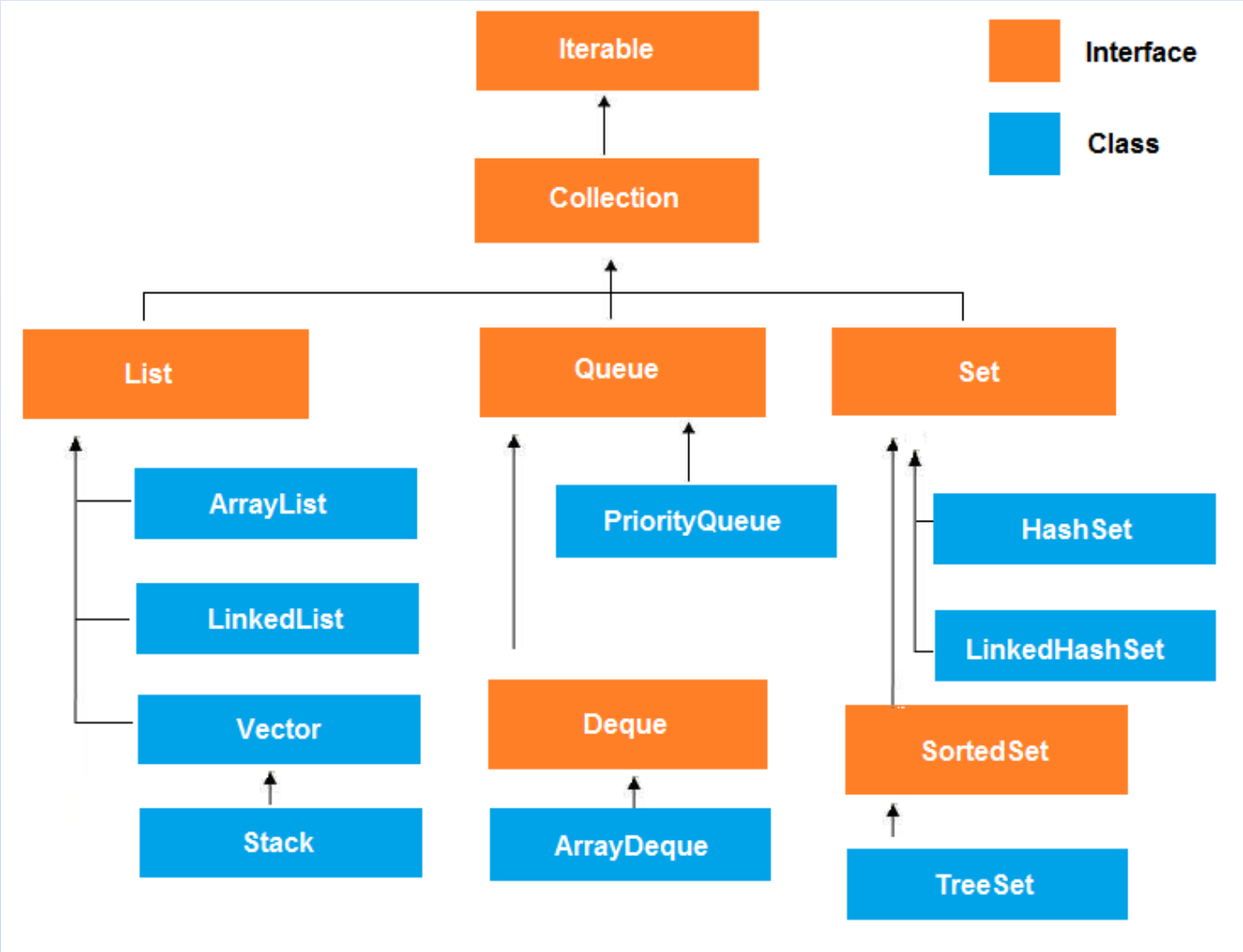
Граф. Граф зберігає колекцію елементів у нелінійний спосіб. Граф складається зі скінченної множини вузлів, також відомих як вершини, і ліній, що їх з'єднують, також відомих як ребра. Вони корисні для представлення реальних систем, таких як комп'ютерні мережі.

Геш-таблиця, також відома як hashmap, зберігає набір елементів в асоціативному масиві, який відображає ключі до значень. Геш-таблиця використовує геш-функцію для перетворення індексу в масив відер, які містять потрібний елемент даних.

Визначення структур даних

	Обчислювальна складність							
	Середнє				Найгірше			
	Індекс	Пошук	Вставка	Видалення	Індекс	Пошук	Вставка	Видалення
ArrayList	O(1)	O(n)	O(n)	O(n)	O(1)	O(n)	O(n)	O(n)
Vector	O(1)	O(n)	O(n)	O(n)	O(1)	O(n)	O(n)	O(n)
LinkedList	O(n)	O(n)	O(1)	O(1)	O(n)	O(n)	O(1)	O(1)
Hashtable	n/a	O(1)	O(1)	O(1)	n/a	O(n)	O(n)	O(n)
HashMap	n/a	O(1)	O(1)	O(1)	n/a	O(n)	O(n)	O(n)
LinkedHashMap	n/a	O(1)	O(1)	O(1)	n/a	O(n)	O(n)	O(n)
TreeMap	n/a	O(log(n))	O(log(n))	O(log(n))	n/a	O(log(n))	O(log(n))	O(log(n))
HashSet	n/a	O(1)	O(1)	O(1)	n/a	O(n)	O(n)	O(n)
LinkedHashSet	n/a	O(1)	O(1)	O(1)	n/a	O(n)	O(n)	O(n)
TreeSet	n/a	O(log(n))	O(log(n))	O(log(n))	n/a	O(log(n))	O(log(n))	O(log(n))

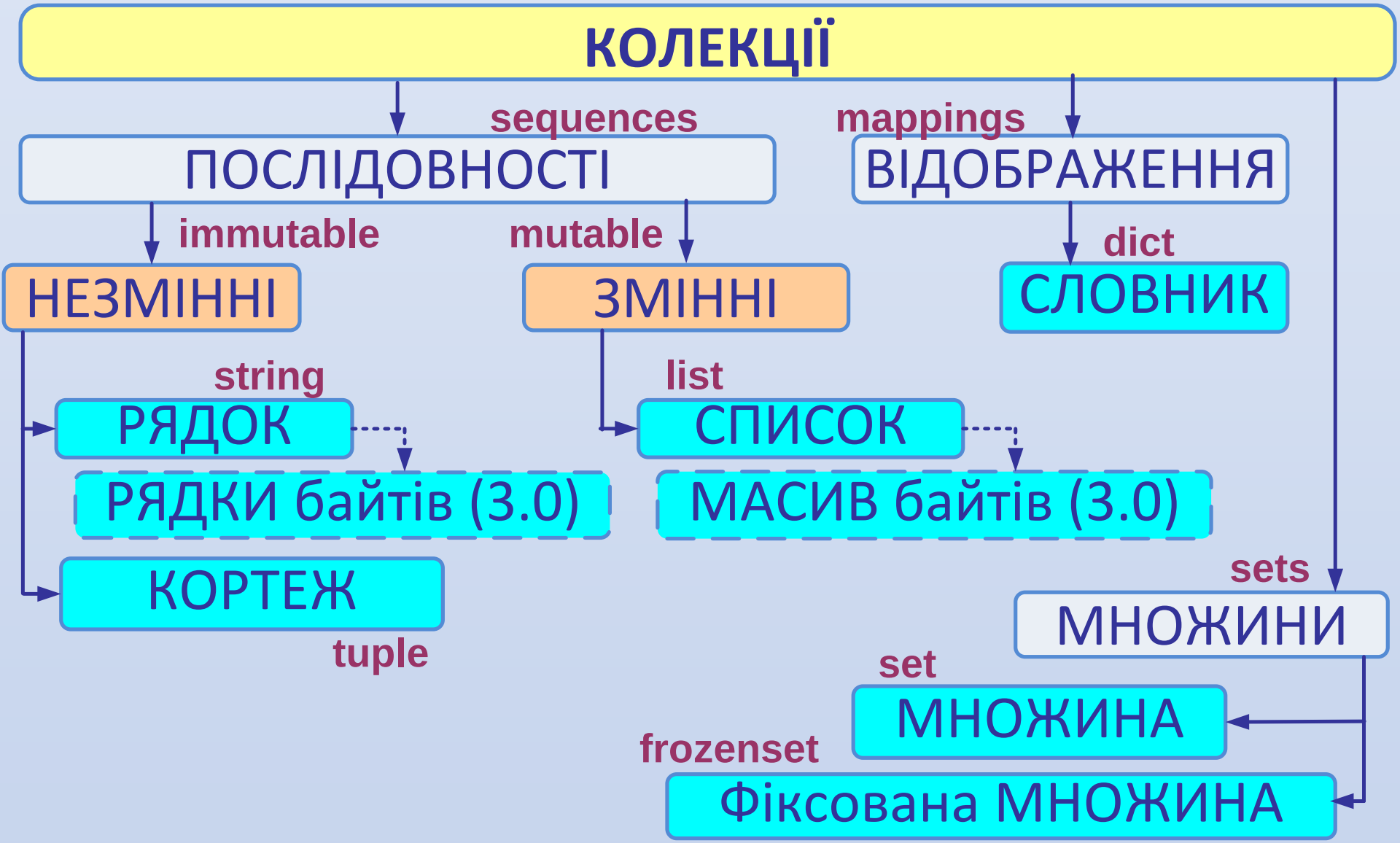
Визначення структур даних



КОЛЕКЦІЇ

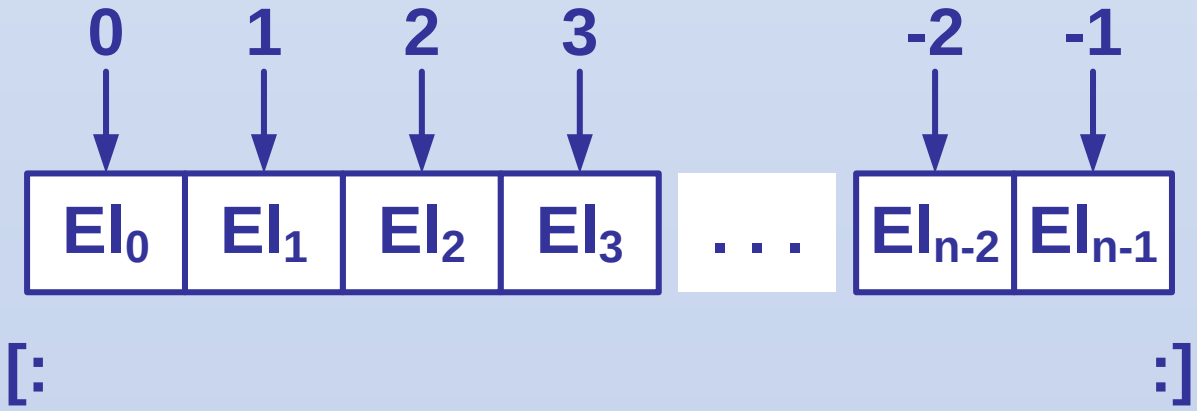
КОЛЕКЦІЯ → **програмний об'єкт (змінна–контейнер), що зберігає значення одного або різних типів та дозволяє звертатися до цих значень а також використовувати вбудовані функції і методи.**

КОЛЕКЦІЇ


КОЛЕКЦІЇ
sequences
ПОСЛІДОВНОСТІ
mappings
ВІДОБРАЖЕННЯ
immutable
НЕЗМІННІ
mutable
ЗМІННІ
dict
СЛОВНИК
string
РЯДОК
РЯДКИ байтів (3.0)
list
СПИСОК
МАСИВ байтів (3.0)
КОРТЕЖ
tuple
sets
МНОЖИНИ
set
МНОЖИНА
frozenset
Фіксована МНОЖИНА

КОЛЕКЦІЇ. Властивості

Індексованість **- кожен елемент колекції має свій порядковий номер - індекс. Це дозволяє звертатися до елементу по його порядковому індексу, проводити слайсінг («нарізку») - брати частину колекції вибираючи виходячи з їх індексу.**



КОЛЕКЦІЇ. Властивості

Змінність колекції **- дозволяє додавати в колекцію нових членів або видаляти їх після створення колекції.**

Незмінні (***числа, рядки, кортежі, фіксовані множини***): не підтримують можливість безпосередньої зміни значення об'єкта, однак завжди можна створити нові об'єкти за допомогою виразів і привласнювати їх необхідним змінним.

Змінні (***списки, словники, множини***): завжди можуть змінюватися безпосередньо, за допомогою операцій, які не створюють нові об'єкти. Змінні об'єкти можуть бути скопійовані, але вони підтримують і можливість безпосереднього зміни.

КОЛЕКЦІЇ. Властивості

Унікальність **- кожен елемент колекції може зустрічатися в ній тільки один раз. Це породжує вимогу незмінності використовуваних типів даних для кожного елемента, наприклад, таким елементом не може бути список.**

КОЛЕКЦІЇ. Властивості

Тип	Змінність	Індексованість	Унікальність	Створення
list	+	+	-	*[ ]* *list()*
tuple	-	+	-	*()* *tuple()*
string	-	+	-	*‘ ‘* *“ “*
set	+	-	+	*{el1, el2, }* *set()*
frosenset	-	-	+	*frosenset()*
dict	+ елементи - ключі + значення	-	+ елементи + ключі - значення	*{}* *{key: value}* *dict()*

КОЛЕКЦІЇ. Стандартні функції

	Функція	Дія
1	**type()**	Тип колекції
2	**print()**	Друкування елементів колекції
3	**len()**	Кількість членів колекції
4	**X in S**	Перевірка входження елемента **X** в колекцію **S**
5	**min()**	Пошук мінімального елемента
6	**max()**	Пошук максимального елемента
7	**sum()**	Сума елементів (числових)

КОЛЕКЦІЇ. Стандартні методи

	.count ()	.index()	.copy()	.clear()
list	+	+	- (<3.3) + (>=3.3)	- (<3.3) + (>=3.3)
tuple	+	+	-	-
string	+	+	-	-
set	-	-	+	+
frosenset	-	-	+	-
dict	-	-	+	+

КОЛЕКЦІЇ. Стандартні методи

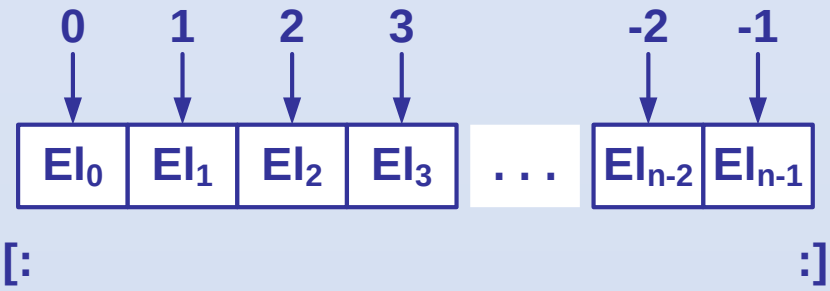
.count () - метод підрахунку певних елементів для неунікальний колекцій (рядок, список, кортеж), повертає скільки разів елемент зустрічається в колекції.

.index () - повертає мінімальний індекс переданого елемента для індексованих колекцій (рядок, список, кортеж)

.copy () - метод повертає неглибоку копію колекції (список, словник, обидва типи множини).

.clear () - метод змінюваних колекцій (список, словник, множина), що видаляє з колекції все елементи і перетворює її в порожню колекцію.

ПОСЛІДОВНОСТІ. Індексування



Для всіх індексованих колекцій можна отримати значення елемента по його індексу в квадратних дужках.

! можна задавати **негативний індекс** (зворотній порядок індексації).

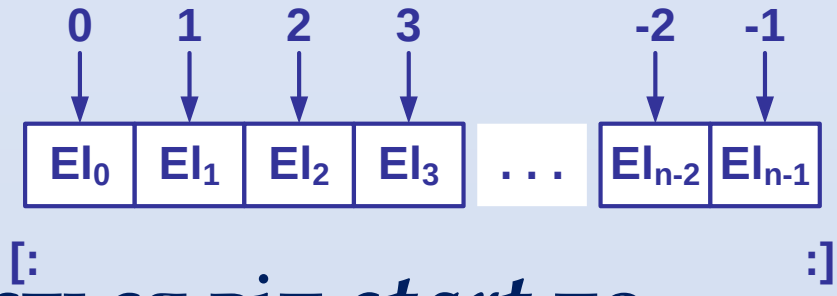
При завданні негативного індексу, останній елемент має індекс -1, передостанній -2 і так далі до першого елемента індекс якого дорівнює значенню довжини колекції з негативним знаком, тобто

-len (mycollection).

ПОСЛІДОВНОСТІ. Зрізи (slice)

Slice index

[*start* : *stop* : *step*]



Індекс елемента змінюється від ***start*** до ***stop-1*** включно з кроком ***step***

Варіанти

[: *stop* : *step*] – від 0 до ***stop-1*** з кроком ***step***

[*start* : : *step*] – від ***start*** до ***len()-1*** з кроком ***step***

[: *stop*] – від 0 до ***stop-1*** з кроком ***1***

[*start* :] – від ***start*** до ***len()-1*** з кроком ***1***

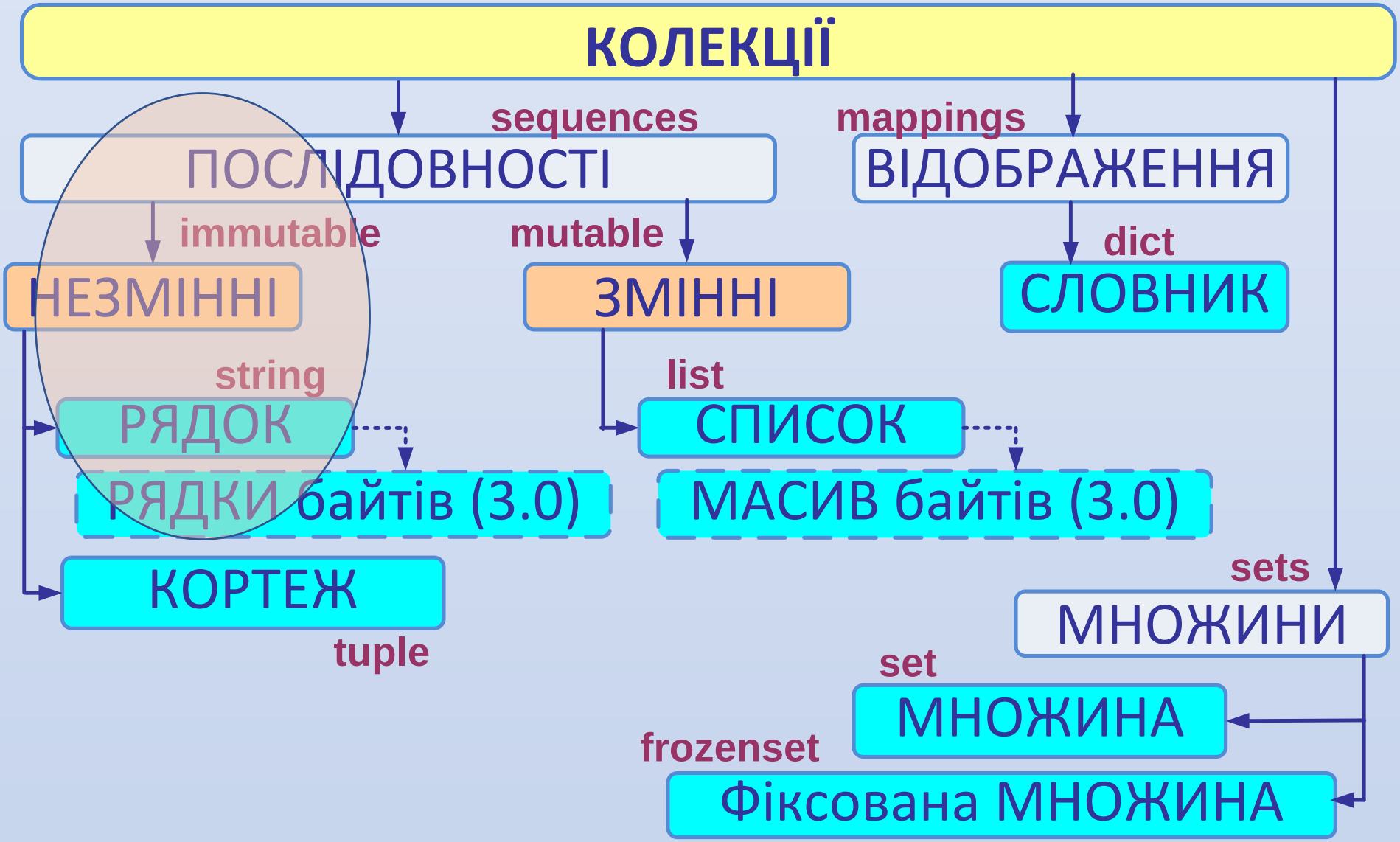
[:] – вся послідовність

ПРИМІТКА : [stop] не включається в результат

ПОСЛІДОВНОСТІ. Зрізи. Приклади

List →	A	B	C	D	E	F	G	Result
	0 (-7)	1 (-6)	2 (-5)	3 (-4)	4 (-3)	5 (-2)	6 (-1)	
[:] →	+	+	+	+	+	+	+	ABCDEFG
[::-1] ←	+	+	+	+	+	+	+	GFEDCBA
[::2] →	+		+		+		+	ACEG
[1::2] →		+		+		+		BDF
[:1]	+							A
[-1:]							+	G
[3:4]				+				D
[-3:] →					+	+	+	EFG
[-3:1:-1] ←			+	+	+			EDC
[2:5] →			+	+	+			CDE

РЯДКИ


КОЛЕКЦІЇ
sequences
ПОСЛІДОВНОСТІ
mappings
ВІДОБРАЖЕННЯ
immutable
mutable
dict
НЕЗМІННІ
ЗМІННІ
СЛОВНИК
string
list
РЯДОК
СПИСОК
РЯДКИ байтів (3.0)
МАСИВ байтів (3.0)
КОРТЕЖ
tuple
sets
МНОЖИНИ
set
МНОЖИНА
frozenset
Фіксована МНОЖИНА

РЯДКИ

Рядок = колекція → незмінна послідовність одно символьних рядків

Рядок як послідовність підтримує порядок розміщення елементів, які вона містить (символи в Unicode !), зліва направо: елементи зберігаються і витягуються виходячи з їх позиції в послідовності.

Тип	Змінність	Індексованість	Унікальність	Створення
string	-	+	-	‘ ‘ “ “

РЯДКИ. Базові операції

ЛІТЕРАЛИ	
S=‘‘ S=“”	Пустий рядок (апострофи, лапки)
S=“spam’s”	Рядок в лапках
S=‘s\np\ta\x00m	Екранований рядок
Block = “””... “””	Блок (потроєні лапки)
S= r’\temp\spam’	Неформатований рядок
S= b’spam’	Рядок байтів (>=3.0)
S= u’spam’	Рядок в unicode

Екранований рядок – екрановані пари

СЛЕШ СИМВОЛ => СПЕЦСИМВОЛ (!!! Один байт)

\n - символ new line (ASCII cod = 10)

\t – символ tabulation

\r – повернення каретки

РЯДКИ. Базові операції

Конкатенація	+	‘ab’+”123”->’ab123’
Дублювання	*	‘abc’*2 -> ‘abcabc’
Розмір рядка	len()	Кількість символів
Вибірка за індексом	[i]	*i*-й символ рядку
Зріз	[start: stop: step]	частина рядку

РЯДКИ. Функції

	Функція	Дія
1	print()	Друкування рядку
2	len()	Кількість символів в рядку
3	X in S	Перевірка входження елемента підрядку **X** в рядок **S**
4	min()	Пошук мінімального символу
5	max()	Пошук максимального символу
6		

РЯДКИ. Методи (>20)

	ВЕРТАЄ
S.lower()	Копію рядка, всі символи малі (lowercase)
S.upper()	Копію рядка, всі символи великі (uppercase)
S.swapcase()	Копію рядка, символи змінені (великі<-> малі)
S.split (sep, maxsplit)	Кількість слів в рядку. Роздільник **sep**
…	
S.is… () **S.isalpha()** **S.isdecimal()** **S.islower()** **. . .**	**True** коли виконується, **False** інакше Усі символи рядка є букви Усі символи рядка є десяткові цифри Усі символи рядка є в нижньому регістрі
…	
S.find (sub[,start[,end]])	Найменший індекс в рядку, де підрядок **sub** знаходиться в зрізі [**start:end**]
S.format(*args, *kwargs)	Форматування рядка

Форматування рядків (1)

Вирази форматування – базується на моделі функції ***printf*** мови **C.**

Оператор **%** - дає можливість множинної підстановки рядків. Використання:

1. Зліва від оператора% вказати рядок формату, що містить один або більше специфікаторів формату, кожен з яких починається з символу% (наприклад,% d).

2. Праворуч від оператора% вказати об'єкт (або об'єкти, у вигляді кортежу), значення якого має бути підставлено на місце специфікатору (або специфікаторів) в лівій частині виразу.

Форматування рядків (1)

%[(*name*)][*flags*][*width*][.*precision*]*code*

name - ключ

flags - список признаків (+, -, 0, …)

width , *precision* – кількість символів

code	
s	Рядок
c	Символ
d	Десяткове (ціле) число
i	Ціле число
f	Дійсне число
e	Дійсне в експоненціальній формі

Форматування рядків (1)

Приклади:

Num = 1234

Res= ‘integers: … %d…%-6d…%06d’ %(Num,Num,Num)

print (Res)

Out:

integers: …1234…1234…001234

fln = 98.23456789

print ('%e %f %.4f' %(fln,fln,fln))

Форматування рядків (2)

Метод форматування **s.format()** починаючи з версії 3.0

Ідея: записується рядок-шаблон **s**, який викликає метод формат **.format(),** в який передаються відповідні позиційні та іменовані аргументи

В рядку-шаблоні {0} {1} {2} …. /позиційні/, {nam1}: {name2} : … /іменовані/ змінні приймають відповідні аргументи методу.

Приклад:

template = '{mt}: {0} - {fd}'

print(*template.format('ham', mt='spam',fd='123')*)

spam:ham - 123

Форматування рядків (2)

Метод форматування **s.format()** починаючи з версії 3.0

Ідея: записується рядок-шаблон **s**, який викликає метод формат **.format(),** в який передаються відповідні позиційні та іменовані аргументи

В рядку-шаблоні {0} {1} {2} …. /позиційні/, {nam1}: {name2} : … /іменовані/ змінні приймають відповідні аргументи методу.

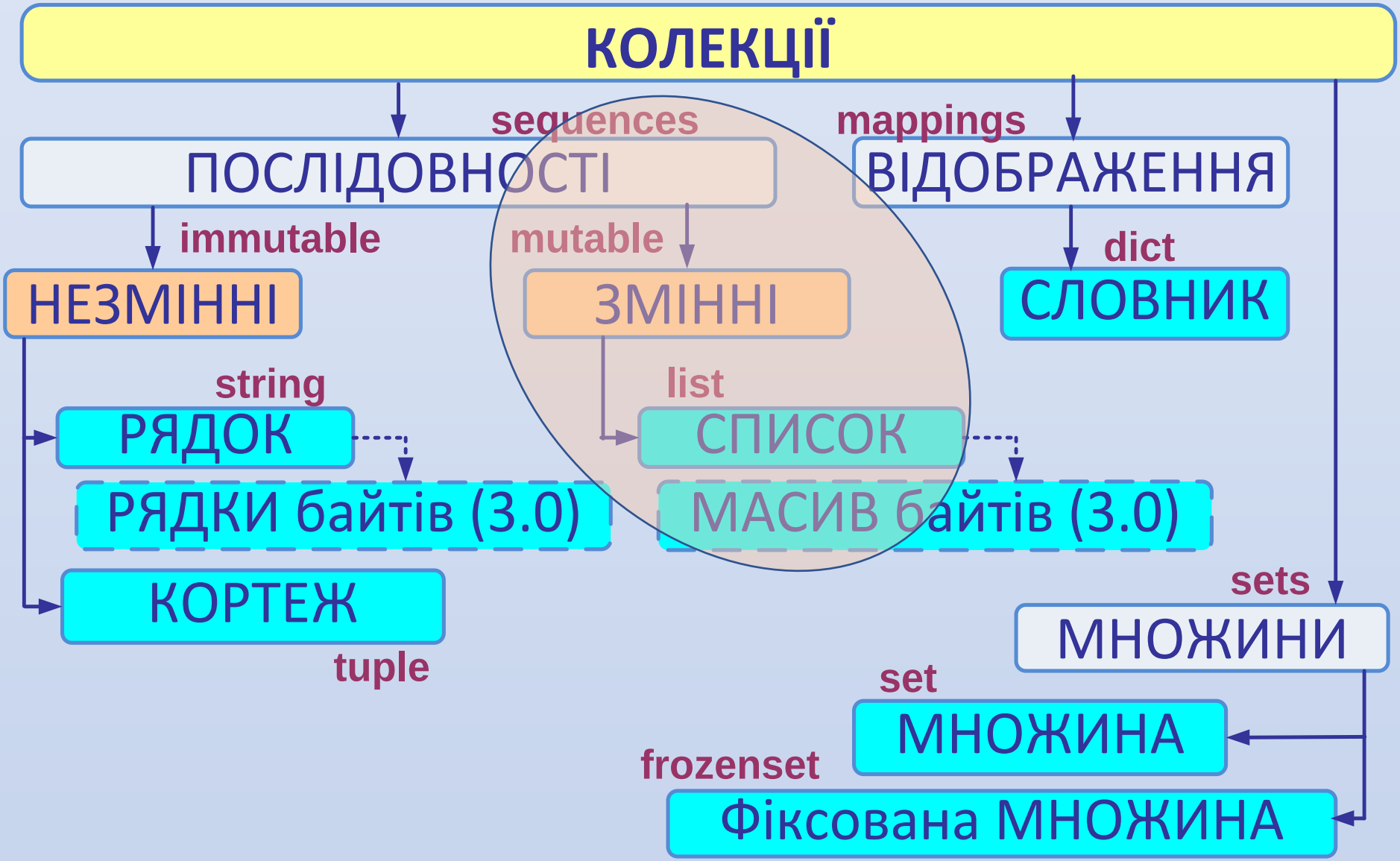
Приклад:

template = '{mt}: {0} - {fd}'

print(*template.format('ham', mt='spam',fd='123')*)

spam:ham - 123

СПИСКИ


КОЛЕКЦІЇ
sequences
mappings
ПОСЛІДОВНОСТІ
ВІДОБРАЖЕННЯ
immutable
mutable
dict
НЕЗМІННІ
ЗМІННІ
СЛОВНИК
string
list
РЯДОК
СПИСОК
РЯДКИ байтів (3.0)
МАСИВ байтів (3.0)
КОРТЕЖ
tuple
sets
МНОЖИНИ
set
МНОЖИНА
frozenset
Фіксована МНОЖИНА

СПИСКИ

Список = → упорядкована колекція об'єктів довільного типу

Список як послідовність підтримує порядок розміщення елементів, які вона містить. Доступ до елементів за зміщенням (**індексом**).

Змінна кількість елементів.

Довільне число рівнів **вкладеності**.

Тип	Змінність	Індексованість	Унікальність	Створення
list	+	+	-	*[ ]* *list()*

СПИСКИ. Створення

Створення списку

	Дія
L=[]	Пустий список
L=[5, 6, 7, 8]	Чотири елементи з індексами 0..3
L=[‘abc’,[‘def’,ghi]]	Вкладені списки
L=list(range(-4,4))	Створення списку
L=list(‘abcdef’)	Створення списку

СПИСКИ. Базові операції/функції

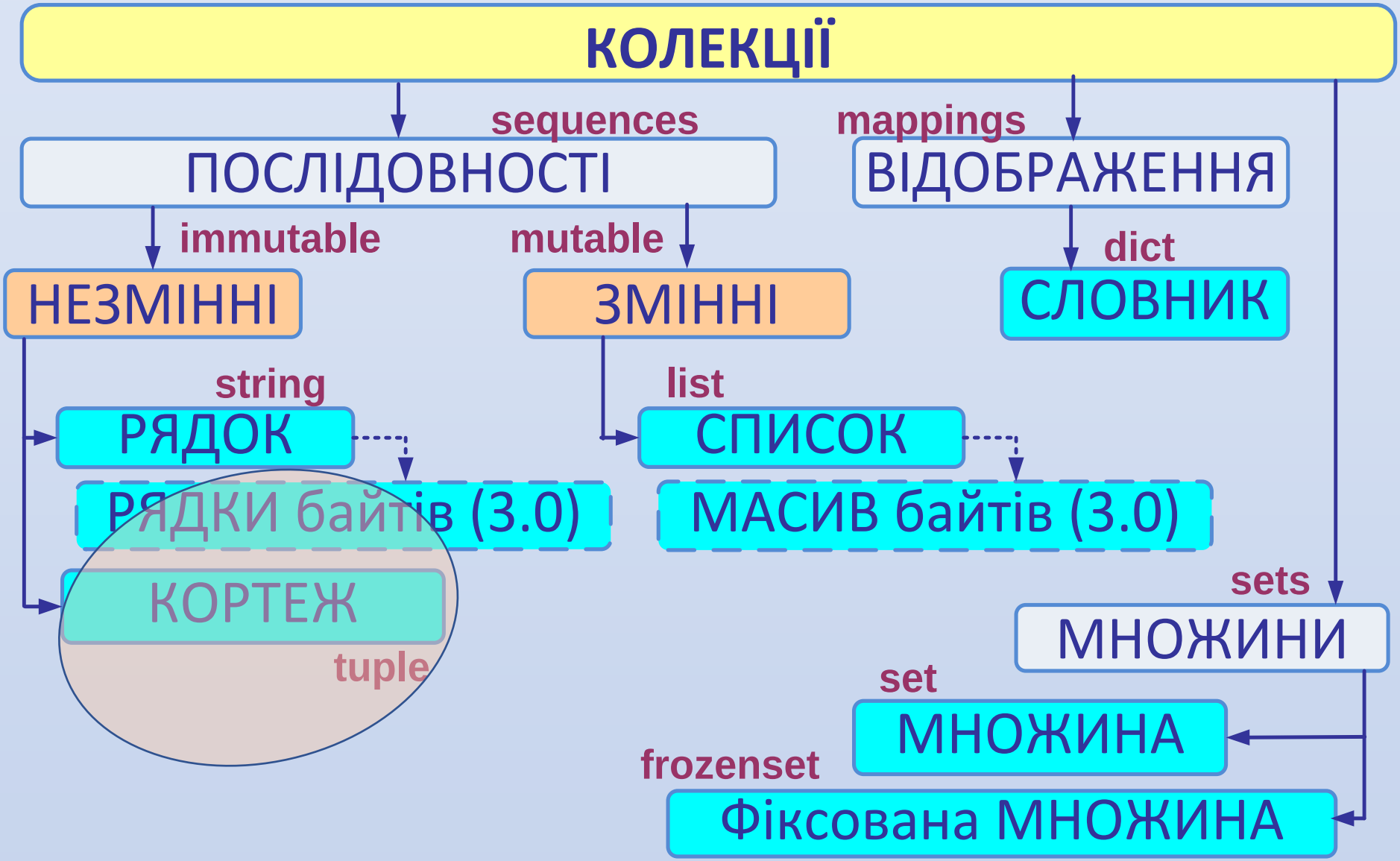
Операція		
Конкатенація	L1+L2	
Дублювання	L * N	N-разів повторення
Вибірка за індексом	L[i]	*i*-й об'єкт списку
Зріз	L[start: stop: step]	Новий список = зрізу

Функція	Дія
print(L)	*Друкування елементів списку*
len(L)	*Кількість об'єктів в списку*
X **in** L	*Перевірка входження об'єкту X в список S*
min(L)	*Пошук мінімального елемента*
max(L)	*Пошук максимального елемента*
sum(L)	*Сума елементів (числових)*

СПИСКИ. Методи

Метод	Дія
L[i] = … L[sr:st:sp] = L = [generator]	*Присвоєння за індексом* *L[5]=34* *L[1:3:1]= 34,15,18*
L.**append**() L.**extend**() L.**insert**()	*Додавання об'єктів до списку*
del L[k] L.**pop**() L.**remove**() L[sr:st:sp] =[]	*Зменшення об'єктів в списку*
L.**sort** ()	*Сортування*
L.**reverse**()	*Зміна порядку на зворотній*

КОРТЕЖІ


КОЛЕКЦІЇ
sequences
mappings
ПОСЛІДОВНОСТІ
ВІДОБРАЖЕННЯ
immutable
mutable
dict
НЕЗМІННІ
ЗМІННІ
СЛОВНИК
string
list
РЯДОК
СПИСОК
РЯДКИ байтів (3.0)
МАСИВ байтів (3.0)
КОРТЕЖ
tuple
sets
МНОЖИНИ
set
МНОЖИНА
frozenset
Фіксована МНОЖИНА

КОРТЕЖ

Кортеж = → упорядкована незмінна колекція об'єктів довільного типу

Кортеж як послідовність підтримує порядок розміщення елементів, які вона містить. Доступ до елементів за зміщенням (індексом).

Незмінна кількість елементів.

Довільне число рівнів вкладеності.

Тип	Змінність	Індексованість	Унікальність	Створення
tuple	-	+	-	*()* *tuple()*

Кортеж – масив указників на елементі

КОРТЕЖ. Створення

Створення кортежу

	Дія
T=()	Пустий кортеж
T=(5, ‘6’, 7, ‘8’)	Чотири елементи з індексами 0..3
T=(‘abc’,(‘def’,ghi))	Вкладений кортеж
T=tuple(‘abcdef’)	Створення кортежу
T=tuple(range(…))	Створення кортежу

КОРТЕЖ . Базові операції/функції

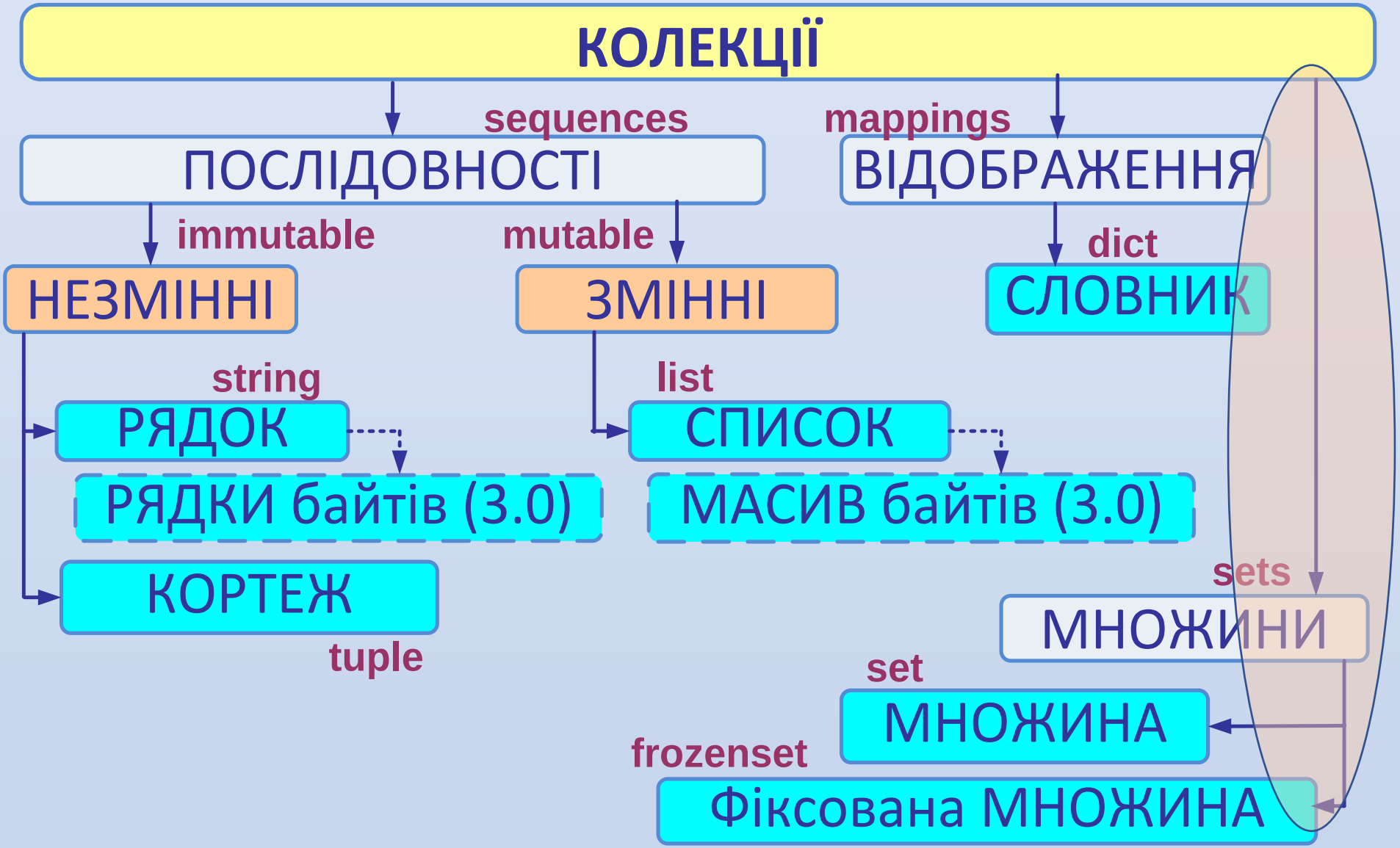
Операція		
Конкатенація	T1+T2	
Дублювання	T * N	N-разів повторення
Вибірка за індексом	T[i]	*i*-й об’єкт кортежу
Зріз	T[start: stop: step]	Новий кортеж = зрізу

Функція	Дія
print(T)	*Друкування елементів кортежу T*
len(T)	*Кількість об'єктів в кортежу T*
X **in** T	*Перевірка входження об'єкту X в кортеж T*
min(T)	*Пошук мінімального елемента кортежу T*
max(T)	*Пошук максимального елемента кортежу T*
sum(T)	*Сума елементів (числових) кортежу T*

КОРТЕЖ. Методи

Метод	Дія
T.**index**(**EL**)	*Індекс елементу* EL *в кортежі* T
T.**count** (**EL**)	*Кількість елементів* EL *в кортежі* T

МНОЖИНА


КОЛЕКЦІЇ
sequences
ПОСЛІДОВНОСТІ
mappings
ВІДОБРАЖЕННЯ
immutable
mutable
dict
НЕЗМІННІ
ЗМІННІ
СЛОВНИК
string
list
РЯДОК
СПИСОК
РЯДКИ байтів (3.0)
МАСИВ байтів (3.0)
КОРТЕЖ
tuple
sets
МНОЖИНИ
set
МНОЖИНА
frozenset
Фіксована МНОЖИНА

МНОЖИНА (> 3.0)

Множина → НЕупорядкована змінна - set (незмінна - frosenset)) колекція унікальних об'єктів довільного типу

Тип	Змінність	Індексованість	Унікальність	Створення
set	+	-	+	*{el1, el2, } set()*
frosen set	-	-	+	*frozenset()*

МНОЖИНА. Створення

Створення множини

	Дія
St={el1 , el2 , , }	Множина елементів
St=set('gsgjsdh')	Множина елементів
St=set([*iterable*])	Множина елементів (ітератор)
St=frozenset([*iterable*])	Фіксована множина елементів (ітератор)

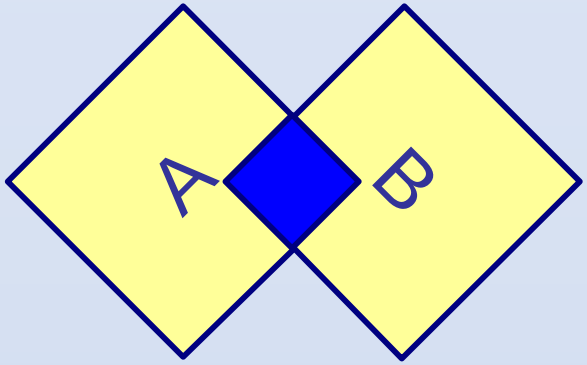
https://docs.python.org/3/library/stdtypes.html?highlight=set#set

МНОЖИНА. Методи

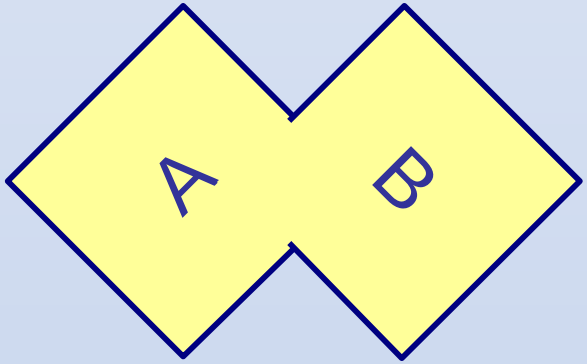
Метод	Дія
St.**copy**() Fst.**copy**()	*Вертає копію множини*
St.**clear**()	*Видалення всіх елементів множини*
St.**add**(el)	*Додає* **el** *до множини*
St.**discard**(el)	*Видаляє* **el,** *якщо він є*
St.**remove**(el)	*Видаляє* **el,** *якщо відсутній KeyError*
St.**pop**()	*Видаляє перший елемент та вертає його*

Метод	Дія
set_a.**isdisjoint**(set_b)	***Set_A*** *неперерізний з* ***Set_B ???***
set_a.**issubset**(set_b)	***Set_A*** *підмножена* ***Set_B ???***
set_a.**issuperset**(set_b)	***Set_A*** *надмножена* ***Set_B ???***

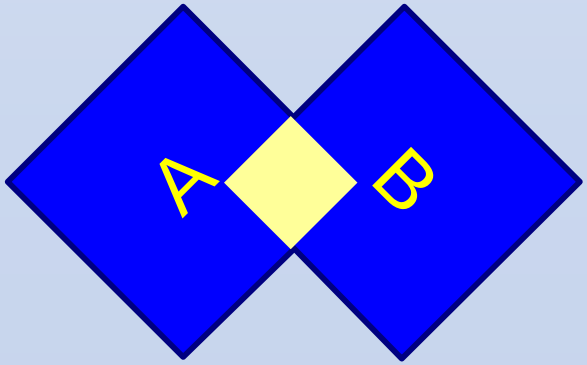
МНОЖИНИ. Методи



Метод
set_c=set_a.**intersection**(set_b) set_c=set_a **&** set_b

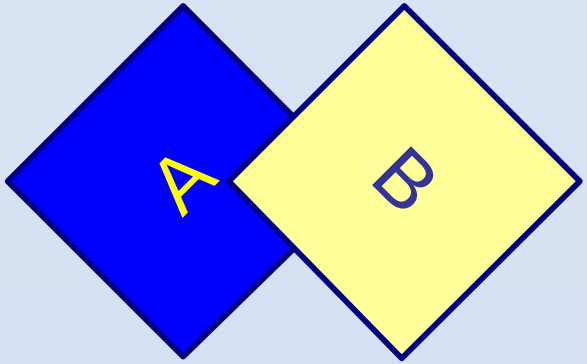


Метод
set_c=set_a.**union**(set_b) set_c=set_a **|**set_b

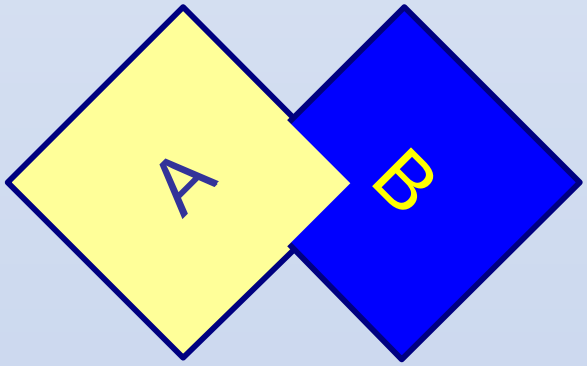


Метод
set_c=set_a.**symmetric_difference** (set_b) set_c=set_a **^** set_b

МНОЖИНИ. Методи



Метод
set_c=set_a.**difference**(set_b)

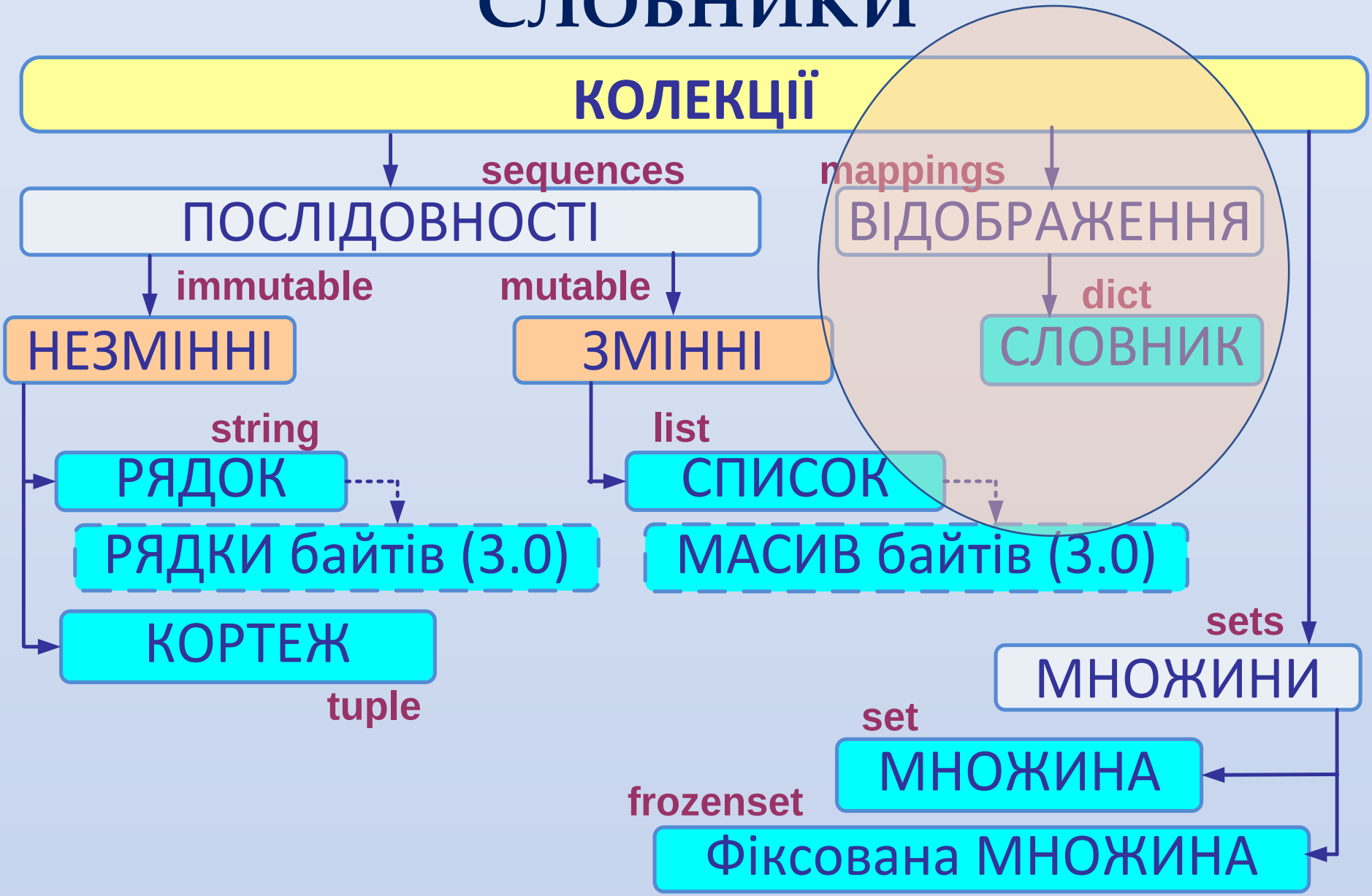


Метод
set_c=set_b.**difference**(set_a)

МНОЖИНИ. Методи

Метод (тільки для set)	Дія
set_a.**union_update**(set_b)	**Змінює set_a**
set_a.**intersection_update** (set_b)	
set_a.**difference_update** (set_b)	
set_a.**symmetric_difference_update** (set_b)	

СЛОВНИКИ


КОЛЕКЦІЇ
sequences
ПОСЛІДОВНОСТІ
mappings
ВІДОБРАЖЕННЯ
immutable
mutable
dict
НЕЗМІННІ
ЗМІННІ
СЛОВНИК
string
list
РЯДОК
СПИСОК
РЯДКИ байтів (3.0)
МАСИВ байтів (3.0)
КОРТЕЖ
tuple
sets
МНОЖИНИ
set
МНОЖИНА
frozenset
Фіксована МНОЖИНА

СЛОВНИКИ

Словник = → Неупорядкована змінна колекція об'єктів довільного типу

Словник забезпечує доступ до елементів за ключем.

В словнику КЛЮЧ це індекс!!!

Змінна кількість елементів.

Довільне число рівнів вкладеності.

Тип	Змінність	Індексованість	Унікальність	Створення
dict	+ елементи - ключі + значення	-	+ елементи + ключі - значення	*{}* *{key: value}* *dict()*

СЛОВНИКИ. Створення

Створення словнику

	Дія
D={}	Пустий словник
D={5:’as’, 6:’is’, 7:’if’}	Словник з трьох елементів
D={’as’:5, ’is’:25, ’if’:’OK’}	Словник з трьох елементів
D=dict(name=‘Piter’, age=35)	Функція створення словнику
D=dict(zip(kyelist, vallist))	Функція створення словнику
D={5:’as’,6:{’as’:5, ’is’:25} 7:’if’}	Словник з вкладеним словником

СЛОВНИКИ. Базові операції/функції

Операція		
Вибірка ключем	D[key]	*key*-й об'єкт словнику
Вибірка ключем	D[6][‘is’]	Вибірка з вбудованого словника

Функція	Дія
print(D)	*Друкування елементів словнику*
len(D)	*Кількість об'єктів в словнику*
key **in** D	*Перевірка на входження об'єкту з ключем* ***key*** *в словник* D

СЛОВНИКИ. Методи

Метод	Дія
D[key] =	*Додавання ключа + значення*
D.**items**() D.**keys**() D.**values**()	*Список ключів та значень* *Список ключів* *Список значень*
D.**copy**()	*Копіювання словника*
D.**get**(key[,default])	*Витяг за ключем, якщо ключа немає вертається* ***default*** *or* ***None***
D.**update**(D2)	*Оновлення словника додавання пар з D2*
D.**pop**(key)	*Повернення значення та видалення*

Рекомендована ЛІТЕРАТУРА

- **Програмування числових методів мовою Python:** підруч. / А. В. Анісімов, А. Ю. Дорошенко, С. Д. Погорілий, Я. Ю. Дорогий ; за ред. А. В. Анісімова. – К. : Видавничо-поліграфічний центр "Київський університет", 2014. – 640 с.
- **Програмування числових методів мовою Python:** навч. посіб. / А. Ю. Дорошенко, С. Д. Погорілий, Я. Ю. Дорогий, Є. В. Глушко ; за ред. А. В. Анісімова. – К. : Видавничо-поліграфічний центр "Київський університет", 2013. – 463 с.
- **Основи програмування Python:** Підручник для студ. спеціальності 122 «Компютерні науки» / А.В.Яковенко; КПІ.- Київ: КПІ, 2018 . – 195 с.
- **Лутц М.** Изучаем Python, 4-е издание. - СПб.: Символ-Плюс. 2011.- 1280 с.: ил.

Контрольні запитання

- Надайте визначення колекції в мові Python, наведіть перелік вбудованих типів колекцій, вкажіть базові властивості колекцій.
- Надайте визначення зрізу для послідовностей, наведіть приклади формування зрізів.
- Надайте перелік основних операцій із рядками, вкажіть їх призначення та наведіть відповідні приклади.
- Надайте перелік основних функцій об'єктів типу рядок, вкажіть їх призначення та наведіть відповідні приклади.
- Надайте перелік основних методів об'єктів типу рядок, вкажіть їх призначення та наведіть відповідні приклади.
- Вкажіть способи форматування рядків, наведіть відповідні приклади.

Контрольні запитання

- Надайте визначення **кортежу** в мові Python, вкажіть властивості кортежу, варіанти створення кортежу. Наведіть приклади.
- Надайте перелік основних операцій із **кортежами**, вкажіть їх призначення та наведіть відповідні приклади.
- Надайте перелік основних функцій об'єктів типу **кортеж**, вкажіть їх призначення та наведіть відповідні приклади.
- Надайте перелік основних методів об'єктів типу **кортеж**, вкажіть їх призначення та наведіть відповідні приклади.

Контрольні запитання

- Надайте визначення **множини** в мові Python, вкажіть властивості множини, варіанти створення множини. Наведіть приклади.
- Надайте перелік основних операцій із **множиною**, вкажіть їх призначення та наведіть відповідні приклади.
- Надайте перелік основних функцій об'єктів **множина**, вкажіть їх призначення та наведіть відповідні приклади.
- Надайте перелік основних методів об'єктів типу **множина**, вкажіть їх призначення та наведіть відповідні приклади.

Контрольні запитання

- Надайте визначення **списку** в мові Python, вкажіть властивості списку, варіанти створення списку. Наведіть приклади.
- Надайте перелік основних **операцій** із **списками**, вкажіть їх призначення та наведіть відповідні приклади.
- Надайте перелік основних **функцій** об'єктів типу **список**, вкажіть їх призначення та наведіть відповідні приклади.
- Надайте перелік основних **методів** об'єктів типу **список**, вкажіть їх призначення та наведіть відповідні приклади.

Контрольні запитання

- Надайте визначення **словника** в мові Python, вкажіть властивості словника, варіанти створення словника. Наведіть приклади.
- Надайте перелік основних операцій із **словниками**, вкажіть їх призначення та наведіть відповідні приклади.
- Надайте перелік основних **функцій** об'єктів типу **словник**, вкажіть їх призначення та наведіть відповідні приклади.
- Надайте перелік основних **методів** об'єктів типу **словник**, вкажіть їх призначення та наведіть відповідні приклади.

The END
Модуль 6. Лекція 6.2.