

СУЧАСНІ ТЕХНОЛОГІЇ ПРОГРАМУВАННЯ

**122 «Комп'ютерні науки»
КН-18**

2019 / 2020 навчальний рік

PYTHON #2

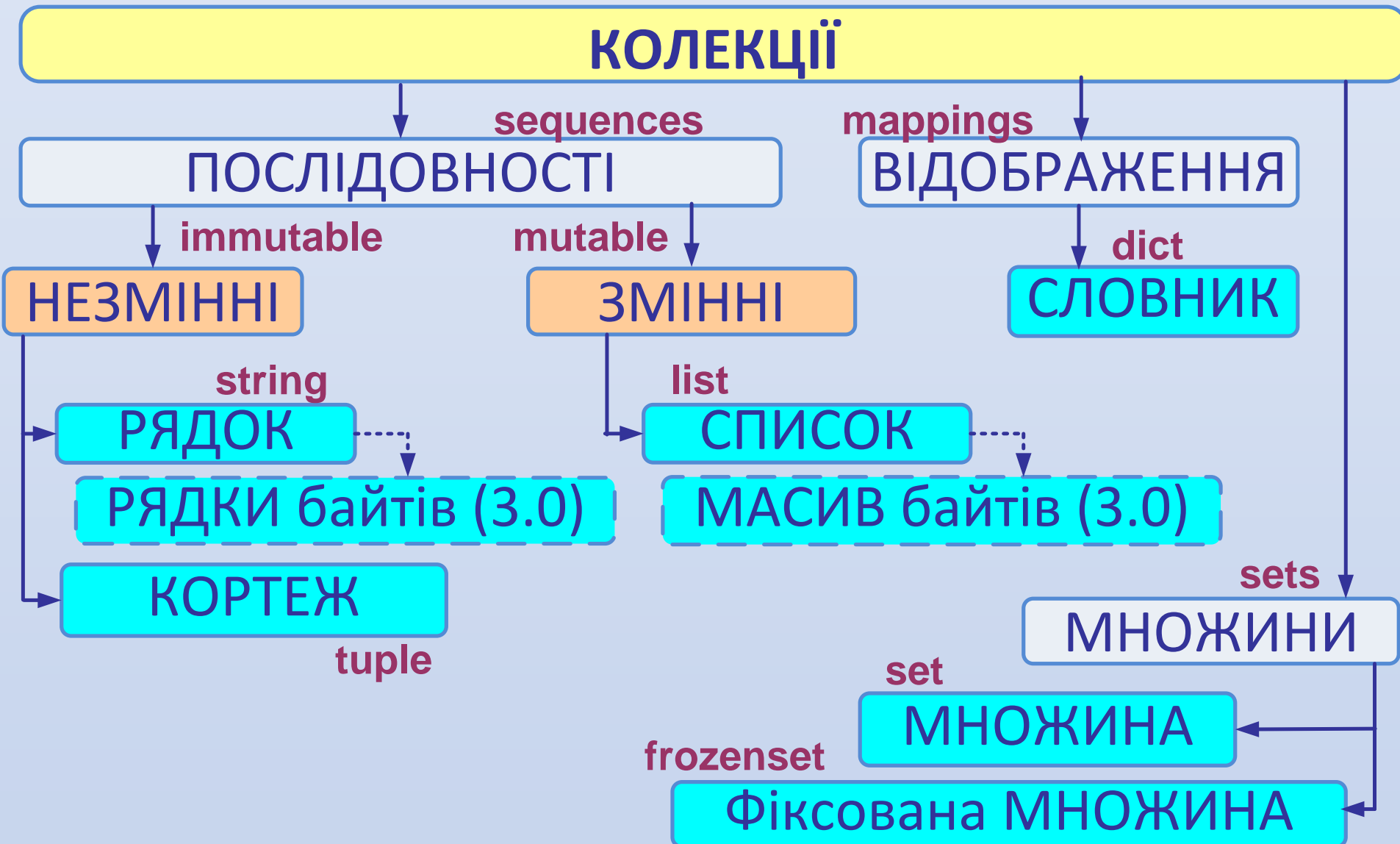
1. Колекції: загальні відомості
2. Рядки (string)

https://github.com/eabshkvprof/2020_Mod_Prog_Tech

КОЛЕКЦІЇ

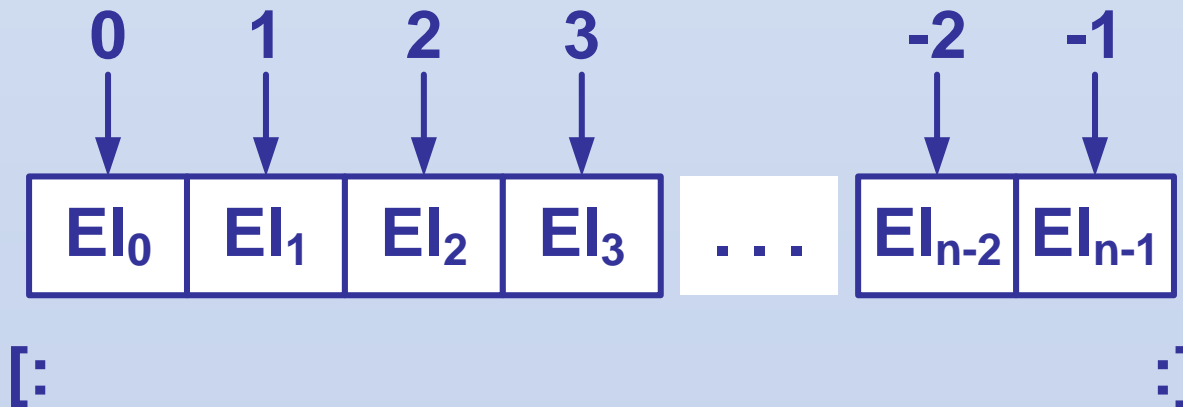
КОЛЕКЦІЯ → програмний об'єкт (змінна-контейнер), що зберігає значення одного або різних типів та дозволяє звертатися до цих значень а також використовувати вбудовані функції і методи.

КОЛЕКЦІЇ



КОЛЕКЦІЇ. 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	+	+	-	<code>[]</code> <code>list()</code>
tuple	-	+	-	<code>()</code> <code>tuple()</code>
string	-	+	-	<code>' '</code> <code>" "</code>
set	+	-	+	<code>{el1, el2, }</code> <code>set()</code>
frozen set	-	-	+	<code>frozenset()</code>
dict	+ елементи - ключі + значення	-	+ елементи + ключі - значення	<code>{}</code> <code>{key: value}</code> <code>dict()</code>

КОЛЕКЦІЇ. Стандартні функції

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

КОЛЕКЦІЇ. Стандартні методи

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

КОЛЕКЦІЇ. Стандартні методи

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

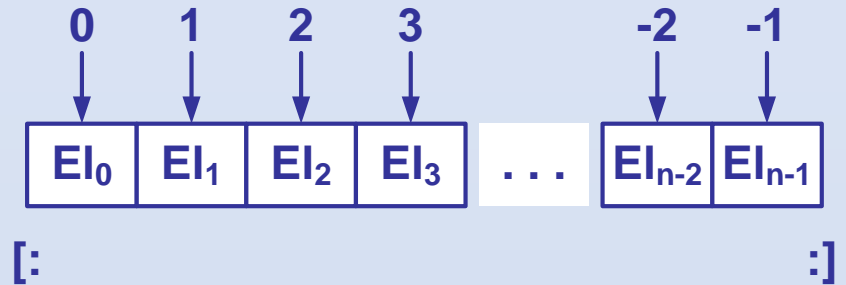
.index () - повертає мінімальний індекс переданого елемента для індексованих колекцій (рядок, список, кортеж)

.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. Індекссування

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



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

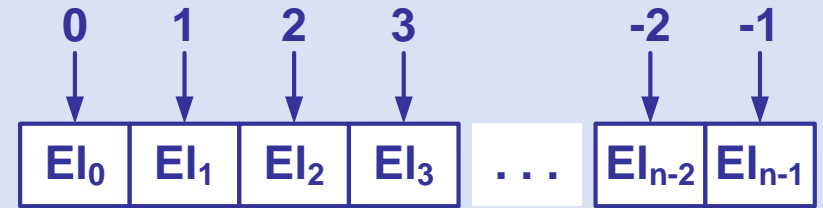
При завданні негативного індексу, останній елемент має індекс -1, передостанній -2 і так далі до першого елемента індекс якого дорівнює значенню довжини колекції з негативним знаком, тобто

-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

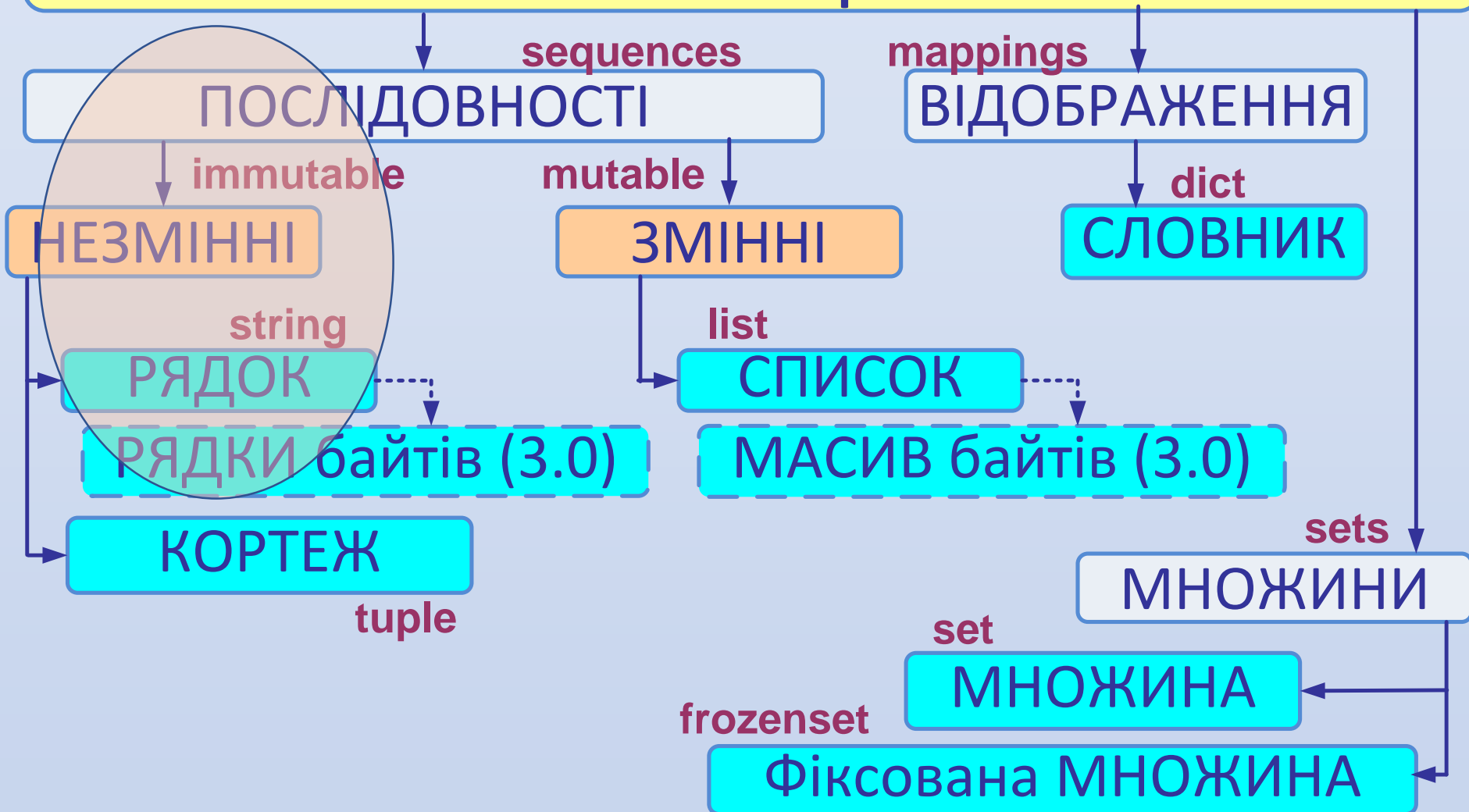
ПРИМІТКА : $[stop]$ не включається в результат

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)	
[:] →	+	+	+	+	+	+	+	ABCDEFGG
[::-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



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 СИВОЛ => СЕЦСИМВОЛ (!!! 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	<code>print()</code>	Друкування рядку
2	<code>len()</code>	Кількість символів в рядку
3	<code>X in S</code>	Перевірка входження елемента підрядку X в рядок S
4	<code>min()</code>	Пошук мінімального символу
5	<code>max()</code>	Пошук максимального символу
6	<code>print()</code>	Друкування рядку

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 признаков (+, -, 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 `{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
```


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.
- **Бейдер Д. Чистый Python. Тонкости программирования для профи.-СПб.: Питер. 2018.-288 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.

The END
Mod 1. Lec 3.