***Форматування рядків у середовищі Python***

***Повторюємо***

Вважають, що програмісти добре знаються в математиці, тому що працюють з числами. Насправді, більшість програмістів працюють з текстовими рядками набагато частіше, ніж з числами.

Будь-яка послідовність символів, укладена в лапки, в Python вважається рядком, при цьому рядки можуть бути укладені як в одинарні, так і в подвійні лапки.

**Рядок (String)** – це складний тип даних, який представляє собою  упорядковані послідовності символів, що використовуються для зберігання і представлення текстової інформації.

***Запитання для повторення:***

а) Що таке рядок та рядковий тип даних? б) Як здійснюється індексація символів у рядку; в) Як можна записати апостроф чи лапки усередині рядкової величини? г) Як вилучити потрібний символ з рядка? д) Що таке зрізи і як їх виконують? е) Як дізнатися довжину рядка? є) Які операції і за допомогою яких методів (функцій) можна виконувати над рядками?

***Підстановка значень у рядки***

Досить часто виникає необхідність поєднувати рядок з числовими даними. Наприклад, провести обчислення, а потім оголосити результат та прокоментувати його (скажімо, задача на обчислення ідеальної маси тіла). Можна, звичайно, перетворити число на текст функцією **str** та виконати конкатенацію рядків або, користуючись print, записати вирази через кому, але це все буде якось громіздко та незручно:

|  |
| --- |
| **m=int(input("Ваша маса в кг: "))**  **h=int(input("Ваш зріст в см: "))**  **h=h/100**  **IMT=int(m/h\*\*2)**  **mesage="Індекс маси тіла становить " + str(IMT) + " балів"**  **print(message)** |

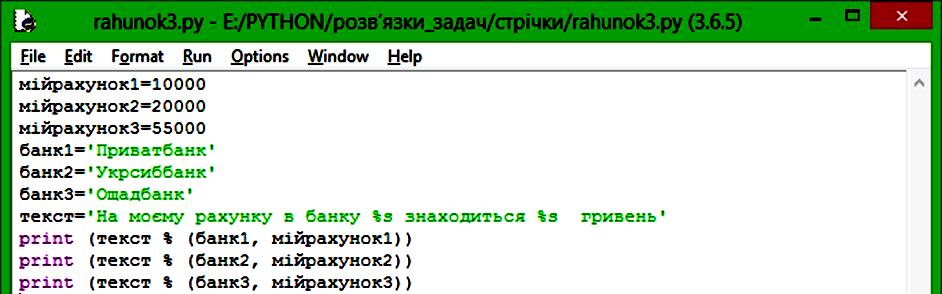
Набагато зручніше використати метод підстановки значень у рядки. Щоб вивести повідомлення, що містить певне значення, застосовується знак **%s** – маркер на позначення величини, яку пізніше можна буде додати у рядок.

***Наприклад:***

**mesage="Індекс маси тіла становить %s балів"**

**print(message % IMT)**

Можна підставляти різні значення на місце тимчасового замінника , користуючись різними змінними.

****Можна використовувати в одному рядку більше одного замінника, при цьому обов’язково брати значення для підстановки у дужки. Порядок значень у дужках збігається з порядком введення їх у рядок.

*Наприклад*, створимо програмний код, який виводитиме на друк інформацію про назву банку та суму рахунку, що зберігається на рахунку вкладника.

|  |
| --- |
| **money1=10000**  **money2=20000**  **money3=55000**  **bank1='Приватбанк'**  **bank2='Укрсиббанк'**  **bank3='Ощадбанк'**  **text="У мене в банку %s є %s гривень"**  **print(text % (bank1, money1))**  **print(text % (bank2, money2))**  **print(text % (bank3, money3))** |

Зверніть увагу – код **text="У мене в банку %s є %s гривень"** є своєрідним «шаблонним» текстом, при виведенні його на друк залежно від параметра послідовності **%s** замінюються на потрібний нам текст. Скажімо, в коді **text % (bank1, money1)** перший маркер **%s** замінюється на фразу «Приватбанк», а другий маркер **%s** – на суму рахунку (10000).

Висновок – у багатьох випадках набагато зручніше користуватися методом підстановки значень у рядки, а не розбивати рядок на окремі фрази з конкатенацією їх в єдине ціле. Так здебільшого і чинять професійні програмісти.

***Форматування рядків. Метод format.***

Іноді (а точніше, доволі часто) виникають ситуації, коли треба зробити рядок, підставивши в нього деякі дані, отримані в процесі виконання програми (запит користувача, дані з файлів і т. д.). Підстановку даних можна зробити за допомогою форматування рядків. Форматування можна зробити за допомогою оператора %, або за допомогою методу format.

Якщо для підстановки потрібно тільки один аргумент, то значення – це сам аргумент:

*Зверніть увагу* – замість фігурних дужок **{}** у текст підставляється фраза, записана у дужках після методу **format**, а саме: 'Vasya'.

Можна використовувати і кілька аргументів. Скажімо, попередній варіант «банківської» вправи можна записати таким чином:

**print("У мене в банку {} є {} гривень".format('Приватбанк', 10000))**

Як бачимо, подібний запис є досить зручним і зрозумілим, а тому широко використовується на практиці.

*Зверніть увагу* – підстановка даних ведеться у тому порядку, в якому записано аргументи. Якщо послідовність з якихось причин треба змінити, застосовують нумерацію (нумерація ведеться від 0).

**print("У мене в банку {0} є {1} гривень".format('Приватбанк', 10000))**

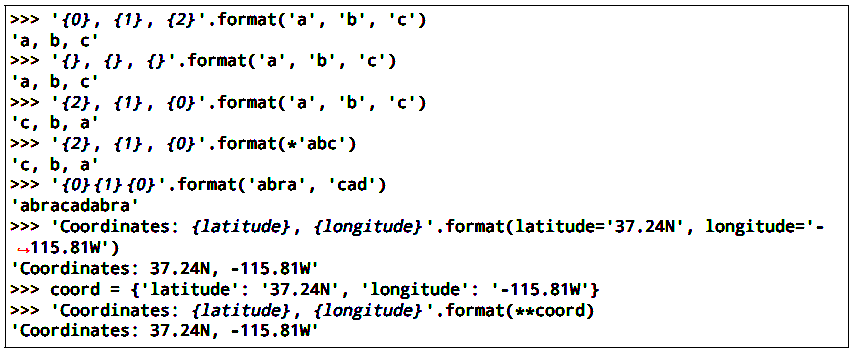
Якщо послідовність не порушено, нумерацію можна і не застосовувати.

Більше того, замість числових індексів можна застосовувати імена:

**print("У мене в банку {bank} є {money} гривень".format(bank='Приватбанк', money=10000))**

Використання подібних імен-синонімів робить програмний код ще більш зручним та читабельним.

Ось декілька прикладів використання:



***Для допитливих*** – у більш пізніх версіях мови Python (3.6 і т. д.) використовують так звані *стрічки*. Їх використання дозволяє, наприклад, виконувати математичні дії безпосередньо усередині текстового рядка.

**m=int(input('Введіть масу тіла: '))**

**g=9.8**

**print(f'Вага тіла становить P = {m\*g} ньютон')**

Ознакою стрічки є наявність перед рядком літери **f**. Запис **{m\*g}** означає, що виконується математична дія (множення величин), а результат додається до рядка.

З використанням методу **format** подібний запис виглядав би так:

**m=int(input('Введіть масу тіла: '))**

**g=9.8**

**P=m\*g**

**print('Вага тіла становить P = {} ньютон'.format(P))**

Як бачимо, використання стрічок робить запис програмного коду ще більш компактним. Але, нагадаємо, старі версії мови (наприклад, Python 4.3.4) подібну можливість не підтримують, а тому варто бути уважним, поширюючи свій програмний код серед користувачів.

**Практичне завдання.**

1. Створіть і збережіть пустий файл для подальшого виконання.
2. Створіть код-програму, яка, використовуючи функцію **input** та метод **format**, вестиме з користувачем діалог, наприклад:

>>> Як тебе звати?

>>> Вася

>>>Привіт, Вася!

1. Аналогічно до «банківської» задачі, створіть програмний код, який виводитиме на екран інформацію про абонента, що користується послугами різних операторів. Фраза може виглядати, наприклад, так:

**За номером +380984749994 у абонента Петренко І. І. на рахунку знаходиться 45 грн.**

Використайте змінні **abonent, number** та **money** для зберігання відповідних даних.

Що використати – маркер **%s**, метод **format** або f-стрічки – вирішіть самі.

1. *Модифікуйте* програму таким чином, щоб вказані дані можна було вводити з клавіатури (за допомогою функції **input**).
2. Виконайте команди використання методу format, наведені вище, прокоментуйте отримані результати та зробіть відповідні висновки.
3. (*високий рівень*): використовуючи метод форматування рядків, створіть програму, яка отримуватиме масу та зріст користувача і за ними виводитиме на екран інформацію про індекс маси тіла (алгоритм розрахунку цього параметра наведено на попередній сторінці). Відповідь програми може бути такою:

**За маси 90 кг та зросту 182 см ваш індекс маси тіла становить 27 балів.**

***Контрольні запитання:***

а) Що таке метод форматування рядків, де і для чого він застосовується? б) Які способи форматування рядків ви знаєте? Які їхні переваги та недоліки? в) Які переваги надає використання f-стрічок і про що слід пам’ятати розробнику, поширюючи свій код? г) Зробіть висновок про те, чому форматування рядків має таке широке використання серед професійних програмістів.