

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

МЕТОДИЧНІ ВКАЗІВКИ

до лабораторних робіт з дисципліни «Основи програмування на Python»

для бакалаврів спеціальності 123 Комп'ютерна інженерія
денної форми навчання

Частина IV

2024

Методичні вказівки до лабораторних робіт з дисципліни «Основи програмування на Python» для бакалаврів спеціальності 123 Комп'ютерна інженерія, денної форми навчання. Частина IV / Укл.: А.В. Тіменко, Н.А. Куликовська – Запоріжжя: НУ «Запорізька політехніка», 2024. – 20 с.

Укладач: А.В. Тіменко, старший викладач
Н.А. Куликовська, старший викладач

Рецензент: М.Ю. Тягунова, к.т.н., доцент

Відповідальний
за випуск: А.В. Тіменко, старший викладач

Затверджено:
На засіданні кафедри
«Комп'ютерні системи та мережі»
Протокол № 7
від 22 березня 2024р.

Рекомендовано до видання
НМК факультету КНТ
Протокол № 8
від 27 березня 2024 р.

ЗМІСТ

1	ЛАБОРАТОРНА РОБОТА №9 ВИКОРИСТАННЯ КЛАСІВ	4
1.1	Теоретичні відомості	4
1.2	Хід роботи.....	6
1.3	Зміст звіту	9
1.4	Контрольні питання	9
2	ЛАБОРАТОРНА РОБОТА №10 РОБОТА З ДАНИМИ	10
2.1	Теоретичні відомості.....	10
2.2	Хід роботи	12
2.3	Зміст звіту.....	14
2.4	Контрольні запитання	14
3	ЛАБОРАТОРНА РОБОТА №11 ВИКОРИСТАННЯМ ФУНКЦІЙ ДОСТУПУ ДО МЕРЕЖІ ІНТЕРНЕТ.....	16
3.1	Теоретичні відомості.....	16
3.2	Завдання до виконання.....	18
3.3	Зміст звіту.....	18
3.4	Контрольні запитання	19
	ЛІТЕРАТУРА.....	20

1 ЛАБОРАТОРНА РОБОТА №9 ВИКОРИСТАННЯ КЛАСІВ

Мета роботи: отримати навички програмування на мові Python з використанням класів.

1.1 Теоретичні відомості

Об'єктно-орієнтоване програмування (ООП) вважається однією з найефективніших методологій створення програмних застосунків.

У об'єктно-орієнтованому програмуванні використовують класи, що описують реально існуючі предмети і ситуації, а потім створюють **об'єкти** на основі цих описів. При написанні класу визначається **загальна поведінка** для цілої категорії об'єктів.

При створенні конкретних об'єктів (**екземплярів**) на основі класів, кожен об'єкт автоматично наділяється загальною поведінкою; після цього можна наділити кожен об'єкт **унікальними особливостями** на свій вибір.

Об'єктно-орієнтовані мови, такі як C++, Java і Python, використовують **класи** (для визначення стану, який може зберігатися в об'єкті), і **методи** (функції для зміни цього стану).

В Python майже все є об'єктом. Рядки, словники, функції, файли і цілі числа - все це об'єкти. Є й винятки: ключові слова (наприклад, **in**) об'єктами не є. Крім того, імена змінних об'єктами також не є, але змінні вказують на об'єкти.

Ви можете написати `num = 20`, щоб створити об'єкт типу **int** зі значенням 20, і присвоїти посилання на нього змінній `num`. І тепер, цілочисельний об'єкт зі значенням 20 може використовувати **методи** на зразок **додавання** і **множення**. Число 9 - це вже інший цілочисельний об'єкт. Це означає, що існує клас **Integer**, якому належать об'єкти 9 і 20, або інакше, 9 і 20 - це екземпляри класу **Integer**.

Наприклад, **String** є вбудованим класом Python, який створює рядкові об'єкти-екземпляри на зразок 'Nairobi' і 'Singapore'. Рядки 'Nairobi' і 'Singapore' мають **методи**, наприклад, **capitalize()** і **replace()**.

Python має багато інших вбудованих класів, що дозволяють створювати інші стандартні типи даних, включаючи списки, словники тощо.

Зрозумілою мовою, об'єкти можна вважати іменниками, а їх методи - дієсловами. Об'єкт представляє окрему річ, а його методи визначають, як вона взаємодіє з іншими речами.

Клас може бути корисний для опису фізичних або концептуальних об'єктів.

Поняття, що є описовими, - прискорення, відстань, температура, середнє арифметичне, колір - не є хорошими кандидатами для класів.

Клас повідомляє Python, як створити об'єкт, який представляє собаку. Після того як клас буде написаний, ми використовуємо його для створення **екземплярів**, кожен з яких представлятиме одну конкретну собаку.

В Python імена класів мають починатися з великої літери і не можуть починатися з цифр. Зазвичай, в якості імен використовуються іменники.

Якщо імена класів складаються з кількох слів, то вони записуються у **«ВерблюжомуРегістрі»**. На відміну від функцій, в іменах яких слова з'єднуються символами підкреслення, у **«ВерблюжомуРегістрі»** кожне слово починається з великої літери, а самі слова з'єднуються без розділювачів.

У кожному екземплярі, створеному на основі класу Dog, буде зберігатися кличка і вік; крім того, у ньому будуть присутні методи sit() і roll_over():

```
class Dog():
    """Проста модель собаки."""
    def __init__(self, name, age):
        """Ініціалізація атрибутів name і age."""
        self.name = name
        self.age = age

    def sit(self):
        """Собака сідає по команді."""
        print(self.name.title() + " is now sitting.")

    def roll_over(self):
        """Собака перекочується по команді."""
```

```
print(self.name.title() + " rolled over!")
```

Можна вважати, що клас - це, свого роду, інструкція по створенню екземплярів. Відповідно, клас Dog - інструкція по створенню екземплярів, які представляють конкретних собак.

Приклад екземпляр, який представляє конкретну собаку, використовуючи клас Dog:

```
my_dog = Dog('jessie', 5)
print("My dog's name is " + my_dog.name.title() + ".")
print("My dog is " + str(my_dog.age) + " years old.")
```

1.2Хід роботи

Завдання 1. Виконайте в інтерактивному інтерпретаторі тренувальні вправи (завдання не обов'язкові):

1. У клас SportResults додайте змінну points зі значенням 25 і виведіть на екран значення атрибуту points для екземпляру diving. А потім надайте значення 150 для points. Знову виведіть на екран значення атрибуту points для екземпляру diving. Чи потрібно було створювати новий екземпляр класу, щоб зробити повторне виведення?

2. Створіть клас, який називається Element, з методом __init__, що має атрибути екземпляру name, symbol і number. Створіть екземпляр цього класу el зі значеннями 'Silicium', 'Si' і 14 і виведіть на екран значення його атрибутів.

3. Створіть словник з наступними ключами і значеннями: 'name': 'Argentum', 'symbol': 'Ag', 'number': 47. Далі створіть екземпляр з ім'ям argentum класу Element за допомогою цього словника і виведіть значення усіх атрибутів.

4. Для класу Element визначте метод з ім'ям dump(), який виводить на екран значення атрибутів екземпляру (name, symbol і number). Створіть екземпляр argentum з цього нового визначення і використайте метод dump(), щоб вивести на екран його атрибути.

5. Визначте три класи: Switzerland, England і Japan. Для кожного з них визначте всього один метод - currency(). Метод повинен повертати значення 'franc' (для Switzerland), 'pound' (для England) або 'yen' (для Japan). Створіть по одному екземпляру кожного класу і виведіть на екран грошову одиницю кожної країни.

6. Визначте три класи: Printer, Lamp і Car. Кожен з них має тільки один метод does(). Він повертає значення 'print' (для Printer), 'glow' (для

Lamp) або 'ride' (для Car). Далі визначте клас Robot з методом `__init__`, який містить по одному екземпляру кожного з цих класів як власні атрибути. Визначте метод `do_it()` для класу Robot, який виводить на екран усі дії, що роблять його компоненти.

Завдання 2. Напишіть програму згідно варіанту (1 або 2) у середовищі програмування для розв'язування таких завдань:

а) Онлайн-магазин.

- 1) Створіть клас з ім'ям Shop(). Метод `__init__()` класу Shop() повинен містити два атрибути: `shop_name` і `store_type`. Створіть метод `describe_shop()`, який виводить два атрибути, і метод `open_shop()`, який виводить повідомлення про те, що онлайн-магазин відкритий. Створіть на основі класу екземпляр з ім'ям store. Виведіть два атрибути окремо, потім викличте обидва методи.
- 2) Створіть три різних екземпляри класу, викличте для кожного екземпляру метод `describe_shop()`.
- 3) Додайте атрибут `number_of_units` зі значенням за замовчуванням 0; він представляє кількість видів товару у магазині. Створіть екземпляр з ім'ям store. Виведіть значення `number_of_units`, а потім змініть `number_of_units` і виведіть знову.
- 4) Додайте метод з ім'ям `set_number_of_units()`, що дозволяє задати кількість видів товару. Викличте метод з новим числом, знову виведіть значення. Додайте метод з ім'ям `increment_number_of_units()`, який збільшує кількість видів товару на задану величину. Викличте цей метод.
- 5) Напишіть клас Discount(), що успадковує від класу Shop(). Додайте атрибут з ім'ям `discount_products` для зберігання списку товарів, на які встановлена знижка. Напишіть метод `get_discounts_products`, який виводить цей список. Створіть екземпляр `store_discount` і викличте цей метод.
- 6) Збережіть код класу Shop() у модулі. Створіть окремий файл, що імпортує клас Shop(). Створіть екземпляр `all_store` і викличте один з методів Shop(), щоб перевірити, що команда **import** працює правильно.

б) Облік користувачів на сайті.

- 1) Створіть клас з ім'ям `User`. Створіть два атрибути `first_name` і `last_name`, а потім ще кілька атрибутів, які зазвичай зберігаються у профілі користувача. Напишіть метод `describe_user` який виводить повне ім'я користувача. Створіть ще один метод `greeting_user()` для виведення персонального вітання для користувача. Створіть кілька примірників, які представляють різних користувачів. Викличте обидва методи для кожного користувача.
- 2) Додайте атрибут `login_attempts` у клас `User`. Напишіть метод `increment_login_attempts()`, що збільшує значення `login_attempts` на 1. Напишіть інший метод з ім'ям `reset_login_attempts()`, обнуляє значення `login_attempts`. Створіть екземпляр класу `User` і викличте `increment_login_attempts()` кілька разів. Виведіть значення `login_attempts`, щоб переконатися у тому, що значення було змінено правильно, а потім викличте `reset_login_attempts()`. Знову виведіть `login_attempts` і переконайтеся у тому, що значення обнулилося.
- 3) Адміністратор - користувач з повними адміністративними привілеями. Напишіть клас з ім'ям `Admin`, що успадковує від класу `User`. Додайте атрибут `privileges` для зберігання списку рядків виду «Allowed to add message», «Allowed to delete users», «Allowed to ban users» і т. д. Напишіть метод `show_privileges()` для виведення набору привілеїв адміністратора. Створіть екземпляр `Admin` і викличте метод.
- 4) Напишіть клас `Privileges`. Клас повинен містити всього один атрибут `privileges` зі списком, який треба забрати із класу `Admin`. Водночас, необхідно перемістити метод `show_privileges()` у клас `Privileges` із класу `Admin`. Створіть екземпляр `priv` як атрибут класу `Admin`. Створіть новий екземпляр `admin` і використайте метод для виведення списку привілеїв.
- 5) Збережіть клас `User` в одному модулі, а класи `Privileges` і `Admin` у іншому модулі. В окремому файлі створіть екземпляр `admin` і викличте метод `show_privileges()`, щоб перевірити, що все працює правильно.

1.3 Міст звіту

Звіт повинен мати такі складові:

- титульний лист;
- мета роботи;
- результат виконання завдань;
- відповіді на контрольні питання.

1.4 Контрольні питання

Дайте пояснення таких понять у Python:

1. Об'єкт;
2. Клас;
3. Екземпляр класу;
4. Метод класу;
5. Атрибут класу;
6. Наслідування;
7. Перевизначення методу;
8. Екземпляри як атрибути.

2 ЛАБОРАТОРНА РОБОТА №10 РОБОТА З ДАНИМИ

Мета роботи: отримати навички програмування на мові Python з використанням даних.

2.1 Теоретичні відомості

У Python вбудовані функції для роботи з символами і їхніми номерами в Unicode:

– **ord()** - якщо рядок представляє один символ Unicode, функція повертає ціле число - код Unicode цього символу. Наприклад, `ord('a')` повертає ціле число 97 та `ord('€')` (символ Євро) повертає 8364.

– **chr()** - повертає рядок у вигляді символу, за його кодом в Unicode. Наприклад, `chr(97)` повертає рядок 'a', а `chr(8364)` повертає рядок '€'.

Кожен із символів у форматі Unicode має унікальне ім'я та ідентифікаційний номер (код). Якщо ви знаєте Unicode ID або назву символу, то можете використовувати його у рядку Python згідно таких правил:

а) Символ `\u`, за яким розташовуються **чотири** шістнадцяткові числа (числа шістнадцяткової системи числення, що містять символи від 0 до 9 і від A до F), визначають символ, що знаходиться в одній із багатомовних площин Unicode .

1) Перші два числа є номером площини (від 00 до FF), а наступні два - кодом символу всередині площини.

2) Площина з номером 00 - це старий формат ASCII, і позиції символів у ньому такі ж, як і у ASCII.

Отже, якщо код символу має 4 цифри і менше, необхідно використовувати малу літеру `\u` і доповнювати код зліва нулями до 4 цифр.

Наприклад, код символу `@` має шістнадцяткове значення 40 у таблиці символів Unicode, тому необхідно дописати два 0 попереду, щоб кількість цифр стала рівною 4, і для використання у Python використовувати символ `\u`:

```
at = '\u0040'
```

```
print(at)
```

Результат:

@

б) Для символів вищих площин потрібно більше бітів.

- 1) Керуюча послідовність для них виглядає як \U, за яким слідує **вісім** шістнадцяткових символів.
- 2) Крайній зліва з них має дорівнювати 0.

Отже, якщо код символу містить більше 4 цифр, необхідно використовувати велику літеру U і доповнити код символу зліва нулями до 8 цифр. Наприклад, код символу 🙋 має шістнадцяткове значення 1F617 у таблиці символів Unicode, тому необхідно дописати три 0 попереду, щоб кількість цифр стала рівною 8, і для використання у Python використовувати символ \U:

```
kissing_face = '\U0001F617'
```

```
print(kissing_face)
```

Результат:

🙋

в) Для усіх символів конструкція \N{character_name} дозволяє вказати символ за допомогою стандартного імені символу.

У Python входить модуль **unicodedata**, що містить функції, які перетворюють символи в обох напрямках:

– **lookup()** приймає незалежне від регістру ім'я і повертає символ Unicode;

– **name()** приймає символ Unicode і повертає його ім'я у верхньому регістрі.

У наступному прикладі напишемо функцію для перевірки, яка приймає рядок із символом Unicode, шукає його ім'я, а потім шукає символ, який відповідає отриманому імені (результат повинен збігтися з оригінальним). Спробуємо перевірити літеру і розділовий знак у форматі ASCII:

```
import unicodedata
```

```
def unicode_test(value):
```

```
    name_unicode = unicodedata.name(value)
```

```
    other_value = unicodedata.lookup(name_unicode)
```

```
    return 'value="{0:s}",          name_unicode="{1:s}",
```

```
other_value="{2:s}"'.format(value, name_unicode, other_value)
```

Результати перевірки будуть наступними:

```

    value="P", name_unicode="LATIN CAPITAL LETTER P",
other_value="P"
    value="!", name_unicode="EXCLAMATION MARK",
other_value="!"

```

А тепер застосуємо функцію до кількох рядків, що містять символи у форматі Unicode (символи валютних знаків '\u00a2', '\u20ac' і '\u20B4'):

```

print(unicode_test("\u00a2'))
print(unicode_test("\u20ac'))
print(unicode_test("\u20B4'))

```

Результати роботи функції є наступними:

```

value="¢", name_unicode="CENT SIGN", other_value="¢"
value="€", name_unicode="EURO SIGN", other_value="€"
value="¤", name_unicode="HRVYNIA SIGN", other_value="¤"

```

Проблемою, з якою можна зіткнутися, працюючи із символами Unicode, є обмеження, які накладаються шрифтом. У жодному шрифті немає символів для усіх символів Unicode.

2.2 Хід роботи

Завдання 1. Виконайте в інтерактивному інтерпретаторі тренувальні вправи (завдання не обов'язкові):

1. Створіть рядок Unicode з ім'ям `unknown` і надайте йому значення '\u2653'. Виведіть на екран значення рядка `unknown`. Знайдіть ім'я Unicode для `unknown`.

2. Створіть рядок Unicode з ім'ям `unknown2` і надайте йому значення '\U000057A6'. Виведіть на екран значення рядка `unknown2`. Знайдіть ім'я Unicode для `unknown2`.

3. Закодуйте рядок `unknown` з використанням кодування UTF-8 у змінну типу `bytes` з ім'ям `unknown_bytes`. Виведіть на екран значення змінної `unknown_bytes`.

4. Використовуючи кодування UTF-8, декодуйте змінну `unknown_bytes` у рядок `unknown_string`. Виведіть на екран значення змінної `unknown_string`. Чи однакові значення мають змінні `unknown_string` і `unknown`?

5. Продовжіть і надрукуйте рядок `I'm ...`, використовуючи стиль форматування за допомогою функції `format()`. Підставте рядки 'hungry', 'in a good mood', 'looking forward to it', 'worried' і 'thirsty' у цей рядок.

6. Запишіть наступний лист за формою за допомогою стилю форматування з використанням функції `format()` (або за допомогою `f`-рядків). Збережіть рядок з ім'ям `letter`. Створіть словник з ім'ям `response`, що має значення для ключів: `'addressee'`, `'amount'`, `'product'`, `sender`, `'post'`, `'institution'`. Виведіть на екран значення змінної `letter`, у яку підставлені значення зі словника `response`.

Dear Mr. {addressee}:

With reference to our telephone conversation today, I am writing to confirm your order for: {amount} x {product} (Ref. No. 856).

Please contact us again if we can help in any way.

Yours sincerely,

{sender}

{post} of {institution}

Завдання 2. Напишіть програми у середовищі програмування для розв'язування таких завдань відповідно варіанту:

1. Використайте регулярні вирази для пошуку рядків у тексті. Текст, у якому відбуватиметься пошук, можна скопіювати із сайту Project Gutenberg's і зберегти у файл. Імпортуйте модуль `re`, щоб використовувати функції регулярних виразів у Python. Вивести на екран усі слова, які починаються з літери «с», чотирилітерні слова, які починаються з літери `s`, слова, які закінчуються на букву `г`, слова, які містять чотири літери `time` поспіль.

2. Виведіть інформацію про однопиксельний прозорий GIF-файл. Використайте функцію `unhexlify()` для того, щоб перетворити шістнадцятковий рядок у змінну типу `bytes` з ім'ям `gif`. Байти, що містяться у змінній `gif`, визначають однопиксельний прозорий GIF-файл. Коректний файл формату GIF починається з рядка `GIF89a`. Чи є цей файл коректним? Ширина файла формату GIF є шістнадцятибітним цілим числом з прямим порядком байтів (`little-endian`, специфікація GIF), яке починається зі зсуву 6 байт. Його висота має такий же розмір і починається зі зсуву 8 байт. Отримайте і виведіть на екран ці значення для змінної `gif`. Чи рівні вони 1?

```
'4749463839610100010080000000000000ffff21f90401000000002c0
00000000100010000020144003b'
```

3. Запишіть у текстовий файл з ім'ям `validate.txt` інформацію про ширину і висоту зображення, отримані із GIF-файла (на сторінці

розміри файла зменшені, відкрийте зображення у новій вкладці браузера для відображення реальних розмірів). Скористайтеся відомостями із попереднього завдання.

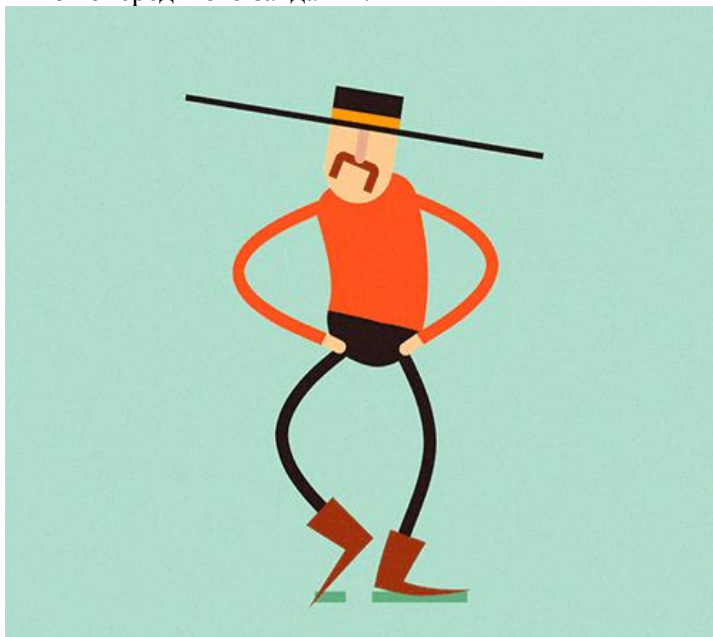


Рисунок 2.1 – Приклад зображення

2.3 Зміст звіту

Звіт повинен мати такі складові:

- титульний лист;
- мета роботи;
- результат виконання завдань;
- відповіді на контрольні питання.

2.4 Контрольні запитання

1. Що таке «система кодування»?
2. Опишіть характеристики форматів ASCII і Unicode.
3. Що таке UTF-8?
4. Для чого використовуються «регулярні вирази»?

5. Що називають «бінарними даними»? Які типи у Python призначені для роботи з бінарними даними?

6. Які інструменти для аналізу даних присутні в Python? Для чого використовується Jupyter Notebook?.

3 ЛАБОРАТОРНА РОБОТА №11 ВИКОРИСТАННЯМ ФУНКЦІЙ ДОСТУПУ ДО МЕРЕЖІ ІНТЕРНЕТ

Мета роботи: отримати навички програмування на мові Python з використанням доступу до мережі Інтернет.

3.1 Теоретичні відомості

Python 3 у власній стандартній бібліотеці містить такі модулі для роботи в Інтернеті:

а) **http** - керує всіма особливостями клієнт-серверної взаємодії за протоколом HTTP:

- 1) **client** виконує всю роботу на стороні клієнта.
- 2) **server** допомагає написати веб-сервер.
- 3) **cookies** і **cookiejar** керують cookies .

б) **urllib** працює на базі **http**:

- 1) **request** - обробляє клієнтські запити.
- 2) **response** - обробляє відповіді сервера.
- 3) **parse** - розбиває URL на частини.

Скористаємося модулем **urllib** стандартної бібліотеки Python, щоб отримати певну інформацію із сайту. URL у наступному прикладі повертає код веб-сторінки в Інтернеті у вигляді HTML (разом з CSS і JavaScript):

```
import urllib.request as ur
url = 'http://openweathermap.org'
conn = ur.urlopen(url)
data = conn.read()
print(conn.status)
print(conn.getheader('Content-Type'))
print(data)
```

Цей фрагмент коду відкрив з'єднання з віддаленим сервером, створив запит HTTP і отримав HTTP-відповідь. Відповідь містить не лише сам вміст веб-сторінки. Одна з найбільш важливих частин відповіді - це **код статусу** HTTP: на екран виводиться значення 200. Це означає, що з'єднання пройшло успішно і отримано відповідь:

200

```
text/html; charset=UTF-8
b'<!DOCTYPE html>\n<html lang=\'en\'>\n...
```

```
...
...</body>\n</html>'
```

Таких кодів статусу HTTP є багато і вони об'єднуються в групи у відповідності з їх першою цифрою:

- 1xx (*інформація*) - сервер отримав запит, але має деяку додаткову інформацію для клієнта;
- 2xx (*успіх*) - спрацювало, кожен код успіху, крім 200, повідомляє додаткові деталі;
- 3xx (*перенаправлення*) - ресурс був переміщений, тому відповідь повертає клієнту новий URL;
- 4xx (*помилка клієнта*) - деякі проблеми на стороні клієнта, на зразок **помилки 404 (ресурс не знайдений)**;
- 5xx (*помилка сервера*) - код 500 - це загальна помилка.

Формат даних, які відправляє сервер, вказується значенням Content-Type у заголовку відповіді HTTP. Рядок text/html є **МІМЕ-типом** і означає, що дані до нас прийшли у вигляді HTML. Значення charset=UTF-8 вказує на кодування веб-сторінки.

Щоб дізнатися, які ще заголовки HTTP нам відправив сервер, доповнимо попередній код:

```
for key, value in conn.getheaders():
    print(key, value)
```

Даний цикл буде відбирати заголовки HTTP-запитів і розмішувати у словнику Python. Назва заголовку - ключ словника, значення заголовку - значення словника. Результати виконання програми доповняться наступними:

```
...
Server openresty/1.9.7.1
Date Thu, 11 Jul 2019 09:39:43 GMT
Content-Type text/html; charset=UTF-8
Transfer-Encoding chunked
Connection close
X-Powered-By PHP/7.1.7
Cache-Control private, must-revalidate
...
```

3.2 Завдання до виконання

Завдання 1. Виконайте в інтерактивному інтерпретаторі тренувальні вправи (завдання не обов'язкові):

1. За допомогою модуля requests виконайте запит до сайту gutenberg.org.
2. Виведіть на екран інформацію про статус запиту.
3. Виведіть на екран інформацію про кодування сторінки.

Завдання 2. Створіть каркас простого сайту з використанням фреймворка Flask.

Встановіть фреймворк. Використайте для роботи сервера за адресою localhost порт 5000. Якщо комп'ютер вже використовує цей порт, використайте інший. Додайте функцію `home()` для обробки запитів до домашньої сторінки, яка, наприклад, повертає рядок `This is a web page project. It works!`. Створіть шаблон `jinja2`, який називається `project.html`. Шаблон має зберігатися у каталозі `templates` і мати наступний вміст:

```
<html>
<head>
<title>This is a web page project. It works!</title>
<body>
  This will be a {{name}}, where I will post {{videos}} and
  {{articles}}.
</body>
</html>
```

Модифікуйте функцію `home()`, щоб вона використовувала шаблон `project.html`. Передайте їй три параметра для команди GET: `name`, `videos` і `articles`.

3.3 Зміст звіту

Звіт повинен мати такі складові:

- титульний лист;
- мета роботи;
- результат виконання завдань;
- відповіді на контрольні питання.

3.4 Контрольні запитання

1. Для чого використовують бібліотеки `urllib` і `requests`?
2. Об'єкт якого типу повертає функція `requests.get()`? Яким чином можна отримати доступ до результату роботи цієї функції у вигляді рядкового значення?
3. Як отримати код статусу HTTP із відповіді на запит модуля `requests`?
4. Що таке API?
5. Що таке «скрапінг»? Як можна застосувати бібліотеку `beautifulsoup` для скрапінгу?
6. Назвіть фреймворки для Python. У яких випадках необхідно їх використовувати?

JITEPATYPA

1. Barry P. Head First Python: A Brain-Friendly Guide / Paul Barry. – [S. l.] : O'Reilly Media, Incorporated, 2023.
2. Borjigin C. Python Data Science / Chaolemen Borjigin. – [S. l.] : Springer, 2023.
3. Chongchitnan S. Exploring University Mathematics with Python [Electronic resource] / Siri Chongchitnan. – Cham : Springer Nature Switzerland, 2023. – Mode of access: <https://doi.org/10.1007/978-3-031-46270-2>
4. Deep Learning and Its Applications Using Python / Niha K. Basha [et al.]. – [S. l.] : Wiley & Sons, Incorporated, John, 2023.
5. Hazrat R. A Course in Python [Electronic resource] / Roozbeh Hazrat. – Cham : Springer Nature Switzerland, 2023. – Mode of access: <https://doi.org/10.1007/978-3-031-49780-3>
6. Korites B. Python Graphics [Electronic resource] / Bernard Korites. – Berkeley, CA : Apress, 2023. – Mode of access: <https://doi.org/10.1007/978-1-4842-9660-8>
7. Kouh M. Thermal Physics Tutorials with Python Simulations / Minjoon Kouh, Taejoon Kouh. – [S. l.] : Taylor & Francis Group, 2023.
8. Matuszek D. Quick Python 3 / David Matuszek. – [S. l.] : CRC Press LLC, 2023.
9. Miller R. Computer Scripting Techniques with Python / Rick Miller. – [S. l.] : Pulp Free Press, 2023.
10. Nelli F. Python Data Analytics [Electronic resource] / Fabio Nelli. – Berkeley, CA : Apress, 2023. – Mode of access: <https://doi.org/10.1007/978-1-4842-9532->
11. Python Programming Recipes for IoT Applications / Madhusudan Ganuji Lanjewar [et al.]. – [S. l.] : Springer, 2023.