

Python Essential

Робота з файлами

Python

Після уроку обов'язково



Повторіть цей урок у відео форматі на [ITVDN.com](http://itvdn.com)



Перевірте, як Ви засвоїли даний матеріал
на [TestProvider.com](http://testprovider.com)

Python Essential

Тема

Робота з файлами

Введення та виведення

Поняття файлів та потоків

Файл – іменована область даних на носії інформації.

Файловий об'єкт (потік) – об'єкт, що надає файло-орієнтований API (методи `read()`, `write()` тощо) для доступу до ресурсу. Залежно від способу створення файловий об'єкт може надавати доступ до реального файлу на диску або іншого виду пристрою зберігання або передачі даних (стандартні потоки вводу/виводу, буфери в пам'яті, сокети і т. д.).

Стандартні потоки:

- `sys.stdin`
- `sys.stdout`
- `sys.stderr`



Введення та виведення

Байтові рядки та Unicode-рядки

	Python 2	Python 3
	<p>В Python 2 клас <code>str</code> являє собою рядок як послідовність байтів, він же і використовується в тих випадках, коли потрібно представити в пам'яті послідовність байтів, яка не є рядком.</p> <p>У Python 3 клас <code>str</code> представляє Unicode-рядки, і саме Unicode є стандартним кодуванням для інтерпретатора.</p>	
Байтові рядки	<code>str</code>	<code>bytes</code>
Unicode-рядки	<code>unicode</code>	<code>str</code>

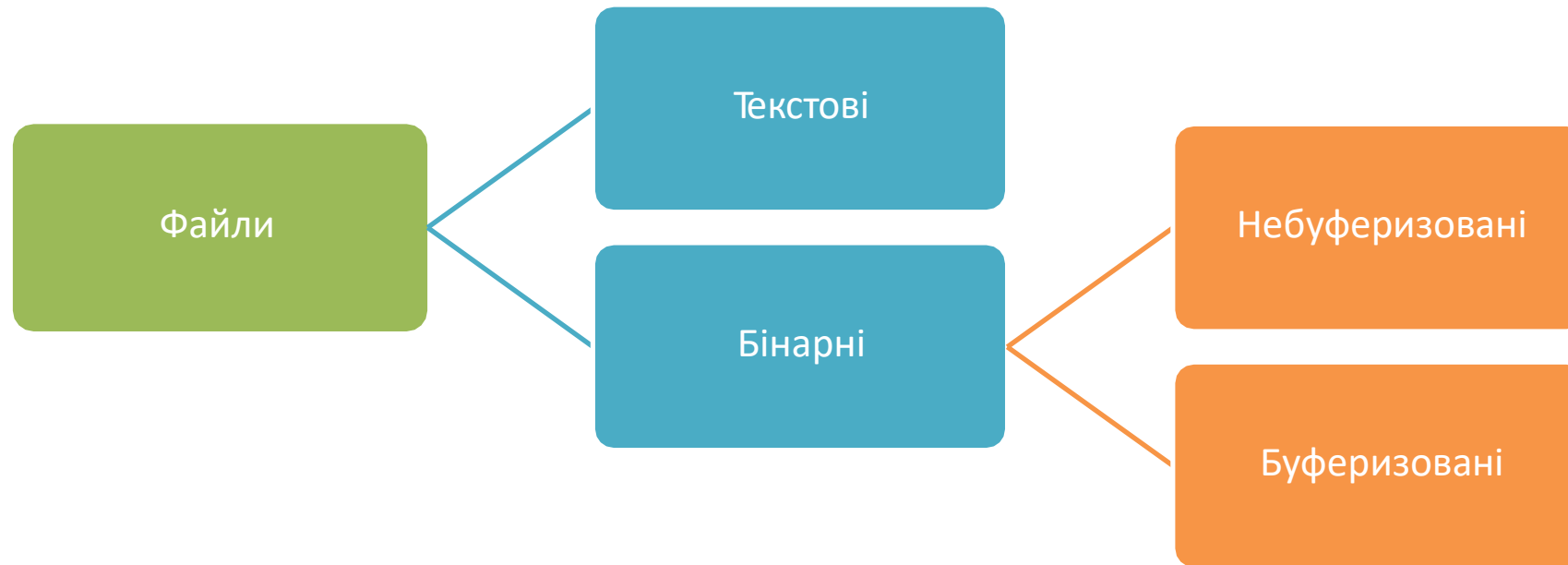
Введення та виведення

Змінювані масиви байтів і символів

	Python 2	Python 3
Змінювані масиви байтів		<code>bytearray</code>
Змінювані масиви Unicode-символів		<code>array.array('u')</code>

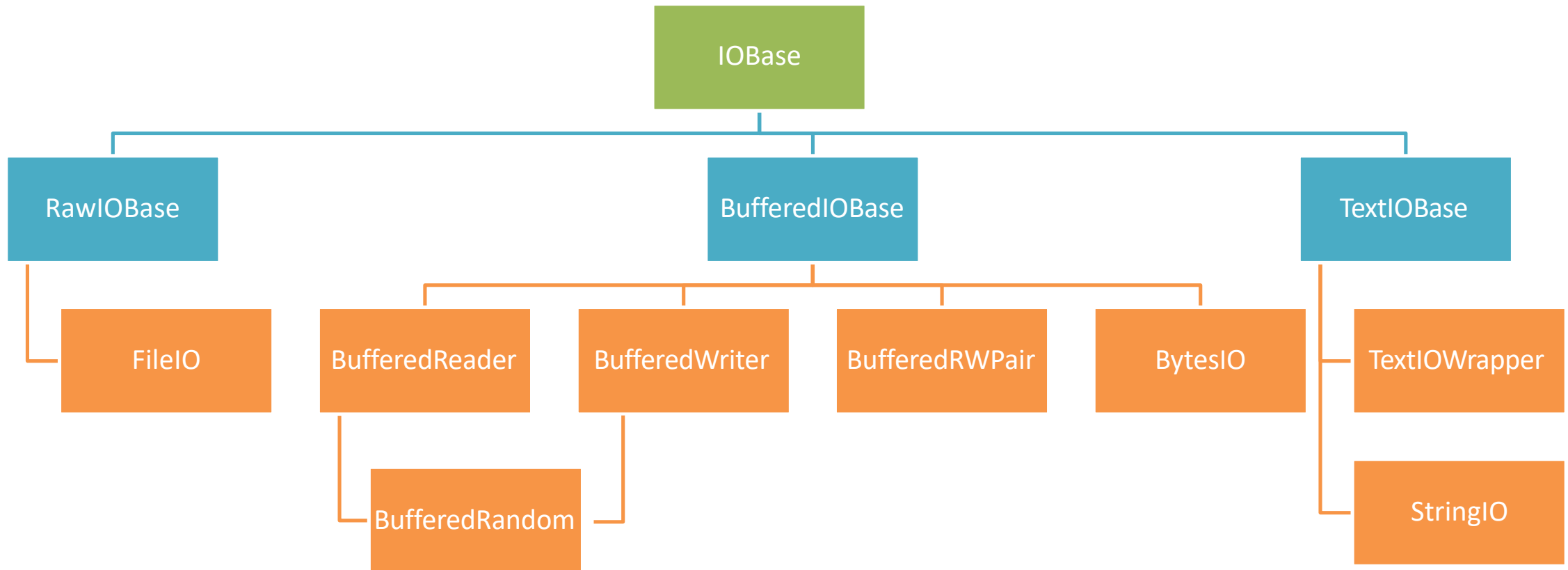
Введення та виведення

Види файлів у Python



Введення та виведення

Ієрархія класів модулю IO



Введення та виведення

Відкриття файлу

Python 2 `open(file, mode='r', buffering=-1)`

Python 3 `open(file, mode='r', buffering=-1,
 encoding=None, errors=None,
 newline=None, closefd=True,
 opener=None)`

Основні параметри:

- `file` – ім'я файлу або файловий дескриптор;
- `mode` – режим відкриття файлу;
- `encoding` – кодування файлу;
- `buffering` – чи використовувати буферизацію і якщо так, яким має бути розмір буфера



Найчастіше за все функція `open()` використовується з двома параметрами

Введення та виведення

Режими відкриття файлів

	Символ	Значення
перший символ	'r'	відкрити для читання (за замовчуванням)
	'w'	відкрити для запису, видаливши вміст файлу
	'x'	відкрити для виняткового створення (помилка, якщо файл існує)
	'a'	відкрити для запису, додаючи вміст у кінець файлу, якщо він існує
другий або третій символ	'b'	бінарний режим
	't'	текстовий режим (за замовчуванням)
	'+'	відкрити для читання та запису



Порядок символів 'b'/'t' та '+' не має значення

Введення та виведення

Закриття файлів

Після закінчення роботи з файлом слід обов'язково його закрити за допомогою методу `close()`, особливо якщо він був відкритий для запису. При використанні буферизованого виводу дані, що записуються у файл, не потрапляють у нього відразу, а записуються до буфера. Вміст буфера записується у файл під час його заповнення чи виклику методів `flush()` або `close()`. Крім того, якщо файл відкритий для запису, він буде заблокований для відкриття для запису іншими процесами до моменту закриття. Всі відкриті файли автоматично закриваються при видаленні відповідних файлових об'єктів з пам'яті збирачем сміття інтерпретатора Python і при завершенні роботи самого інтерпретатора, проте слід тримати файли відкритими мінімально потрібний час.



Введення та виведення

Оператор with

Файлові об'єкти є менеджерами контексту.

Менеджер контексту (context manager) – це об'єкт, який описує певний контекст, який встановлюється під час виконання оператора with. Менеджери контексту використовуються для збереження та відновлення глобального стану, блокування та розблокування ресурсів, автоматичного закриття файлів тощо.

Тобто, при відкритті файлу таким чином:

```
with open(filename, mode) as file:  
    ...  
    # perform actions on file  
    ...
```

він гарантовано буде закритий, як тільки перестане бути потрібним.

Введення та виведення

Читання файлів

Читання одного рядка тексту	<code>file.readline()</code>
Читання списку рядків тексту	<code>file.readlines()</code>
Порядкове читання	<code>for line in file: pass</code>
Читання заданої чи максимально можливої кількості даних (символів чи байтів)	<code>file.read(100)</code> <code>file.read()</code>
Читання даних бінарного файлу в змінюваний масив	<code>file.readinto(arr)</code>

Введення та виведення

Запис до файлів

Запис рядків тексту	<code>file.writelines(lines)</code>
Запис даних (символів або байтів)	<code>file.write(data)</code>
Запис даних у текстовий файл за допомогою функції або оператора print	<code>print(*args, file=text_file) # Python 3</code> <code>print >>text_file, arg1, arg2, ... # Python 2</code>

Введення та виведення

Управління позицією читання/запису

Отримання поточної позиції (відступ від початку файлу в символах або байтах)	<code>file.tell()</code>
Встановлення позиції як зміщення щодо початку файлу	<code>file.seek(offset)</code> <code>file.seek(offset, o.SEEK_SET)</code>
Встановлення позиції як зміщення від поточної	<code>file.seek(offset, io.SEEK_CUR)</code>
Встановлення позиції як зміщення від кінця файлу	<code>file.seek(offset, io.SEEK_END)</code>

Введення та виведення

Серіалізація об'єктів

Серіалізація – це процес збереження об'єктів у двійковому чи рядковому вигляді для зберігання, передачі та відновлення. Зворотний процес називається **десеріалізацією**.



Введення та виведення

Модуль json

- Для збереження різноманітних даних у текстовому вигляді можна використовувати формат JSON (JavaScript Object Notation), який є стандартом де-факто у веб-додатках і є досить популярним в інших областях.
- Він підтримується (на рівні мови, стандартної бібліотеки мови або, принаймні, існуючих сторонніх бібліотек) всіма сучасними мовами програмування та знайомий багатьом розробникам, що робить його чудовим вибором для збереження даних для передачі між додатками, написаними різними мовами.
- Модуль json дозволяє серіалізувати словники, списки, кортежі, рядки, цілі та дійсні числа, перерахування, булевські значення та None. Для підтримки інших типів даних слід розширити класи JSONEncoder та JSONDecoder.



Введення та виведення

Модуль pickle

- Модуль pickle реалізує бінарні протоколи для серіалізації та десеріалізації об'єктів Python.
- Pickle підтримує серіалізацію величезної кількості типів даних, включаючи створені користувачем (багато з яких підтримуються автоматично, але у складніших випадках об'єкт повинен реалізувати певні спеціальні методи), коректно обробляє посилання на вже серіалізовані об'єкти і серіалізацію рекурсивних типів даних.
- На відміну від JSON, який може бути прочитаний будь-де, pickle специфічний для Python і не може бути використаний для обміну даними з додатками, написаними іншими мовами.



Недоліком pickle є його небезпечність: десеріалізувати можна лише ті дані у форматі pickle, які отримані з надійного джерела, інакше є ризик виконання довільного коду.

Дивіться наші уроки у відео форматі

ITVDN.com



Перегляньте цей урок у відео форматі на освітньому порталі [ITVDN.com](http://itvdn.com) для закріплення пройденого матеріалу.

Усі курси записані сертифікованими тренерами, які працюють у навчальному центрі CyberBionic Systematics



Перевірка знань

TestProvider.com

TestProvider

Мы помогаем людям оценить себя

Регистрация Войти

Поиск сертификата

Главная Услуги и цены Центр Тестирования Поддержка О нас

Мы в социальных сетях

Тестирование

Языки программирования и информационные технологии

Microsoft

C# ASP.NET MVC JavaScript Patterns OF Design SQL Architecture Guide WCF HTML&CSS XML SEO WPF HTML5&CSS3 JQuery XNA SharePoint GUI for Android Windows Azure Platform Microsoft Patterns&Practices TFS SCRUM ReSharper TDD WWF LINQ Entity Framework Windows Forms Refactoring Microsoft Expression Blend 4 Windows Phone 8 Windows 8 AppStore Visual Studio Tips&Tricks MSF MEF SilverLight AJAX MEF Service Oriented Architecture

Добро пожаловать на TestProvider.com!

Сайт перенесен на новую облачную платформу с использованием системы единой авторизации Single Sign On. Если вы хотите восстановить статистику по предыдущим экзаменам обратитесь в [службу поддержки](#). Для восстановления информации с предыдущей версии сайта, просба написать в службу поддержки Ваш старый и новый логины.

ITVDN PROMETRIC TEST CENTER CyberBionic Microsoft Partner Windows Azure Cloud Partner EBA

TestProvider – це online сервіс перевірки знань з інформаційних технологій. З його допомогою Ви можете оцінити Ваш рівень та виявити слабкі місця. Він буде корисним як у процесі вивчення технології, так і для загальної оцінки знань IT-спеціаліста.

Після кожного уроку проходите тестування для перевірки знань на TestProvider.com

Успішне проходження фінального тестування дозволить Вам отримати відповідний Сертифікат.



Python Essential

Q&A

Інформаційний відеосервіс для розробників програмного забезпечення

