

Тема. Повторення та узагальнення вивченого за рік

Після цього заняття потрібно:

- Розуміти поняття двійкового коду, називати одиниці вимірювання його довжини.
- Пояснювати взаємодію складових обчислювальних пристроїв, описувати їхні основні характеристики.
- Розуміти призначення мови програмування та основних її елементів, наводити приклади сучасних мов програмування.
- Порівнювати особливості різних середовищ програмування.
- Пояснювати структуру програми.
- Мати уявлення про принципи та можливості опрацювання текстових даних, мати уявлення про структуру документа.
- Наводити приклади засобів автоматизованого створення веб-сторінок.

Ознайомтеся з інформацією

Двійкове кодування - кодування повідомлень з використанням сигналів лише двох видів.

Біт – найменша порція дискретної інформації. Приймає значення 0 (вимкнено) або 1 (ввімкнено).

Байт – двійковий код порції інформації.

1 байт = 8 біт

Для кодування інформації в комп'ютері використовується восьмизначний двійковий код.

1 символ = 1 байт

Таблиці кодів символів

- ASCII, Windows-1251, KOI8-U – $2^8=256$ символів, 1 символ=1 байт
- Unicode (Юнікод) – $17 \cdot 2^{16} = 1\,114\,112$ символів, 1 символ=2 байти

Довжина двійкового коду повідомлення — це кількість байтів у його двійковому коді. Її можна вказувати у кратних одиницях вимірювання інформації, тобто у **бітах, байтах, кілобайтах** і т. д. Для визначення довжини двійкового коду документа достатньо ознайомитись із його властивостями та визначити його розмір у байтах.

Перегляньте відео за посиланням:

<https://youtu.be/GS9GylNOemc>

Ознайомтеся з інформацією

Мову, яку використовують для запису алгоритмів, призначених для виконання комп'ютером, називають **мовою програмування**.

Кожна мова програмування має такі **складові**:

1. **Алфавіт** - множина символів, з яких можна утворювати слова і речення цієї мови.
2. **Словник** - набір слів, які використовуються в цій мові.
3. **Синтаксис** - правила складання і запису мовних конструкцій: не словникових слів і речень
4. **Семантику** - встановлене однозначне тлумачення мовних конструкцій, правил їх виконання.

Використання символів, що не входять до алфавіту мови програмування, неправильне написання словникових слів, порушення синтаксичних правил призводять до неможливості виконання комп'ютером відповідної команди. Такі порушення називають *синтаксичними помилками*.

За останні 70 років створено близько трьох тисяч різних мов програмування. Деякі з них уже не використовують, для деяких постійно з'являються нові версії, що зручніші для складання програм і мають більші можливості. Постійно створюються нові мови програмування.

Мови програмування можна поділити на дві групи: **мови низького рівня** та **мови високого рівня**.

Алфавіт мови Python

У мові Python при створенні програм можуть використовуватися такі символи:

- ✓ літери латинського алфавіту A..Z, a..z; цифри 0..9;
- ✓ знаки арифметичних операцій, спеціальні символи: + - * / \^ = < > () . , : ; ' # _;
- ✓ комбінації символів: <=, >=, <>, = =; !=, **;
- ✓ службові слова, що мають фіксований для Python зміст, наприклад: *and, elif, if, print, as, else, import* тощо.

Окремий інформаційний об'єкт (число, символ, рядок та ін.) називають величиною. Основними характеристиками величин є назва, вид, тип і значення. Вид величини визначає спосіб використання величини в програмі. Величина може бути константою (тобто постійною) або змінною.

Константи — це величини, значення яких не можуть змінюватися в ході виконання програми. Прикладом константи може бути число (5, 1.23) або рядок: "Це рядок!".

Змінні — величини, значення яких можуть змінюватися в ході виконання програми.

Коментарі в програмі

Коментар — це текст, призначений для читання людиною, а не комп'ютером.

Коментар — це підказка для нас, яку дію виконує програма. Вставляючи коментарі в текст програми, ми спрощуємо собі та іншим її читання і розуміння.

Щоб комп'ютер відрізняв команди від коментарів, у мові Python перед текстом коментаря ставиться знак '#'.

Типи величин в мові Python

Змінні зручно представити у вигляді "поштових скриньок" (комірок пам'яті комп'ютера), на які навішені ярлики з їх іменами. Для різних величин створюються «скриньки» різного розміру, який залежить від типу величини. Тип величини визначається обсягом пам'яті, необхідним для її збереження, множиною припустимих значень величини, та операціями, які можна над нею виконувати. Основними типами величин є числа і рядки.

В Python є два типи числових даних:

- ✓ цілі числа (int), тобто числа без дробової частини;
- ✓ і дійсні (float) — дробові числа з десятковою крапкою.

Рядок — це взята в одинарні лапки послідовність будь-яких символів — цифр, літер, розділових знаків. У змінних рядкового типу ми зберігатимемо фрагменти тексту.

Оператори Python

Математичні символи, такі як + (плюс) і – (мінус), називаються операторами, так як вони оперують (або виконують обчислення) числами в рівнянні.

Для тих, хто хоче знати більше

10 популярних мов програмування у 2022 році

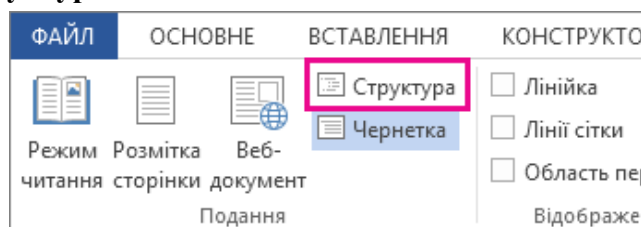
<https://uaspectr.com/2022/02/11/10-populyarnyh-mov-programuvannya/>

Ознайомтеся з інформацією

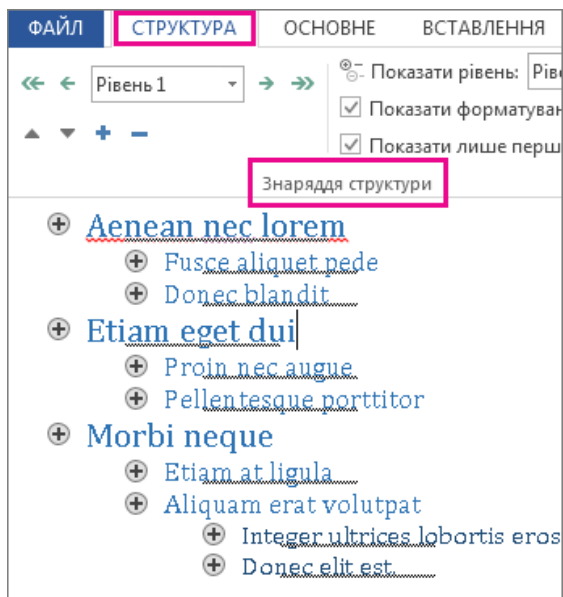
Створення структури документа в поданні структури

Якщо у вас є складний документ для впорядкування подання структури (**подання > Структура**) допомагає вам зосередитися на її структурі. Створення або редагування заголовків, настройте рівнів заголовків за допомогою подання структури та перевпорядкування вмісту, поки все правильно потрібне місце.

Послідовно виберіть елементи **вигляд > Структура**. Це автоматично створює структури та відкриває засоби **Структура**.



Якщо документ має заголовки (будь-які рівні заголовків від Н1 до Н9), ви побачите ці заголовки впорядковані за рівнями.



Якщо в документі ще немає заголовків і підзаголовків, уся структура відображатиметься як маркірований список з окремим пунктом для кожного абзацу. Текст окремих пунктів позначатиметься як основний, доки ви не додасте заголовки.

Використання подання структури для керування заголовками й упорядкування тексту

Подання структури дає змогу керувати структурою та заголовками документа, не ризикуючи втратити текст. У поданні структури (на вкладці **Подання** натисніть кнопку **Структура**) можна скористатися **інструментами структури**, щоб переміщати й редагувати заголовки, змінювати їхні рівні, а також переміщати текст. Також можна керувати деталізацією даних.

Ось як працювати з документом у поданні структури:

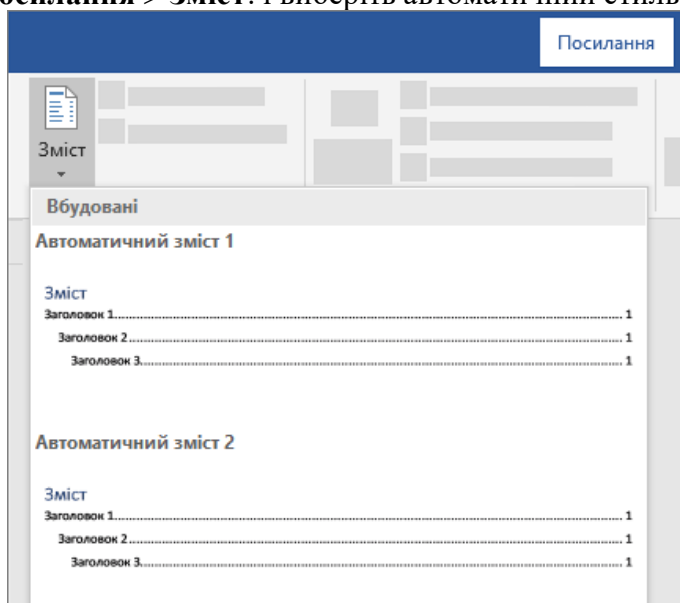
Завдання	Дія	Сполучення клавіш
Створення заголовка	Виділіть основний текст у поданні структури, а потім на вкладці Структура у групі Знаряддя структури клацніть поле Рівень і виберіть потрібний рівень заголовка.	Alt + Shift + стрілка вліво
Редагування заголовка	Виділіть текст у поданні структури та внесіть необхідні зміни.	
Змінення рівня заголовку	У розділі Структура , у розділі Знаряддя для структури клацніть стрілку вправо та вліво біля поля Рівень структури , щоб підвищити або Понизити рівень заголовків.	Alt + Shift + стрілка вліво або Alt + Shift + стрілка вправо
Переміщення тексту вгору або вниз	Виділіть текст і натисніть кнопку вгору або за допомогою стрілок у розділі створення структури .	Alt + Shift + стрілка вгору або Alt + Shift + стрілка вниз
Розгортання або згортання тексту	Виберіть заголовок і натисніть кнопку на + або - символи у розділі Структура .	Alt + Shift + "+" або Alt + Shift + "-"
Відображення або приховання деяких заголовків	У полі Показати рівень у розділі Структура виберіть найнижчий рівень заголовків, який необхідно відобразити. Усі рівні, нижчі за вибраний, будуть приховані.	Alt+Shift+n – відображення всіх заголовків до рівня n включно
Відображення першого рядка тексту або всього тексту	На вкладці Структура установіть або зніміть прапорець Показати лише перший рядок .	Alt+Shift+L

Закриття подання структури	На вкладці Структура клацніть піктограму "X" на кнопці Закрити режим структури .	
Перехід між поданнями	Натисніть кнопку подання та виберіть потрібний тип подання, який потрібно перейти, наприклад Розмітки або Режим читання .	Alt+Ctrl+P – подання друку, Alt+Ctrl+N – подання чернетки, Alt+Ctrl+O – повернення до подання структури.

Вставлення змісту

Створення змісту

1. Розташуйте курсор там, де потрібно додати зміст.
2. Виберіть **Посилання > Зміст**. і виберіть автоматичний стиль.



3. Якщо ви вносите зміни до документа, який впливає на зміст, оновіть зміст, клацнувши правою кнопкою миші зміст і вибравши **поле оновити**.

Створення та публікація веб-ресурсів

Веб-сторінка (англ. Web-page) — інформаційний ресурс, доступний у мережі World Wide Web, який можна переглянути у веб-браузері.

Веб-сайт (англ. website, місце, майданчик в Інтернеті), також сайт (англ. site, місце, майданчик) — сукупність веб-сторінок, доступних в Інтернеті, які об'єднані як за змістом, так і навігаційно. Фізично сайт може розміщуватися як на одному, так і на кількох серверах.

Для того щоб сайт став доступним широкому колу відвідувачів, йому необхідно призначити доменне ім'я і розмістити в мережі Інтернет. Розміщення сайту на сервері та подальше його адміністрування називають хостингом. Наданням такої послуги займаються спеціальні організації. Хостинг буває платним і безкоштовним. Можливість створювати веб-сторінки та організовувати форуми й чати в автоматизованому режимі (крім власне розміщення сайту) часто надається на серверах, що забезпечують хостинг.

Для створення і наповнення даними веб-сторінок можуть використовуватись різні техніки: від стандартного **Блокноту** до **автоматизованих онлайн-засобів** (Google-сайти, WordPress, Wix, Weebly, Tilda тощо).

Пройдіть тестування за посиланням:

<https://naurok.com.ua/test/join?gamecode=5823869>