

3. Асинхронність. Прототипи.

Контрольні запитання

- 03.01. Що таке асинхронна операція?
- 03.02. Для чого потрібен об'єкт **Promise**?
- 03.03. Що таке AJAX?
- 03.04. Що таке прототип об'єкта?
- 03.05. Яким чином здійснюється доступ до прототипу з конкретного об'єкта? Як це зробити з функції-конструктора?
- 03.06. Що таке ланцюжок прототипів?
- 03.07. Як працює оператор **instanceof**?
- 03.08. Як здійснити наслідування в прототипно-орієнтованій моделі?

Завдання для аудиторної роботи

[A]03.01. Опишіть функцію **delay(ms)**, яка повертає **Promise**, що виконається через **ms** мілісекунд. Викличте її в асинхронній функції.

[A]03.02. Напишіть програму, яка отримує дані постів з API

```
https://jsonplaceholder.typicode.com/posts/{number}
```

за номером поста **number**, відображає назву посту на сторінці та обробляє мережеві помилки.

Номер поста вводить користувач в окремому полі на сторінці.

- а) Виконайте задачу використовуючи об'єкт **Promise**.
- б) Виконайте задачу використовуючи **async/await**.

Вказівка. Використайте функцію **delay(ms)** з задачі [A]03.01 щоб уповільнити виконання AJAX-запитів.

[A]03.03. Використовуючи прототипно-орієнтовану модель (без синтаксису **class**), опишіть такі конструктори:

- **Animal(name)** для створення об'єктів з методом **speak()**, який виводить **"\${name} видає звук!"**.
- **Dog(name, breed)** з перевизначеним методом **speak()**, який виводить **"\${name} породи \${breed} гавкає!"**.

Всі методи мають знаходитися в прототипі.

Продемонструйте роботу програми створивши декілька об'єктів заданих конструкторів.

Завдання для самостійної роботи

[B]03.01. Опишіть дві асинхронні функції, кожна з яких кидає виключення з імовірністю 50%. Викличте їх паралельно через `Promise.any()`. Виведіть повідомлення про успішність виконання програми.

Вказівка. Щоб кинути виключення, використайте інструкцію

```
throw new Error("Щось пішло не так...");
```

[B]03.02. Зробіть три AJAX-запити з API

```
https://jsonplaceholder.typicode.com/posts/{number}
```

які виконуються послідовно одне за одним. Виведіть їх результати на сторінці у правильному порядку.

Вказівка. Використайте функцію `delay(ms)` з задачі [A]03.01 щоб уповільнити виконання AJAX-запитів.

[B]03.03. Використовуючи прототипно-орієнтовану модель (без синтаксису `class`), опишіть такі конструктори:

- `Shape()` для створення об'єктів з методом `area()`, який повертає площу фігури.
- `Ellipse(a, b)` з методом `area()`, який повертає площу еліпса.
- `Circle(r)` з методом `area()`, який повертає площу круга.

Всі методи мають знаходитися в прототипі.

Продемонструйте роботу програми створивши декілька об'єктів заданих конструкторів.