3. Асинхронність в JavaScript. AJAX.

Контрольні запитання

03.01. Що таке асинхронна операція?
03.02. Чим відрізняється синхронний код від асинхронного?
03.03. Для чого потрібен об'єкт `Promise`? Які він має стани?
03.04. Яка різниця між `Promise.all()` та `Promise.any()`?
03.05. Що повертає `async` функція? Для чого потрібне ключове слово `await`?
03.06. Що таке AJAX?
03.07. Для чого застосовується функція `fetch()`?

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.

a) Виконайте задачу використовуючи об'єкт `Promise`.
b) Виконайте задачу використовуючи `async/await`.

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

[A]03.03. Напишіть програму, яка отримує дані NASA з API

```
https://api.nasa.gov/planetary/apod?api_key=DEMO_KEY
```

надсилаючи сьогоднішню або обрану в окремому полі `<input type=date>` дату і розміщує отримане покликання як фото на HTML-сторінці.

a) Для виконання задачі використайте функцію `fetch()`, `Promise`, `async/await`, `bootstrap` (за бажання).
b) Використайте функцію `setTimeout`, щоб скрипт оновлював фото кожні 10 секунд, показуючи фото зроблені кожного попереднього дня (інверсивна ітерація списку дат).

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. Напишіть програму для отримання мемів з API

```
https://api.imgflip.com/get_memes
```

Структура API має вигляд

```
{
  "success": true,
  "data": {
    "memes": [
      {
        ...
        "url": "meme_link"
        ...
      },
      ...
    ]
  }
}
```

Обробіть відповідь з зазначеною структурою та оберіть випадковий мем через

```
Math.floor(Math.random() * array.length)
```

Відобразіть отримане зображення на сторінці.
Застосуйте функцію `setTimeout()` для того щоб динамічно оновлювати його кожні 10 секунд.