Q: Чому в JavaScript потрібен асинхронний код?

a) Для прискорення виконання всього коду.

b) Щоб уникнути блокування основного потоку виконання під час "повільних" операцій, таких як завантаження даних.

c) Для спрощення синтаксису.

d) JavaScript не підтримує синхронний код.

A: 2

Q: Що представляє собою Promise (проміс) в JavaScript?

a) Функцію, яка виконує асинхронну операцію.

b) Об'єкт, що представляє поточний стан асинхронної операції.

c) Синхронну операцію.

d) Значення, яке завжди відоме на момент створення.

A: 2

Q: Скільки основних станів може мати проміс?

a) Один

b) Два

c) Три

d) Чотири

A: 3

Q: Як називається початковий стан промісу під час його створення?

a) Fulfilled

b) Rejected

c) Settled

d) Pending

A: 4

Q: В який стан переходить проміс, якщо асинхронна операція виконана успішно з будь-яким результатом?

a) Pending

b) Rejected

c) Fulfilled

d) Settled

A: 3

Q: В який стан переходить проміс, якщо асинхронна операція відхилена з помилкою?

a) Pending

b) Fulfilled

c) Rejected

d) Settled

A: 3

Q: Чи може стан промісу змінитися після того, як він перейшов у стан "виконано" (fulfilled) або "відхилено" (rejected)?

a) Так, може повернутися до стану "очікування".

b) Так, може перейти з "виконано" у "відхилено".

c) Ні, після переходу в стан fulfilled або rejected він більше не змінює свій стан.

d) Залежить від типу асинхронної операції.

A: 3

Q: Яким терміном описують проміс, який перебуває в будь-якому стані, крім "очікування" (pending)?

a) Active

b) Resolved

c) Completed

d) Settled

A: 4

Q: Як створюється новий проміс в JavaScript?

a) Викликом функції createPromise().

b) Як екземпляр класу Promise за допомогою new Promise().

c) За допомогою ключового слова async.

d) Шляхом передачі колбеків у функцію.

A: 2

Q: Як називається функція, яка передається як аргумент конструктору new Promise() і відразу викликається?

a) Handler

b) Callback

c) Executor

d) Resolver

A: 3

Q: Яка функція викликається всередині виконавчої функції промісу (executor), щоб повідомити про успішне виконання операції та передати її результат?

a) reject()

b) catch()

c) resolve()

d) then()

A: 3

Q: Яка функція викликається всередині виконавчої функції промісу (executor), щоб повідомити про помилку виконання операції?

a) resolve()

b) then()

c) reject()

d) finally()

A: 3

Q: Який метод промісу використовується для реєстрації обробників успішного виконання та/або помилки?

a) .catch()

b) .finally()

c) .then()

d) .resolve()

A: 3

Q: Скільки аргументів (callback-функцій) може приймати метод .then()?

a) Один (для успіху)

b) Два (перший для успіху, другий для помилки)

c) Три (для успіху, помилки, і завершення)

d) Жодного

A: 2

Q: Який метод промісу на практиці найчастіше використовується для обробки тільки помилок (відхиленого стану)?

a) .then() з двома аргументами

b) .catch()

c) .finally()

d) try...catch

A: 2

Q: Після якого методу промісу зазвичай розташовується метод .catch() при побудові ланцюжка обробників?

a) Перед .then().

b) Перед .finally().

c) Після .then().

d) Порядок не має значення.

A: 3

Q: Який метод промісу дозволяє виконати код незалежно від того, чи проміс виконався успішно чи з помилкою (тобто перейшов у стан settled)?

a) .then()

b) .catch()

c) .finally()

d) .resolve()

A: 3

Q: Чи отримує callback-функція, передана в метод .finally(), якісь аргументи?

a) Так, значення успішного виконання.

b) Так, об'єкт помилки.

c) Так, об'єкт зі статусом промісу.

d) Ні, вона не отримує жодних аргументів.

A: 4

Q: Яка особливість методу .then() дозволяє формувати послідовність асинхронних операцій, зв'язуючи проміси в ланцюжок?

a) Він завжди виконується негайно.

b) Він повертає новий проміс.

c) Він приймає будь-яку кількість аргументів.

d) Він автоматично обробляє всі помилки.

A: 2

Q: Що відбувається у ланцюжку промісів, якщо виникає помилка в одному з .then() обробників?

a) Виконання ланцюжка продовжується, і помилка ігнорується.

b) Виконання всіх наступних .then() скасовується, а управління передається найближчому методу .catch().

c) Весь код зупиняється з фатальною помилкою.

d) Помилка передається як значення наступному .then().

A: 2