Q: У якому атрибуті тега script вказується шлях до файлу скрипта?

a) src

b) scr

c) defer

A: 1

Q: Який індекс першого елемента рядка?

a) -1

b) 0

c) 1

A: 2

W: Яке розширення має файл, що містить JavaScript-код, який буде виконано на сторінці?

A: js

MQ: Назвіть способи підключення скрипта JavaScript до HTML-сторінки:

a) вбудований скрипт через тег <script>;

b) зовнішній скрипт через тег <script>;

c) через тег <style>;

d) зовнішній скрипт через тег <link>.

A: 1,2

Q: Яка властивість рядка містить значення її довжини?

a) size

b) length

c) width

d) chars

A: 2

Q: Що буде виведено в консоль під час виконання коду? console.log("false" + null)

a) null

b) Помилка

c) Рядок “falsenull”

d) Рядок “false”

A: 3

Q: Визнач, чи є помилка у прикладі коду: const username = "Mango"; username = "Poly";

a) так

b) ні

A: 1

Q: Вибери той варіант імені змінної, що записаний у camelCase нотації.

a) userage

b) user\_age

c) userAge

d) UserAge

A: 3

Q: Яка директива активує суворий режим?

a) ‘strict mode’

b) ‘use strict’

c) ‘safe mode’

A: 2

Q: За допомогою якого ключового слова оголошується змінна, якій не можна задати нове значення після її оголошення?

a) const

b) let

A: 1

MQ: Вибери існуючі примітивні типи даних в мові JavaScript:

a) String

b) Char

c) Number

d) Float

e) Boolean

A: 1,3,5

Q: Якої помилки припустилися в цьому фрагменті коду під час оголошення змінної: username = "Mango"; console.log(username);

a) Неприпустиме ім'я змінної

b) Неприпустиме значення змінної

c) Не вистачає ключового слова const або let

A: 3

Q: Який метод використовується для виведення інформації з коду в консоль інструментів розробника?

a) print()

b) output()

c) console.log()

A: 3

Q: Яким буде фінальне значення змінної username? let username = "Poly"; username[3] = "a";

a) “Poly”

b) “Pola”

c) Буде помилка

A: 1

Q: Що буде виведено в консоль інструментів розробника після виконання коду? let username; console.log(username);

a) ReferenceError: username is not defined

b) null

c) undefined

A: 3

Q: Яким буде результат виразу typeof "true"?

a) String

b) Boolean

c) True

A: 1

Q: Яким буде результат виразу? null || true

a) true

b) false

c) null

d) 0

A: 1

Q: Яким буде результат виразу? "false" && 0

a) true

b) false

c) null

d) 0

e) "false"

A: 4

Q: Яким буде результат виразу? Boolean("")

a) true

b) false

A: 2

Q: Яким буде результат виразу? null && true

a) true

b) false

c) null

A: 3

Q: Яким буде результат виразу? "Mango" && "Poly"

a) true

b) false

c) “Mango”

d) “Poly”

A: 4

Q: Яким буде результат виразу? "Mango" || "Poly"

a) true

b) false

c) “Mango”

d) “Poly”

A: 3

Q: Яким буде результат виразу? !0

a) true

b) false

c) 1

d) 0

A: 1

Q: Яким буде результат виразу? !"Poly"

a) true

b) false

c) 1

d) 0

e) "Poly"

f) "not Poly"

A: 2

Q: Яким буде результат виразу? Boolean("false")

a) true

b) false

A: 1

Q: Яким буде результат виразу? "false" || 0

a) true

b) false

c) "false"

d) 0

A: 3

Q: Вибери правильний синтаксис літерала масиву

a) <>

b) []

c) {}

d) ()

A: 2

Q: Який із наведених способів правильно створює масив у JavaScript?

a) let arr = "apple", "banana", "orange";

b) let arr = ["apple", "banana", "orange"];

c) let arr = array("apple", "banana", "orange");

d) let arr = {"apple", "banana", "orange"};

A: 2

Q: Який індекс першого елемента масиву?

a) -1

b) 0

c) 1

A: 2

Q: Вибери правильний синтаксис звернення до елемента масиву users з індексом 2

a) [2]users

b) users[2]

c) users.2

A: 2

Q: Що поверне наступний код? let fruits = ["apple", "banana", "orange"]; console.log(fruits[1]);

a) apple

b) undefined

c) banana

d) orange

A: 3

Q: Як визначається індекс останнього елемента масиву?

a) Він завжди дорівнює значенню довжини масиву

b) Він завжди на одиницю менший за довжину масиву

c) Він завжди на одиницю більший за довжину масиву

A: 2

Q: Яким буде фінальне значення змінної arr2? const arr1 = [1, 2, 3]; const arr2 = arr1; arr1[1] = 1; arr2[2] = 1;

a) [1, 2, 3]

b) [1, 1, 1]

c) [1, 1, 3]

A: 2

Q: Яким буде значення змінної result? const arr1 = [1, 2, 3]; const arr2 = [1, 2, 3]; const result = arr1 === arr2;

a) true

b) false

A: 2

Q: Яким буде значення змінної result? const array= [false]; const result = array ? "A" : "B";

a) true

b) false

c) "А"

d) "В"

A: 3

Q: Що повертає метод indexOf, якщо елемент у масиві не знайдено?

a) false

b) -1

c) null

A: 2

Q: Що поверне наступний код? let numbers = [1, 2, 3, 4, 5]; console.log(numbers.length);

a) 5

b) 4

c) 6

d) undefined

A: 1

Q: Що в коді використовується в умові для зупинки циклу під час ітерацій по масиву? const planets = ["Earth", "Mars", "Venus"]; for (let i = 0; i < planets.length; i += 1) { console.log(planets[i]); }

a) Значення довжини масиву

b) Значення елементів масиву

A: 1

Q: Який метод додає новий елемент у кінець масиву?

a) .push()

b) .pop()

c) .shift()

d) .unshift()

A: 1

Q: Який метод видаляє перший елемент масиву?

a) .push()

b) .pop()

c) .shift()

d) .unshift()

A: 3

Q: Що повертає метод .join() у JavaScript?

a) Масив, де елементи з'єднані заданим роздільником.

b) Рядок, де елементи масиву з'єднані заданим роздільником.

c) Число, яке відповідає кількості елементів у масиві. Об'єкт з властивостями масиву.

d) Об'єкт з властивостями масиву.

A: 2

Q: Що поверне наступний код? let fruits = ["apple", "banana", "orange"]; console.log(fruits.join(", "));

a) "apple, banana, orange"

b) "apple,banana,orange"

c) ["apple", "banana", "orange"]

d) "apple banana orange"

A: 1

Q: Що відбудеться, якщо викликати .join() без аргументу?

a) Елементи будуть з'єднані пробілом.

b) Елементи будуть з'єднані через кому.

c) Елементи залишаться у вигляді масиву.

d) Буде викинута помилка.

A: 2

Q: Що поверне наступний код? let numbers = [1, 2, 3]; console.log(numbers.join(" - "));

a) "1 - 2 - 3"

b) "1-2-3"

c) "[1, 2, 3]"

d) "1, 2, 3"

A: 1

Q: Що повертає метод includes?

a) Індекс елемента або -1

b) Значення елемента або undefined

c) Буль true або false

A: 3

Q: Який метод об'єднує два або більше масивів в один?

a) .concat()

b) .join()

c) .map() .

d) .filter()

A: 1

Q: Чи можна отримати доступ до лічильника в циклі for...of?

a) так

b) ні

A: 2

Q: Вибери правильний синтаксис імпорту змінної А під ім'ям В?

a) import B from " ... "

b) import { A to B } from " ... "

c) import { A as B } from " ... "

d) import A as B from " ... "

A: 3

Q: Вибери правильний синтаксис іменованого імпорту змінної username

a) import { username } from " ... "

b) import { anyName } from " ... "

c) import username from " ... "

A: 1

Q: Вибери правильний синтаксис експорту для замовчування для зміни імені користувача username default

a) export default username

b) export default

c) export username

d) export username default

A: 1

Q: Вибери синтаксис імпорту простору імен усіх експортів модуля в об'єкт з ім'ям everything

a) import everything as \* from " ... "

b) import \* as everything from " ... "

c) import everything from " ... "

d) import \* from " ... "

A: 2

Q: Що таке модуль в JS-проекті?

a) Будь-який JavaScript-файл проекту

b) Будь-який файл налаштування проєкту

c) Тільки файли залежностей проекту

A: 1

Q: Вибери правильний синтаксис імпорту за замовчуванням

a) from " ... " import anyName

b) import { anyName } from " ... "

c) import anyName

d) import anyName from " ... "

A: 4

Q: Вибери правильний синтаксис іменованого експорту змінної username

a) username export

b) export const username = "Mango"

c) export default username

A: 2

Q: Що робить функція setTimeout?

a) Викликає передану функцію

b) Планує в асинхронній черзі відкладений виклик функції через вказаний інтервал часу

c) Чекає зазначеного часу, перед тим як викликати функцію

A: 2

Q: Що станеться, якщо викликати setInterval з нульовим інтервалом? (setInterval(fn, 0))

a) Викинеться помилка

b) Функція буде викликатись з мінімально можливою затримкою

c) Функція буде викликатись безперервно без затримки

d) Функція виконається лише один раз

A: 2

Q: Що потрібно передати функції clear Timeout?

a) Посилання на функцію, яка була передана в setTimeout

b) Затримку, після якої буде зупинено таймер

c) Ідентифікатор таймера, який повернув виклик setTimeout

A: 3

Q: Як виконується синхронний код?

a) По черзі, кілька інструкцій виконуються в міру можливості інтерпретатора

b) Паралельно, інструкції не чекають завершення поточної операції

c) Послідовно, кожна наступна інструкція чекає завершення поточно

A: 3

Q: Що повертає функція setInterval?

a) Значення затримки до виклику функції

b) Посилання на відкладену функцію

c) Числовий ідентифікатор інтервалу

d) Нічого

A: 3

Q: Що потрібно передати функції clearlnterval?

a) Ідентифікатор інтервалу, який повернув setInterval

b) Значення часу, через яке буде повторюватися інтервал

c) Посилання на функцію, яка була передана в setlnterval

A: 1

Q: Скільки разів буде викликана функція fоо для setinterval(foo, 2000)?

a) Кожні 2 секунди

b) Один раз одразу і один раз через 2 секунди

c) Один раз через 2 секунди

A: 1

Q: Який із варіантів коду зупинить виконання setTimeout? const timer = setTimeout(() => console.log("Hello"), 5000);

a) timer = null;

b) clearTimeout(timer);

c) stop Timeout(timer);

d) clearInterval(timer);

A: 2

Q: Що повертає виклик функції setTimeout?

a) Числовий ідентифікатор таймера

b) Нічого

c) Посилання на відкладену функцію

d) Значення затримки до виклику функції

A: 1

Q: Що бере асинхронний код у JavaScript?

a) Код, який може бути незалежною від основного потоку

b) Код, який працює тільки в браузері

c) Код, який виконується разом, інструкція за інструкцією

d) Код, який завжди виконується швидше за синхронний

A: 1

Q: Що отримує функція, яка передається аргументом у метод catch()?

a) Значення, з яким виконався проміс

b) Помилку, з якою виконався проміс

A: 2

Q: У який стан переходить проміс, що успішно виконався?

a) Pending

b) Fulfilled

c) Settled

d) Rejected

A: 2

Q: Що отримує функція, яка передається аргументом у метод finally?

a) Нічого

b) Значення, з яким виконався проміс

c) Помилку, з якою виконався проміс

A: 1

Q: Який метод промісу використовується для обробки його результату?

a) process

b) then

c) catch

d) execute

A: 2

Q: Коли виконається колбек-функція, яка передається першим аргументом у метод then?

a) Після того, як проміс буде виконано успішно

b) Щойно проміс буде створено

c) Після того, як проміс буде виконано успішно або з помилкою

d) Після того, як проміс буде виконано з помилкою

A: 1

Q: Чи виконається метод finally, якщо вже виконався метод then()?

a) ні

b) так

A: 2

MQ: Вибери всі стани, які існують у промісу

a) Rejected

b) Success

c) Settled

d) Loading

e) Fulfilled

f) Pending

A: 1,5,6

Q: Вибери правильний синтаксис створення промісу

a) const promise = Promise(new)

b) const promise = Promise(executor)

c) const promise = new Promise(executor)

A: 3

Q: Чи виконається метод catch, якщо проміс виконався без помилки?

a) ні

b) так

A: 1

Q: Що повертає метод then()?

a) Проміс

b) undefined

c) Значення, яке поверне його колбек-функція

A: 1

Q: Що повертає виклик new Promise()?

a) undefined

b) Функцію

c) Об'єкт

A: 3

Q: У якому стані спочатку перебуває проміс?

a) Fulfilled

b) Rejected

c) Settled

d) Pending

A: 4

Q: Яким буде значення промісу, який поверне then()?

a) Завжди буде undefined

b) Значення, яке поверне його колбек-функція

A: 2

Q: Коли виконається функція, яка передається аргументом у метод finally()?

a) Після того, як проміс буде виконано успішно або з помилкою

b) Щойно проміс буде створено

c) Після того, як проміс буде виконано з помилкою

d) Після того, як проміс буде виконано успішно

A: 1

Q: Що таке проміс?

a) Об'єкт, що зберігає стан асинхронної операції

b) Об'єкт, що зберігає стан синхронної операції

c) Функція, яка повертає об'єкт

d) Функція, яка виконує асинхронну операцію

A: 1