Q: Яка головна мета використання циклів у програмуванні?

a) Оголошення змінних.

b) Виконання однотипної дії багато разів.

c) Визначення умов виконання коду.

d) Створення нових функцій.

A: 2

Q: Як називається одне виконання тіла циклу?

a) Умова виходу.

b) Ітерація.

c) Розгалуження.

d) Повторення.

A: 2

Q: Яка конструкція циклу виконує блок коду в тілі циклу доти, доки умова для виходу оцінюється як true?

a) for

b) do...while

c) while

d) for...of

A: 3

Q: Де обчислюється умова в циклі while?

a) Після кожної ітерації циклу.

b) Перед кожною ітерацією циклу.

c) Тільки перед першою ітерацією.

d) Тільки після останньої ітерації.

A: 2

Q: Що відбувається, якщо умова в циклі while оцінюється як false перед першою ітерацією?

a) Тіло циклу виконується один раз.

b) Виникає помилка.

c) Виконання циклу переривається, і скрипт продовжує виконувати інструкції після циклу.

d) Цикл перетворюється на for.

A: 3

Q: Яка особливість циклу do...while порівняно з циклом while?

a) Умова перевіряється перед першою ітерацією.

b) Тіло циклу гарантовано виконується хоча б один раз, оскільки умова перевіряється після першої ітерації.

c) Тіло циклу не виконується взагалі, якщо умова хибна.

d) Цикл не має умови виходу.

A: 2

Q: Де розміщується умова в циклі do...while?

a) Перед ключовим словом do.

b) Після ключового слова while після тіла циклу.

c) У круглих дужках одразу після ключового слова do.

d) У фігурних дужках тіла циклу.

A: 2

Q: Який цикл вважається найбільш поширеним і часто використовується, коли відома кількість ітерацій або потрібен доступ до лічильника?

a) while

b) do...while

c) for

d) for...in

A: 3

Q: З яких трьох частин складається заголовок циклу for у круглих дужках?

a) Ініціалізація лічильника, умова, ітерація лічильника.

b) Умова, тіло циклу, ініціалізація лічильника.

c) Ітерація, умова, тіло циклу.

d) Ініціалізація, тіло циклу, умова.

A: 1

Q: Що відбувається на кожній ітерації циклу for після виконання тіла циклу?

a) Перевіряється умова, а потім ініціалізується лічильник.

b) Виконується вираз ітерації лічильника.

c) Цикл негайно зупиняється.

d) Обчислюється умова виходу.

A: 2

Q: Що відбувається, якщо умова в циклі for стає хибною?

a) Виконується тіло циклу ще один раз.

b) Виконання циклу переривається.

c) Лічильник скидається до нуля.

d) Цикл переходить у стан "очікування".

A: 2

Q: Що робить оператор break всередині циклу?

a) Продовжує виконання циклу до наступної ітерації.

b) Негайно припиняє виконання поточного циклу і передає керування на наступну інструкцію після циклу.

c) Пропускає поточну ітерацію і переходить до наступної.

d) Викликає помилку.

A: 2

Q: Чи перериває оператор break виконання функції, в якій він знаходиться?

a) Так, завжди.

b) Ні, він перериває тільки цикл, але не функцію.

c) Тільки якщо цикл знаходиться в корені функції.

d) Залежить від типу циклу.

A: 2

Q: Який оператор використовується для переривання виконання і циклу, і функції, в якій він знаходиться, з можливістю повернути значення?

a) break

b) continue

c) return

d) stop

A: 3

Q: Що робить оператор continue всередині циклу?

a) Негайно припиняє виконання поточного циклу.

b) Пропускає решту коду в тілі циклу для поточної ітерації і переходить до наступної ітерації циклу.

c) Зупиняє виконання функції.

d) Викликає помилку, якщо умова хибна.

A: 2

Q: Коли доцільно використовувати оператор continue?

a) Коли потрібно повністю вийти з циклу.

b) Коли потрібно пропустити обробку певних елементів або випадків в ітерації, але продовжити цикл.

c) Коли потрібно повернути значення з функції.

d) Коли потрібно перетворити цикл на розгалуження.

A: 2

Q: Якщо умова в циклі while ніколи не стає хибною, що відбудеться?

a) Цикл виконається один раз.

b) Цикл завершиться автоматично через 100 ітерацій.

c) Цикл стане нескінченним, що може призвести до зависання програми.

d) Виникне синтаксична помилка.

A: 3

Q: У прикладі for (let i = 0; i <= 5; i += 1), скільки разів виконається тіло циклу?

a) 5

b) 6

c) 4

d) Нескінченно

A: 2

Q: Що буде виведено в консоль в результаті виконання наступного коду? let i = 0; while (i < 3) { console.log(i); i += 1; }

a) 0 1 2

b) 1 2 3

c) 0 1 2 3

d) 3 2 1

A: 1

Q: Що буде виведено в консоль в результаті виконання наступного коду? for (let j = 1; j <= 5; j += 1) { if (j === 3) { break; } console.log(j); }

a) 1 2 3

b) 1 2

c) 3 4 5

d) 1 2 4 5

A: 2

Q: Для чого використовуються розгалуження в програмуванні?

a) Для виконання одного і того ж коду багато разів.

b) Для оголошення нових змінних.

c) Для виконання різного коду залежно від умови.

d) Для створення функцій.

A: 3

Q: Яка інструкція дозволяє виконати блок коду, якщо умова є істинною, і пропустити його, якщо умова хибна?

a) switch

b) for

c) while

d) if

A: 4

Q: Що відбувається, якщо умова в інструкції if оцінюється як false?

a) Код у тілі if виконується.

b) Виникає помилка.

c) Код у тілі if пропускається, і виконання переходить до інструкцій після блоку if.

d) Програма зупиняється.

A: 3

Q: Яка інструкція використовується для виконання альтернативного блоку коду, якщо умова в інструкції if виявилася хибною?

a) switch

b) else

c) while

d) for

A: 2

Q: Яка послідовність перевірки умов є правильною при використанні if...else if...else?

a) Спочатку перевіряється else, потім if.

b) Умови перевіряються послідовно зверху вниз. Виконується код першого блоку, умова якого виявилася істинною.

c) Перевіряється тільки остання умова.

d) Перевіряються всі умови одночасно.

A: 2

Q: Чи є обов'язковим наявність блоку else після інструкції if або if...else if?

a) Так, завжди.

b) Ні, блок else є необов'язковим.

c) Тільки якщо використовується else if.

d) Тільки якщо умова в if завжди хибна.

A: 2

Q: Який оператор є більш лаконічним способом написання простого розгалуження типу if...else і часто використовується для простого присвоєння або повернення значення?

a) Логічне АБО (||)

b) Тернарний оператор (? :)

c) Логічне І (&&)

d) Оператор switch

A: 2

Q: Який синтаксис має тернарний оператор?

a) condition ? valueIfTrue : valueIfFalse

b) if (condition) { valueIfTrue } else { valueIfFalse }

c) condition && valueIfTrue || valueIfFalse

d) condition ? (valueIfTrue, valueIfFalse)

A: 1

Q: Який оператор розгалуження призначений для перевірки значення виразу на сувору рівність з декількома можливими варіантами?

a) if...else

b) Тернарний оператор

c) switch

d) Логічні оператори

A: 3

Q: З яким оператором порівняння switch суворо порівнює значення виразу з усіма значеннями в блоках case?

a) == (несуворе порівняння)

b) != (не дорівнює)

c) > (більше)

d) === (суворе порівняння)

A: 4

Q: Що робить оператор break у блоці case оператора switch?

a) Продовжує перевірку наступних блоків case.

b) Негайно перериває виконання оператора switch і передає керування коду після нього.

c) Викликає помилку.

d) Перезапускає оператор switch.

A: 2

Q: Як називається поведінка в операторі switch, коли після виконання коду в одному блоці case виконання продовжується в наступних блоках case, якщо відсутній оператор break?

a) Циклічність.

b) Ланцюг.

c) Провалювання (fall-through).

d) Рекурсія.

A: 3

Q: Для чого використовується блок default в операторі switch?

a) Для визначення початкового значення.

b) Для виконання коду за замовчуванням, якщо жодного збігу значень у блоках case не відбулося.

c) Для обробки помилок.

d) Для повторення оператора switch.

A: 2

Q: Чи потрібен оператор break після блоку default в операторі switch?

a) Так, завжди.

b) Ні, оскільки це останній блок, який може бути виконаний.

c) Тільки якщо після switch є інший код.

d) Залежить від кількості блоків case.

A: 2

Q: Що буде виведено в консоль в результаті виконання наступного коду? const age = 15; if (age >= 18) { console.log("Adult"); } else { console.log("Minor");}

a) Adult

b) Minor

c) Помилка

d) Нічого не буде виведено

A: 2

Q: Що буде значенням змінної message в результаті виконання наступного коду? const isLoggedIn = true; const message = isLoggedIn ? "Welcome!" : "Please log in.";

a) "Please log in."

b) "Welcome!"

c) `true`

d) `isLoggedIn`

A: 2

Q: Що буде виведено в консоль в результаті виконання наступного коду? const dayOfWeek = 3; switch (dayOfWeek) { case 1: console.log("Monday"); case 2: console.log("Tuesday"); default: console.log("Another day"); }

a) Another day

b) Tuesday

c) Monday Tuesday Another day

d) Tuesday Another day

A: 1

Q: Якщо потрібно перевірити діапазон значень (наприклад, чи число більше за 100), яка інструкція розгалуження є більш доцільною?

a) switch (оскільки він тільки для суворої рівності)

b) if...else if

c) Тернарний оператор

d) Обидва if...else if та switch однаково доцільні

A: 2

Q: Що буде виведено в консоль в результаті виконання наступного коду? const score = 75; if (score > 90) { console.log("Excellent"); } else if (score > 70) { console.log("Good"); } else { console.log("Satisfactory"); }

a) Excellent

b) Good

c) Satisfactory

d) Good Satisfactory

A: 2

Q: Яка інструкція розгалуження дозволяє об'єднати кілька значень case, щоб виконати один і той самий блок коду (використовуючи "провалювання")?

a) if...else if

b) Тернарний оператор

c) switch

d) Логічне АБО (||)

A: 3

Q: Що означає логічне перетворення типів в JavaScript?

a) Зміна типу даних на число.

b) Приведення значення будь-якого типу даних до логічного (булевого) значення true або false.

c) Перетворення рядка на масив.

d) Зміна типу даних на об'єкт.

A: 2

Q: Які значення завжди перетворюються на false при логічному перетворенні?

a) Будь-яке число, окрім 0.

b) Рядки, які не є порожніми.

c) 0, NaN, null, undefined, порожній рядок "".

d) Будь-який об'єкт.

A: 3

Q: Які значення перетворюються на true при логічному перетворенні?

a) Тільки число 1.

b) Тільки непорожні рядки.

c) Усі значення, окрім тих, що перетворюються на false ("truthy").

d) Тільки об'єкти та масиви.

A: 3

Q: Який логічний оператор повертає true, якщо обидва операнди є істинними (або можуть бути перетворені на true), і false в іншому випадку?

a) Логічне АБО (||)

b) Логічне І (&&)

c) Логічне заперечення (!)

d) Логічне виключне АБО (^)

A: 2

Q: Яка особливість поведінки логічного оператора І (&&) щодо "лінивих" обчислень (short-circuiting)?

a) Він завжди обчислює обидва операнди.

b) Він обчислює тільки другий операнд, якщо перший є істинним.

c) Він обчислює тільки другий операнд, якщо перший є хибним.

d) Він обчислює тільки перший операнд, якщо перший є хибним, і повертає його значення, не обчислюючи другий.

A: 4

Q: Якщо перший операнд логічного оператора І (&&) є хибним (або приводиться до false), що повертає цей оператор?

a) Значення першого операнда.

b) Значення другого операнда.

c) true.

d) undefined.

A: 1

Q: Який логічний оператор повертає true, якщо хоча б один з операндів є істинним (або може бути перетворений на true), і false тільки якщо обидва операнди є хибними?

a) Логічне І (&&)

b) Логічне АБО (||)

c) Логічне заперечення (!)

d) Логічне виключне АБО (^)

A: 2

Q: Яка особливість поведінки логічного оператора АБО (||) щодо "лінивих" обчислень (short-circuiting)?

a) Він завжди обчислює обидва операнди.

b) Він обчислює тільки другий операнд, якщо перший є хибним.

c) Він обчислює тільки другий операнд, якщо перший є істинним.

d) Він обчислює тільки перший операнд, якщо перший є істинним, і повертає його значення, не обчислюючи другий.

A: 4

Q: Якщо перший операнд логічного оператора АБО (||) є істинним (або приводиться до true), що повертає цей оператор?

a) Значення першого операнда.

b) Значення другого операнда.

c) false.

d) undefined.

A: 1

Q: Який логічний оператор повертає протилежне логічне значення свого операнда (перетворює true на false і false на true)?

a) Логічне І (&&)

b) Логічне АБО (||)

c) Логічне заперечення (!)

d) Оператор порівняння (==)

A: 3

Q: Що повертає логічне заперечення (!) при застосуванні до значення, яке перетворюється на true ("truthy")?

a) true.

b) false.

c) Оригінальне значення.

d) undefined.

A: 2

Q: Що повертає логічне заперечення (!) при застосуванні до значення, яке перетворюється на false ("falsy")?

a) false.

b) true.

c) Оригінальне значення.

d) null.

A: 2

Q: Що буде виведено в консоль в результаті обчислення 5 > 3 && 10 < 20?

a) true

b) false

c) Помилка

d) undefined

A: 1

Q: Що буде виведено в консоль в результаті обчислення 5 < 3 || 10 < 20?

a) true

b) false

c) Помилка

d) undefined

A: 1

Q: Що буде виведено в консоль в результаті обчислення !true?

a) true

b) false

c) 0

d) 1

A: 2

Q: Що буде виведено в консоль в результаті обчислення !"" (логічне заперечення порожнього рядка)?

a) false

b) true

c) ""

d) null

A: 2

Q: Що буде виведено в консоль в результаті обчислення 0 || "Default Value"?

a) 0

b) "Default Value"

c) true

d) false

A: 2

Q: Що буде виведено в консоль в результаті обчислення "Hello" && 123?

a) "Hello"

b) 123

c) true

d) false

A: 2

Q: Якщо потрібно виконати дію, тільки якщо обидві умови є істинними, який логічний оператор слід використовувати?

a) Логічне АБО (||)

b) Логічне І (&&)

c) Логічне заперечення (!)

d) Оператор порівняння.

A: 2

Q: Якщо потрібно виконати дію, якщо хоча б одна з умов є істинною, який логічний оператор слід використовувати?

a) Логічне І (&&)

b) Логічне АБО (||)

c) Логічне заперечення (!)

d) Оператор присвоєння.

A: 2