

Рекомендації з використання керуючих конструкцій

Якщо потрібно перевірити умову, треба використати *if-else*

Якщо при виконанні/невиконанні умови треба виконати тільки одну команду(оператор)

Якщо треба виконувати якусь одну команду **як при виконанні**, так і одну команду при **невиконанні** деякої умови

```
if (<умова>
    <оператор1>
else
    <оператор2>
```

```
// 3-клавіатури вводитьься вік. Треба вивести
// чи може користувач водити авто (18 років)
const MIN_DRIVER_AGE = 18
const userAge = parseInt(prompt('Введіть кількість років'))

let resultMessage
if (userAge >= MIN_DRIVER_AGE)
    resultMessage = 'Користувача може водити'
else
    resultMessage = 'Користувача не може водити'

document.write(resultMessage)
```

Якщо треба виконувати одну команду при **виконанні** деякої умови. Якщо умова **не виконується, то нічого** робити не треба

```
if (<умова>
    <оператор1>
```

```
// 3-клавіатури вводитьься розмір заробітної плати.
// Якщо розмір заробітної плати менше мінімальної (3500)
// змінити розмір заробітної плати на мінімальну

const MIN_SALARY_VALUE = 3500
let userSalary = parseFloat(prompt('Введіть розмір зароб. плати'))

if (userSalary < MIN_SALARY_VALUE)
    userSalary = MIN_SALARY_VALUE

document.write(`Кінцева зар.плата = ${userSalary}`)
```

Якщо потрібно перевірити умову, треба використати *if-else*

Якщо при виконанні/невиконанні умови треба виконати декілька команд(операторів) (використати фігурні дужки)

Якщо треба виконувати декілька команд **як при виконанні**,
так і невиконанні деякої умови

```
if (<умова>)  
{  
    <оператор1.1>  
    . . . . .  
    <оператор1.N>  
}  
else  
{  
    <оператор2.1>  
    . . . . .  
    <оператор2.M>  
}
```

```
// 3-клавіатури вводить кількість одиниць товару, що хоче придбати користувач.  
// Якщо кількість більша за 10, то привітати, повідомити про елітний статус і визначити статус "1"  
// інаше -- привітати користувача, і визначити статус "i"  
const MIN_DISCOUNT_LIMIT = 10, DISCOUNT_VALUE = 20  
const ELIT_USER_STATUS = 1, REGULAR_USER_STATUS = 2  
  
const productsNumber = parseInt(  
    prompt('Введіть кількість одиниць товару')  
)  
  
let userStatus, invitationMessage  
if (productsNumber >= MIN_DISCOUNT_LIMIT) {  
    invitationMessage = `Вітаємо! Ви є елітним користувачем`  
    userStatus = ELIT_USER_STATUS  
}  
else {  
    invitationMessage = `Вітаємо!`  
    userStatus = REGULAR_USER_STATUS  
}  
  
alert(invitationMessage)  
document.write(`Статус користуча : ${userStatus}`)
```

Якщо потрібно перевірити умову, треба використати *if-else*

Якщо при виконанні/невиконанні умови треба виконати декілька команд(операторів) (використати фігурні дужки)

Якщо треба виконувати деякі команди
при виконанні деякої умови. Якщо умова
не виконується, то нічого робити не треба

```
if (<умова>)  
{  
    <оператор1>  
    . . . . .  
    <операторN>  
}
```

```
//Користувач вносить деяку кількість грошей на рахунок. Якщо розмір внеску  
//більше 500 грн, то повідомити користувача про знижку і збільшити баланс на 20%.  
  
const MIN_DISCOUNT_LIMIT = 500, DISCOUNT_VALUE=20  
let userMoney = parseFloat(prompt('Скільки коштів ви вносите?'))  
  
if (userMoney > MIN_DISCOUNT_LIMIT) {  
    alert('Ви маєте знижку у розмірі 20%')  
    userMoney += userMoney*DISCOUNT_VALUE/100  
}  
  
document.write(`Кінцева зар.плата = ${userSalary}`)
```

Якщо значення виразу може дорівнювати **одному зі 2-х можливий значень** у залежності від умови - **тернарний оператор**

`умова ? значення1 (умова викон.) : значення2 (умова не викон.)`

```
//З клавіатури вводиться ім'я користувачки та статус (чи заміжня).  
//Визначити префікс (Miss - незаміжня/Mrs - заміжня) та привітати користувачку  
  
const userName = prompt('What is your name?')  
const isMerriedAnswer = prompt('Are yu merried (Y/N)?')  
  
//..... префікс = ..... якщо_заміжня ..... то "Mrs" інакше "Miss"  
const userPrefix = isMerriedAnswer === 'Y' ? 'Mrs' : 'Miss'  
  
document.write(`Welcome ${userPrefix} ${userName} !`)
```

Якщо потрібно виконати якісь команди у залежності від значення деякої величини(селектора), для якої кількість значень невелика (їх можна перерахувати), то можна використати switch

```
switch (селектор) {  
    case значення 1 : оператори  
        break;  
    case значення 2 : оператори  
        break;  
    . . . . .  
    case значення n : оператори  
        break;  
  
    default:  
        оператори  
}
```

```
//З клавіатури вводиться скорочене позначення мови (ua, en, fr, de).  
//Вивести назву цієї мови.  
//(ua-українська, en-англійська, fr-французька, de-німецька).  
const shortLocaleTitle = prompt('Input locale (ua, en, fr, de)')  
  
let localeTitle  
switch (shortLocaleTitle) {  
    case 'ua': localeTitle = 'українська'  
        break  
    case 'en': localeTitle = 'англійська'  
        break  
    case 'fr': localeTitle = 'французька'  
        break  
    case 'de': localeTitle = 'німецька'  
        break  
    default: localeTitle = 'невідома мова'  
        break  
}  
  
document.write(localeTitle)
```

Цикли. Коли відома кількість повторень

Загальна форма	<pre>for (<u>лічильник</u> = 0; <u>лічильник</u> < <u>кількість повторень</u> ; <u>лічильник</u> ++) { оператор ; }</pre>
Приклад, розв'язаний з використанням for	<pre>//----- 100 разів вивести на екран слово «Мир» ----- <u>for(let i = 0 ; i < 100 ; i++)</u> { document.write('Мир'); }</pre>

Цикли. Коли відома кількість повторень

Загальна
форма

```
for ( лічильник = 0; лічильник < кількість повторень ; лічильник ++ )  
{  
    оператор ;  
}
```

Задача. Користувач вводить кількість потрібних випадкових чисел від 1 до 100. Вивести ці числа у формі нумерованого списку (з використанням)

Цикли. Коли відома кількість повторень

Загальна
форма

```
for ( лічильник = 0; лічильник < кількість повторень ; лічильник ++ )  
{  
    оператор ;  
}
```

Задача. Користувач вводить кількість потрібних випадкових чисел від 1 до 100. Вивести ці числа у формі нумерованого списку (з використанням)

```
//----- алгоритм -----  
// ввести кількість випадкових чисел requiredRandomNumber  
// виводимо відкриваючий тег <ol> . . . . . <-- тільки один раз  
// . . повторити requiredRandomNumber разів  
// . . . . . - згенерувати випадкове число randomNumber  
// . . . . . - вивести <li> randomNumber </li>  
// виводимо закриваючий тег </ol> . . . . . <-- тільки один раз
```

Цикли. Коли відома кількість повторень

Загальна форма	<pre>for (лічильник = 0; лічильник < кількість повторень; лічильник++) { оператор; }</pre>
----------------	---

```
//----- алгоритм -----  
// ввести кількість випадкових чисел requiredRandomNumber  
// виводимо відкриваючий тег <ol> ..... <-- тільки один раз  
// повторити requiredRandomNumber разів  
// ..... згенерувати випадкове число randomNumber  
// ..... вивести <li> randomNumber </li>  
// виводимо закриваючий тег </ol> ..... <-- тільки один раз
```

Задача. Користувач вводить кількість потрібних випадкових чисел від 1 до 100. Вивести ці числа у формі нумерованого списку (з використанням)

```
const MIN_RANDOM_VALUE = 1,  
      MAX_RANDOM_VALUE = 100  
  
const requiredRandomNumber = parseInt(  
  prompt('Кількість випадкових чисел')  
)  
document.write('<ol>')  
for (let i = 0; i < requiredRandomNumber; i++) {  
  const randomNumber =  
    MIN_RANDOM_VALUE +  
    Math.floor(Math.random() * (MAX_RANDOM_VALUE - MIN_RANDOM_VALUE + 1))  
  document.write(`<li>${randomNumber}</li>`)  
}  
document.write('</ol>')
```

Цикли. Коли відома кількість повторень

Задача. Тестування додавання.

Користувач вводить кількість прикладів та максимальне число. Провести перевірку користувача і підрахувати кількість правильних

Цикли. Коли відома кількість повторень

Задача. Тестування додавання.

Користувач вводить кількість прикладів та максимальне число. Провести перевірку користувача і підрахувати кількість правильних

```
//----- алгоритм -----  
  
// ввести кількість прикладів testNumber  
// спочатку кількість правильних correctAnswersNumber = 0  
// повторити testNumber разів  
// ..... згенерувати перше число firstNumber  
// ..... згенувати друге число secondNumber  
// ..... обчислити правильний результат  
// ..... вивести запитання і ввести відповідь userAnswer  
// ..... якщо відповідь правильна (userAnswer === correctAnswer)  
// ..... то  
// ..... привітати користувача  
// ..... кількість_правильних збільшити на 1  
// ..... інакше  
// ..... вивести правильну відповідь  
// вивести кількість правильних
```

```
//----- алгоритм -----
```

```
// ввести кількість прикладів testNumber  
// спочатку кількість правильних correctAnswersNumber = 0  
// повторити testNumber разів  
// ----- згенерувати перше число firstNumber  
// ----- згенерувати друге число secondNumber  
// ----- обчислити правильний результат  
// ----- вивести запитання і ввести відповідь userAnswer  
// ----- якщо відповідь правильна (userAnswer === correctAnswer)  
// ----- то  
// ----- привітати користувача  
// ----- кількість_правильних збільшити на 1  
// ----- інакше  
// ----- вивести правильну відповідь  
// вивести кількість правильних
```

Цикли. Коли відома кількість повторень

```
const testNumber = parseInt(prompt('Кількість прикладів'))  
const maxNumber = parseInt(prompt('Максимальне число'))  
  
let correctAnswersNumber = 0  
for (let i = 0; i < testNumber; i++) {  
  const firstNumber = Math.floor(Math.random() * (maxNumber + 1))  
  const secondNumber = Math.floor(Math.random() * (maxNumber + 1))  
  const correctAnswer = firstNumber + secondNumber  
  const userAnswer = parseInt(  
    prompt(`${firstNumber} + ${secondNumber} =`)  
  )  
  if (userAnswer === correctAnswer) {  
    alert('Ok')  
    correctAnswersNumber++  
  } else  
    alert(`Помилка! ${firstNumber} + ${secondNumber} = ${correctAnswer}`)  
}  
document.write(  
  `Тест закінчено. Кількість правильних : ${correctAnswersNumber}`  
)
```

Цикли. Коли відомі початкове, кінцеве значення та крок зміни деякої величини (параметра)

Загальна форма	<pre>for (параметр = <u>поч.значення</u>; параметр <= <u>кін.значення</u>; параметр = параметр + <u>крок</u>) { оператор ; }</pre>
Приклад, розв'язаний з використанням for	<pre>//----- Вивести на екран усі числа кратні трьом <u>від 6 до 28</u>. for (<u>var i = 6</u>; <u>i <= 28</u>; <u>i=i+3</u>) { document.write(<u>i</u>); }</pre>

Коли відома умова продовження

```
while( умова )  
{  
    оператор1  
    . . . . .  
    операторN  
}
```

Задача. Користувач вводить наявну кількість грошей. Поки є гроші (не менше 25 грн.) пропонувати товари
(1.Сік - 25 грн. 2. Коктейль - 120 грн. 3.Фреш - 70 грн.)
Підрахувати кількість куплених товарів.

Коли відома умова продовження

```
while( умова )  
{  
    оператор1  
    . . . . .  
    операторN  
}
```

Задача. Користувач вводить наявну кількість грошей. Поки є гроші (не менше 25 грн.) пропонувати товари
(1.Сік - 25 грн. 2. Коктейль - 120 грн. 3.Фреш - 70 грн.)
Підрахувати кількість куплених товарів.

```
//----- алгоритм -----  
//Ввести кількість наявних грошей у користувача (userMoney)  
//Кількість куплених товарів спочатку дорівнює 0 (selectedProductNumber=0)  
//поки (кількість грошей більше або рівне 25) повторювати  
//.....- запропонувати товар і отримати відповідь userChoice  
//.....- проаналізувати вибір користувача userChoice  
//.....* випадок "1" тоді відняти від суми грошей 25 грн  
//.....* випадок "2" тоді відняти від суми грошей 120 грн  
//.....* випадок "3" тоді відняти від суми грошей 70 грн  
//.....- кількість товарів збільшити на 1 (selectedProductNumber++)  
//Вивести кількість куплених товарів
```


Коли відома умова продовження

Задача. Користувач вводить наявну кількість грошей. Поки є гроші (не менше 25 грн.) пропонувати товари (1. Сік - 25 грн. 2. Коктейль - 120 грн. 3. Фреш - 70 грн.)
Підрахувати кількість куплених товарів.

```
//----- алгоритм -----  
//Ввести кількість наявних грошей у користувача (userMoney)  
//Кількість куплених товарів спочатку дорівнює 0 (selectedProductNumber)  
//поки (кількість грошей більше або рівне 25) повторювати  
//..... запропонувати товар і отримати відповідь userChoice  
//..... проаналізувати вибір користувача userChoice  
//..... * випадок "1" тоді відняти від суми грошей 25 грн  
//..... * випадок "2" тоді відняти від суми грошей 120 грн  
//..... * випадок "3" тоді відняти від суми грошей 70 грн  
//..... кількість товарів збільшити на 1 (selectedProductNumber++)  
//Вивести кількість куплених товарів
```

```
while( умова )  
{  
    оператор1  
    . . . . .  
    операторN  
}
```

```
const MIN_MONEY_LIMIT = 120  
let userMoney = parseFloat(prompt('Кількість грошей'))  
let selectedProductNumber = 0  
while (userMoney >= MIN_MONEY_LIMIT) {  
    const userAnswer = parseInt(  
        prompt('1. Сік - 25 грн.\n2. Коктейль - 120грн.\n3. Фреш - 70грн.')  
    )  
    switch (userAnswer) {  
        case 1:  
            userMoney -= 25  
            break  
        case 2:  
            userMoney -= 120  
            break  
        case 3:  
            userMoney -= 70  
            break  
        default:  
            alert('Немає такого товару!')  
            break  
    }  
    selectedProductNumber++  
}  
document.write(`Було куплено : ${selectedProductNumber} товарів`)
```

Коли відома умова продовження. Але принаймні один раз треба виконати

```
do  
{  
    оператор  
}  
while ( умова )
```

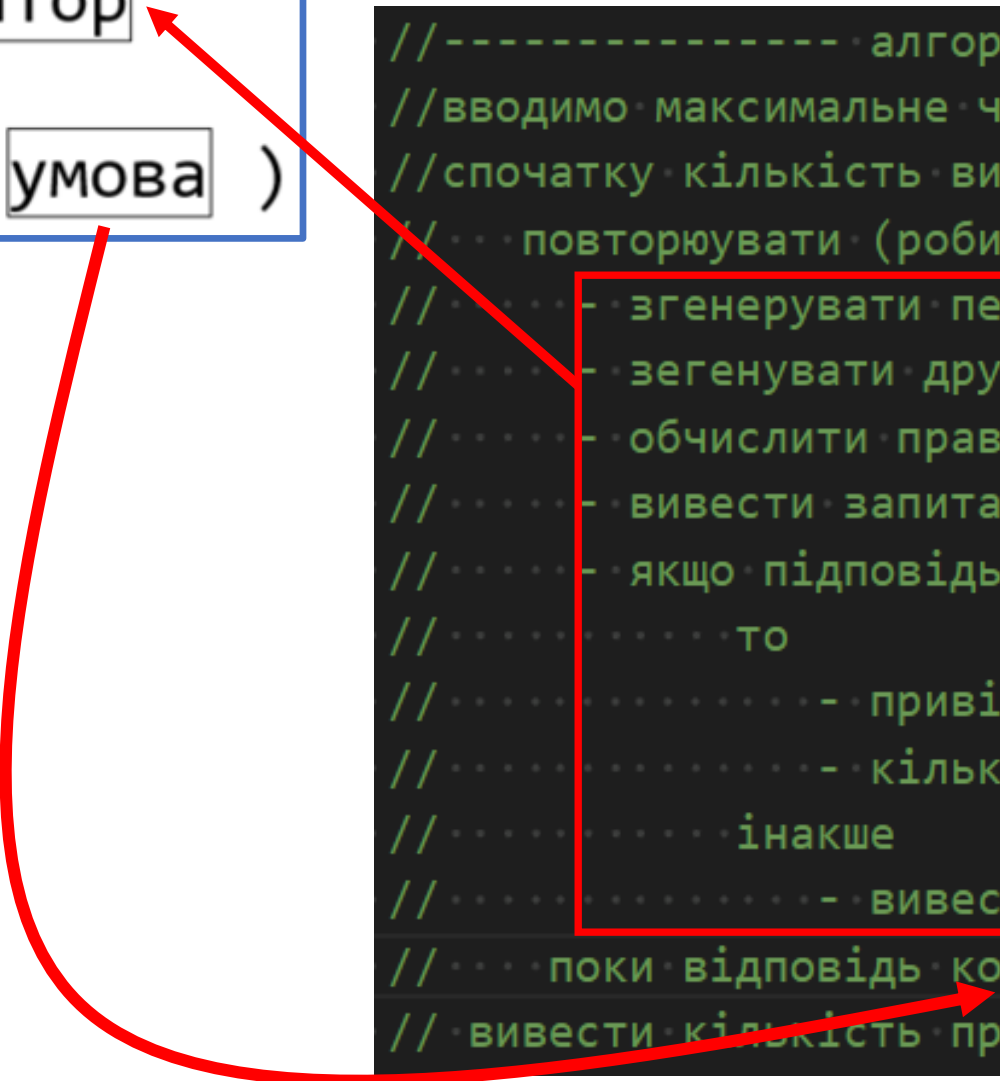
Коли відома умова продовження. Але принаймні один раз треба виконати

```
do
{
    оператор
}
while ( умова )
```

Задача. Тестування додавання. Користувач вводить максимальне число. Генеруємо приклади поки користувач не помилиться. Підрахувати кількість виконаних прикладів

Коли відома умова продовження. Але принаймні один раз треба виконати

```
do
{
    оператор
}
while ( умова )
```



Задача. Тестування додавання. Користувач вводить максимальне число. Генеруємо приклади поки користувач не помилиться. Підрахувати кількість виконаних прикладів

```
//----- алгоритм -----
//вводимо максимальне число maxTestNumer
//спочатку кількість виконаних прикладів completedTestsNumber = 0
//... повторювати (робити)
//... - згенерувати перше число (firstNumber)
//... - згенерувати друге число (secondNumber)
//... - обчислити правильний результат (correctAnswer)
//... - вивести запитання і ввести відповідь (userAnswer)
//... - якщо відповідь правильна (userAnswer == correctAnswer)
//... то
//... - привітати користувача
//... - кількість_правильних збільшити на 1
//... інакше
//... - вивести правильну відповідь
//... поки відповідь користувача є правильною
// вивести кількість правильних
```

Коли відома умова продовження. Але принаймні один раз треба виконати

```
//----- алгоритм -----  
//вводимо максимальне число maxTestNumber  
//спочатку кількість виконаних прикладів completedTestsNumber = 0  
//... повторювати (робити)  
//..... згенерувати перше число (firstNumber)  
//..... згенерувати друге число (secondNumber)  
//..... обчислити правильний результат (correctAnswer)  
//..... вивести запитання і ввести відповідь (userAnswer)  
//..... якщо відповідь правильна (userAnswer === correctAnswer)  
//..... то  
//..... привітати користувача  
//..... кількість_правильних збільшити на 1  
//..... інакше  
//..... вивести правильну відповідь  
//... поки відповідь користувача є правильною  
// вивести кількість правильних
```

```
const maxNumber = parseInt(prompt('Максимальне число'))  
  
let userAnswer, correctAnswer  
  
let completedTestsNumber = 0  
do {  
  const firstNumber = Math.floor(Math.random() * (maxNumber + 1))  
  const secondNumber = Math.floor(Math.random() * (maxNumber + 1))  
  correctAnswer = firstNumber + secondNumber  
  userAnswer = parseInt(prompt(`${firstNumber} + ${secondNumber} =`))  
  if (userAnswer === correctAnswer) {  
    alert('Ok')  
  } else  
    alert(`Помилка! ${firstNumber} + ${secondNumber} = ${correctAnswer}`)  
  completedTestsNumber++  
} while (userAnswer !== correctAnswer)  
  
document.write(`Виконано прикладів : ${completedTestsNumber}`)
```