

Лабораторна робота №4.

Асинхронний режим у JS

Мета роботи

Робота із скриптовою мовою програмування JS в асинхронному режимі.

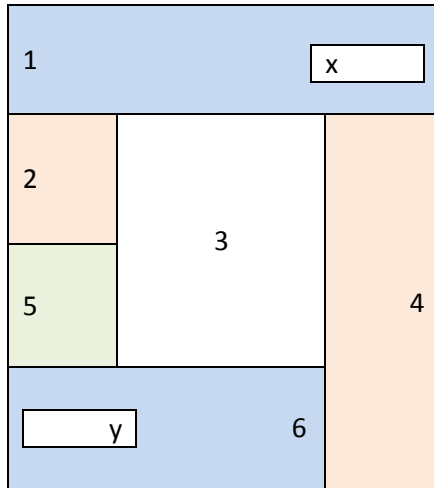
Тривалість роботи

2 години

Завдання

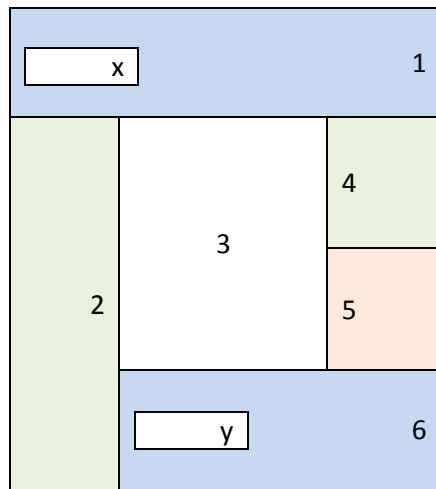
1. У вас вже є обраний із 16 варіантів власний варіант виконання роботи відповідно до вашого порядкового номера у списку групи, подальші маніпуляції виконуйте на зверстаній сторінці веб-сайта з л.р. №2.
2. За допомогою мови програмування JS виконайте вказані обчислення та проведіть необхідні маніпуляції з документом згідно з вашим варіантом, зберігаючи контент веб-сторінки, з урахуванням позиціювання та візуалізації окремих елементів сторінки (як в л.р.№2).
3. Для запису скриптів JS використовуйте будь-які відомі вам способи.
4. Продемонструйте роботу новостворених скриптів через локальну файлову систему або через обраний в л.р.№1 хостінг.

Варіант 1



1. Послідовно в порядку нумерації зчитайте вміст номерних блоків контенту. Переставте кожен зчитаний контент на місце попереднього так, щоб кожне переставлення відбувалося із наростаючою (+5сек.) затримкою виконання. Після вичерпання блоків контенту зупиніть виконання скрипта.
2. Змініть колір тексту у блоках «1» і «6» через 5 секунд після настання вашої події із завдання №5 л.р. №3. Забезпечте зміну кольору тексту в блоці «3» через кожні 5 секунд.
3. За допомогою асинхронного з'єднання без перезавантаження всієї сторінки зчитайте коміти із Github для вказаного користувачем аккаунта та репозиторія у формі на цій сторінці. Якщо за вказаними користувачем даними неможливо зчитати коміти – зафіксуйте помилку. Вивід отриманих комітів оформіть у вигляді нenumerованого списку у вигляді "**<commit.author.name>: <commit.message>**". Вивід помилки оформіть написом у вигляді "**Error: <error.status> (<error.statusText>)**" в межах прямокутної червоної плашки. Для доступу до Github використовуйте їхній API.
4. Напишіть функцію, яка приймає два колбеки та викликає їх по порядку. Функцію реалізуйте за шаблоном колбека. Для підтвердження роботи кожної функції фіксуйте повідомлення у консолі.
5. Напишіть функцію сортування списку значень, отриманого із форми користувача, з допомогою Insertion sort. Зайві (не числові) значення фільтруйте за допомогою регулярного виразу.

Варіант 2

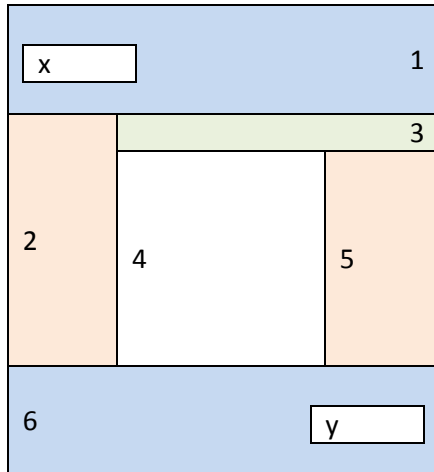


1. Послідовно в порядку нумерації зчитайте вміст номерних блоків контенту. Переставте кожен зчитаний контент на місце попереднього так, щоб кожне переставлення відбувалося із наростаючою (+5сек.) затримкою виконання. Після вичерпання блоків контенту зупиніть виконання скрипта.

2. Змініть колір фону у блоках «1» і «6» через 5 секунд після настання вашої події із завдання №5 л.р. №3. Забезпечте зміну кольору фону в блоці «3» через кожні 5 секунд.

3. За допомогою асинхронного з'єднання без перезавантаження всієї сторінки зчитайте комміти із Github для вказаного користувачем аккаунта та репозиторія у формі на цій сторінці. Якщо за вказаними користувачем даними неможливо зчитати комміти – зафіксуйте помилку. Вивід отриманих коммітів оформіть у вигляді нумерованого списку у вигляді "**<commit.author.name>: <commit.message>**". Вивід помилки оформіть написом у вигляді "**Error: <error.status> (<error.statusText>)**" в межах прямокутної червоної плашки. Для доступу до Github використовуйте їхній API.
4. Напишіть функцію, яка приймає два колбеки та викликає їх по порядку. Функцію реалізуйте за шаблоном колбека. Для підтвердження роботи кожної функції фіксуйте повідомлення у консолі.
5. Напишіть функцію сортування списку значень, отриманого із форми користувача, з допомогою Quick sort. Зайві (не числові) значення фільтруйте за допомогою регулярного виразу.

Варіант 3



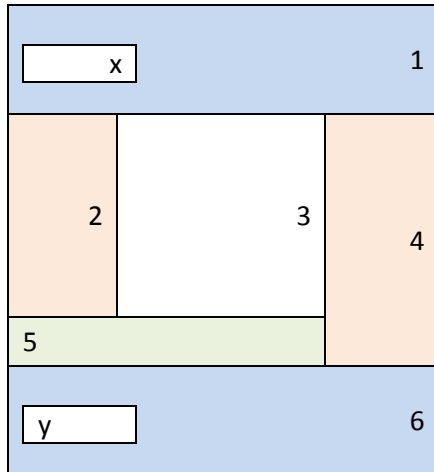
1. Послідовно в порядку нумерації зчитайте вміст номерних блоків контенту. Переставте кожен зчитаний контент на місце наступного так, щоб кожне переставлення відбувалося із наростаючою (+5сек.) затримкою виконання. Після вичерпання блоків контенту зупиніть виконання скрипта.

2. Змініть властивість «жирний» для текстів у блоках «1» і «6» через 5 секунд після настання вашої події із завдання №5 л.р. №3.

Забезпечте зміну властивості «жирний» для тексту в блоці «4» через кожні 5 секунд.

3. За допомогою асинхронного з'єднання без перезавантаження всієї сторінки зчитайте комміти із Github для вказаного користувачем аккаунта та репозиторія у формі на цій сторінці. Якщо за вказаними користувачем даними неможливо зчитати комміти – зафіксуйте помилку. Вивід отриманих коммітів оформіть у вигляді нумерованого списку у вигляді "**<commit.author.name>:
<commit.message>**". Вивід помилки оформіть написом у вигляді "**Error: <error.status> (<error.statusText>)**" в межах прямокутної червоної плашки. Для доступу до Github використовуйте їхній API.
4. Напишіть функцію, яка приймає два колбеки та викликає їх по порядку. Функцію реалізуйте за шаблоном колбека. Для підтвердження роботи кожної функції фіксуйте повідомлення у консолі.
5. Напишіть функцію сортування списку значень, отриманого із форми користувача, з допомогою Quick sort. Зайві (не числові) значення фільтруйте за допомогою регулярного виразу.

Варіант 4

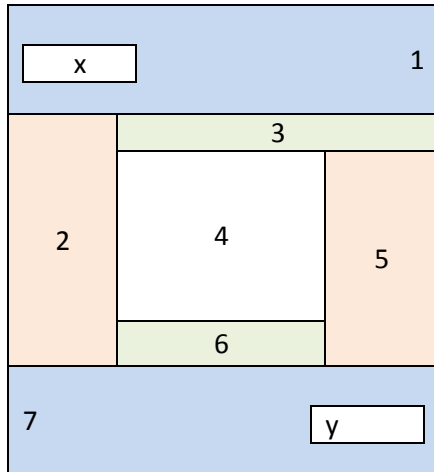


1. Послідовно в порядку нумерації зчитайте вміст номерних блоків контенту. Переставте кожен зчитаний контент на місце наступного так, щоб кожне переставлення відбувалося із 5-секундною затримкою виконання. Після вичерпання блоків контенту зупиніть виконання скрипта.

2. Змініть колір тексту у блоків «1» і «7» через 5 секунд після настання вашої події із завдання №5 л.р. №3. Забезпечте зміну кольору тексту блоку «4» через кожні 5 секунд.

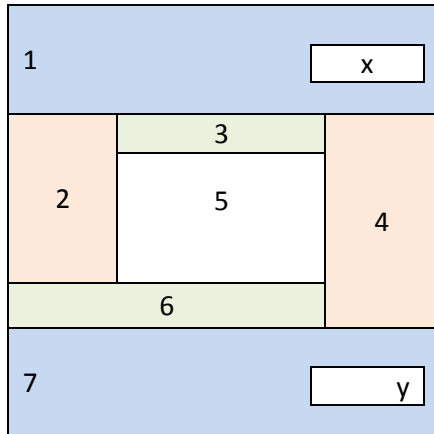
3. За допомогою асинхронного з'єднання без перезавантаження всієї сторінки зчитайте комміти із Github для вказаного користувачем аккаунта та репозиторія у формі на цій сторінці. Якщо за вказаними користувачем даними неможливо зчитати комміти – зафіксуйте помилку. Вивід отриманих коммітів оформіть у вигляді нумерованого списку у вигляді "**<commit.author.name>: <commit.message>**". Вивід помилки оформіть написом у вигляді "**Error: <error.status> (<error.statusText>)**" в межах прямокутної червоної плашки. Для доступу до Github використовуйте їхній API.
4. Напишіть функцію, яка приймає два колбеки та викликає їх по порядку. Функцію реалізуйте за шаблоном колбека. Для підтвердження роботи кожної функції фіксуйте повідомлення у консолі.
5. Напишіть функцію сортування списку значень, отриманого із форми користувача, з допомогою Selection sort. Зайві (не числові) значення фільтруйте за допомогою регулярного виразу.

Варіант 5



1. Послідовно в порядку нумерації зчитайте вміст номерних блоків контенту. Переставте кожен зчитаний контент на місце наступного так, щоб кожне переставлення відбувалося із 5-секундною затримкою виконання. Після вичерпання блоків контенту зупиніть виконання скрипта.
2. Змініть колір тексту у блоків «1» і «7» через 5 секунд після настання вашої події із завдання №5 л.р. №3. Забезпечте зміну кольору тексту блоку «4» через кожні 5 секунд.
3. За допомогою асинхронного з'єднання без перезавантаження всієї сторінки зчитайте комміти із Github для вказаного користувачем аккаунта та репозиторія у формі на цій сторінці. Якщо за вказаними користувачем даними неможливо зчитати комміти – зафіксуйте помилку. Вивід отриманих коммітів оформіть у вигляді нумерованого списку у вигляді "**<commit.author.name>:
<commit.message>**". Вивід помилки оформіть написом у вигляді "**Error:
<error.status> (<error.statusText>)**" в межах прямокутної червоної плашки. Для доступу до Github використайте їхній API.
4. Напишіть функцію, яка приймає два колбеки та викликає їх по порядку. Функцію реалізуйте за шаблоном колбека. Для підтвердження роботи кожної функції фіксуйте повідомлення у консолі.
5. Напишіть функцію сортування списку значень, отриманого із форми користувача, з допомогою Selection sort. Зайві (не числові) значення фільтруйте за допомогою регулярного виразу.

Варіант 6



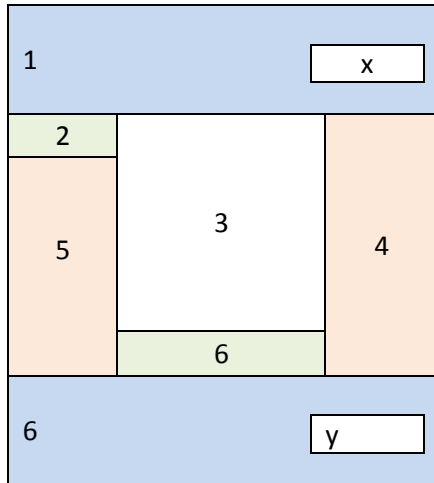
1. Послідовно в порядку нумерації зчитайте вміст номерних блоків контенту. Переставте кожен зчитаний контент на місце наступного так, щоб кожне переставлення відбувалося із наростаючою (+5сек.) затримкою виконання. Після вичерпання блоків контенту зупиніть виконання скрипта.

2. Встановіть властивість вирівнювання по лівому краю для текстів у блоках «1» і «7» через 5 секунд після настання вашої події із завдання №5 л.р. №3.

Забезпечте циклічне вмикання/вимикання властивості «курсив» для тексту в блоці «5» через кожні 5 секунд.

3. За допомогою асинхронного з'єднання без перезавантаження всієї сторінки зчитайте комміти із Github для вказаного користувачем аккаунта та репозиторія у формі на цій сторінці. Якщо за вказаними користувачем даними неможливо зчитати комміти – зафіксуйте помилку. Вивід отриманих коммітів оформіть у вигляді нумерованого списку у вигляді "**<commit.author.name>**
<commit.message>". Вивід помилки оформіть написом у вигляді "**Error: <error.status> (<error.statusText>)**" в межах прямокутної червоної плашки. Для доступу до Github використовуйте їхній API.
4. Напишіть функцію, яка приймає два колбеки та викликає їх по порядку. Функцію реалізуйте за шаблоном колбека. Для підтвердження роботи кожної функції фіксуйте повідомлення у консолі.
5. Напишіть функцію сортування списку значень, отриманого із форми користувача, з допомогою Quick sort. Зайві (не числові) значення фільтруйте за допомогою регулярного виразу.

Варіант 7



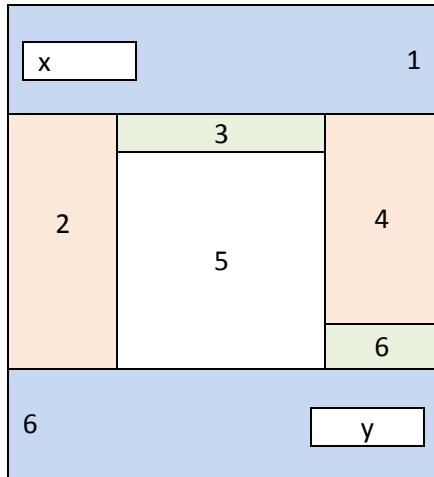
1. Послідовно в порядку нумерації зчитайте вміст номерних блоків контенту. Переставте кожен зчитаний контент на місце наступного так, щоб кожне переставлення відбувалося із 5-секундною затримкою виконання. Після вичерпання блоків контенту зупиніть виконання скрипта.

2. Встановіть властивість вирівнювання по лівому краю для текстів у блоках «1» і «6» через 5 секунд після настання вашої події із завдання №5

л.р. №3. Забезпечте циклічне вмикання/вимикання властивості вирівнювання по лівому краю для тексту в блоці «3» через кожні 5 секунд.

3. За допомогою асинхронного з'єднання без перезавантаження всієї сторінки зчитайте комміти із Github для вказаного користувачем аккаунта та репозиторія у формі на цій сторінці. Якщо за вказаними користувачем даними неможливо зчитати комміти – зафіксуйте помилку. Вивід отриманих коммітів оформіть у вигляді нумерованого списку у вигляді "**<commit.author.name>:
<commit.message>**". Вивід помилки оформіть написом у вигляді "**Error: <error.status> (<error.statusText>)**" в межах прямокутної червоної плашки. Для доступу до Github використовуйте їхній API.
4. Напишіть функцію, яка приймає два колбеки та викликає їх по порядку. Функцію реалізуйте за шаблоном колбека. Для підтвердження роботи кожної функції фіксуйте повідомлення у консолі.
5. Напишіть функцію сортування списку значень, отриманого із форми користувача, з допомогою Selection sort. Зайві (не числові) значення фільтруйте за допомогою регулярного виразу.

Варіант 8



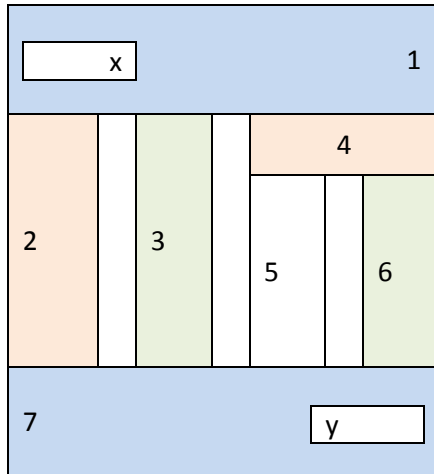
1. Послідовно в порядку нумерації зчитайте вміст номерних блоків контенту. Переставте кожен зчитаний контент на місце наступного так, щоб кожне переставлення відбувалося із 5-секундною затримкою виконання. Після вичерпання блоків контенту зупиніть виконання скрипта.

2. Встановіть властивість «жирний» для текстів у блоках «1» і «7» через 5 секунд після настання вашої події із завдання №5 л.р. №3.

Забезпечте циклічне вмикання/вимикання властивості «жирний» для тексту в блоці «5» через кожні 5 секунд.

3. За допомогою асинхронного з'єднання без перезавантаження всієї сторінки зчитайте комміти із Github для вказаного користувачем аккаунта та репозиторія у формі на цій сторінці. Якщо за вказаними користувачем даними неможливо зчитати комміти – зафіксуйте помилку. Вивід отриманих коммітів оформіть у вигляді нумерованого списку у вигляді "**<commit.author.name>:
<commit.message>**". Вивід помилки оформіть написом у вигляді "**Error: <error.status> (<error.statusText>)**" в межах прямокутної червоної плашки. Для доступу до Github використовуйте їхній API.
4. Напишіть функцію, яка приймає два колбеки та викликає їх по порядку. Функцію реалізуйте за шаблоном колбека. Для підтвердження роботи кожної функції фіксуйте повідомлення у консолі.
5. Напишіть функцію сортування списку значень, отриманого із форми користувача, з допомогою Selection sort. Зайві (не числові) значення фільтруйте за допомогою регулярного виразу.

Варіант 9



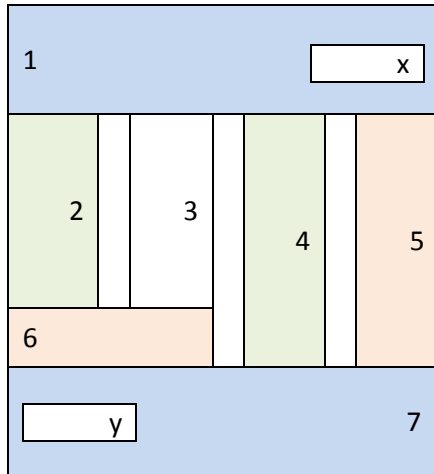
1. Послідовно в порядку нумерації зчитайте вміст номерних блоків контенту. Переставте кожен зчитаний контент на місце наступного так, щоб кожне переставлення відбувалося із 5-секундною затримкою виконання. Після вичерпання блоків контенту зупиніть виконання скрипта.

2. Встановіть вирівнювання по правому краю для текстів у блоках «1» і «7» через 5 секунд після настання вашої події із завдання №5 л.р. №3.

Забезпечте циклічне вмикання/вимикання властивості «жирний» для тексту в блоці «5» через кожні 5 секунд.

3. За допомогою асинхронного з'єднання без перезавантаження всієї сторінки зчитайте комміти із Github для вказаного користувачем аккаунта та репозиторія у формі на цій сторінці. Якщо за вказаними користувачем даними неможливо зчитати комміти – зафіксуйте помилку. Вивід отриманих коммітів оформіть у вигляді нумерованого списку у вигляді "**<commit.author.name>**
<commit.message>". Вивід помилки оформіть написом у вигляді "**Error: <error.status> (<error.statusText>)**" в межах прямокутної червоної плашки. Для доступу до Github використайте їхній API.
4. Напишіть функцію, яка приймає два колбеки та викликає їх по порядку. Функцію реалізуйте за шаблоном колбека. Для підтвердження роботи кожної функції фіксуйте повідомлення у консолі.
5. Напишіть функцію сортування списку значень, отриманого із форми користувача, з допомогою Selection sort. Зайві (не числові) значення фільтруйте за допомогою регулярного виразу.

Варіант 10



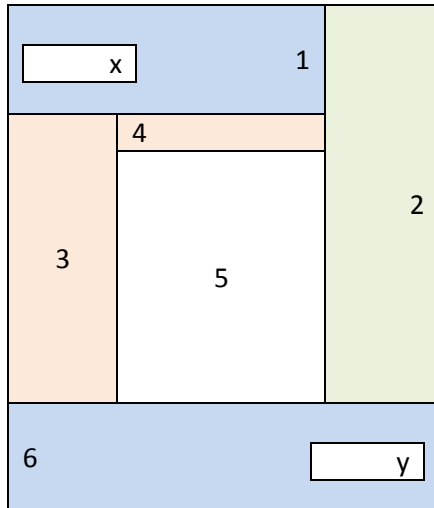
1. Послідовно в порядку нумерації зчитайте вміст номерних блоків контенту. Переставте кожен зчитаний контент на місце попереднього так, щоб кожне переставлення відбувалося із 5-секундною затримкою виконання. Після вичерпання блоків контенту зупиніть виконання скрипта.

2. Встановіть властивість «курсив» для текстів у блоках «1» і «6» через 5 секунд після настання вашої події із завдання №5 л.р. №3.

Забезпечте циклічне вмикання/вимикання властивості «курсив» для тексту в блоці «5» через кожні 5 секунд.

3. За допомогою асинхронного з'єднання без перезавантаження всієї сторінки зчитайте комміти із Github для вказаного користувачем аккаунта та репозиторія у формі на цій сторінці. Якщо за вказаними користувачем даними неможливо зчитати комміти – зафіксуйте помилку. Вивід отриманих коммітів оформіть у вигляді нумерованого списку у вигляді "**<commit.author.name>:
<commit.message>**". Вивід помилки оформіть написом у вигляді "**Error: <error.status> (<error.statusText>)**" в межах прямокутної червоної плашки. Для доступу до Github використайте їхній API.
4. Напишіть функцію, яка приймає два колбеки та викликає їх по порядку. Функцію реалізуйте за шаблоном колбека. Для підтвердження роботи кожної функції фіксуйте повідомлення у консолі.
5. Напишіть функцію сортування списку значень, отриманого із форми користувача, з допомогою Selection sort. Зайві (не числові) значення фільтруйте за допомогою регулярного виразу.

Варіант 11



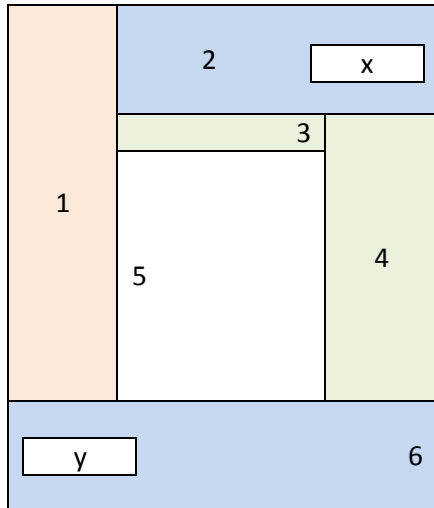
1. Послідовно в порядку нумерації зчитайте вміст номерних блоків контенту. Переставте кожен зчитаний контент на місце попереднього так, щоб кожне переставлення відбувалося із 5-секундною затримкою виконання. Після вичерпання блоків контенту зупиніть виконання скрипта.

2. Встановіть властивість «курсив» для текстів у блоках «1» і «6» через 5 секунд після настання вашої події із завдання №5 л.р. №3. Забезпечте циклічне вмикання/вимикання

властивості «курсив» для тексту в блоці «5» через кожні 5 секунд.

3. За допомогою асинхронного з'єднання без перезавантаження всієї сторінки зчитайте комміти із Github для вказаного користувачем аккаунта та репозиторія у формі на цій сторінці. Якщо за вказаними користувачем даними неможливо зчитати комміти – зафіксуйте помилку. Вивід отриманих коммітів оформіть у вигляді нумерованого списку у вигляді "**<commit.author.name>:
<commit.message>**". Вивід помилки оформіть написом у вигляді "**Error: <error.status> (<error.statusText>)**" в межах прямокутної червоної плашки. Для доступу до Github використовуйте їхній API.
4. Напишіть функцію, яка приймає два колбеки та викликає їх по порядку. Функцію реалізуйте за шаблоном колбека. Для підтвердження роботи кожної функції фіксуйте повідомлення у консолі.
5. Напишіть функцію сортування списку значень, отриманого із форми користувача, з допомогою Insertion sort. Зайві (не числові) значення фільтруйте за допомогою регулярного виразу.

Варіант 12



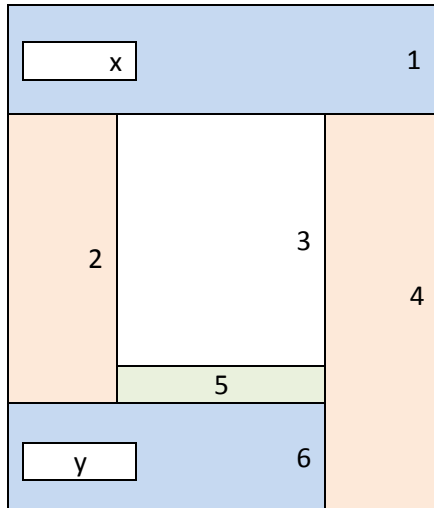
1. Послідовно в порядку нумерації зчитайте вміст номерних блоків контенту. Переставте кожен зчитаний контент на місце попереднього так, щоб кожне переставлення відбувалося із 5-секундною затримкою виконання. Після вичерпання блоків контенту зупиніть виконання скрипта.

2. Змініть регістр кожного слова в блоках «2» і «6» через 5 секунд після настання вашої події із завдання №5 л.р. №3. Забезпечте циклічну зміну регістру кожного слова в блоці «5» через кожні 5

секунд.

3. За допомогою асинхронного з'єднання без перезавантаження всієї сторінки зчитайте комміти із Github для вказаного користувачем аккаунта та репозиторія у формі на цій сторінці. Якщо за вказаними користувачем даними неможливо зчитати комміти – зафіксуйте помилку. Вивід отриманих коммітів оформіть у вигляді нумерованого списку у вигляді "**<commit.author.name>**: **<commit.message>**". Вивід помилки оформіть написом у вигляді "**Error: <error.status> (<error.statusText>)**" в межах прямокутної червоної плашки. Для доступу до Github використовуйте їхній API.
4. Напишіть функцію, яка приймає два колбеки та викликає їх по порядку. Функцію реалізуйте за шаблоном колбека. Для підтвердження роботи кожної функції фіксуйте повідомлення у консолі.
5. Напишіть функцію сортування списку значень, отриманого із форми користувача, з допомогою Bubble sort. Зайві (не числові) значення фільтруйте за допомогою регулярного виразу.

Варіант 13

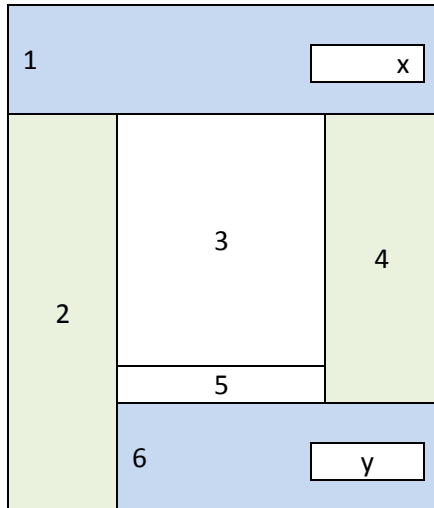


1. Послідовно в порядку нумерації зчитайте вміст номерних блоків контенту. Переставте кожен зчитаний контент на місце наступного так, щоб кожне переставлення відбувалося із 5-секундною затримкою виконання. Після вичерпання блоків контенту зупиніть виконання скрипта.

2. Змініть першу літеру кожного слова в блоках «1» і «6» через 5 секунд після настання вашої події із завдання №5 л.р. №3. Забезпечте циклічну зміну першої літери кожного слова в блоці «3» через кожні 5 секунд.

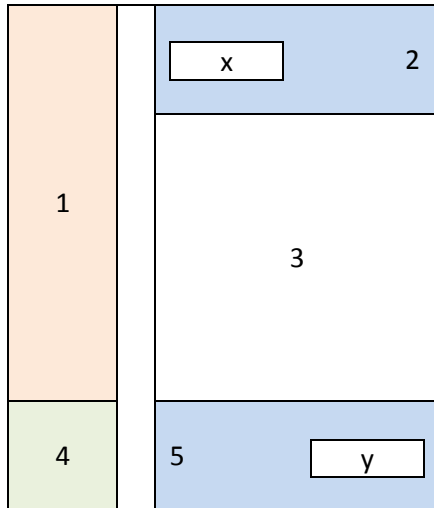
3. За допомогою асинхронного з'єднання без перезавантаження всієї сторінки зчитайте комміти із Github для вказаного користувачем аккаунта та репозиторія у формі на цій сторінці. Якщо за вказаними користувачем даними неможливо зчитати комміти – зафіксуйте помилку. Вивід отриманих коммітів оформіть у вигляді нумерованого списку у вигляді "**<commit.author.name>**: **<commit.message>**". Вивід помилки оформіть написом у вигляді "**Error: <error.status> (<error.statusText>)**" в межах прямокутної червоної плашки. Для доступу до Github використовуйте їхній API.
4. Напишіть функцію, яка приймає два колбеки та викликає їх по порядку. Функцію реалізуйте за шаблоном колбека. Для підтвердження роботи кожної функції фіксуйте повідомлення у консолі.
5. Напишіть функцію сортування списку значень, отриманого із форми користувача, з допомогою Bubble sort. Зайві (не числові) значення фільтруйте за допомогою регулярного виразу.

Варіант 14



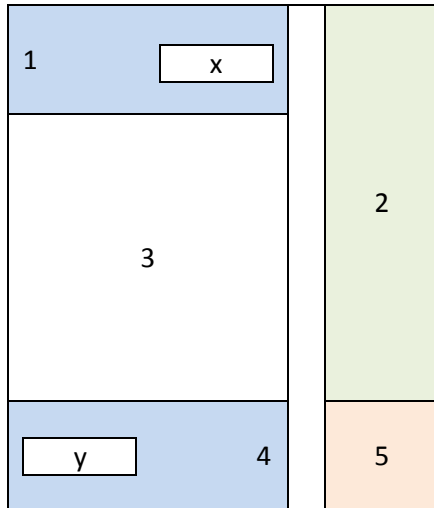
1. Послідовно в порядку нумерації зчитайте вміст номерних блоків контенту. Переставте кожен зчитаний контент на місце попереднього так, щоб кожне переставлення відбувалося із 5-секундною затримкою виконання. Після вичерпання блоків контенту зупиніть виконання скрипта.
2. Змініть колір границі в блоків «1» і «6» через 5 секунд після настання вашої події із завдання №5 л.р. №3. Забезпечте зміну кольору границі блоку «2» через кожні 5 секунд.
3. За допомогою асинхронного з'єднання без перезавантаження всієї сторінки зчитайте комміти із Github для вказаного користувачем аккаунта та репозиторія у формі на цій сторінці. Якщо за вказаними користувачем даними неможливо зчитати комміти – зафіксуйте помилку. Вивід отриманих коммітів оформіть у вигляді ненумерованого списку у вигляді "**<commit.author.name>: <commit.message>**". Вивід помилки оформіть написом у вигляді "**Error: <error.status> (<error.statusText>)**" в межах прямокутної червоної плашки. Для доступу до Github використайте їхній API.
4. Напишіть функцію, яка приймає два колбеки та викликає їх по порядку. Функцію реалізуйте за шаблоном колбека. Для підтвердження роботи кожної функції фіксуйте повідомлення у консолі.
5. Напишіть функцію сортування списку значень, отриманого із форми користувача, з допомогою Insertion sort. Зайві (не числові) значення фільтруйте за допомогою регулярного виразу.

Варіант 15



1. Послідовно в порядку нумерації зчитайте вміст номерних блоків контенту. Переставте кожен зчитаний контент на місце наступного так, щоб кожне переставлення відбувалося із наростаючою в 5 секунд затримкою виконання. Після вичерпання блоків контенту зупиніть виконання скрипта.
2. Змініть фоновий колір блоків «2» і «5» через 5 секунд після настання вашої події із завдання №5 л.р. №3. Забезпечте зміну фонового кольору блоку «1» через кожні 5 секунд.
3. За допомогою асинхронного з'єднання без перезавантаження всієї сторінки зчитайте комміти із Github для вказаного користувачем аккаунта та репозиторія у формі на цій сторінці. Якщо за вказаними користувачем даними неможливо зчитати комміти – зафіксуйте помилку. Вивід отриманих коммітів оформіть у вигляді нумерованого списку у вигляді "**<commit.author.name>: <commit.message>**". Вивід помилки оформіть написом у вигляді "**Error: <error.status> (<error.statusText>)**" в межах прямокутної червоної плашки. Для доступу до Github використайте їхній API.
4. Напишіть функцію, яка приймає два колбеки та викликає їх по порядку. Функцію реалізуйте за шаблоном колбека. Для підтвердження роботи кожної функції фіксуйте повідомлення у консолі.
5. Напишіть функцію сортування списку значень, отриманого із форми користувача, з допомогою Quick sort. Зайві (не числові) значення фільтруйте за допомогою регулярного виразу.

Варіант 16



6. Послідовно в порядку нумерації зчитайте вміст номерних блоків контенту. Переставте кожен зчитаний контент на місце попереднього так, щоб кожне переставлення відбувалося із 5-секундною затримкою виконання. Після вичерпання блоків контенту зупиніть виконання скрипта.
7. Змініть колір тексту у блоків «1» і «4» через 5 секунд після настання вашої події із завдання №5 л.р. №3. Забезпечте зміну кольору тексту блоку «2» через кожні 5 секунд.
8. За допомогою асинхронного з'єднання без перезавантаження всієї сторінки зчитайте комміти із Github для вказаного користувачем аккаунта та репозиторія у формі на цій сторінці. Якщо за вказаними користувачем даними неможливо зчитати комміти – зафіксуйте помилку. Вивід отриманих коммітів оформіть у вигляді нумерованого списку у вигляді "**<commit.author.name>: <commit.message>**". Вивід помилки оформіть написом у вигляді "**Error: <error.status> (<error.statusText>)**" в межах прямокутної червоної плашки. Для доступу до Github використайте їхній API.
9. Напишіть функцію, яка приймає два колбеки та викликає їх по порядку. Функцію реалізуйте за шаблоном колбека. Для підтвердження роботи кожної функції фіксуйте повідомлення у консолі.
10. Напишіть функцію сортування списку значень, отриманого із форми користувача, з допомогою Selection sort. Зайві (не числові) значення фільтруйте за допомогою регулярного виразу.