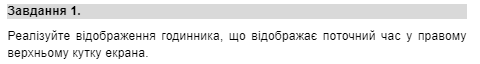
Лабораторна робота №8

Тема**Функції setInterval та setTimeout.**

***Мета роботи:*** *вивчити особливості відкладеного виконання програмного коду*



HTML

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <link rel="stylesheet" href="style.css">

</head>

<body>

    <div id="clock"></div>

</body>

<script defer src="script.js"></script>

</html>

JavaScript

function updateClock() {

    const clock = document.getElementById('clock');

    const now = new Date();

    clock.textContent = now.toLocaleTimeString();

}

setInterval(updateClock, 1000);

updateClock();

CSS

#clock {

    position: absolute;

    top: 10px;

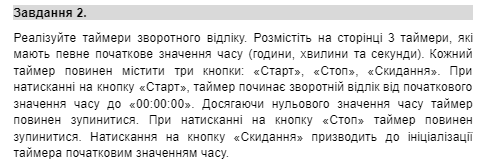
    right: 10px;

    font-size: 24px;

    font-weight: bold;

}





HTML

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <link rel="stylesheet" href="style.css">

</head>

<body>

    <div id="timers">

        <div class="timer" data-hours="0" data-minutes="1" data-seconds="30">

            <div class="time-display">00:01:30</div>

            <div class="buttons">

                <button class="start">Старт</button>

                <button class="stop">Стоп</button>

                <button class="reset">Скидання</button>

            </div>

        </div>

        <div class="timer" data-hours="0" data-minutes="2" data-seconds="0">

            <div class="time-display">00:02:00</div>

            <div class="buttons">

                <button class="start">Старт</button>

                <button class="stop">Стоп</button>

                <button class="reset">Скидання</button>

            </div>

        </div>

        <div class="timer" data-hours="0" data-minutes="0" data-seconds="45">

            <div class="time-display">00:00:45</div>

            <div class="buttons">

                <button class="start">Старт</button>

                <button class="stop">Стоп</button>

                <button class="reset">Скидання</button>

            </div>

        </div>

    </div>

</body>

<script defer src="script.js"></script>

</html>

JavaScript

document.querySelectorAll('.timer').forEach(timer => {

    const timeDisplay = timer.querySelector('.time-display');

    let hours = parseInt(timer.getAttribute('data-hours'));

    let minutes = parseInt(timer.getAttribute('data-minutes'));

    let seconds = parseInt(timer.getAttribute('data-seconds'));

    let initialTime = { hours, minutes, seconds };

    let countdown;

    function formatTime(h, m, s) {

        return `${String(h).padStart(2, '0')}:${String(m).padStart(2, '0')}:${String(s).padStart(2, '0')}`;

    }

    function updateDisplay() {

        timeDisplay.textContent = formatTime(hours, minutes, seconds);

    }

    function countdownTick() {

        if (seconds > 0) {

            seconds--;

        } else if (minutes > 0) {

            minutes--;

            seconds = 59;

        } else if (hours > 0) {

            hours--;

            minutes = 59;

            seconds = 59;

        } else {

            clearInterval(countdown);

            countdown = null;

            return;

        }

        updateDisplay();

    }

    timer.querySelector('.start').addEventListener('click', () => {

        if (!countdown) {

            countdown = setInterval(countdownTick, 1000);

        }

    });

    timer.querySelector('.stop').addEventListener('click', () => {

        clearInterval(countdown);

        countdown = null;

    });

    timer.querySelector('.reset').addEventListener('click', () => {

        clearInterval(countdown);

        countdown = null;

        hours = initialTime.hours;

        minutes = initialTime.minutes;

        seconds = initialTime.seconds;

        updateDisplay();

    });

    updateDisplay();

});

CSS

.timer {

    margin: 20px 0;

}

.time-display {

    font-size: 32px;

    margin-bottom: 10px;

}

.buttons {

    display: flex;

    gap: 10px;

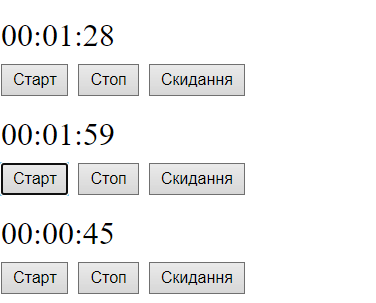
}

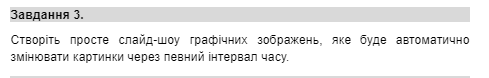
button {

    padding: 5px 10px;

    font-size: 16px;

}





HTML

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <link rel="stylesheet" href="style.css">

</head>

<body>

    <div id="slideshow">

        <img src="1.1.jpg" class="active" alt="Slide 1">

        <img src="2.1.jpg" alt="Slide 2">

        <img src="3.1.jpg" alt="Slide 3">

        <img src="4.1.jpg" alt="Slide 4">

        <img src="5.1.jpg" alt="Slide 5">

    </div>

</body>

<script defer src="script.js"></script>

</html>

JavaScript

let currentSlide = 0;

const slides = document.querySelectorAll("#slideshow img");

const totalSlides = slides.length;

function showNextSlide() {

    slides[currentSlide].classList.remove("active");

    currentSlide = (currentSlide + 1) % totalSlides;

    slides[currentSlide].classList.add("active");

}

setInterval(showNextSlide, 3000);

CSS

#slideshow {

    width: 500px;

    height: 300px;

    overflow: hidden;

    margin: 20px auto;

    position: relative;

}

#slideshow img {

    width: 100%;

    height: 100%;

    position: absolute;

    opacity: 0;

    transition: opacity 1s;

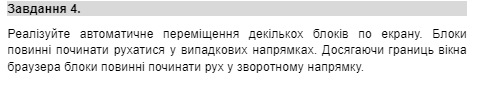
}

#slideshow img.active {

    opacity: 1;

}





HTML

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <link rel="stylesheet" href="style.css">

</head>

<body>

</body>

<script defer src="script.js"></script>

</html>

JavaScript

const numberOfBlocks = 5;

const blocks = [];

const directions = [];

function createRandomBlock() {

    const block = document.createElement('div');

    block.classList.add('block');

    const x = Math.floor(Math.random() \* (window.innerWidth - 50));

    const y = Math.floor(Math.random() \* (window.innerHeight - 50));

    block.style.left = `${x}px`;

    block.style.top = `${y}px`;

    document.body.appendChild(block);

    blocks.push(block);

    directions.push({

        x: Math.random() > 0.5 ? 1 : -1,

        y: Math.random() > 0.5 ? 1 : -1

    });

}

for (let i = 0; i < numberOfBlocks; i++) {

    createRandomBlock();

}

function moveBlocks() {

    blocks.forEach((block, index) => {

        const direction = directions[index];

        let x = parseFloat(block.style.left);

        let y = parseFloat(block.style.top);

        x += direction.x \* 2;

        y += direction.y \* 2;

        if (x <= 0 || x + block.offsetWidth >= window.innerWidth) {

            direction.x \*= -1;

        }

        if (y <= 0 || y + block.offsetHeight >= window.innerHeight) {

            direction.y \*= -1;

        }

        block.style.left = `${x}px`;

        block.style.top = `${y}px`;

    });

    requestAnimationFrame(moveBlocks);

}

moveBlocks();

CSS

body {

    overflow: hidden;

    height: 100vh;

    margin: 0;

    position: relative;

}

.block {

    width: 50px;

    height: 50px;

    position: absolute;

    background-color: #3498db;

    border-radius: 5px;

}



Висновок: Я вивчив *особливості відкладеного виконання програмного коду*