Лабораторная работа № 9

Тема: Таймеры.

Цель: Познакомиться с работой функций setTimeout, setInterval.

🔷 Вариант 1

Задание 1. setTimeout

Создайте HTML-страницу с кнопкой и параграфом.

При нажатии на кнопку через 2 секунды в параграф выводится сообщение:

👉 «Привет, мир!».

Задание 2. setInterval

Создайте «часы»: каждую секунду в параграфе обновляется текущее время (new Date().toLocaleTimeString()).

Добавьте кнопку «Остановить часы», которая прекращает обновление времени.

Задание 3. Дрейф интервалов

Создайте setInterval, который запускается каждые 1000 мс.

Внутри колбэка:

Сначала выведите в консоль текущее время (new Date().toLocaleTimeString()).

Затем сделайте «тяжёлую работу» (например, цикл, который блокирует поток на ~2 секунды).

После «работы» снова выведите время.

Наблюдайте:

Интервал назначен на 1000 мс.

Но фактический запуск функции откладывается, так как Call Stack занят тяжёлой работой.

В консоли вы увидите смещение времени — вызовы происходят не ровно раз в секунду.

👉 Подсказка: используйте такой фрагмент для «нагрузки»:

```
const end = Date.now() + 2000; // 2 секунды
while (Date.now() < end) {}
```

Задание 4. Рекурсивный setTimeout

Сделайте всё то же самое, что в варианте 1.

Добавьте сравнение с рекурсивным setTimeout:

Перепишите пример так, чтобы вместо setInterval использовать рекурсивный setTimeout с задержкой в 1000 мс.

Запустите ту же «тяжёлую работу» внутри.

Сравните результаты:

В случае setInterval вызовы будут накладываться и накапливаться (дрейф).

В случае рекурсивного setTimeout новый вызов начнётся только после завершения текущего, то есть задержки будут более «честными».

👉 Шаблон для рекурсивного setTimeout:

```
function tick() {
  console.log("start", new Date().toLocaleTimeString());
  const end = Date.now() + 2000;
  while (Date.now() < end) {}
  console.log("end", new Date().toLocaleTimeString());

  setTimeout(tick, 1000);
}

setTimeout(tick, 1000);
```

Задание 5. Простая анимация.

Сделайте div-шарик, который двигается слева направо по экрану.

Каждые 20 мс позиция увеличивается на несколько пикселей.

При достижении края — останавливается.

Дополнительное задание: Добавьте кнопки **«Старт / Стоп».**

🔷 Вариант 2

Задание 1. setTimeout

Создайте кнопку «Отсчёт». При клике запускается обратный отсчёт от 5 до 0, каждую секунду число уменьшается в параграфе.

Задание 2. setInterval

Сделайте «светофор»: каждые 2 секунды меняется цвет блока (div) по циклу: красный → жёлтый → зелёный → красный.

Добавьте кнопку «Стоп», которая выключает светофор.

Задание 3. Дрейф интервалов (визуализация)

Создайте setInterval, который каждую секунду добавляет новое сообщение в список (<ul>).

Сообщение должно содержать текст:

👉 «Сообщение в X:XX:XX» (время вставки).

Внутри обработчика добавьте «тяжёлую работу» на 2 секунды (цикл while).

Посмотрите в браузере: из-за блокировки сообщения будут приходить неровно раз в секунду — иногда сразу несколько подряд.

Вопрос к студентам: почему сообщения не приходят точно по таймеру?

Задание 4. Рекурсивный setTimeout (точный таймер)

Перепишите задание 3 с использованием рекурсивного setTimeout.

Сделайте так, чтобы новое сообщение добавлялось через 1 секунду после завершения тяжёлой работы, а не по фиксированному интервалу.

Сравните с результатами задания 3:

В setInterval сообщения сдвигаются и накапливаются (дрейф).

В setTimeout — сообщения приходят строго «по порядку», но общий тайминг сдвигается (работа учитывается в задержке).

Задание 5. Простая анимация.

Сделайте div-квадрат, который мигает (например, то красный, то зеленый).

Дополнительное задание: Добавьте кнопки **«Старт / Стоп».**

⚡ **Отчет должен содержать (см. образец):**
- номер и тему лабораторной работы;
- фамилию, номер группы студента и вариант задания;
- скриншоты окна VSC с исходным кодом программ;
- скриншоты с результатами выполнения программ;
- пояснения, если необходимо;
- выводы.

Отчеты в формате **pdf** отправлять на email:
colledge20education23@gmail.com

Можно использовать шаблоны:

```
◈ Вариант 1
Задание 1. setTimeout
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <title>setTimeout — Привет, мир!</title>
</head>
<body>
  <button id="btn">Показать сообщение</button>
  <p id="output"></p>

  <script>
    const btn = document.getElementById("btn");
    const output = document.getElementById("output");

    btn.addEventListener("click", () => {
      setTimeout(() => {
        output.textContent = "Привет, мир!";
      }, 2000);
    });
  </script>
</body>
</html>

Задание 2. setInterval (часы)
```

```html
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <title>setInterval — Часы</title>
</head>
<body>
  <p id="clock"></p>
  <button id="stop">Остановить часы</button>

  <script>
    const clock = document.getElementById("clock");
    const stopBtn = document.getElementById("stop");

    const timer = setInterval(() => {
      clock.textContent = new Date().toLocaleTimeString();
    }, 1000);

    stopBtn.addEventListener("click", () => {
      clearInterval(timer);
    });
  </script>
</body>
</html>
```

Задание 3. Дрейф интервалов

```html
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <title>Дрейф интервалов</title>
</head>
<body>
  <script>
    setInterval(() => {
      console.log("start", new Date().toLocaleTimeString());

      // тяжёлая работа (~2 сек)
      const end = Date.now() + 2000;
      while (Date.now() < end) {}

      console.log("end", new Date().toLocaleTimeString());
    }, 1000);
  </script>
</body>
</html>
```

Задание 4. Рекурсивный setTimeout

```html
<!DOCTYPE html>
<html lang="ru">
<head>
```

```
  <meta charset="UTF-8">
  <title>Рекурсивный setTimeout</title>
</head>
<body>
  <script>
    function tick() {
      console.log("start", new Date().toLocaleTimeString());

      // тяжёлая работа (~2 сек)
      const end = Date.now() + 2000;
      while (Date.now() < end) {}

      console.log("end", new Date().toLocaleTimeString());

      setTimeout(tick, 1000);
    }

    setTimeout(tick, 1000);
  </script>
</body>
</html>

Задание 5. Двигающийся шарик
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <title>Анимация шарика</title>
  <style>
    #ball {
      width: 50px;
      height: 50px;
      border-radius: 50%;
      background: crimson;
      position: absolute;
      top: 100px;
      left: 0;
    }
  </style>
</head>
<body>
  <div id="ball"></div>

  <script>
    const ball = document.getElementById("ball");
    let pos = 0;
    const step = 5; // шаг движения

    setInterval(() => {
      pos += step;
      if (pos > window.innerWidth - 60) {
```

```
        pos = 0; // если достигли края, начинаем сначала
      }
      ball.style.left = pos + "px";
    }, 20);
  </script>
</body>
</html>
```

◈ Вариант 2

Задание 1. setTimeout (обратный отсчёт)

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <title>Обратный отсчёт</title>
</head>
<body>
  <button id="start">Отсчёт</button>
  <p id="counter"></p>

  <script>
    const btn = document.getElementById("start");
    const counter = document.getElementById("counter");

    btn.addEventListener("click", () => {
      let n = 5;

      function countdown() {
        counter.textContent = n;
        if (n > 0) {
          n--;
          setTimeout(countdown, 1000);
        }
      }

      countdown();
    });
  </script>
</body>
</html>
```

Задание 2. setInterval (светофор)

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <title>Светофор</title>
  <style>
    #light {
      width: 100px;
      height: 100px;
```

```
      border-radius: 50%;
      background: red;
    }
  </style>
</head>
<body>
  <div id="light"></div>
  <button id="stop">Стоп</button>

  <script>
    const light = document.getElementById("light");
    const stopBtn = document.getElementById("stop");

    const colors = ["red", "yellow", "green"];
    let i = 0;

    const timer = setInterval(() => {
      light.style.background = colors[i];
      i = (i + 1) % colors.length;
    }, 2000);

    stopBtn.addEventListener("click", () => {
      clearInterval(timer);
    });
  </script>
</body>
</html>
```

Задание 3. Дрейф интервалов (визуализация)

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <title>Дрейф интервалов</title>
</head>
<body>
  <ul id="messages"></ul>

  <script>
    const list = document.getElementById("messages");

    setInterval(() => {
      const li = document.createElement("li");
      li.textContent = "Сообщение в " + new Date().toLocaleTimeString();
      list.appendChild(li);

      // тяжёлая работа (~2 сек)
      const end = Date.now() + 2000;
      while (Date.now() < end) {}
    }, 1000);
  </script>
```

```
</body>
</html>

Задание 4. Рекурсивный setTimeout (точный таймер)
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <title>Рекурсивный setTimeout</title>
</head>
<body>
  <ul id="messages"></ul>

  <script>
    const list = document.getElementById("messages");

    function tick() {
      const li = document.createElement("li");
      li.textContent = "Сообщение в " + new Date().toLocaleTimeString();
      list.appendChild(li);

      // тяжёлая работа (~2 сек)
      const end = Date.now() + 2000;
      while (Date.now() < end) {}

      setTimeout(tick, 1000);
    }

    setTimeout(tick, 1000);
  </script>
</body>
</html>

Задание 5. Мигающий квадрат
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <title>Мигающий квадрат</title>
  <style>
    #box {
      width: 100px;
      height: 100px;
      background: red;
      margin: 50px auto;
    }
  </style>
</head>
<body>
  <div id="box"></div>
```

```
  <script>
    const box = document.getElementById("box");
    let isRed = true;

    setInterval(() => {
      if (isRed) {
        box.style.background = "green";
      } else {
        box.style.background = "red";
      }
      isRed = !isRed;
    }, 1000); // смена каждую секунду
  </script>
</body>
</html>
```