|  |  |
| --- | --- |
| **Планирование: setTimeout и setInterval**  Мы можем вызвать функцию не в данный момент, а позже, через заданный интервал времени. Это называется «планирование вызова».  **Для этого существуют два метода:**  **setTimeout** позволяет вызвать функцию один раз через определённый интервал времени.  **setInterval** позволяет вызывать функцию регулярно, повторяя вызов через определённый интервал времени.  Эти методы не являются частью спецификации JavaScript. Но большинство сред выполнения JS-кода имеют внутренний планировщик и предоставляют доступ к этим методам. В частности, они поддерживаются во всех браузерах и Node.js.  **setTimeout**  Синтаксис:  let timerId = setTimeout(func|code, [delay], [arg1], [arg2], ...);  **Параметры:**  **func|code**  Функция или строка кода для выполнения. Обычно это функция. По историческим причинам можно передать и строку кода, но это не рекомендуется.  **delay**  Задержка перед запуском в миллисекундах (1000 мс = 1 с). Значение по умолчанию – 0.  **arg1, arg2…**  Аргументы, передаваемые в функцию  **Примеры**  Например, данный код вызывает sayHi() спустя одну секунду:  function sayHi() {  alert('Привет');  }  setTimeout(sayHi, 1000);  **С аргументами:**  function sayHi(phrase, who) {  alert( phrase + ', ' + who );  }  setTimeout(sayHi, 1000, "Привет", "Джон"); // Привет, Джон  **setTimeout("alert('Привет')", 1000);** | **setInterval**  Метод setInterval имеет такой же синтаксис как setTimeout:  let timerId = setInterval(func|code, [delay], [arg1], [arg2], ...);  Все аргументы имеют такое же значение. Но отличие этого метода от setTimeout в том, что функция запускается не один раз, а периодически через указанный интервал времени.  **Чтобы остановить дальнейшее выполнение функции, необходимо вызвать clearInterval(timerId).**  **Следующий пример выводит сообщение каждые 2 секунды. Через 5 секунд вывод прекращается:**  // повторить с интервалом 2 секунды  let timerId = setInterval(() => alert('tick'), 2000);  // остановить вывод через 5 секунд  setTimeout(() => { clearInterval(timerId); alert('stop'); }, 5000);  **Итого**  Методы setInterval(func, delay, ...args) и setTimeout(func, delay, ...args) позволяют выполнять func регулярно или только один раз после задержки delay, заданной в мс.  Для отмены выполнения необходимо вызвать clearInterval/clearTimeout со значением, которое возвращают методы setInterval/setTimeout.  Вложенный вызов setTimeout является более гибкой альтернативой setInterval. Также он позволяет более точно задать интервал между выполнениями.  Планирование с нулевой задержкой setTimeout(func,0) или, что то же самое, setTimeout(func) используется для вызовов, которые должны быть исполнены как можно скорее, после завершения исполнения текущего кода.  Браузер ограничивает 4-мя мс минимальную задержку между пятью и более вложенными вызовами setTimeout, а также для setInterval, начиная с 5-го вызова. |