

Динамическое манипулирование структурой DOM

Кулаков Кирилл Александрович

Динамическое манипулирование DOM

- События JavaScript - это некоторые действия, которыми можно управлять программно и для которых можно описать определенную программную реакцию. Возникновение событий инициируется либо непосредственно браузером, либо пользователем.
- Категории событий
 - события мыши (клик, двойной клик, перемещение курсора и т.д.);
 - события клавиатуры (нажатие или отпускание клавиши);
 - события элементов управления (установка или потеря фокуса на поле формы, отправка данных формы и т.д.);
 - события браузера (завершение построения HTML-страницы, изменение размера окна и т.д.);
 - пользовательские события (созданные программно разработчиком).

Динамическое манипулирование DOM

- Подключение к событиям
 - Атрибуты событий HTML
 - `<button onclick="displayDate()">Try it</button>`
 - События HTML DOM
 - `document.getElementById("myBtn").onclick = displayDate;`

Динамическое манипулирование DOM

- События
 - onload и onunload: пользователь входит или покидает страницу
 - onchange: обработка изменений значений полей
 - onmouseover и onmouseout: наведение указателя на элемент и выход из области элемента
 - onmousedown, onmouseup и onclick: обработка действий указателя
 - и т. д.

https://schoolsw3.com/jsref/dom_obj_event.php

Динамическое манипулирование DOM

- Способы распространения событий в HTML DOM:
 - Восходящий: обработка событий внутреннего элемента и переход ко внешнему
 - Захватывающий: обработка событий внешнего элемента и переход ко внутреннему
- Пример:

```
<div onclick="...">  
  <p onclick="...">  
  </p>  
</div>
```

Прослушиватель событий

- Методы JavaScript HTML DOM
 - `addEventListener()`: добавление обработчика
 - `element.addEventListener(event, function, useCapture);`
 - Пример:
`document.getElementById("myBtn").addEventListener("click", displayDate);`
 - `removeEventListener()`: удаление обработчика
 - `element.removeEventListener(event, function);`
 - Пример:
`element.removeEventListener("mousemove", myFunction);`

Управление событием

- Методы JavaScript HTML DOM
 - `preventDefault()`: предотвращает поведение браузера, определенное для события по умолчанию
 - `stopPropagation()`: предотвращает распространение события вверх по DOM-дереву

```
<form>
```

```
  <input type="text" id="field" />
```

```
</form>
```

```
let field = document.getElementById('field');
```

```
field.addEventListener('keydown', event =>  
event.preventDefault());
```

Управление событием

```
<p id="parent">  
  JavaScript  
  <strong id="child1">Events</strong>  
  <i id="child2">stopPropagation</i>  
</p>  
let parent = document.getElementById('parent');  
let child1 = document.getElementById('child1');  
let child2 = document.getElementById('child2');  
parent.addEventListener('click', event => alert('Parent handler'));  
child1.addEventListener('click', event => alert('Child 1 handler'));  
child2.addEventListener('click', event => {  
  event.stopPropagation();  
  alert('Child 2 handler');  
});
```


jQuery

- jQuery был создан в 2006 году Джоном Ресигом
- разработан для обработки несовместимости браузеров и упрощения манипулирования HTML DOM, обработки событий, анимации и Ajax
- после версии 5 версии JavaScript 5 (2009 г.) большинство утилит jQuery можно решить с помощью нескольких строк стандартного JavaScript

jQuery

- Поиск элемента HTML по идентификатору
 - jQuery: `var myElement = $("#id01");`
 - JavaScript: `var myElement = document.getElementById("id01");`
- Поиск элементов HTML по имени тега
 - jQuery: `var myElements = $("p");`
 - JavaScript: `var myElements = document.getElementsByTagName("p");`
- Поиск элементов HTML по имени класса
 - jQuery: `var myElements = $(".intro");`
 - JavaScript: `var myElements = document.getElementsByClassName("intro");`

jQuery

- Установить текстовое содержимое
 - jQuery: `myElement.text("Привет, Россия!");`
 - JavaScript: `myElement.textContent = "Привет, Россия!";`
- Получить текстовый контент
 - jQuery: `var myText = myElement.text();`
 - JavaScript: `var myText = myElement.textContent || myElement.innerText;`
- Установить HTML контент
 - jQuery: `var myElement.html("<p>Привет, мир</p>");`
 - JavaScript: `var myElement.innerHTML = "<p>Привет, мир</p>";`

jQuery

- Скрытие HTML элементов
 - jQuery: `myElement.hide()`;
 - JavaScript: `myElement.style.display = "none"`;
- Отображение элементов HTML
 - jQuery: `myElement.show()`;
 - JavaScript: `myElement.style.display = ""`;
- Стилизация HTML элементов
 - jQuery: `myElement.css("font-size","35px")`;
 - JavaScript: `myElement.style.fontSize = "35px"`;

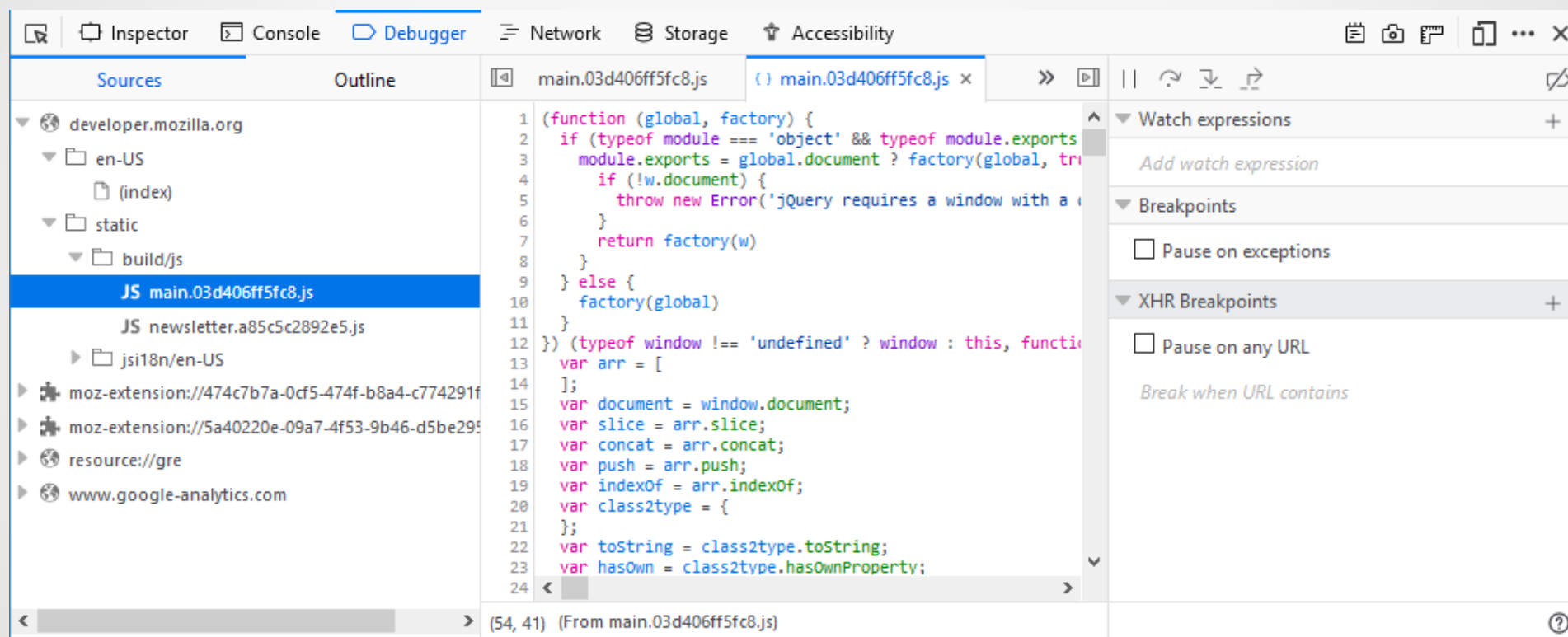
jQuery

- Удаление элементов
 - jQuery: `$("#id").remove();`
 - JavaScript: `element.parentNode.removeChild(element);`
- Получить родительский элемент
 - jQuery: `var myParent = myElement.parent();`
 - JavaScript: `var myParent = myElement.parentNode;`

Инструменты разработчика

- Возможность редактирования дерева DOM прямо в браузере
- Возможность контроля за действиями скрипта
- Возможность управления скриптом
- Возможность управления данными

Инструменты разработчика



Инструменты разработчика

- Инспектор страницы
 - Инструмент для просмотра и редактирования содержимого и макета страницы. Позволяет рассмотреть страницу с разных точек зрения, включая блочную модель, анимацию и grid компоновку.
- Web-консоль
 - Позволяет просматривать отладочные сообщения страницы и взаимодействовать с ней посредством JavaScript.
- Отладчик JavaScript
 - Позволяет остановить, выполнять по шагам, исследовать и изменять JavaScript код выполняемый на странице.

Инструменты разработчика

- Сетевой монитор
 - Показывает сетевые запросы возникающие в процессе загрузки страницы.
- Инструменты производительности
 - Позволяют провести общий анализ отзывчивости вашего сайта, а также производительности кода JavaScript и макета.
- Режим адаптивного дизайна
 - Позволяет увидеть, как будут выглядеть и работать ваши сайты или приложения на различных устройствах или типах сетей.
- Инспектор доступности
 - Предоставляет средства для доступа к дереву доступности страницы, позволяя вам проверить, что отсутствует или иным образом требует внимания.

Инструменты разработчика

- Инспектор доступности

–