

Браузерные события

JavaScript

Введение

- *Событие* – сигнал от браузера о том, что что-то произошло
- Выделяют:
 - События мыши
 - События на элементах управления
 - Клавиатурные события
 - События документа

События мыши

- `click` – кликнули на элемент ЛКМ
- `contextmenu` – кликнули на элемент ПКМ
- `mouseover` – на элемент навели мышь
- `mousedown` и `mouseup` – кнопку мыши нажали или отжали
- `mousemove` – при движении мыши

События на элементах управления

- `submit` – посетитель отправил данные на сервер
- `focus` – посетитель фокусируется на элементе, например нажимает на текстовое поле

Клавиатурные события

- `keydown` – посетитель нажимает клавишу
- `keyup` – посетитель отпускает клавишу
- `keypress` – нажата и отпущена клавиша

События документа

- DOMContentLoaded – HTML загружен и обработан, DOM документа полностью построен и доступен
- load – закончена загрузка документа
- unload – попытка закрытия окна браузера и выгрузка документа

Обработчики событий

- Обработчик события – функция, которая вызывается при наступлении события
- Благодаря обработчикам JavaScript-код может реагировать на действия посетителя
- Есть несколько способов назначить событию обработчик

Использование атрибута

- `<body>`
 - `<script>`
 - `function count() {`
 - `for(var i=1; i<=3; i++) {`
 - `alert("Кролик номер " + i);`
 - `}`
 - `}`
 - `</script>`
 - `<input type="button" onclick="count()" value="Считать!"/>`
- `</body>`

Использование свойства DOM-объекта

- `<body>`
 - `<script>`
 - `elem.onclick = function() {`
 - `alert('Спасибо');`
 - `};`
 - `</script>`
 - `<input id="elem" type="button" value="Нажми" />`
- `</body>`

Доступ к элементу через this

- `<body>`
 - `<button onclick="test(this)">Нажми </button>`
 - `<script>`
 - `function test(elem) {`
 - `alert(elem.innerHTML);`
 - `}`
 - `</script>`
- `</body>`

addEventListener & removeEventListener

- `element.addEventListener(event, handler);`
- `event` – имя события, например `click`
- `handler` – ссылка на функцию, которую надо поставить обработчиком
- `function handler() { alert('Спасибо!'); }`
- `input.addEventListener("click", handler);`
- `input.removeEventListener("click", handler);`

Пример

- `<input id="elem" type="button" value="Нажми" />`
- `<script>`
 - `function handler1() {`
 - `alert('Спасибо!');`
 - `};`
 - `function handler2() {`
 - `alert('Спасибо ещё раз!');`
 - `}`
 - `elem.onclick = function() { alert("Привет"); };`
 - `elem.addEventListener("click", handler1);`
 - `elem.addEventListener("click", handler2);`
- `</script>`

Свойства объекта события

- Свойства объекта event:
 - `event.type` – тип события
 - `event.currentTarget` – элемент, на котором сработал обработчик
 - `event.clientX` / `event.clientY` – координаты курсора в момент клика (относительно окна)

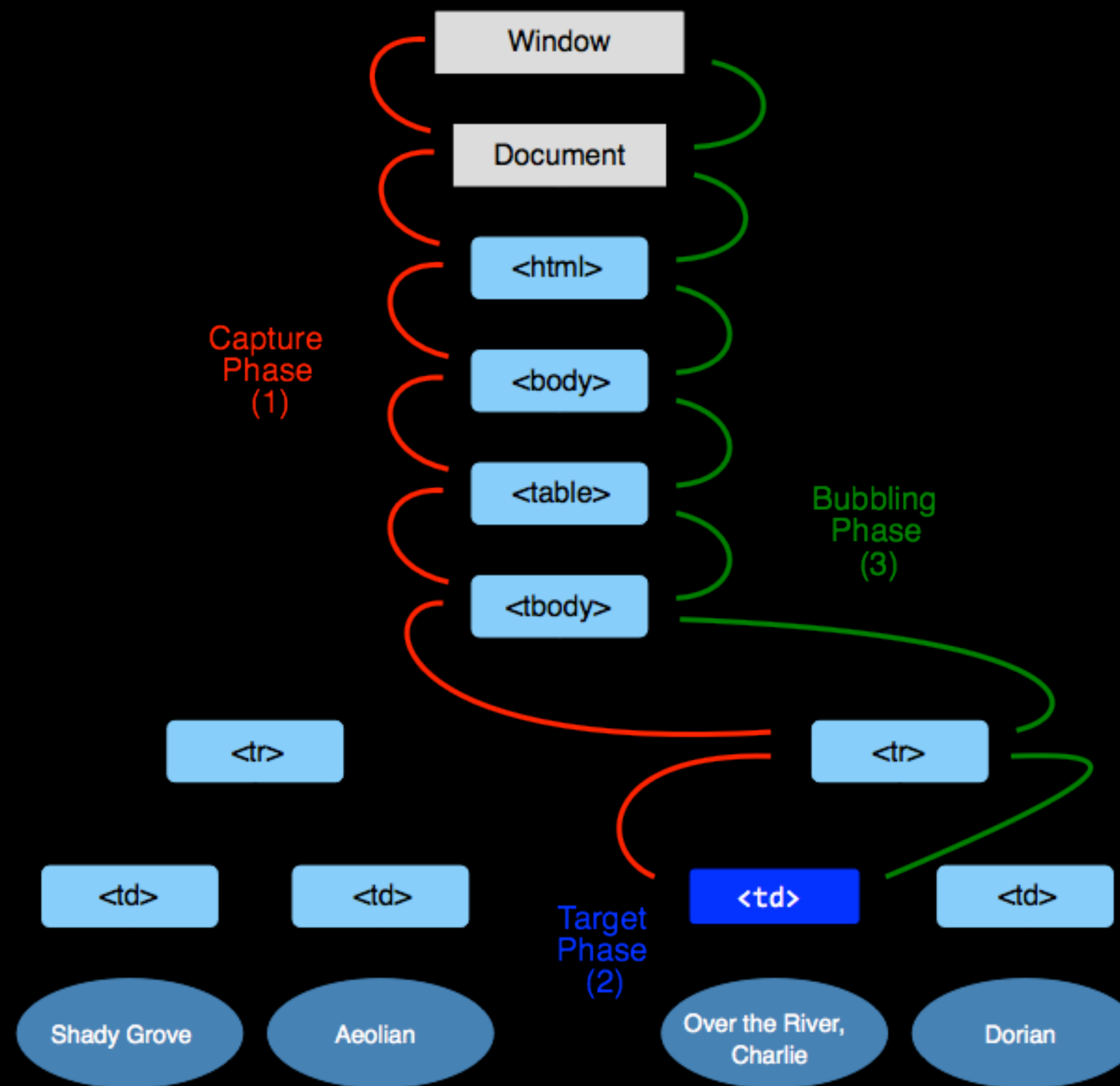
Пример

- `<body>`
 - `<input type="button" value="Нажми" id="elem">`
 - `<script>`
 - `elem.onclick = function(event) {`
 - `alert(event.type + " на " + event.currentTarget);`
 - `alert(event.clientX + ":" + event.clientY);`
 - `}`
 - `</script>`
- `</body>`

Всплытие и погружение

- Современный стандарт выделяет три стадии прохода события:
 - "стадия перехвата" – событие идет сверху вниз (capturing stage)
 - "стадия цели" – событие достигло целевого элемента (target stage)
 - "стадия всплытия" – событие идет снизу вверх (bubbling stage)

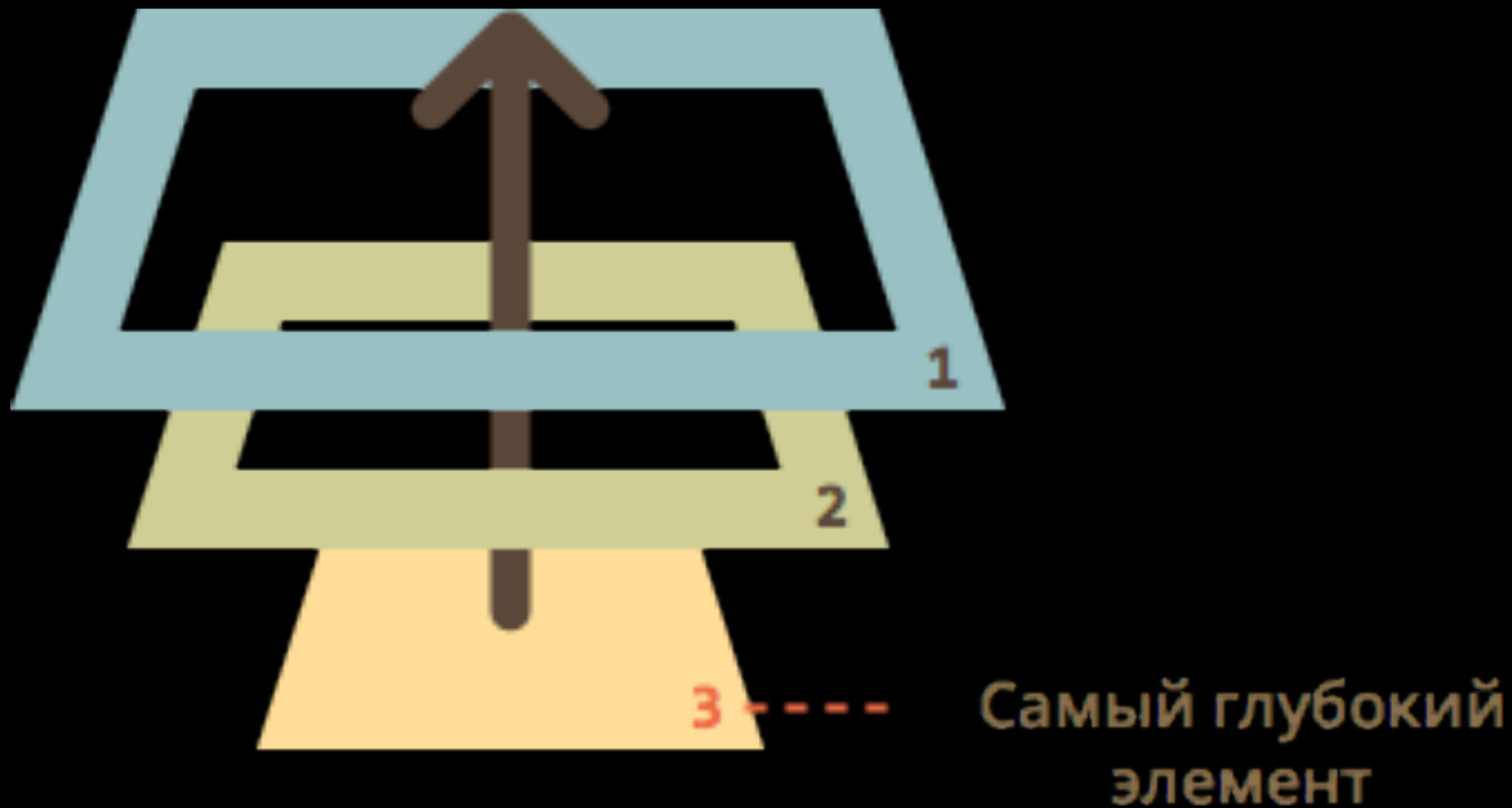
Представление



Пример

- `<style>`
 - `body * { margin: 10px; border: 1px solid blue; }`
- `</style>`
- `<form onclick="alert('form')">FORM`
 - `<div onclick="alert('div')">DIV`
 - `<p onclick="alert('p')">P</p>`
 - `</div>`
- `</form>`

Интерпретация



Прекращение всплытия

- `<style>`
 - `body * { margin: 10px; border: 1px solid blue; }`
- `</style>`
- `<form onclick="form()">FORM`
 - `<div onclick="div()">DIV`
 - `<p onclick="p()">P</p>`
 - `</div>`
- `</form>`
- `<script>`
 - `function form() { alert("form"); }`
 - `function div() { alert("div"); }`
 - `function p() { alert("p"); event.stopPropagation(); }`
- `</script>`

Делегирование событий

Bagua Chart: Direction, Element, Color, Meaning

Northwest Metal Silver Elders	North Water Blue Change	Northeast Earth Yellow Direction
West Metal Gold Youth	Center All Purple Harmony	East Wood Blue Future
Southwest Earth Brown Tranquility	South Fire Orange Fame	Southeast Wood Green Romance

Часть html-файла

- `<table id="bagua-table">`
 - `<tr>`
 - `<th colspan="3">Bagua Chart: Direction, Element, Color, Meaning</th>`
 - `</tr>`
 - `<tr>`
 - `<td class="nw">`
 - `Northwest`
 - `
Metal`
 - `
Silver`
 - `
Elders`
 - `</td>`
 - `<td class="n">`
 - `North`
 - `
Water`
 - `
Blue`
 - `
Change`
 - `</td>`
- ...

JavaScript-часть

- `<script>`
 - `var table = document.getElementById('bagua-table');`
 - `var selectedTd;`
 - `table.onclick = function(event) {`
 - `var target = event.target;`
 - `while (target != this) {`
 - `if (target.tagName == 'TD') {`
 - `highlight(target);`
 - `return;`
 - `}`
 - `target = target.parentNode;`
 - `}`
 - `}`
 - `function highlight(node) {`
 - `if (selectedTd) {`
 - `selectedTd.classList.remove('highlight');`
 - `selectedTd = false;`
 - `}`
 - `else {`
 - `selectedTd = node;`
 - `selectedTd.classList.add('highlight');`
 - `}`
 - `}`
- `</script>`

table.onclick

- table.onclick = function(event) {
 - var target = event.target;
 - while (target !== this) {
 - if (target.tagName === 'TD') {
 - highlight(target);
 - return;
 - }
 - target = target.parentNode;
 - }
- }

function highlight(node)

- function highlight(node) {
 - if (selectedTd) {
 - selectedTd.classList.remove('highlight');
 - selectedTd = false;
 - }
 - else {
 - selectedTd = node;
 - selectedTd.classList.add('highlight');
 - }
- }

Проектирование поведения

- Счётчик 1: `<button data-counter>1</button>`
- Счётчик 2: `<button data-counter>2</button>`
- `<script>`
 - `document.onclick = function(event) {`
 - `if (!event.target.hasAttribute('data-counter')) {`
 - `return;`
 - `}`
 - `var counter = event.target;`
 - `counter.innerHTML++;`
 - `}`
- `</script>`