**ЛР 17**

**События в JS**

**Контрольные вопросы**

1. \*\*Что понимается под событием на веб-странице?\*\*

Событие на веб-странице — это любое действие, которое происходит в процессе взаимодействия с элементами страницы, например, клик мышью, ввод с клавиатуры, изменение размера окна и т. д.

2. \*\*Перечислите известные вам типы событий.\*\*

- `click` — клик мышью по элементу.

- `keydown` — нажатие клавиши.

- `keyup` — отпускание клавиши.

- `mouseover` — наведение мыши на элемент.

- `mouseout` — убирание мыши с элемента.

- `submit` — отправка формы.

- `resize` — изменение размера окна.

- `scroll` — прокрутка страницы.

- `focus` — получение фокуса элементом.

3. \*\*Пример простейшего обработчика событий.\*\*

В этом примере обработчик реагирует на клик по кнопке:

```javascript

document.getElementById("myButton").addEventListener("click", function() {

alert("Button clicked!");

});

```

4. \*\*Как осуществляется передача параметров в обработчик события?\*\*

Для передачи параметров можно использовать анонимные функции или функции-обработчики, которые обрабатывают событие и передают нужные параметры.

Пример:

```javascript

function handleClick(message) {

alert(message);

}

document.getElementById("myButton").addEventListener("click", function() {

handleClick("Button clicked!");

});

```

5. \*\*Метод `addEventListener` позволяет добавлять несколько обработчиков на одно событие одного элемента.\*\*

Пример добавления нескольких обработчиков для одного события:

```javascript

const button = document.getElementById("myButton");

button.addEventListener("click", function() {

console.log("First click handler");

});

button.addEventListener("click", function() {

console.log("Second click handler");

});

```

Оба обработчика сработают при клике на кнопку.

6. \*\*Распространение событий (восходящие и нисходящие события, их отличие друг от друга).\*\*

- \*\*Нисходящее (capturing)\*\* — событие начинает распространяться от верхнего уровня DOM до целевого элемента.

- \*\*Восходящее (bubbling)\*\* — событие начинает распространяться от целевого элемента к верхнему уровню DOM.

Пример:

```javascript

document.body.addEventListener('click', function() {

console.log('Body clicked (bubbling)');

}, true); // capturing

document.getElementById("myButton").addEventListener('click', function() {

console.log('Button clicked');

}); // bubbling

```

7. \*\*Назначение объекта `Event`. Доступ к его свойствам. Примеры кода работы с объектом `Event`.\*\*

Объект `Event` содержит информацию о событии, например, тип события, целевой элемент и другие параметры.

Пример:

```javascript

document.getElementById("myButton").addEventListener("click", function(event) {

console.log(event.type); // Тип события (например, 'click')

console.log(event.target); // Целевой элемент, на который было кликнуто

});

```

Свойства объекта `Event` включают:

- `type` — тип события.

- `target` — целевой элемент, на котором произошло событие.

- `preventDefault()` — предотвращает стандартное поведение события.

- `stopPropagation()` — останавливает распространение события.