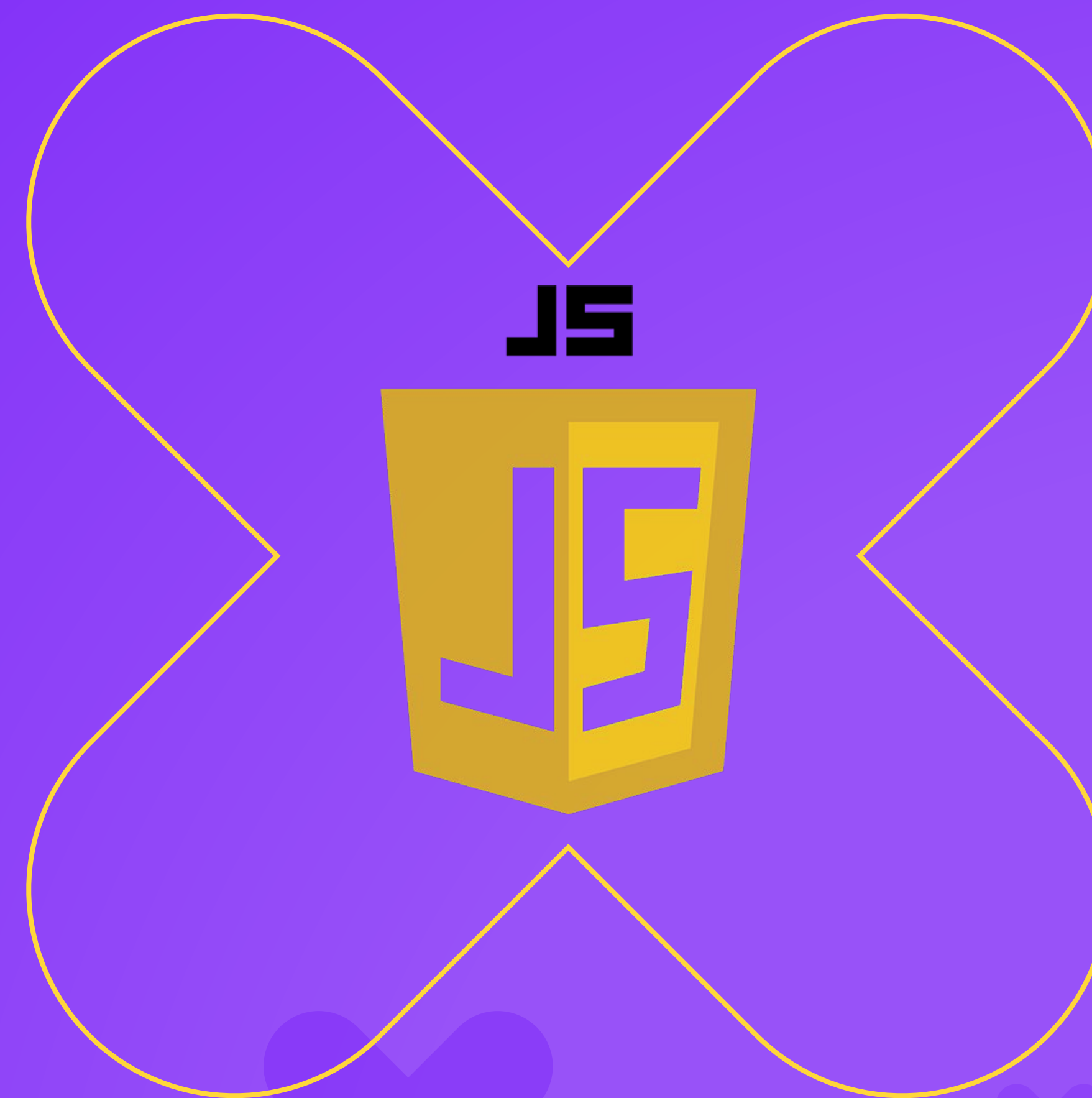




ЗАНЯТИЕ №20-22

События



Введение

События

События - это внешние действия по отношению к элементу.

Событийная система - это основа Javascript.

Примеры событий: щелчок кнопкой мыши, наведение, нажатие клавиши на клавиатуре и т.д.

Обработчик события

Обработчиком события называют функцию, привязанную к событию. Данная функция выполнится в момент, когда произойдёт событие.

Обработчики события - это обычные функции, и писать их нужно как обычные функции.

В дальнейшем в презентации подразумевается, что есть некая функция fn , которая выполняет определенные действия.

Названия события

В DOM существуют заранее определенные события. Вот некоторые из них:

`click` - щелчок левой кнопкой мыши

`contextmenu` - вызов контекстного меню (щелчок правой кнопкой)

`mouseenter` - курсор входит в пределы элемента

`mouseleave` - курсор покидает пределы элемента

`mousemove` - курсор движется

Способы назначения событий

1. Через html код в атрибутах - атрибут записывается как on + название события.

```
<button onclick = "fn()">Press</button>
```

Когда произойдет событие, выполнится код, записанный в onclick, поэтому функция fn записана со скобками.

Данный подход назначения событий не совсем удобен, так как нет разделения HTML и JS.

Способы назначения событий

2. Свойство объекта Element, полученного через JS. Свойство называется on + название события.

```
elem.onclick = fn
```

Данный способ удобнее и быстрее всех, однако имеет одно неудобство - через onclick нельзя назначить несколько обработчиков.

Способы назначения событий

3. С использованием метода `addEventListener` объекта `Element`, полученного в JS.

Первым параметром в метод передаётся строковое значение с названием события, вторым параметром - функция.

```
elem.addEventListener("click", fn)
```


Удаление обработчика события

Обработчик, назначенный через `addEventListener` можно удалить с помощью метода `removeEventListener`.

```
elem.removeEventListener("click", fn).
```

Важно! Если обработчик события был назначен с помощью анонимной функции, удалить его не получится.

`removeEventListener` требует имя функции для удаления.

Конец

