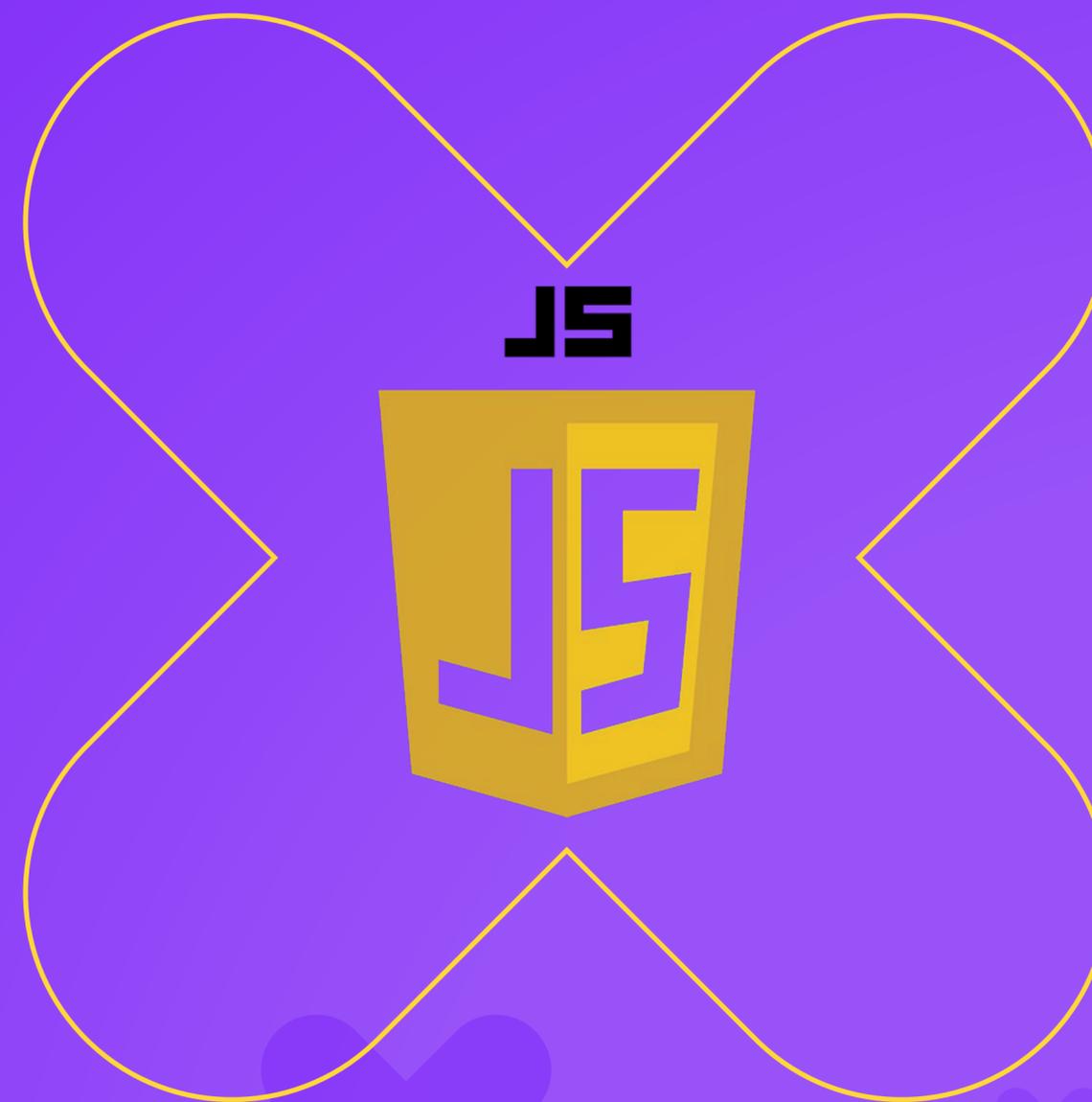




ЗАНЯТИЕ №20-22

# Объект события



# Классы событий

---

События можно поделить на несколько групп:

- события мыши
- события клавиатуры
- события DOM
- события элементов управления

# Объект события

---

В функцию - обработчик события в момент срабатывания передаётся объект события.

Объект события - это объект с информацией о произошедшем событии.

В зависимости от класса события объект несет разную информацию.

# Работа с объектом события

---

```
elem.onclick = function(ev){  
    console.log(ev);  
}
```

`ev` - имя параметра указанного для функции, можно заменить любым другим именем.

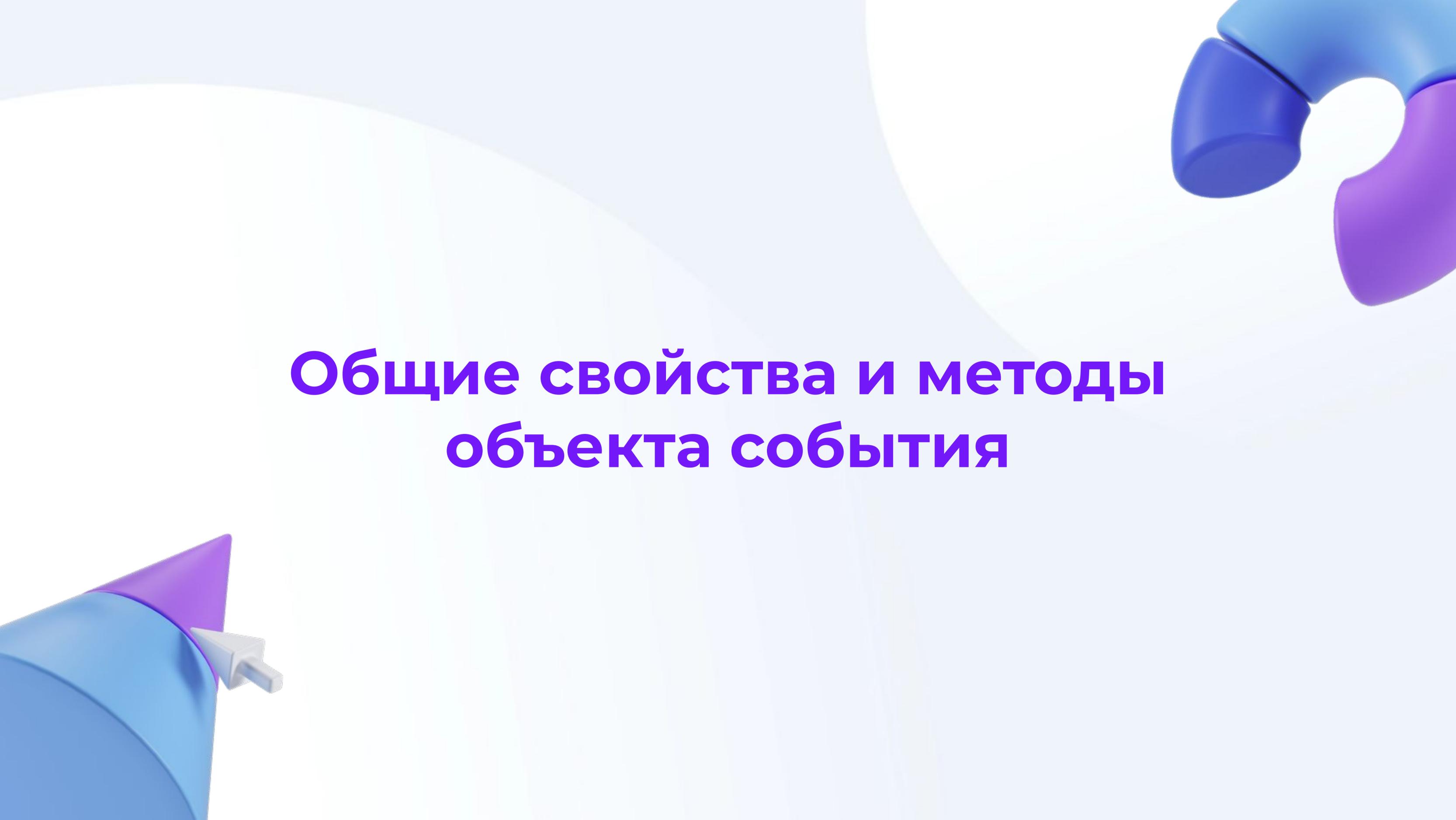
Этот параметр и будет объектом события.

# Работа с объектом события

---

Если вы задаете событие с помощью атрибутов в HTML, то объект события будет доступен с помощью переменной `event`.

```
<button onclick="console.log(event)">Press</button>
```

The background features abstract 3D shapes in shades of light blue, dark blue, and purple. A large, light blue curved shape arches across the top. In the bottom left, there is a blue and purple object with a white rectangular protrusion. In the top right, there is a blue and purple ring-like structure.

# **Общие свойства и методы объекта события**

# 1. Отмена действия по умолчанию

---

За отмену действия по умолчанию отвечает метод `preventDefault` объекта события.

```
elem.onclick = function(ev){  
    ev.preventDefault();  
}
```

## 2. Отмена всплытия

---

Всплытие события - это ситуация, когда событие одного элемента, происходит и с другим. Рассмотрим пример:

```
<div id="div" onclick="alert('div') ">  
  <p onclick="alert('p') ">  
    <button onclick="alert('button') ">Press</button>  
  </p>  
</div>
```

## 2. Отмена всплытия

---

Так как кнопка находится внутри тега `p`, который находится внутри `div`, то по щелчку по кнопке событие `click` сработает и у тега `p`, и у тега `div`.

Данное событие сначала происходит у самого “глубокого” элемента, а затем поднимается выше (поэтому всплытие).

Метод `stopPropagation` у объекта события отменяет всплытие, то есть дальше событие не всплывает.

# Пример

```
<div id="div" onclick="alert('div')">  
  <p onclick="alert('p')">  
    <button onclick="event.stopPropagation();  
alert('button')">Press</button>  
  </p>  
</div>
```

## 3. target vs currentTarget

---

Оба свойства ссылаются на источник события, однако `target` - это элемент, на котором событие возникло, а `currentTarget` - элемент, на который добавили обработчик (как с `this`).

`currentTarget` появляется, когда есть всплытие события.

Далее указан код, на котором можно отследить разницу.

# Пример

---

```
<div id="div" onclick="console.log(event.currentTarget,  
event.target) ">  
  <p onclick="console.log(event.currentTarget, event.target) ">  
    <button onclick="console.log(event.currentTarget,  
event.target) ">Press</button>  
  </p>  
</div>
```

# Другие свойства объекта события

---

- `timeStamp` - время в мс, прошедших с момента загрузки страницы до события
- `type` - имя события
- `defaultPrevented` - указывает, был ли вызван метод `preventDefault`

# События мыши



# События мыши

---

Дополнительные события мыши

- `dblclick` - двойной клик
- `mousedown` - кнопка мыши опущена вниз
- `mouseup` - кнопка мыши поднята вверх

# Свойства объекта события мыши

---

Координаты:

- `pageX`, `pageY` - координаты от начала документа (как и `position: absolute`)
- `clientX`, `clientY` - координаты от окна (прокрутка не учитывается)
- `offsetX`, `offsetY` - координаты от начала самого элемента

# Свойства объекта события мыши

---

Дополнительные клавиши:

- `ctrlKey` - true, если была нажата клавиша `ctrl`
- `altKey` - true, если была нажата клавиша `alt`
- `shiftKey` - true, если была нажата клавиша `shift`

`which` - какая кнопка была нажата (работает с `mousedown` и `mouseup`).

1 - левая, 3 - правая, 2 - колесико (средняя)

# buttons и button

---

<https://developer.mozilla.org/ru/docs/Web/API/MouseEvent/buttons>

<https://developer.mozilla.org/ru/docs/Web/API/MouseEvent/button>

# События клавиатуры



# События клавиатуры

---

- `keydown` - клавиша клавиатуры опущена вниз, нажата (предпочтительный вариант)
- `keyup` - клавиша клавиатуры поднята наверх, вернулась в исходное состояние
- `keypress` - нажата символьная клавиша

# Добавление обработчика события

---

Событие “нажатие клавиши” может происходить только в элементах, в которых нажатие клавиши работает (элементы управления), либо его можно добавить на глобальный объект `window`

# Основные свойства объекта события

---

- `key` - обозначение нажатой клавиши в виде строки (например, `"w"`)
- `code` - код нажатой клавиши в виде строки (например, `"keyW"`)
- `keyCode` - численный код нажатой клавиши
- `altKey`, `ctrlKey`, `shiftKey` - по аналогии с событиями мыши

# События загрузки страницы

---

```
window.onload = fn;
```

- Срабатывает в момент когда ВСЕ данные загружены (включая картинки, файлы css)
- Можно добавлять обработчик в свойства и через `addEventListener`
- Есть у объекта `window`

```
document.addEventListener("DOMContentLoaded", fn);
```

- Срабатывает в момент, когда загружено дерево DOM, без картинок и стилей
- Добавляется только через `addEventListener`
- Есть у объекта `document`

**Конец**

