JAVASCRIPT

Работа с DOM

DOM

Навигация - Свойства

```
document.body
```

- *это элемент, который включает в себя содержимое страницы.*
- *может быть заменено, но это удалит все его дочерние элементы.*

```
document.forms
document.forms[index]
```

```
document.forms['id']
document.forms['name']
```

```
Node.firstChild
```

- *возвращает первый узел в списке своих прямых детей или null.*

```
Node.lastChild
```

- *возвращает последний узел в списке своих прямых детей или null.*

```
Node.childNodes
```

- *возвращает коллекцию (NodeList) дочерних элементов данного элемента.*

```
Node.children
```

- *возвращает коллекцию (HTMLCollection) дочерних элементов узла.*

DOM

Навигация - Свойства

```
Node.parentNode
```

- *возвращает родителя определенного элемента DOM дерева.*
- ***null**, если элемент только был создан и еще не добавлен в DOM дерево.*

```
Node.nextSibling
```

- *возвращает узел, непосредственно следующий за данным узлом в списке **childNodes** его родительского элемента*
- ***null** если данный узел последний в этом списке*

```
Node.previousSibling
```

- *возвращает узел предшедствующий указанному в родительском элементе **childNodes***
- ***null**, если указанный узел первый в своём родителе*

DOM

Навигация – Методы

```
Document.createElement()
```

- *создает указанный в аргументе элемент*
- ***HTMLUnknownElement**, если элемент неизвестен.*

```
Document.createTextNode()
```

- *Создает новый текстовый узел*

```
Node.appendChild()
```

- *Добавляет элемент в конец списка дочерних элементов родителя.*
- *Если элемент уже существует он удаляется из текущего родителя и вставляется заново*

```
Node.insertBefore()
```

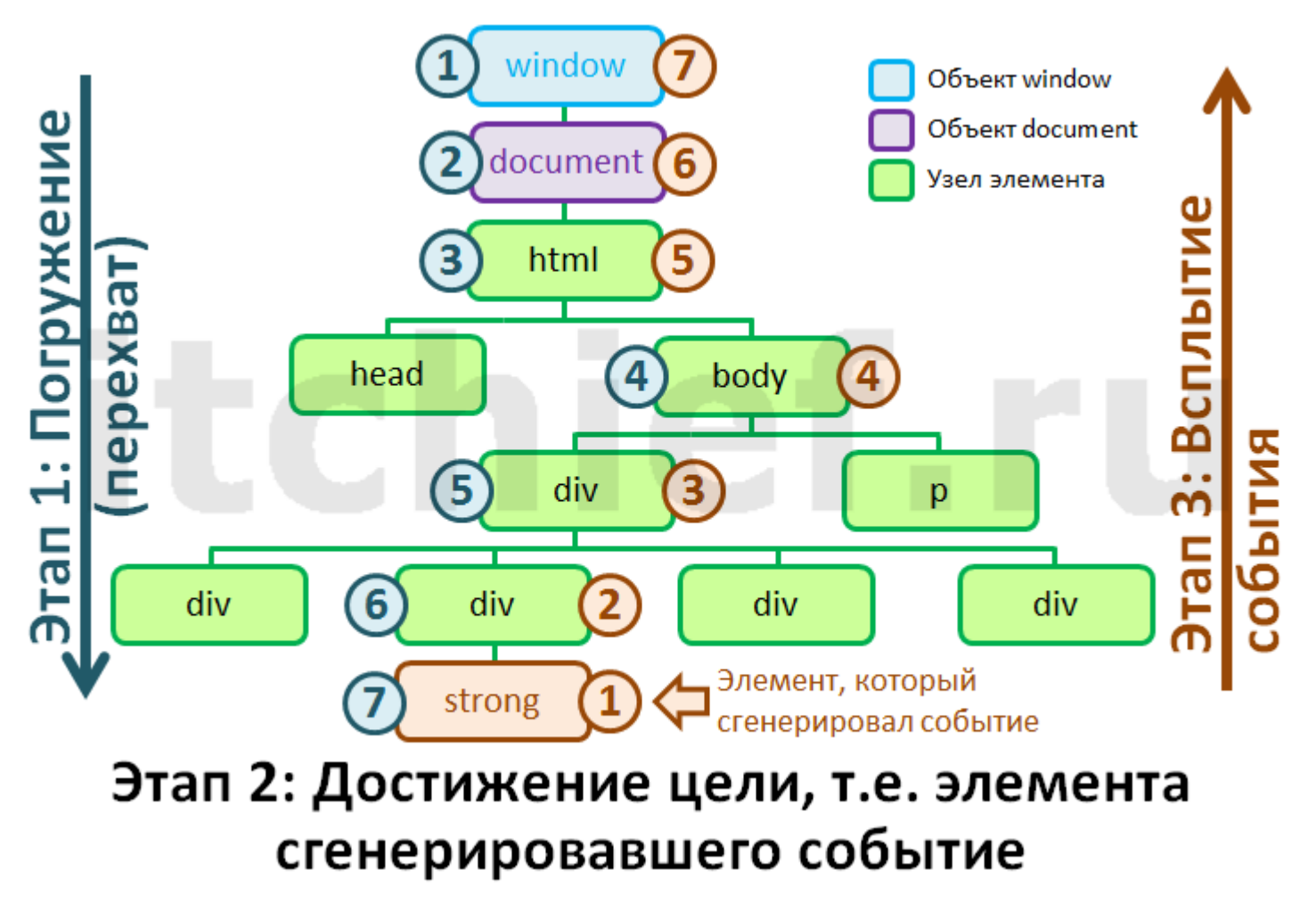
- *добавляет элемент в список дочерних элементов родителя перед указанным элементом*

```
Node.cloneNode()
```

- *возвращает дубликат узла, из которого этот метод был вызван*

DOM

Погружение и всплытие событий



- ***event.stopPropagation()***

препятствует продвижению события дальше, но на текущем элементе все обработчики отработают.

- ***event.stopImmediatePropagation()***

не только предотвращает всплытие, но и останавливает обработку событий на текущем элементе

DOM

Действия браузера по умолчанию

- *Клик по ссылке инициирует переход на новый URL.*
- *Нажатие на кнопку «отправить» в форме – отсылку ее на сервер.*
- *Двойной клик на тексте – инициирует его выделение.*

Для отмены действия браузера:

- *вызвать* ***event.preventDefault()***
- *можно просто вернуть* ***false*** *из обработчика.*

DOM

Делегирование

```
var table = document.querySelector('table');
table.addEventListener('click', function(event) {
  var target = event.target; // где был клик

// если это не TD, нас не интересует
  if (target.tagName !== 'TD') {
    return;
  }
    alert(target.tagName);
  }
});
```

Внутри обработчика:

- ***this*** *– элемент, на который повешен обработчик*
- ***event.target*** *– элемент, на котором был клик.*

DOM

Делегирование

```
var table = document.querySelector('table');
table.addEventListener('click', function(event) {
  var target = event.target;

  // цикл двигается вверх от target к родителям до table
  while (target != this) {
    if (target.tagName == 'TD') {
      // нашли элемент, который нас интересует!
      alert(target.tagName);
      return;
    }
    target = target.parentNode;
  }

  // возможна ситуация, когда клик был вне <td>
  // если цикл дошёл до table и ничего не нашёл,
  // то обработчик просто заканчивает работу
})
```

DOM

Resource Events

- ***load*** *Ресурс и его зависимые ресурсы завершили загрузку*
- ***beforeunload*** *Window, document и его зависимые ресурсы должны будут быть выгружены, срабатывает до* ***unload***
- ***unload*** *document собирается быть выгружен (уход со страницы).*

Session History Events

- ***pagehide*** *когда пользователь уходит со страницы, но воспрепятствует запись в кэш*
- ***pageshow*** *тоже что и load, но срабатывает каждый раз при загрузке страницы (даже из кэша)*

DOM

View Events

- ***resize*** *(window)при изменении размеров окна браузера*
- ***scroll*** *(window) при прокрутке окна браузера*

DOM

Mouse Events

- ***mouseenter*** *при попадании курсора мыши на сам элемент или его дочерний элемент*
- ***mouseover*** *при попадании курсора мыши на элемент, даже если курсор мыши пришёл с дочернего элемента.*
- ***mouseleave*** *при покидании курсором мыши самого элемента и его дочерних элементов*
- ***mouseout*** *при покидании курсором мыши элемента, даже если курсор мыши ушёл на дочерний элемент*
- ***mousemove*** *при движении курсора мыши по элементу*
- ***mousedown*** *при нажатии кнопки мыши на элементе*
- ***mouseup*** *при отпускании кнопки мыши на элементе*
- ***click*** *при щелчке левой кнопкой мыши на элементе*
- ***dblclick*** *при двойном щелчке мышью на элементе*
- ***contextmenu*** *при щелчке правой кнопкой мыши по элементу (before the context menu is displayed).*
- ***wheel*** *при вращении колеса мыши над элементом*

DOM

Touch events

- ***touchstart*** *при прикосновении пальцем к элементу*
- ***touchend*** *при отпускании пальца от элемента*
- ***touchmove*** *при перемещении прижатого к элементу пальца*
- ***touchcancel*** *когда во время касания элемента веб-страница теряет фокус (например, при входящем звонке)*

DOM

Mouse Events

- ***dragstart*** *у перетаскиваемого элемента, когда пользователь нажал левую кнопку мыши, т.е. начал перетаскивать элемент*
- ***drag*** *у перетаскиваемого элемента при каждом перемещении мыши*
- ***dragend*** *у перетаскиваемого элемента, когда пользователь отпустил кнопку мыши, т.е. закончил перетаскивать элемент (независимо от того, брошен элемент в приёмник или нет)*
- ***dragenter*** *у элемента-приёмника, когда в него входит перетаскиваемый элемент*
- ***dragover*** *у элемента-приёмника, когда в нём находится перетаскиваемый элемент, при каждом перемещении мыши*
- ***dragleave*** *у элемента-приёмника, когда из него выходит перетаскиваемый элемент*
- ***drop*** *у элемента-приёмника, когда в нём находится перетаскиваемый объект и левая кнопка мыши отпускается*

DOM

Mouse Events

- ***keydown*** *(input, textarea, select, a, button) при нажатии клавиши*
- ***keyup*** *(input, textarea, select, a, button) при отпускании клавиши*
- ***keypress*** *при вжатии клавиши, соответствующей печатному символу*

(нажатия управляющих клавиш, таких как Shift или F1, не вызывают события keypress)

таким образом, при вжатии клавиши с печатным символом происходят подряд события keydown и keypress, а при отпускании — keyup; при автоповторе (автоматическом многократном срабатывании клавиши при её удержании) повторяются пары событий keydown и keypress. input, textarea, select, a, button

DOM

Form's element Events

- ***select*** *(input type=text, textarea) при выделении текста в текстовом поле*
- ***focus*** *и* ***focusin*** *(input, textarea, select, a, button)*

при получении элементом фокуса, т.е. выделении элемента мышью или клавиатурой, или вызове метода элемент.focus()

- ***blur*** *и* ***focusout*** *(input, textarea, select, a, button)*

при потере элементом фокуса, т.е. выделении другого элемента или вызове метода элемент.blur()

- ***change*** *(input type=text, textarea, select)*

при изменении значения элемента ввода; для текстовых полей возникает только при потере элементом фокуса

- ***copy, cut, paste*** *(input type=text, textarea)*

при операциях копирования, вырезания, вставки (например, при нажатии клавиш Ctrl+C, Ctrl+X, Ctrl+V, или выборе мышью из выпадающего меню соответствующего пункта)

- ***submit*** *(form) при отсылке формы на сервер, но не при вызове форма.submit()*
- ***reset*** *(form) при сбросе содержимого формы, но не при вызове форма.reset()*

JAVASCRIPT

Работа с DOM