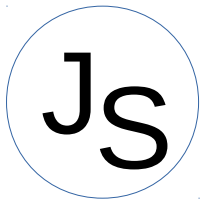


# Document Object Model

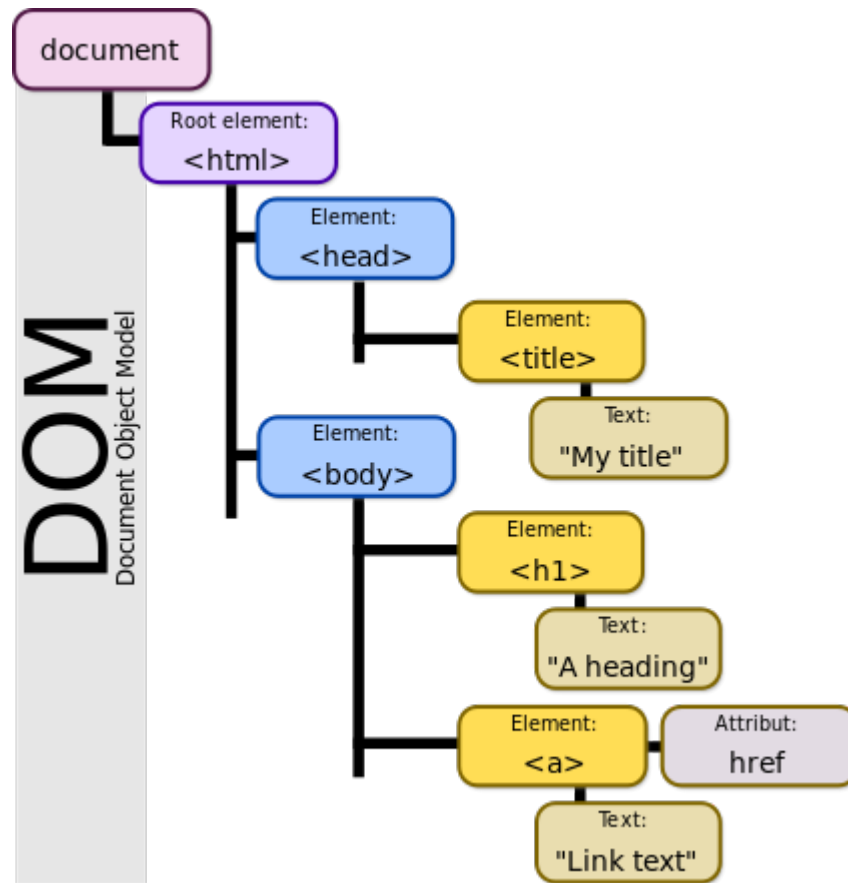


- DOM
- DOM Level 0
- W3C DOM
- Универсальное представление дерева
- Реализация DOM в HTML
- Версии W3C DOM
- Совместимость DOM в браузерах
- Обход DOM-дерева
- Поиск элементов в документе
- Модификация документа
- Добавление элементов в документ
- Свойство innerHTML

# DOM

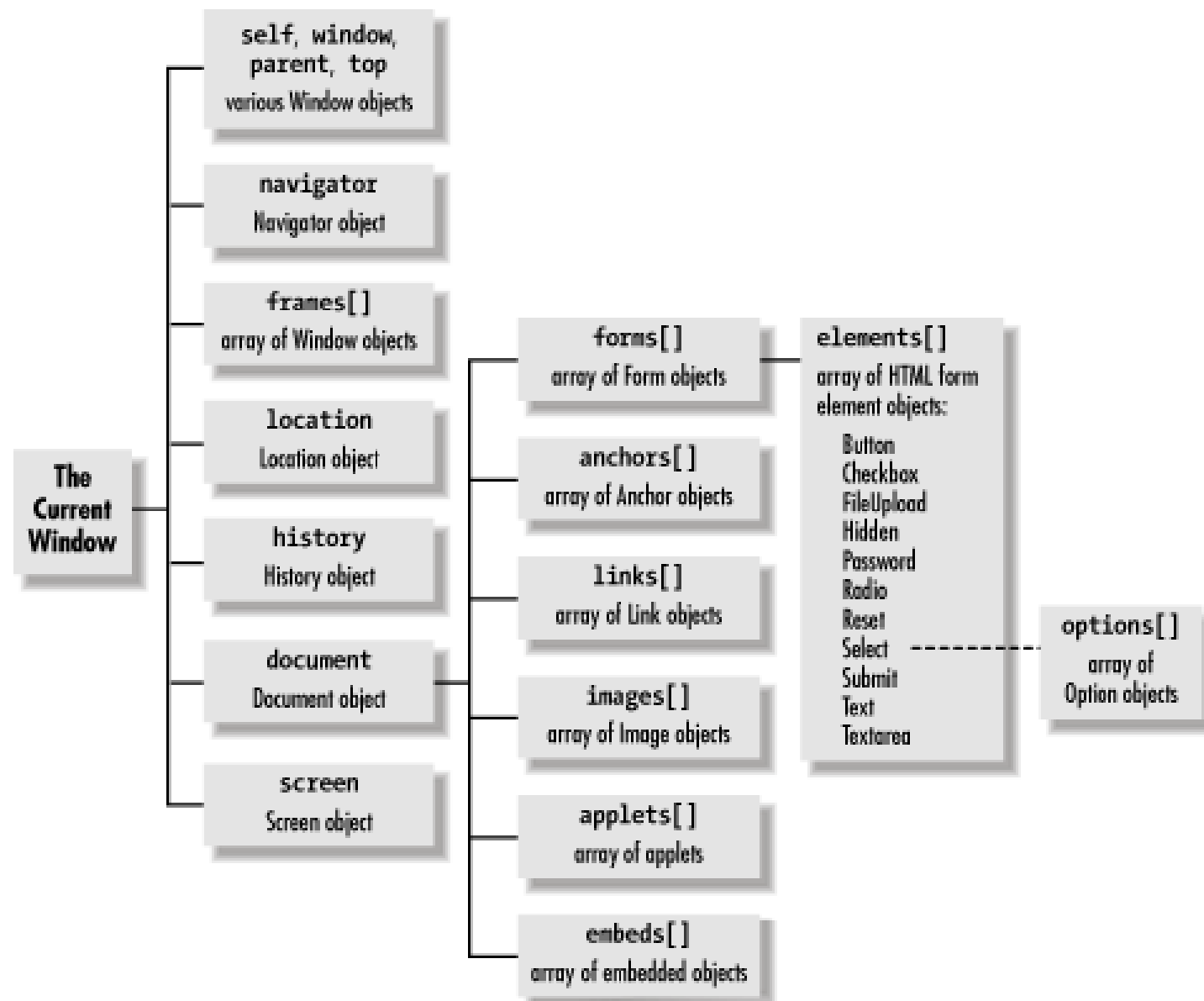


window.document



# Level 0 DOM

JS

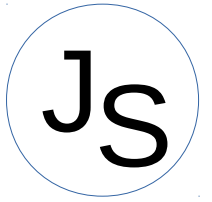


# W3C DOM



- Level 0 – до стандартизации
- Level 1 – 1988, 2000
- Level 2 – 2000, 2003 (HTML)
  - getElementById
  - модель событий
  - XML namespace
  - CSS
- Level 3 – 2003-2004
  - XPath
  - события клавиатуры
  - сериализация как XML
- DOM 4 – проект

# Универсальное представление дерева



## Node

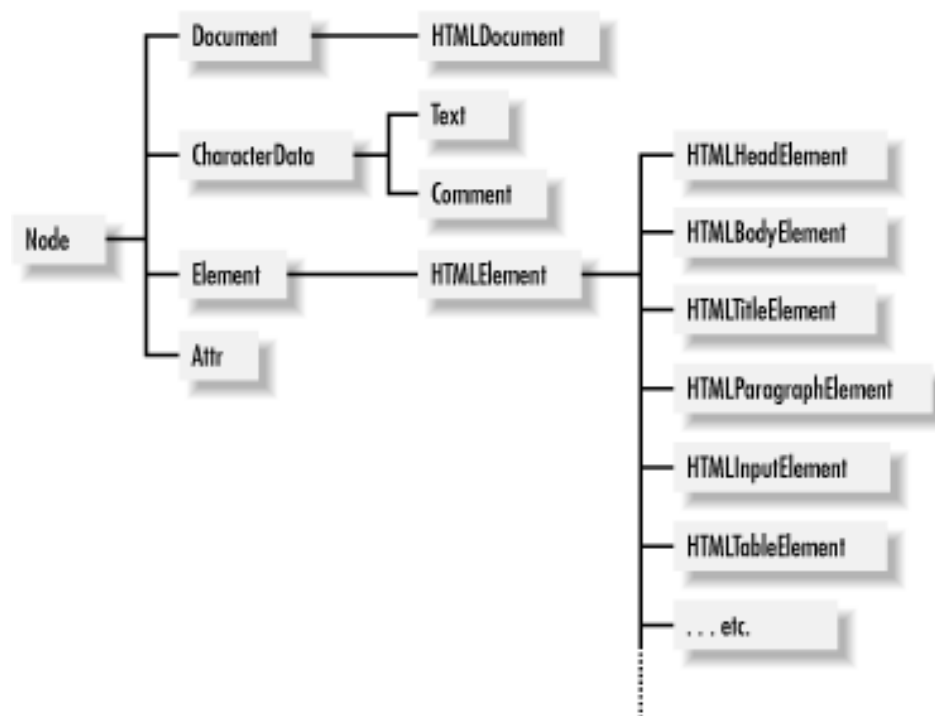
- nodeType
- nodeValue
- appendChild
- replaceChild
- removeChild
- insertBefore
- childNodes
- firstChild
- lastChild
- nextSibling
- previousSibling
- parentNode

# Реализация DOM в HTML

JS

## Типы узлов

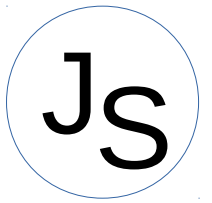
- Документ
- Фрагмент документа
- Элемент
- Атрибут
- Текстовый узел
- Комментарий



Node.DOCUMENT\_NODE  
Node.DOCUMENT\_FRAGMENT\_NODE  
Node.ELEMENT\_NODE

Node.ATTRIBUTE\_NODE  
Node.TEXT\_NODE  
Node.COMMENT\_NODE

# Совместимость DOM в браузерах



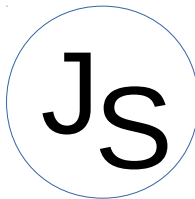
## Полифилл (polyfill)

```
if (document.documentElement.firstChild ===
undefined) {

    Element.prototype.firstChild = function() {
        var el = this.firstChild;
        do {
            if (el.nodeType === Node.ELEMENT_NODE) {
                return el;
            }
            el = el.nextSibling;
        } while (el);
        return null;
    };
}
```

```
<script
  src="//cdn.polyfill.io/v1/polyfill.js?features=es6"
></script>
```

# Обход дерева DOM



```
function displayIds (node) {  
    if (node.id != '') {  
        console.log (node.id) ;  
    }  
    for (  
        let i=0;  
        i<node.childNodes.length;  
        i++  
    ) {  
        displayIds (node.childNodes[i]) ;  
    }  
}  
  
displayIds (document) ;
```



# Поиск элементов в DOM



## getElementById или id

```
<div id="content-holder">
  <div id="content">Элемент</div>
</div>

<script>
  console.log(content);
  console.log(window['content-holder']);
</script>
```

```
<div id="content">Выделим этот элемент</div>

<script>
  var elem = document.getElementById('content');
  elem.style.background = 'red';
  console.log(elem == content); // true
  content.style.background = ""; // тот же элемент
</script>
```

# Поиск элементов в DOM



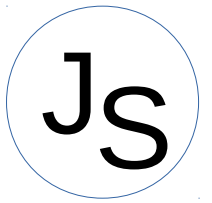
## element.getElementsByTagName

```
var tableElem = document.getElementById('age-table');  
var elems = tableElem.getElementsByTagName('input');  
var allElems = tableElem.getElementsByTagName('*');
```

## element.getElementsByClassName (кроме IE8-)

```
<div class="article">Статья</div>  
<div class="long article">Длинная статья</div>  
  
<script>  
  var articles =  
document.getElementsByClassName('article');  
  alert( articles.length ); // 2  
</script>
```

# Поиск элементов в DOM



## element.querySelectorAll

```
var elements = document.querySelectorAll(':hover');
```

## element.querySelector

```
document.querySelector('#list > li:last-child').  
    style.display = 'none';
```

## element.matches

```
if(elem.matches('a[href$="zip"]'))  
    elem.title = 'Скачать архив';
```

## element.closest

```
document.querySelector('td').closest('tr').  
    style.backgroundColor = '#55f';
```

# Поиск элементов в DOM



## element.querySelectorAll

```
var elements = document.querySelectorAll(':hover');
```

## element.querySelector

```
document.querySelector('#list > li:last-child').  
    style.display = 'none';
```

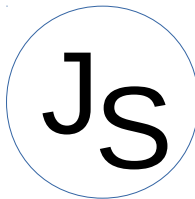
## element.matches

```
if(elem.matches('a[href$=".zip"]'))  
    elem.title = 'Скачать архив';
```

## element.closest

```
document.querySelector('td').closest('tr').  
    style.backgroundColor = '#55f';
```

# Модификация документа



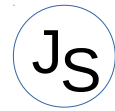
```
var div = document.createElement('div');
div.className = "alert alert-success";
div.innerHTML = "<strong>Внимание!</strong>";
document.body.insertBefore(
    div, document.body.firstChild
);
let text = "Важное сообщение.";
div.appendChild(document.createTextNode(text));
```

- createElement(tag)
- createTextNode(value)
- cloneNode(deep)

- innerHTML

- appendChild(elem)
- parent.insertBefore(elem, nextSibling)
- removeChild(elem)
- replaceChild(newElem, elem)

## Document Object Model

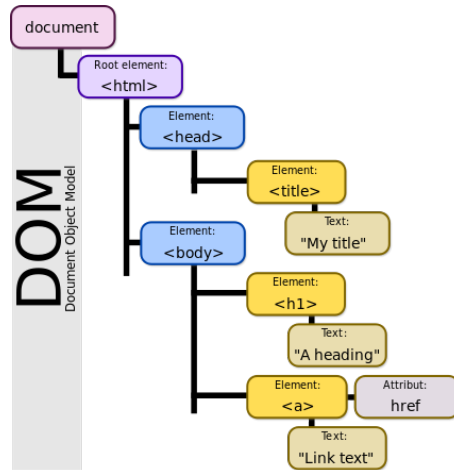


- DOM
- DOM Level 0
- W3C DOM
- Универсальное представление дерева
- Реализация DOM в HTML
- Версии W3C DOM
- Совместимость DOM в браузерах
- Обход DOM-дерева
- Поиск элементов в документе
- Модификация документа
- Добавление элементов в документ
- Свойство innerHTML

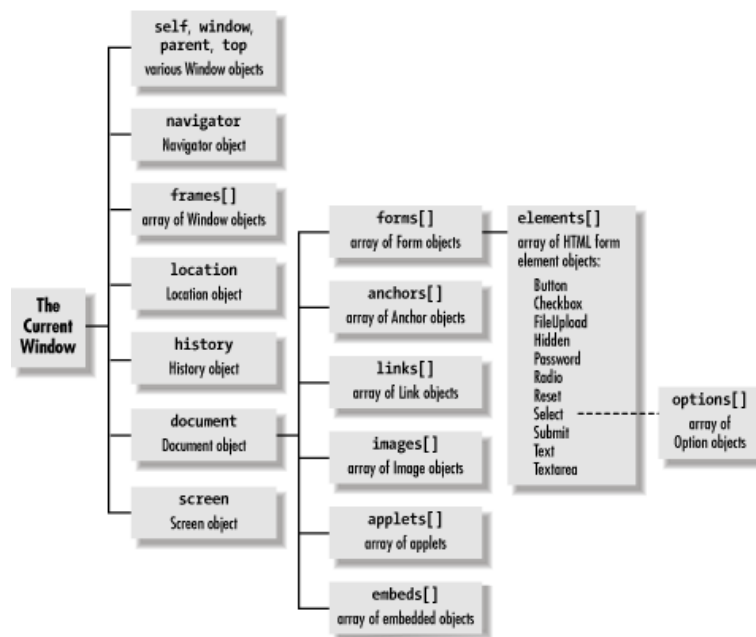
# DOM



window.document



# Level 0 DOM



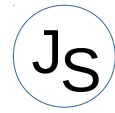


## W3C DOM



- Level 0 – до стандартизации
- Level 1 – 1988, 2000
- Level 2 – 2000, 2003 (HTML)
  - getElementById
  - модель событий
  - XML namespace
  - CSS
- Level 3 – 2003-2004
  - XPath
  - события клавиатуры
  - сериализация как XML
- DOM 4 – проект

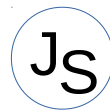
# Универсальное представление дерева



## Node

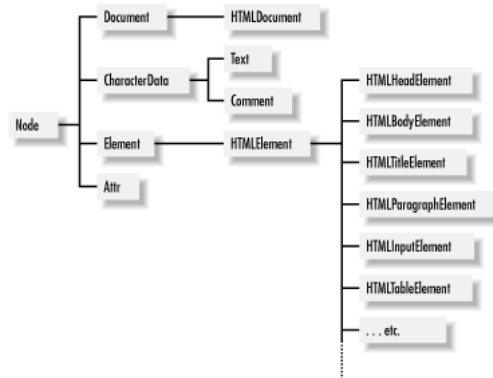
- nodeType
- nodeName
- parentNode
- firstChild
- lastChild
- nextSibling
- previousSibling
- childNodes
- appendChild
- replaceChild
- removeChild
- insertBefore

# Реализация DOM в HTML



## Типы узлов

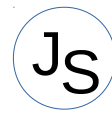
- Документ
- Фрагмент документа
- Элемент
- Атрибут
- Текстовый узел
- Комментарий



Node.DOCUMENT\_NODE  
Node.DOCUMENT\_FRAGMENT\_NODE  
Node.ELEMENT\_NODE

Node.ATTRIBUTE\_NODE  
Node.TEXT\_NODE  
Node.COMMENT\_NODE

# Совместимость DOM в браузерах



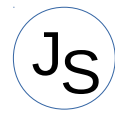
## Полифилл (polyfill)

```
if (document.documentElement.firstChild ===
undefined) {

  Element.prototype.firstChild = function() {
    var el = this.firstChild;
    do {
      if (el.nodeType === Node.ELEMENT_NODE) {
        return el;
      }
      el = el.nextSibling;
    } while (el);
    return null;
  }
};
```

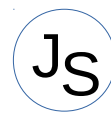
```
<script
  src="//cdn.polyfill.io/v1/polyfill.js?features=es6"
></script>
```

## Обход дерева DOM



```
function displayIds (node) {  
  if (node.id !== '') {  
    console.log (node.id);  
  }  
  for (  
    let i=0;  
    i<node.childNodes.length;  
    i++  
  ) {  
    displayIds (node.childNodes[i]);  
  }  
}  
  
displayIds (document);
```

## Поиск элементов в DOM



### getElementById или id

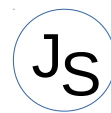
```
<div id="content-holder">
  <div id="content">Элемент</div>
</div>

<script>
  console.log(content);
  console.log(window['content-holder']);
</script>
```

```
<div id="content">Выделим этот элемент</div>

<script>
  var elem = document.getElementById('content');
  elem.style.background = 'red';
  console.log(elem == content); // true
  content.style.background = ""; // тот же элемент
</script>
```

## Поиск элементов в DOM



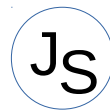
### element.getElementsByTagName

```
var tableElem = document.getElementById('age-table');  
var elems = tableElem.getElementsByTagName('input');  
var allElems = tableElem.getElementsByTagName('*');
```

### element.getElementsByClassName (кроме IE8-)

```
<div class="article">Статья</div>  
<div class="long article">Длинная статья</div>  
  
<script>  
  var articles =  
document.getElementsByTagName('article');  
  alert( articles.length ); // 2  
</script>
```

## Поиск элементов в DOM



### element.querySelectorAll

```
var elements = document.querySelectorAll(':hover');
```

### element.querySelector

```
document.querySelector('#list > li:last-child').  
  style.display = 'none';
```

### element.matches

```
if(elem.matches('a[href$=".zip"]'))  
  elem.title = 'Скачать архив';
```

### element.closest

```
document.querySelector('td').closest('tr').  
  style.backgroundColor = '#55f';
```



# Поиск элементов в DOM



## element.querySelectorAll

```
var elements = document.querySelectorAll(':hover');
```

## element.querySelector

```
document.querySelector('#list > li:last-child').  
style.display = 'none';
```

## element.matches

```
if(elem.matches('a[href$=".zip"]'))  
    elem.title = 'Скачать архив';
```

## element.closest

```
document.querySelector('td').closest('tr').  
style.backgroundColor = '#55f';
```

# Модификация документа



```
var div = document.createElement('div');
div.className = "alert alert-success";
div.innerHTML = "<strong>Внимание!</strong>";
document.body.insertBefore(
    div, document.body.firstChild
);
let text = "Важное сообщение.";
div.appendChild(document.createTextNode(text));
```

- createElement(tag)
- createTextNode(value)
- cloneNode(deep)

- innerHTML

- appendChild(elem)
- parent.insertBefore(elem, nextSibling)
- removeChild(elem)
- replaceChild(newElem, elem)