Function call

- Ordinary
- Object method like
- apply() & call()

Constructor

- Ordinary
- Object method like

Prototype

- The Object.prototype property represents the Object prototype object
- Nearly all objects in JavaScript are instances of Object
- Typical object inherits properties (including methods) from Object.prototype, although these properties may be shadowed (a.k.a. overridden).
- An Object may be deliberately created for which this is not true (e.g. by Object.create(null)), or it may be altered so that this is no longer true (e.g. with Object.setPrototypeOf).

Prototype Properties Section

1. Object.prototype.constructor
   - Specifies the function that creates an object's prototype.
2. Object.prototype.__proto__
   - Points to the object which was used as prototype when the object was instantiated.
3. Object.prototype.__noSuchMethod__
   - Allows a function to be defined that will be executed when an undefined object member is called as a method.

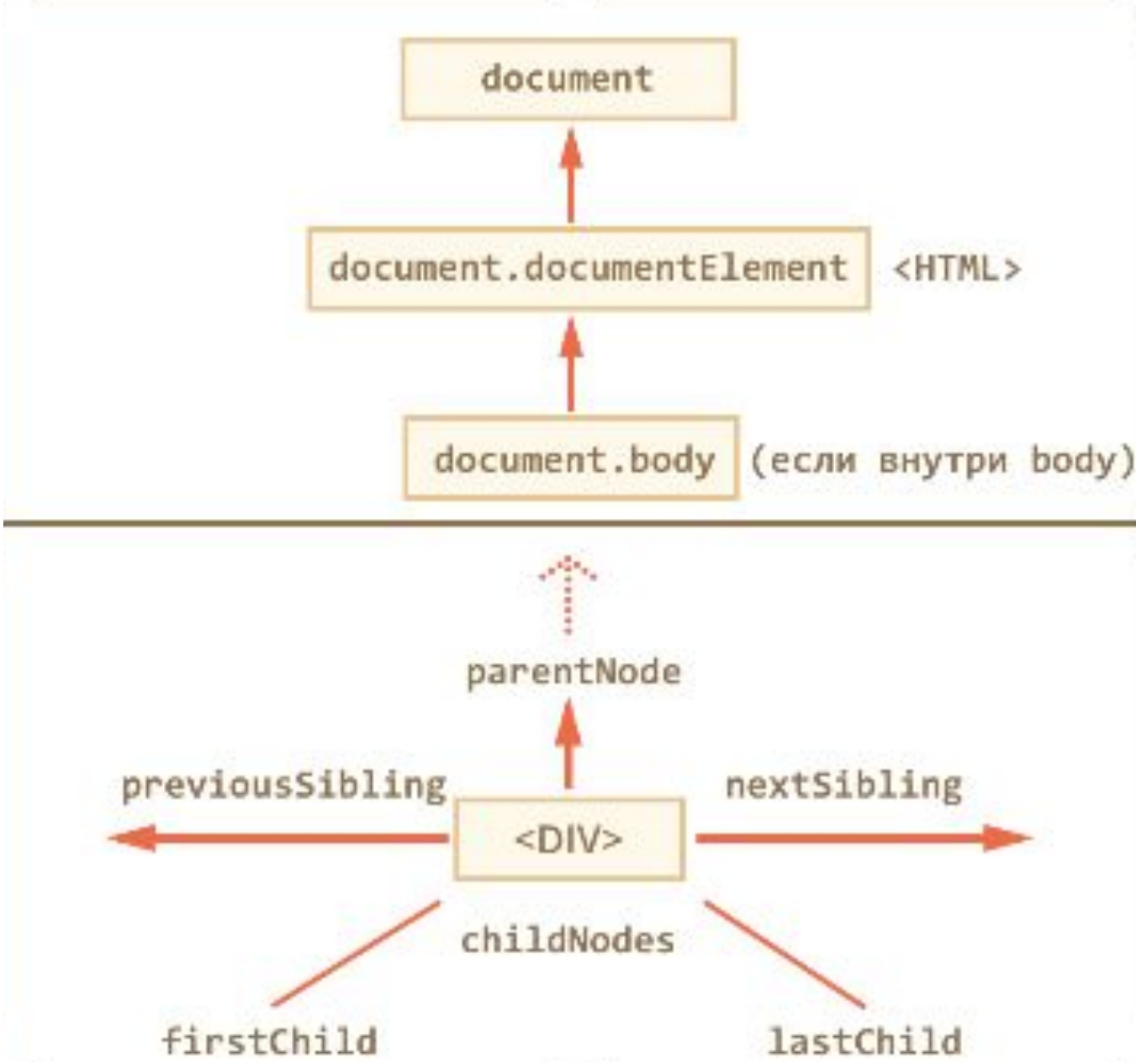
Prototype Methods

Main:

- Object.prototype.hasOwnProperty()
- Object.prototype.isPrototypeOf()
- Object.prototype.toString()

__Proto__

- While Object.prototype.__proto__ is supported today in most browsers, its existence and exact behavior has only been standardized in the ECMAScript 2015 specification as a legacy feature to ensure compatibility for web browsers. For better support, it is recommended that only Object.getPrototypeOf() be used instead.


document
document.documentElement
<HTML>
document.body
(если внутри body)
parentNode
previousSibling
nextSibling
<DIV>
childNodes
firstChild
lastChild

Element access

```
interface Node {
  //Different value of nodeType
  const unsigned short ELEMENT_NODE = 1;
  const unsigned short ATTRIBUTE_NODE = 2;
  const unsigned short TEXT_NODE = 3;
  const unsigned short CDATA_SECTION_NODE = 4;
  const unsigned short ENTITY_REFERENCE_NODE = 5;
  const unsigned short ENTITY_NODE = 6;
  const unsigned short PROCESSING_INSTRUCTION_NODE = 7;
  const unsigned short COMMENT_NODE = 8;
  const unsigned short DOCUMENT_NODE = 9;
  const unsigned short DOCUMENT_TYPE_NODE = 10;
  const unsigned short DOCUMENT_FRAGMENT_NODE = 11;
  const unsigned short NOTATION_NODE = 12;
  ...}
```

HTMLCollection

```
1.  // 1 var images = document.images;
2.  console.log(images.length);//1
3.  var img = document.createElement('img');
4.  document.body.appendChild(img);
5.  console.log(images.length);//2

6.  // function foreach(){
7.      var collections = document.body.childNodes;
8.      [].forEach.call(collections, function(item, index){
9.          console.log(item);
10.         console.log(index);
11.     });
12. }
```

document.head, document.title, document.links, document.images

```
function head() {
    console.log(document.head);
    document.head = '<h1>Test</h1>';
    console.log(document.head);
}
```

```
function title() {
    console.log(document.title);//Navigation
    document.title = 'Test';
    console.log(document.title);//Test
}
```

```
function links(){
    console.log(document.links);
    document.links = [];
    console.log(document.links);
}
```

```
function images(){
    console.log(document.images);
    document.images = [];
    console.log(document.images);
}
```

document.documentElement

```
<html>
▼<head>
    <title>Navigation</title>
  </head>
▼<body>
  ▼<div id="container">
    ▶<div class="content">…</div>
    ▼<div class="content">
      ▶<div class="title">…</div>
      ▶<div class="message">…</div>
      </div>
    </div>
    <script src="navigation.js"></script>
  ▶<script>…</script>
  </body>
</html>
```

```
1 console.log(document.documentElement);
2 //root element in document (html)
```

childNodes

- console.log(document.body.childNodes);

Methods

- children – the only child of the element nodes, that is, the corresponding tags
- firstElementChild, lastElementChild -respectively, the first and last children-elements
- previousElementSibling, nextElementSibling- neighbors-elements
- parentElement-parent-element

Property

Node.firstChild

- Read-only property
- Returns the first child of a node in the tree, or null if the node is childless
- If the node is a document, it returns the first node in the list of its direct children

Node.lastChild

- Read-only property
- Returns the last child of a node in the tree, or null if the node is childless
- If the node is a document, it returns the last node in the list of its direct children

Search elements

1. getElementById
2. getElementsByTagName
3. getElementsByName
4. getElementsByClassName
5. querySelectorAll
6. querySelector
7. closest