

Модуль - JavaScript

2/15 занятие

Массивы и циклы

Тип данных **Массив** (Array) позволяет хранения упорядоченные коллекции

Например: строк, чисел, нулей, объектов и html-элементов

```
let arr = new Array()
```

```
let arr = []
```

```
let arr = [1, 2, 'Яблоко']
```

Каждый элемент массива имеет индекс(порядковый номер), начиная с 0

По нему можно обратиться и получить ячейку данных

```
arr[2] === 'Яблоко'
```

Методы и свойства массива

`arr = []`

`arr.length` - свойство, содержит кол-во ячеек массива

методы:

`arr.push(что-то)` - добавляет элемент в конец массива

`arr.unshift(что-то)` - добавляет элемент в начало массива(медленно)

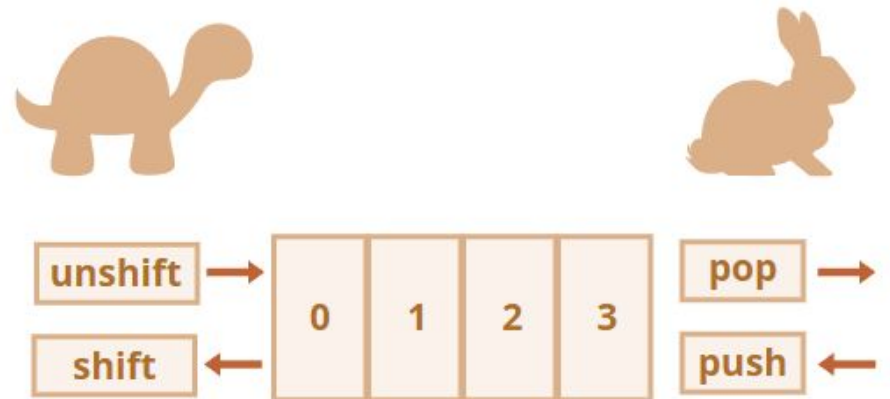
`arr.shift()` - удаляет элемент в начале, так что второй элемент становится первым

`arr.shift()` - удаляет первый элемент

`arr.toString()` - возвращает строку

`arr.join("")` - возвращает строку с раздел

`arr.indexOf(что-то)` - номер ячейки



Разбор кода

Let a = []

Let a = [].length

Let a = [1, 3, 5]

Let a = new Array().push(1)

Let a = [1, 3, 5].pop()

Многомерные массивы

```
let matrix = [  
    [1, 2, 3],  
    [4, 5, 6],  
    [7, 8, 9]  
]
```

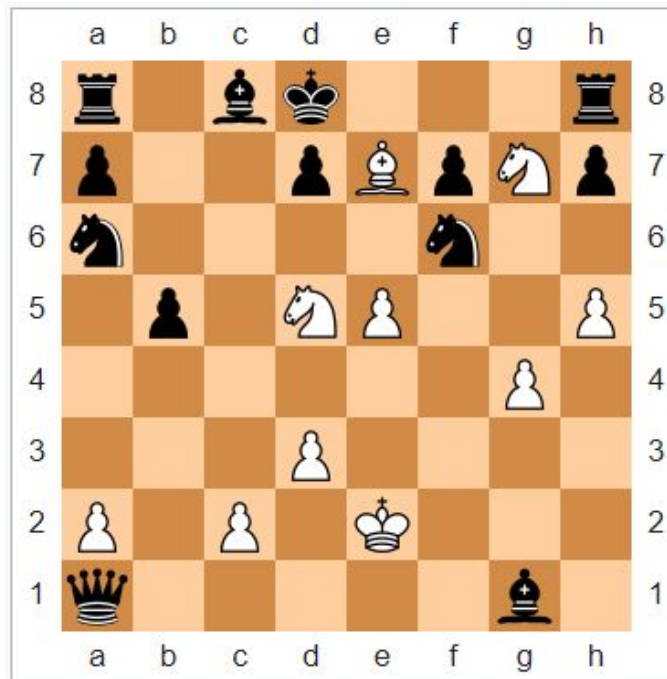
```
let num = matrix[1][1]
```

Правая часть читается в 3 действия - берем массив, затем выбираем элемент по индексу в первых скобках, затем из него уже берем элемент по индексу из вторых скобок

Задание 1

Создайте новый скрипт `chess.js`

Сохраните результат партии в виде массива `board`, называть фигуры можно произвольно, например “black queen”



Задание 2

Создайте новый скрипт bank.js

Решите задачу с помощью массивов и их методов

В банке работает всего 1 окошко и стоит очередь из 4 человек

```
const queue = ['Иван', 'Мария', 'Саша', 'Даша']
```

```
const outside = ['Амфибрахий']
```

За несколько часов из банка вышли Иван и Даша, но позже пришел Амфибрахий и занял последнее место в очереди.

Напишите такой код, чтобы в консоле было следующее

В очереди: Мария, Саша, Амфибрахий

Вне банка: Иван, Даша

```
console.log('В очереди: ' + queue + '\n ' + 'Вне банка: ' + outside)
```

Циклы

for - перебор ограниченной длины

++ - инкремент, увеличивает число на 1

```
for (let i = 0; i < 10; i++) {
```

```
    // тело цикла - код который будет выполнен 10 раз, при этом i каждый
```

```
    // раз(итерацию) будет на +1 больше
```

```
    console.log(i)
```

```
}
```

```
alert()    //этот код выполнится только после завершения цикла
```


for of

```
let arr = [1, 4, 6, 'Яблоко']
```

```
for (let item of arr) {
```

```
    console.log(item)
```

```
}
```

```
// 1
```

```
// 4
```

```
// 6
```

```
// 'Яблоко'
```

while

цикл while - перебор по условию для остановки

```
let i = 0
```

```
while (i < 3) {
```

```
    console.log( i++ )
```

```
    // выводит 0, затем 1, затем 2
```

```
}
```

Прерывание

break - прерывание цикла

continue - прерывание итерации

```
let i = 0
while (i < 6) {
  if (i === 1) {
    i++
    continue
  }
  if (i === 3) break
  console.log( i++ )
  // выводит 0, затем 2
}
```

while - опасный цикл, если убрать **i++** - будет выполняться бесконечно

Вложенные циклы

```
for (let i = 0; i < 10; i++) {  
  console.log(i)  
  for (let k = 0; k < 10; k++) {  
    console.log(k)  
    //этот код будет выполняться  $i * k$  раз  
  }  
}
```

Задание 3

Дополните chess.js из задания 1 таким образом, чтобы у вас в консоль выводились координаты каждой фигуры, например

Черный слон - g1

Черная ладья - a8

Белый король - e2

Примитивные типы данных

string(строка) - содержит произвольный набор символов внутри кавычек(')

number(число) - содержит цифры в числовом представлении

boolean(логический) - содержат бинарное true или false (да\нет)

undefined(неопределенный) - пустая ячейка, которая будет заполнена

null(ничто) - пустая ячейка, которая заполнена ничем(зануление)

symbol(символ) - уникальная строка

bigint(большое число) - для хранения больших чисел

```
Const name      = 'Кота зовут Барсик'    //string
Const age       = 3                      //number
Const isDog     = false                  //boolean
Const howMuchFoodWillCatEatToday          //undefined
Const barking   = null                   //Null
```

< JS >

перерыв

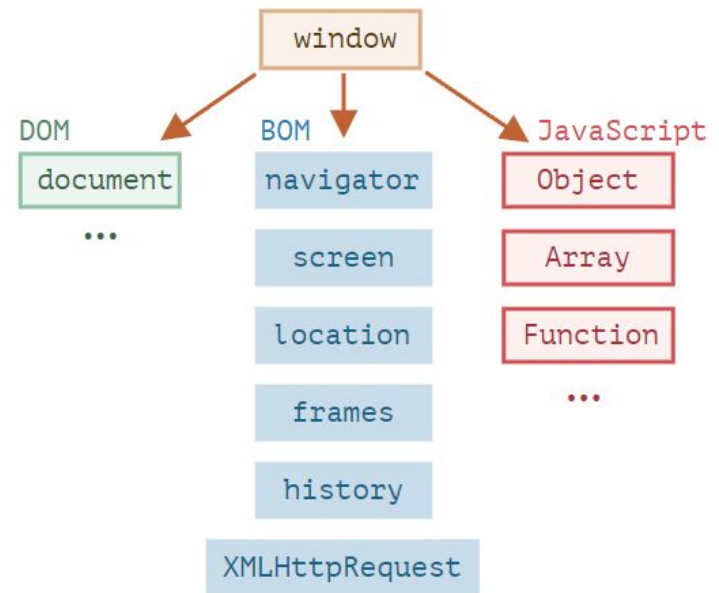
Работа с DOM и BOM

Объектная модель браузера (Browser Object Model, BOM) – это дополнительные объекты, предоставляемые браузером (окружением)

document - объект DOM

Из него можем получить любой элемент:

document.querySelector('css selector')



Практика

Исследуйте в консоли document, window, navigator, location, history

Измените свойство любого элемента на странице:

```
document.querySelector('div').style.backgroundColor = 'red'
```

Выбор DOM элементов

<code>Document.querySelector('.row .column')</code>	<code>//Element</code>
<code>Document.querySelectorAll('.row .column')</code>	<code>//NodeList</code>
<code>document.getElementById('newsBlock')</code>	<code>// Element</code>
<code>Document.getElementsByClassName('column')</code>	<code>// HTMLCollection</code>
<code>Document.getElementsByTagName('div')</code>	<code>// HTMLCollection</code>

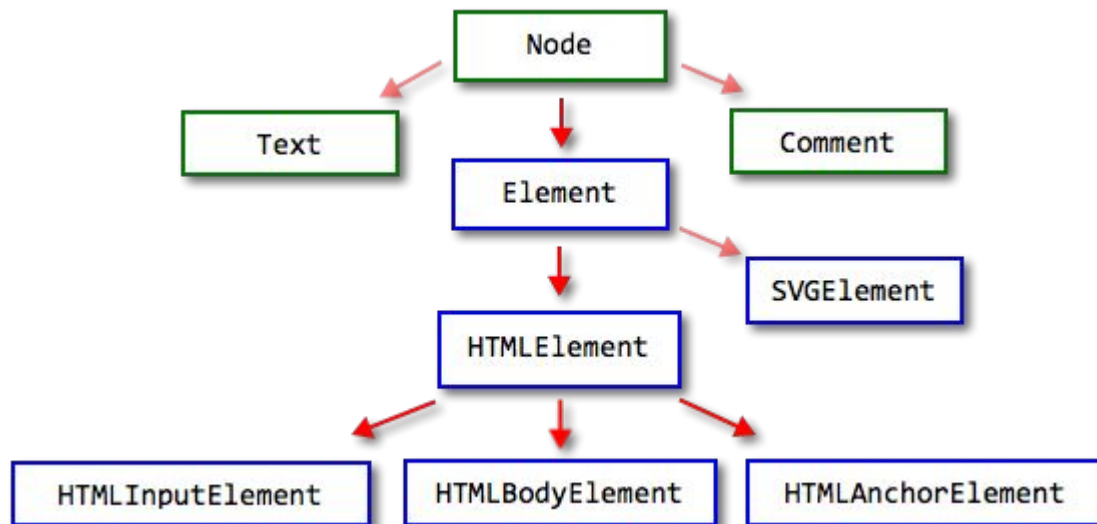
Узлы и элементы DOM

```
<div id="menu">some text <span>First span text</span></div>
```

```
let node = document.getElementById("menu")
```

```
let childNode = node.firstChild
```

```
let childElement = node.firstChild
```



Практика

Создание аккордеона

<https://getbootstrap.com/docs/5.0/components/accordion/>

Где let index = тому элементу, который должен быть развернут

Конец занятия