

# МАССИВЫ, ЦИКЛЫ, ВЕТВЛЕНИЯ, DOM

JavaScript

# СОЗДАНИЕ МАССИВОВ

- `var a=[];`
- `var a=[4, 9, 7];`
- `var a=[4, 'апельсин', 3];`
- `var a={"Иванов":15000, "Павлов":25000};`

Обращение к массиву

`a[1]`

`a["Иванов"]`

# МЕТОДЫ МАССИВОВ

- `reverse()` – изменение порядка элементов в массива;
- `sort()` – сортировка массива по возрастанию;
- `push()` – добавление элемента в конец;
- `unshift()` – добавляет элемент в начале;
- `shift()` – удаляет элемент вначале;
- `pop()` – удаляет элемент вконец;
- `slice(i,j)` – извлекает массив с *i*-го по *j*-ый элементы;
- `toLocaleString()` – преобразует массив в строку;
- `JSON.stringify(a)` – преобразует массив *a* в json формат;
- `JSON.parse(b)` – декодирует Json *b* в объект JavaScript

# ЗАДАНИЕ

- Создайте массив:  
Россия; Афганистан; Туркменистан; Белоруссия.
- Добавьте в конец массива: Казахстан
- Рассортируйте массив в алфавитном порядке;
- Удалите из массива элемент Афганистан

# ВЕТВЛЕНИЕ

- `if (year == 2015) {  
    alert( 'Да вы знаток!' ); }  
else {  
    alert( 'А вот и неправильно!' );`
- `==` и `!=` — проверяет одно значение идентично или не идентично другому.
- `<` и `>` — проверяет одно значение меньше или больше, чем другое.
- `<=` и `>=` — проверяет одно значение меньше или равно, либо больше или равно другому
- `&&` - логическое И
- `||` - логическое ИЛИ

# ЦИКЛЫ

- `while (condition) {`  
    `// код // также называемый "телом цикла«`  
    `}`
- `do {`  
    `// тело цикла }`  
`while (condition);`
- `for (начало; условие; шаг) {`  
    `// ... тело цикла ...`  
    `}`
- Прерывание цикла: «`break`»

# РАБОТА С DOM

## ПОИСК ЭЛЕМЕНТОВ

- `document.getElementById('ID');` //по id
- `document.getElementsByTagName('tag');` // по тегу
- `document.getElementsByClassName('class');` // по классу

## СОЗДАНИЕ ЭЛЕМЕНТОВ

- `element.innerHTML;` //вставить внутренний HTML
- `document.createElement(name);` //Создать элемент
- `parentNode.appendChild(node);`//Привязать элемент к родителю

## Передвижение по DOM

- `Node.parentNode;` //Найти родителя элемента
- `myElement.children` // Возвращает наследников
- `Node.firstChild;`
- `Node.lastChild`

# ЗАДАНИЕ

## ПРАКТИЧЕСКАЯ РАБОТА № 9

- Для страницы теста (практическая работа № 4) создайте обработчик, показывающий результаты тестирования