# Модуль - JavaScript

1/15 занятие

# Цели модуля

Изучим основы языка JavaScript и освоимся в среде выполения

Исследуем и закрепим практикой:

- Синтаксис языка
- Управление DOM и BOM
- Обработку событий
- Функциональное программирование и немного ООП
- Систему контроля версий Git
- Мемы

# Что такое JavaScript

**JavaScript** - интерпретируемый язык, для написания скриптов(программ)

Выполняется **движком**, чаще всего это **V8**, например в **браузере** или в **nodeJs**

**ECMAScript** - это стандарт языка JavaScript

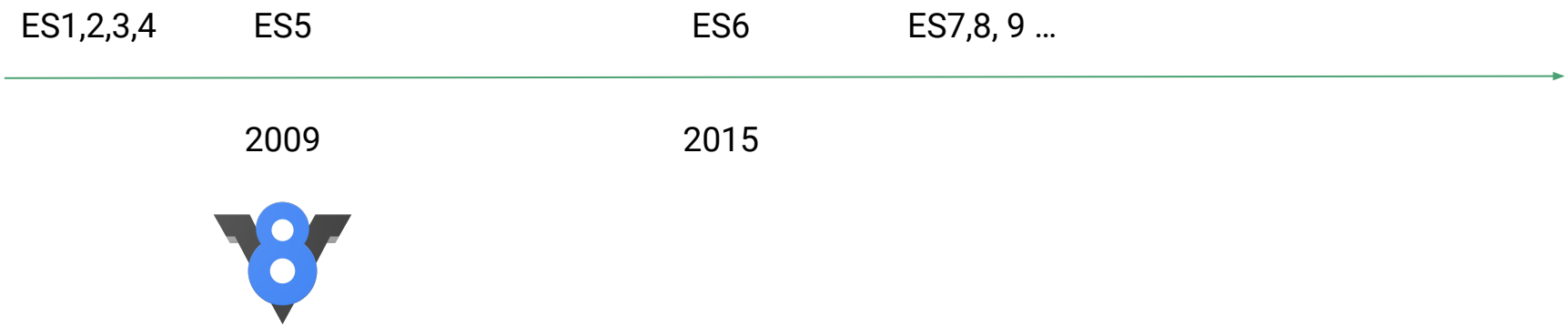
**Java** - компилируемый язык, напрямую не связан с JS

Когда тебя приняли на позицию Java-разработчика, хотя ты в резюме указывал, что ты JavaScript-разработчик

# Версии JavaScript

Стандарт ES5 работает во всех браузерах

Стандарты ES6+ работают в современных браузерах, однако до сих пор практикуется приведение(транспиляция) к стандарту ES5 при помощи утилиты Babel



# Как выполняется JS

Движок V8 запускает *изолированные* **однопоточные** контейнеры выполнения JS, например - вкладка браузера

Цикл выполнения JS кода:

- Код считывается(парсинг)
- Данные сохраняются во временную память
- Можно обращаться к движку и данным напрямую через консоль (DevTool)

# Практика

1. Откройте 2 страницы на разных вкладках браузера
2. В консоли запустите встроенную функцию ***alert()*** на каждой вкладке, но не совершайте действий с появившимся окошком(!!! Не нажимайте ОК !!!)
3. Попробуйте взаимодействовать с сайтом, проверьте клики или hover эффекты
4. Введите в консоле **console.log(123)**
5. Нажимайте ОК в модальном окошке
6. Опишите наблюдения

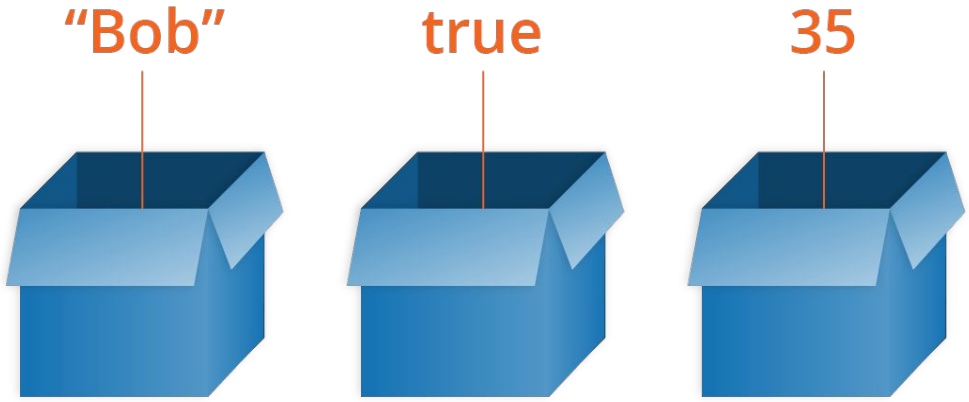
# Подключение

Порядок подключения очень важен, так как содержимое тега Script считывается при формировании DOM

```
<!DOCTYPE html>
<html>
	<head>
		<title>Web site</title>
		<script src="/app/main.js"></script>
	</head>
	<body>
		<div>text</div>
		<script>
			let a = 1;
			alert(a);
			// любой JS код
		</script>
		<div>text</div>
	</body>
</html>
```

# Переменные и директивы

**Объявление переменных** = автоматическое выделение области памяти, чтобы положить туда данные




Директивы(команды):

**Let** - создать переменную

**Const** - создать константу

**Var** - создать переменную в ES5

< JS >

```
let age = 35
Const  name = 'Bob'
Var isItUsedIn2023 = false

//Это однострочный комментарий
//И кстати, точки с запятой в конце строк тоже уже не используются
```

# 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

< JS >

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

# Базовые операторы

**=** Присваивание

**==** Сравнение с приведением типа к одному(чаще к строке)

**===** Строгое сравнение, с учетом типов

**!** Не(что-то), например не равно **!= и строго не равно(сравниваем типы) !==**

< JS >

```
Const age = 3

age = 4             //присвоили новое значение - 4
age == '4'          //true - несмотря на то, что '4' это строка
age === '4'         //false - потому что сравниваем число со строкой
age != '4'          //4 не равно '4', будет false
age !== '4'         //4 строго не равно '4', будет true
```

# Неявная типизация

Иногда приведение типов работает неожиданным образом

```
> typeof NaN
"number"
> 9999999999999999
10000000000000000
> 0.5+0.1==0.6
true
> 0.1+0.2==0.3
false
> Math.max()
-Infinity
> Math.min()
Infinity
> []+[]
""
> []+{}
"[object Object]"
> {}+[]
0
> true+true+true===3
true
> true-true
0
```

```
> true==1
true
> true===1
false
> (!+[]+[]+![]).length
9
> 9+"1"
"91"
> 91-"1"
90
> []==0
true
```

# Математические операторы

**+** сложение

**-** вычитание

**/** деление

***** умножение

**%** Взятие остатка от деления и

****** возведение в степень

**Конкатенация строк**

**+=** или **+** можно использовать для склейки строк (‘Кота зовут ’ + ‘Барсик’)

В результате будет строка ‘Кота зовут Барсик’

# Отладка в браузере

**Console.log(** *..что-нибудь..* **)** - вывод данных(любой тип) в консоли

**Alert(** *..что-нибудь..* **)** - вывод данных(строка) в окошке

**Prompt(** *..что-нибудь..* **)** - получение данных от пользователя(строка)

# Задание 1

Создайте example.html и подключите к нему script1.js:

1.Арифметика

- Объявите переменные *result* со значением 8 и *multiplier*
- Присвойте *multiplier* значение 2.5
- Присвойте *result* вычисление 8 умножить на *multiplier* , затем прибавить 674, все это разделить на 3.3

2. Верно ли утверждение? - *multiplier* !== '2,5'

3. Создайте переменную *cat* со значением ‘Кот Василий’

- Добавьте к ней ‘ лежит на печи’, чтобы значением *cat* стала строка ‘Кот Василий лежит на печи’

4. Сравните между собой оператором == null, undefined, 0, 1, true, false, ''

перерыв

# Условия If / else

```
if (условие) {
    выражение1
}
else {
    выражение2
}
```

# Тернарный оператор и не(!)

**условие ?** выражение1 : выражение2

```
if (a === 1) { console.log(a) } else { null }

a === 1 ? console.log(a) : null
```

Примеры:

```
let isVisible = true

a = isVisible ? 1 : ''  //две одинарных кавычки = пустая строка

isVisible = !isVisible // Не что-то
```

# Switch

```
x = чему-то
switch(x) {
        case 'value1':
                // этот код выполнится, если x === 'value1'
                break
        case 'value2':
                // этот код выполнится, если x === 'value2'
                break
        default:
                break
}
```

**break** - это остановка движения дальше по конструкции

# Задание 2

Решите задачи, подключив script2.js

1. Напишите алгоритм проверки пароля, где в результате будет собрана приветственная надпись text:
   - если пароль не 123532, “Пароль не верный, вы не авторизованы”
   - если пароль совпал с 123532: “Добро *пожаловать*”
   - если это администратор: “*Добро пожаловать,* администратор”
2. Студент за семестр набрал 75(N*) баллов по математике, какая будет итоговая оценка, если:
   - 60 и более - удовлетворительно
   - более 74 - хорошо
   - более 90 - отлично

Результат выведите в консоль

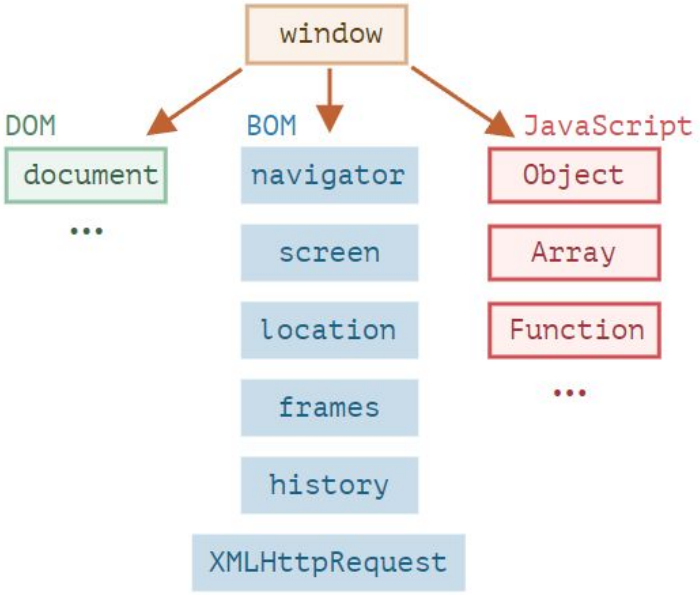
# Работа с DOM и BOM

**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'*

# Конец занятия