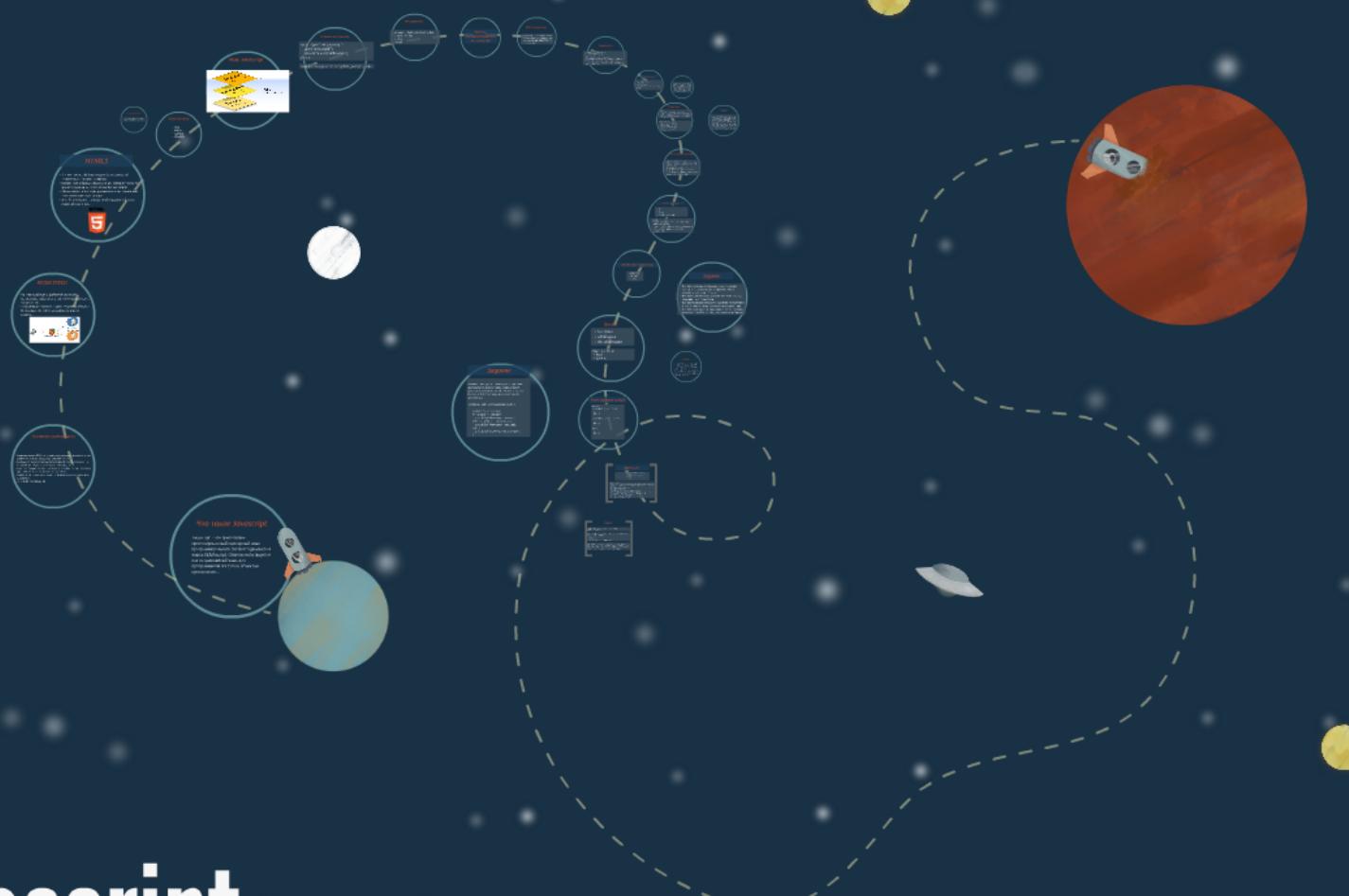


# Javascript

## введение



# Javascript

## введение

## *Что такое Javascript*

Javascript - это прототипно-ориентированный сценарный язык программирования. Являетсяialectом языка ECMAScript. Обычно используется как встраиваемый язык для программного доступа к объектам приложения....



## **Основные возможности**

- Создавать новые HTML-теги, удалять существующие, менять стили элементов, прятать, показывать элементы и т.п.
- Реагировать на действия посетителя, обрабатывать клики мыши, перемещение курсора, нажатие на клавиатуру и т.п.
- Посыпать запросы на сервер и загружать данные без перезагрузки страницы(эта технология называется "AJAX").
- Получать и устанавливать cookie, запрашивать данные, выводить сообщения...
- ...и многое, многое другое!

# НЕДОСТАТКИ

- Не умеет работать с файловой системой компьютера, работать с внешними программами, запускать их...
- Запущенный в одной вкладке, не умеет работать с остальными если они запущены на другом домене...



# **HTML5**

- Чтение/запись файлов на диск (в специальной «песочнице», то есть не любые).
- Встроенная в браузер база данных, которая позволяет хранить данные на компьютере пользователя.
- Многозадачность с одновременным использованием нескольких ядер процессора.
- 2d и 3d-рисование с аппаратной поддержкой, как в современных играх.



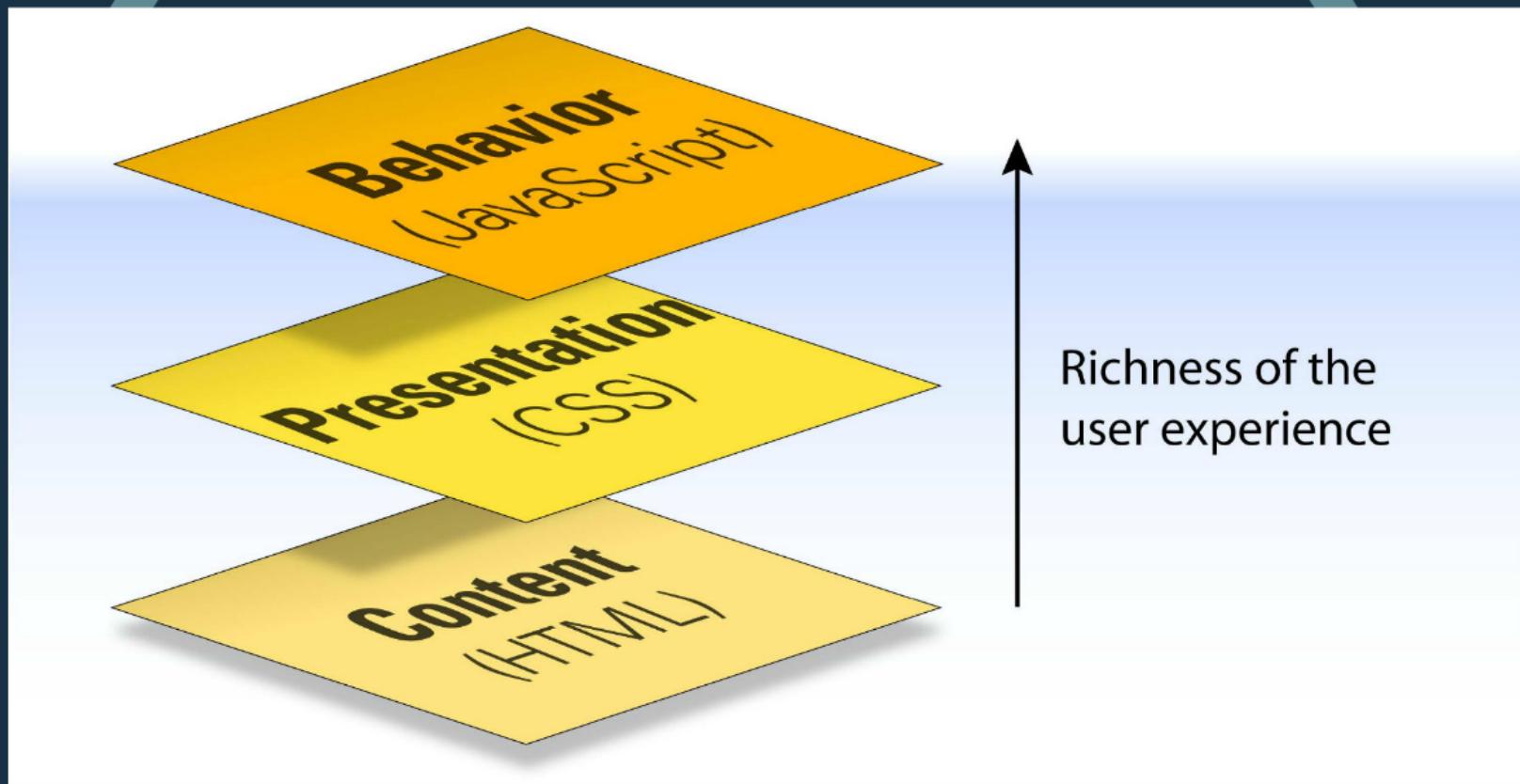
## *Среда исполнения*

- Браузер (Chrome / Firefox/ Opera)
- Сервер (node.js)

## **АЛЬТЕРНАТИВЫ**

- JAVA
- FLASH
- ACTIVEX
- Silverlight

# Роль Javascript



## *Вставка на страницу*

```
<script type="text/javascript">  
    alert("Hello world!");  
    document.write("Hello world");  
</script>
```

```
<script type="text/javascript" src=".js/hello_world.js"></script>
```

## *Инструменты*

- phpStorm / WebStorm /Sublime Text
- Chrome / Firefox
- Firebug
- GitHub

# *Основы программирования на Javascript*

# *Организация кода*

- Команды - строка кода которая создает переменную или изменяет ее, вызывает функцию и т.д...
- Комментарии

# *Переменные*

- Объявляются через директиву var
- Регистрозависимы
- Имеют тип данных
- Могут меняться в течении выполнения кода
- Имя переменной может состоять из: букв, цифр, символов \$ и \_
- Имя переменной не должно начинаться с цифры
- Нельзя использовать для имени зарезервированные слова var, class, return, implements и т.д

## Типы данных

- Числа (numbers)
- Строки (strings)
- Логические ( boolean)
- null ( означает "ничего")
- undefined ( означает что переменной не присвоено значение)
- Object

Создать переменные типа "string". В одной будет содержаться ваше имя, а в другой ваша фамилия. Также создать одну переменную типа "number" в которой будет содержаться ваш возраст и вывести значение этих переменных в консоль.

# *Операторы*

## Термины

- Операнд - цель применения оператора
- Унарный оператор - применяется к одному операнду
- Бинарный оператор - применяется к нескольким операндам

## Типы операторов

- Арифметические операторы
- Присваивание
- Инкремент/декремент ++ --
- Побитовые операторы (&
- Оператор запятая

## Задание

- Создать две переменные типа number с произвольным значением. Выполнить над ними несколько арифметических операций. Результаты всех операций выводить в консоль.
- Результат последней операции присвоить новой переменной.
- Ее значение увеличить на 2 с помощью функции инкремента и вывести результат в консоль в виде сообщения "Последнее значение переменной ..." (используя объединение строк)

# Операторы сравнения

- Больше/меньше:  $a > b$ ,  $a < b$ .
- Больше/меньше или равно:  $a \geq b$ ,  $a \leq b$ .
- Равно  $a == b$ .
- Оператор неравенства  $a != b$
- Оператор строгого сравнения  $a === b$
- При сравнении - результат возвращается в виде значение true или false

## Условные операторы

- If
- else
- If else if
- Тернарные операторы

Принципы:

- Вычисляет значение в скобках и приводит его к логическому значению
- 0, пустая строка "", null и undefined и NaN являются false в контексте логических значений, все остальное true.

## *Логические операторы*

- || (ИЛИ)
- && (И)
- ! (НЕ)

## Задание

- Напишите условие для проверки того что цифра находится в диапазоне двух цифры 10 и 100 и вывидите сообщение об этом.
- Напишите условие которые вычисляет какое из двух цифр больше и выводит его
- Напишите условие которое по значению переменной в которой записан номер месяца выводит время года
- Напишите условие в котором проверяется если цифра делится на 2 и 3 без остатка, то выводится сообщение.

# Циклы

- for Цикл
- while цикл
- do..while цикл

Директивы циклов

- break
- continue

# *Задание*

- Напишите цикл while и for для вывода всех цифр которые делятся на 2 без остатка в промежутке. Промежуток задается двумя переменными.
- Модифицировать цикл чтобы по достижению цифры 30 он прерывался
- Модифицировать цикл чтобы он выводил на экран результат умножения текущей цифры на 10 только в том случае если она равна 10 или 20 используя директиву continue.

# Конструкция *switch*

```
switch(x) {  
    case 'value1': // if (x === 'value1')  
        ...  
        [break]  
  
    case 'value2': // if (x === 'value2')  
        ...  
        [break]  
  
    default:  
        ...  
        [break]  
}
```

# Задание

Напишите конструкцию switch для определения континента по имени страны с использования проваливания и блока default. Учтите что одна из стран может быть одновременно и страной и континентом.

Переписать код с использованием switch:

```
var myName = 'Дмитрий';
if ( myName == 'Алексей' ) {
    console.log( 'Меня зовут Алексей' );
} else if ( myName == 'Александр' ) {
    console.log( 'Меня зовут Александр' );
} else {
    console.log( 'Значит все таки Дмитрий' );
}
```

# ФУНКЦИИ

```
function myFirstFunction(param1, param2) {  
    var var1 = 10;  
    console.log('Моя первая функция');  
    return true;  
}
```

Основное про функции:

- Объявление функции начинается со слова **function**, далее следует **имя функции**, круглые скобки в которых перечислены параметры через запятую и пара фигурных скобок, внутри которых размещено код тела функции.
- Функция имеет свою область видимости
- Внутри функции можно объявлять переменные
- Функция имеет свой контекст исполнения
- Параметры переданные в функцию становятся переменными внутри нее
- Функция может возвращать значения через конструкцию **return**. Функция которая не возвращает ничего - возвращает **undefined**. Функции можно передавать как параметры в другие функции.
- Функция может вызывать сама себя по имени (рекурсия).
- Инструкция **return** прерывает выполнение функции.
- Функция имеет доступ к внешним переменным

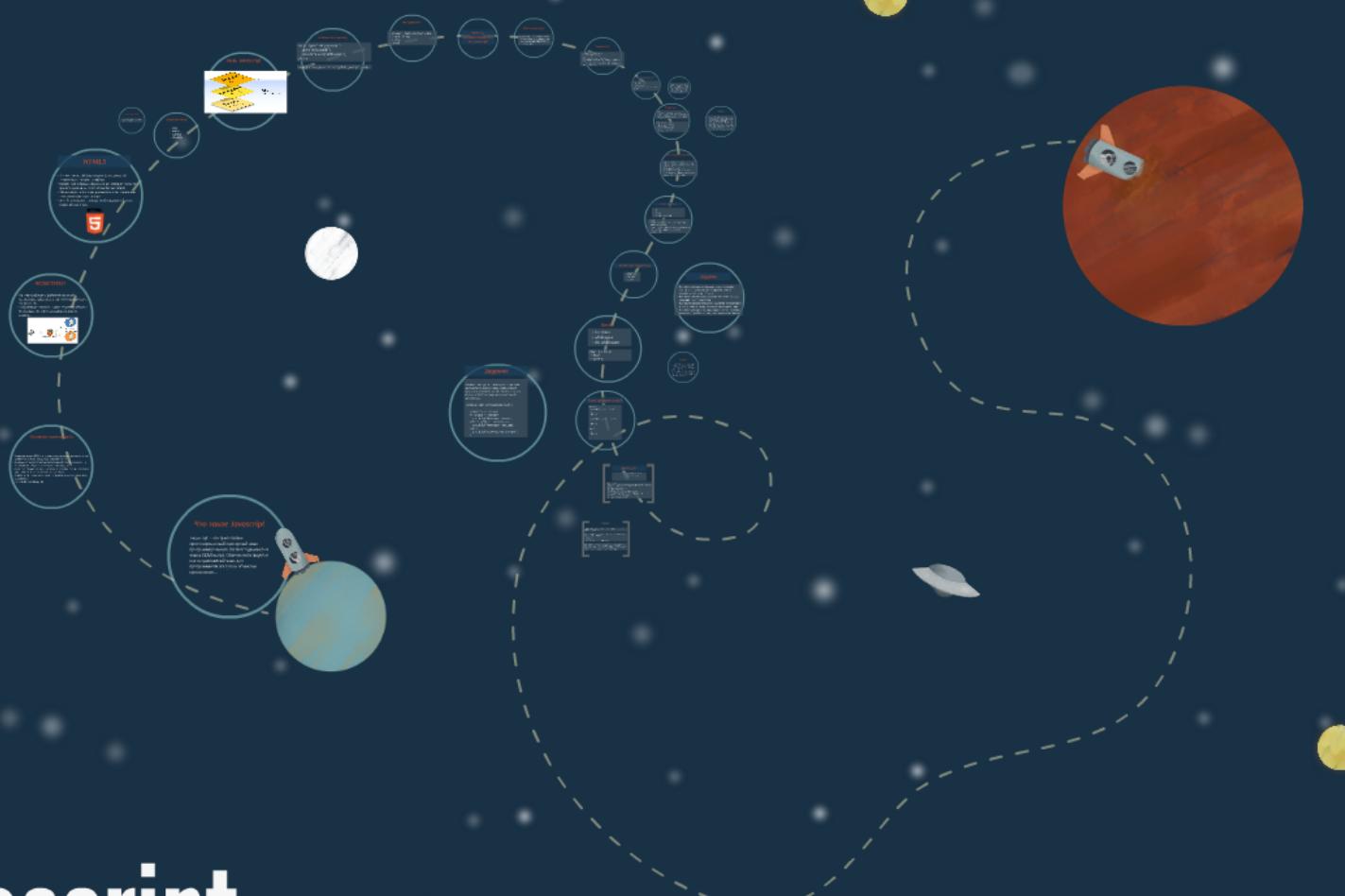
# Задание

1. Написать функцию которая возвращает максимальное значение из 2 параметров переданных ей

2. Напишите функцию калькулятор для сложения положительных чисел которая принимает 3 параметра

- первое число
- второе число
- имя операции (add, minus, multi, divide)

Функция должна проверять что каждое из 2 чисел больше 0 и числа не дробные (остаток от деления равен нулю) , а также имя операции одно из допустимых (switch) . В обратном случае должна возвращать 0 и выводить в консоль ошибку по которой можно понять что произошло.



# Javascript

## введение