STEP ACADEMY

ЗАНЯТИЕ №1

JavaScript Введение в программирование

Что такое язык программирования

“Это формальный язык, предназначенный для записи компьютерных программ. Язык программирования определяет набор лексических, синтаксических и семантических правил, определяющих внешний вид программы и действия, которые выполнит исполнитель под её управлением.”

Появление Javascript

Цель создания языка: добавить страницам динамики, обеспечить взаимодействие с пользователем.

Так как разработкой веб-сайтов в то время нередко занимались не программисты, то язык должен был быть легким.

Язык Javascript появился в 1995 году в браузере NetScape Navigator.

Рабочее название Livescript

ECMAScript

ECMAScript - это название стандартного языка, разработанного ECMA, из первоначальной реализации Javascript.

Javascript, JScript, TypeScript - реализации данного стандарта

ECMAscript выходит ежегодно, первые пакеты обновления назывались ES1, ES2, ES3, ES4, ES5, новые выпуски (начиная с 2015 года) получили название ES2015, ES2016, ES2017 (аббревиатура ES + год выпуска; ES6 == ES2015);

Как работает Javascript?

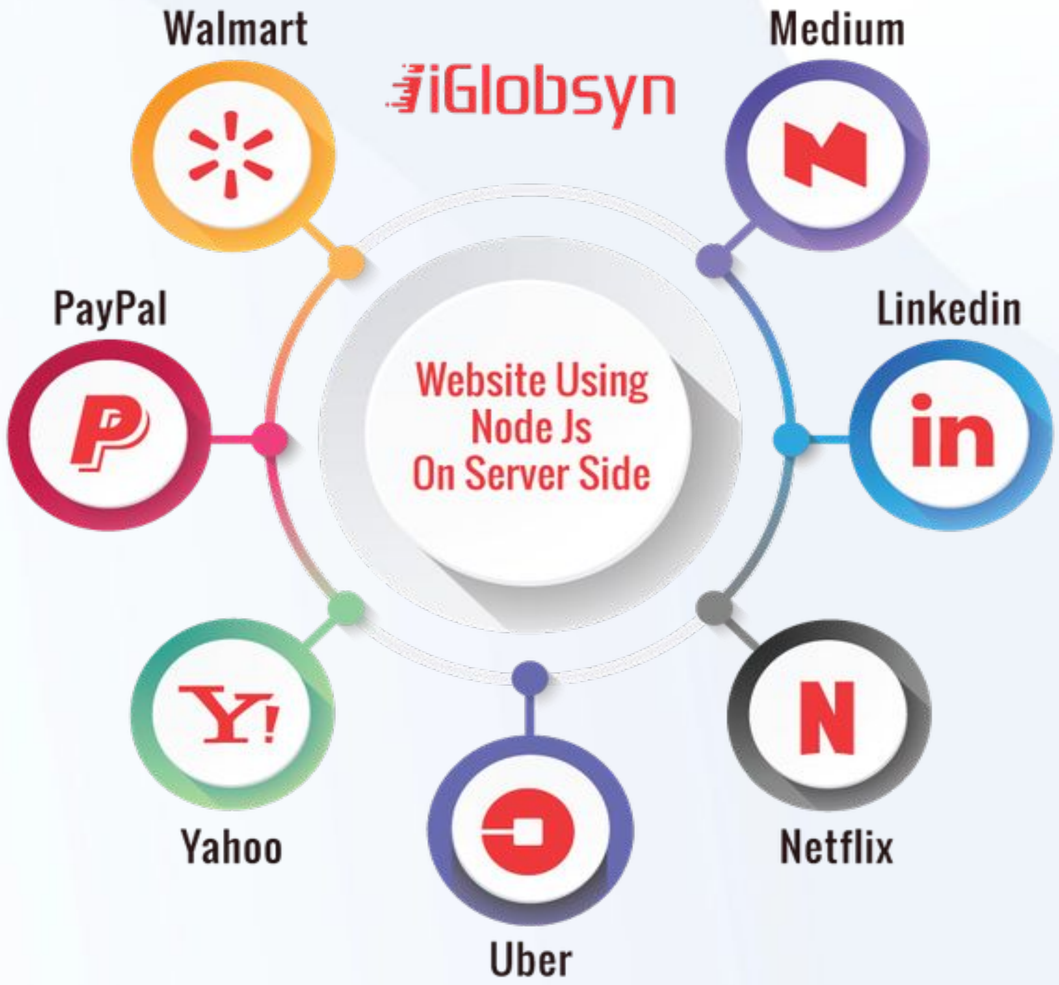
Компиляция – это когда исходный код программы, при помощи специального инструмента, другой программы, которая называется «компилятор», преобразуется в другой язык, как правило – в машинный код. Этот машинный код затем распространяется и запускается. При этом исходный код программы остаётся у разработчика.

Интерпретация – это когда исходный код программы получает другой инструмент, который называют «интерпретатор», и выполняет его «как есть».

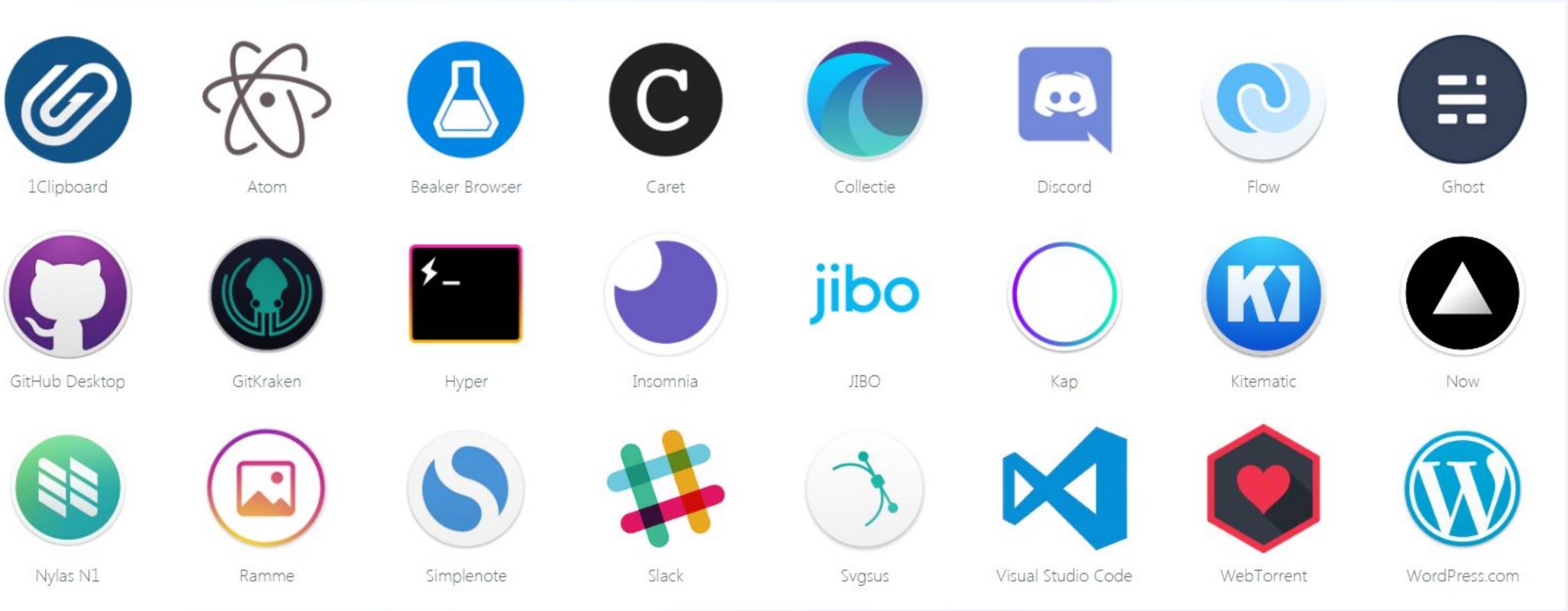
Почему Javascript нужен?

1. Единственный язык, который работает в браузере.
2. Входит в 10 самых популярных языков по индексу TIOBE.
3. Можно разрабатывать и серверные приложения, и десктопные приложения.
4. Frontend фреймворки, которые захватывают интернет.

node.js


Walmart
Medium
iGlobsyn
PayPal
Linkedin
Website Using
Node Js
On Server Side
Yahoo
Uber
Netflix

Electron: десктопные приложения на Javascript



Подключение JS к HTML

Для подключения используется тег script. Тег работает 2-мя способами:

1. Код пишется напрямую в теге script

2. Код пишется в файле, путь к которому указывается с помощью атрибута src

Основы программирования

Алгоритм

Алгоритм - совокупность последовательных шагов, схема действий, приводящих к желаемому результату.

Алгоритм - основа программирования, потому что по сути на языке программирования описываются алгоритмы.

Правила алгоритмов

1. Алгоритм состоит из конечного числа шагов.
2. Алгоритм приводит к результату.
3. Алгоритм решает набор однотипных задач.
4. Алгоритм должен быть однозначно понятным для исполнителя.

Абстракция в программировании

В программировании мы почти никогда не работаем с конкретными данными, это почти всегда данные, полученные от пользователя или других источников, которые мы должны обработать каким-то определенным образом, чтобы выдать ответ.

ОЧЕНЬ ВАЖНО понимать, что мы точно не знаем, что у нас есть, и нам нужно это обработать.

Операторы и операции

Оператор — это элемент языка, задающий полное описание действия, которое необходимо выполнить. Каждый оператор представляет собой законченную фразу языка программирования и определяет некоторый вполне законченный этап обработки данных.

Операция — конструкция в языках программирования, аналогичная по записи математическим операциям, то есть специальный способ записи некоторых действий.

Комментарии

Комментарии используется для добавления текста, который будет игнорироваться при выполнении программы.

Обычно комментарии используются для пометок в коде или для отключения части кода.

Комментарии в JS

// однострочный комментарий

/* многострочный комментарий */

Переменные и типы данных. Операции

Переменные

Переменные - это область оперативной памяти, в которой сохраняются значения в процессе работы программы.

Переменным нужно задавать имена. Имена переменных подчиняются определенным правилам:

Нельзя задавать имя переменным занятое другими командами

Можно использовать цифры, но нельзя начинать название переменных с цифры

Нельзя использовать спецсимволы, кроме _ и $

camelCase или верблюжий стиль

Для названия переменных существуют и рекомендации.

Для названия переменных использовать слова, описывающие суть переменных

Если в названии переменных присутствуют несколько слов, то использовать верблюжий стиль: слова не отделяются пробелами, каждое новое слово начинается с большой буквы.

Пример: myVariableName

Создание переменных в JS

Создание переменных начинается с ключевого слова let.

let myVar;

Также есть вариант с var, но это устаревший способ, о нём поговорим позднее.

Оператор присваивания (=)

Оператор присваивания служит для записи значения в переменные.

let a = 140;

В левой части от знака присваивания находится переменная, в которую запишется значения справа от знака присваивания.

Присваивание значение переменным доступно в любой момент в работе программы (скрипта).

Инициализация переменной - получение переменной её первого значения.

console.log()

Команда для вывода значения в консоль. Значение может задаваться вручную (текст в кавычках), но может и передаваться в виде переменных.

Консоль открывается в инструментах разработчика.

В консоли также записываются все ошибки.

Типы данных

При задании значения переменной в JS, для этой переменной определяется тип данных. Тип данных указывает на способ хранения переменной в оперативной памяти, а также на возможные операции с этими переменными.

Javascript является языком с динамической типизацией, то есть тип данных не указывается в момент создания переменной, а определяется по значению переменной. Также в Javascript переменная может поменять свой тип в процессе работы скрипта.

Типы данных JS

number - числовой тип данных

string - строковый тип данных, для хранения текста

boolean - булевский или логический тип данных, имеет два возможных варианта: true или false

object - объект, сложная структура

Также предусмотрено два специальных значения:

undefined - значение неопределено

null - значение отсутствует

Операции над числовыми типами данных

Сложение +

Вычитание -

Умножение *

Деление /

Нахождение остатка от деления -

Унарные операторы

Унарные операторы выполняют действия над одним операндом (значением)

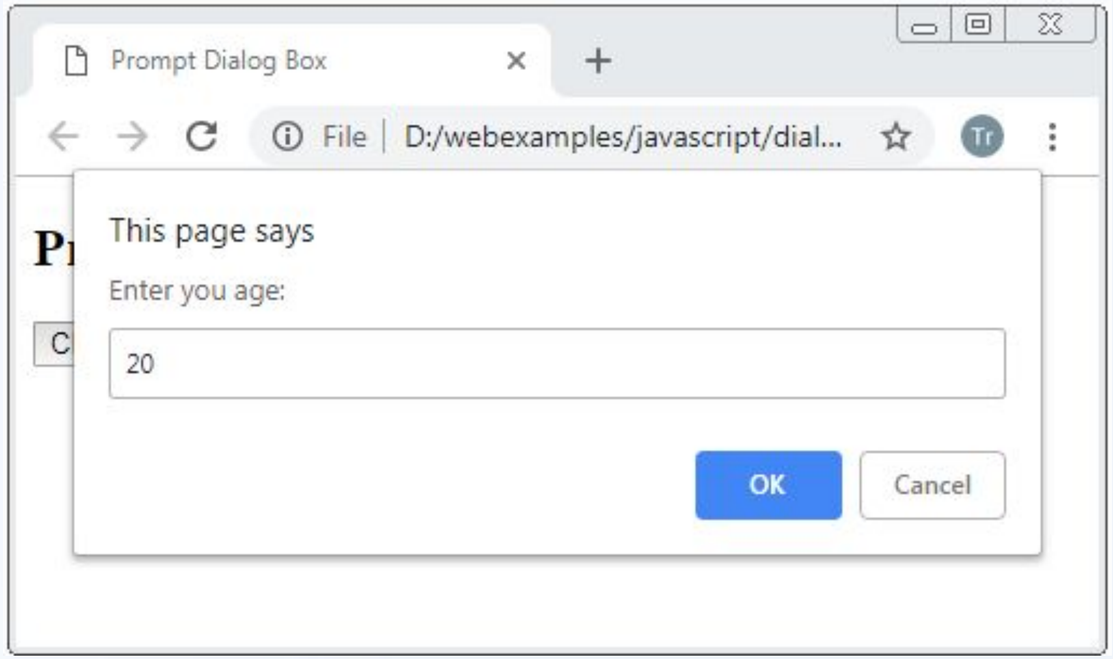
Инкремент - увеличение значения на 1. Префиксный: ++a, постфиксный a++

Декремент - уменьшение значения на 1. Префиксный: --a, постфиксный a--

В постфиксной форме сначала выполняться все остальные действия над переменной, а потом произойдёт увеличение.

В префиксной – наоборот.

Ввод значения в переменную

Для начала будем использовать функцию prompt

let myVar = prompt(“Текст в окошке”);