

JAVASCRIPT

АВТОР РАБОТЫ ТИХОМИРОВ РОМАН "8В"



JavaScript

ЧТО ТАКОЕ JAVASCRIPT?

JavaScript – один из самых популярных языков программирования в мире с более чем двадцатилетней историей. Это также один из трёх главных языков для веб-разработчиков:

- 1) **HTML**: позволяет добавить контент на вашу веб-страницу.
- 2) **CSS**: задаёт стили и внешний вид веб-страницы.
- 3) **JavaScript**: улучшает поведение веб-страницы

HTML – язык
разметки веб-
страниц, а CSS -
язык таблиц
стилей



ИСТОРИЯ JAVASCRIPT

- ✓ JavaScript был создан за 10 дней автором Бренданом Айком (Brandan Eich), работавшим на Netscape в далёком 1995 году. Изначально он был назван Mocha, имя языка было сменено на Mona и потом на LiveScript пока наконец он не стал JavaScript. Первоначальная версия языка была ограничена только браузером Netscape и предлагала узкую функциональность, но со временем он продолжил развиваться частично благодаря сообществу разработчиков, которые не оставляли работу над ним.
- ✓ В 1996 году язык JavaScript был стандартизирован и было дано официальное имя ECMAScript, с ECMAScript 2 вышедшим в 1998 и ECMAScript 3 в следующем 1999 году. Это преобразовалось в сегодняшний JavaScript, которые теперь работает не только в разных браузерах, но также на различных устройствах, включая мобильные и настольные компьютеры.

[Подробнее о истории JavaScript](#)

ПРЕИМУЩЕСТВА JAVASCRIPT

1. Вам не нужен компилятор, потому что веб-браузер интерпретирует его в HTML;
2. Его проще изучать, чем другие языки программирования;
3. Ошибки проще выявить, а значит и исправить;
4. Он может привязываться к специальным элементам страниц или событиям вроде нажатия(click) или наведения мыши(mouseover);
5. JS работает в разных браузерах и на разных платформах;
6. Вы можете использовать JavaScript для валидации входных данных и снижения необходимости ручной проверки данных;
7. Он делает сайт более интерактивным и привлекательным для посетителей;
8. Он быстрее и легче, чем другие языки программирования.

СЛАБЫЕ МЕСТА JAVASCRIPT

Когда язык программирования становится таким популярным как JavaScript, он становится объектом повышенного интереса для хакеров, мошенников и других вредоносных проявлений третьих сторон, которые пытаются найти уязвимости и слабые места в безопасности. Некоторые слабые места:

- 1) Уязвим отношении к эксплойтам (вредоносный код, использующий уязвимости программного продукта);
- 2) Может быть использован для запуска вредоносного кода на компьютере пользователя;
- 3) Не всегда поддерживается некоторыми браузерами или устройствами;
- 4) Фрагменты JS кода могут быть очень большими;
- 5) Может по разному отображаться на разных устройствах, что приводит к отсутствию целостности.

JAVASCRIPT КАК ЯЗЫК ПРОГРАММИРОВАНИЯ

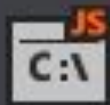
ПРАКТИЧЕСКАЯ ЧАСТЬ РАБОТЫ

Для запуска кода будет
использоваться среда
Node.js совместно с Visual
Studio



НАСТРОЙКА СРЕДЫ РАЗРАБОТКИ

- 1) У вас должна быть установлена Visual Studio старше 2015 и пакет разработки Node.js
- 2) Создайте проект “Пустое консольное приложение Node.js”
- 3) Запустите скрипт с помощью Ctrl + F5



Пустое консольное приложение Node.js в Azure

Пустое приложение Node.js.

JavaScript

Windows

Linux

macOS

Консоль

“HELLO WORLD”

Для начала включим строгий режим (нужен для использования современной версии языка)

```
"use strict";
```

Объявим переменную `message`

```
let message;
```

Видим первое отличие JS от других языков – переменные не имеют типа

А теперь присвоим ей значение “Hello world”

```
message = “Hello world”;
```

Осталось только вывести значение переменной на экран

```
console.log(message);
```

Запуск

```
Hello world
```


АРИФМЕТИЧЕСКИЕ ОПЕРАТОРЫ

Теперь напишем что-нибудь посложнее: выведем произведение чисел 15 и 10

1. `"use strict";` // строгий режим
2. `let message;` // объявляем переменную message
3. `message = 15 * 10;` // присваиваем ей значение 10 * 15
4. `console.log(message);` // выводим полученное число

Двойной слеш (//) означает начало комментария. После него все символы на данной строке игнорируются.

Запуск

УСЛОВНЫЕ ОПЕРАТОРЫ

Напишем программу для определения чётное число или нет.

1. `"use strict";` // строгий режим
2. `let numb;` // объявляем переменную numb
3. `numb = 8;` // будем проверять число 8 на четность
4. `If (numb % 2 == 0) console.log("Число четное");`
5. // оператором % получим остаток от деления нашей переменной на 2 и // сравним его с 0. Если остаток равен нулю, выведем "Число четное"
6. `else console.log("Число нечетное");`
7. // в противном случае выведем "Число нечетное"

Запуск

Число четное