

# Основи на програмирането с JavaScript

Димитър Митев

# Какво ще учим

- Първи стъпки в JS. Типове данни. Операции и изрази
- Логически блокове и цикли
- Функции
- Обектно ориентирано програмиране с JS
- DOM и DOM манипулации
- CSS модификации
- Събития (Events)
- Заявки

# Същност и история на JavaScript

- Създаден от Netscape през 1995г.
- Официално стандартизиран през 1996г. [ECMA](#)
- High-level, динамичен, **нетипизиран**, интерпретеран програмен език
- End-to-end език

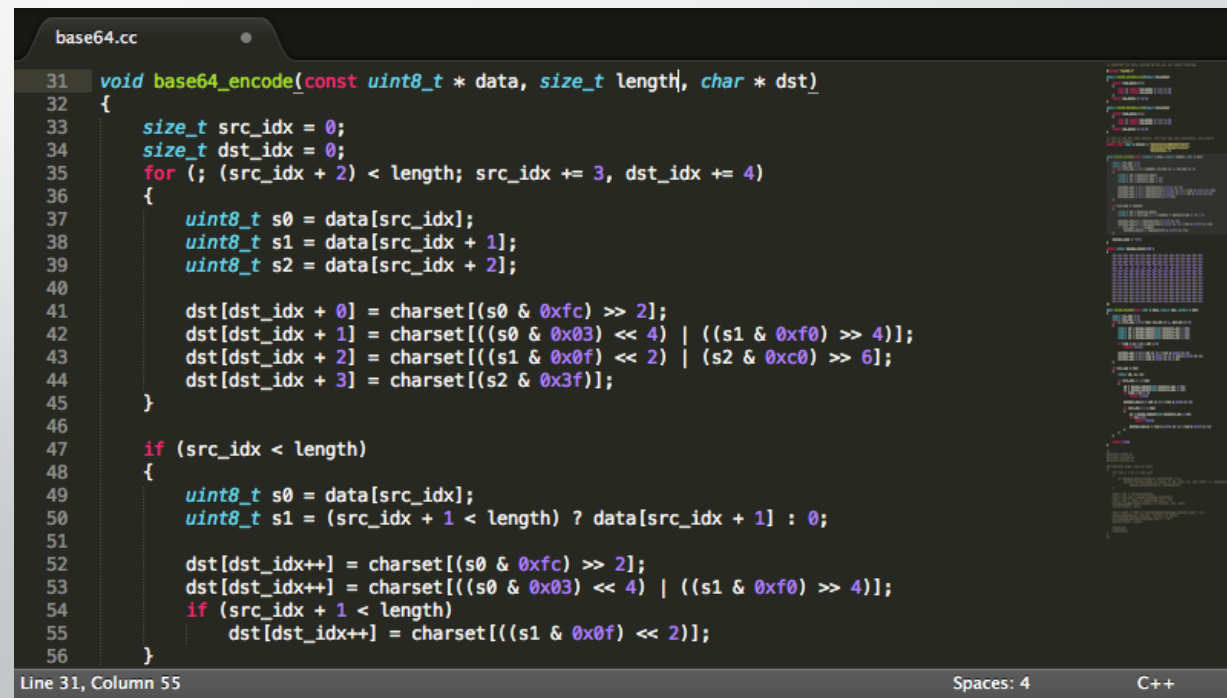
# Приложение на JavaScript

- Уеб приложения
  - Front-end -> динамичен UI/UX
  - Back-end -> nodejs -> сървърна логика с голяма продуктивност
- Мобилни приложения
  - Cross-platform apps с HTML5, CSS3 & JS
- IoT (Internet of the things)
- Мини-компютри (Arduino, Raspberry Pi...) и [роботи](#)

# Инструменти за разработка на JS приложения

JS е интерпретиран език, който най-често бива използван от браузъра, т.е. най-обикновен текстов редактор ще бъде достатъчен за да пишем код. Но все пак можем да си улесним живота като използваме:

- IDE
  - [Visual Studio Community](#)
  - [WebStorm](#)
- Текстови редактори
  - [Notepad++](#)
  - [Visual Studio Code](#)
  - [Adobe Brackets](#)
  - [Sublime Text](#)
  - [Atom](#) -> все още е beta версия за Windows потребители



```
base64.cc
31 void base64_encode(const uint8_t * data, size_t length, char * dst)
32 {
33     size_t src_idx = 0;
34     size_t dst_idx = 0;
35     for (; (src_idx + 2) < length; src_idx += 3, dst_idx += 4)
36     {
37         uint8_t s0 = data[src_idx];
38         uint8_t s1 = data[src_idx + 1];
39         uint8_t s2 = data[src_idx + 2];
40
41         dst[dst_idx + 0] = charset[(s0 & 0xfc) >> 2];
42         dst[dst_idx + 1] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];
43         dst[dst_idx + 2] = charset[((s1 & 0x0f) << 2) | (s2 & 0xc0) >> 6];
44         dst[dst_idx + 3] = charset[(s2 & 0x3f)];
45     }
46
47     if (src_idx < length)
48     {
49         uint8_t s0 = data[src_idx];
50         uint8_t s1 = (src_idx + 1 < length) ? data[src_idx + 1] : 0;
51
52         dst[dst_idx++] = charset[(s0 & 0xfc) >> 2];
53         dst[dst_idx++] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];
54         if (src_idx + 1 < length)
55             dst[dst_idx++] = charset[((s1 & 0x0f) << 2)];
56     }
57 }
```

Line 31, Column 55      Spaces: 4      C++



# Visual Studio Code

- Безплатен текстов редактор.
- Работи на всякакви платформи – Windows, Linux & Mac OSX
- Бърз и лек
- Предоставя много възможности за преконфигуриране (hackable)



# Sublime Text

- Безкраен пробен период
- Лек и бърз
- Тонове приставки (plugins)
- Възможност за преконфигурация (hackable)
- Полезни приставки (plugins)
  - BracketHighlighter – маркира отворени и затворени скоби
  - JavaScript Beautifier – форматира кода
  - SublimeCodeIntel – добавя intellisense
- \*Трябва да имате инсталиран [Package Control](#)

# Какво още ни трябва?

- Последна версия на браузър
  - **Google Chrome**
  - Mozilla Firefox
  - Opera
  - Safari
- Приставки (addons)
  - [Web Developer](#) -> управление на HTML, CSS & JS съдържание на страницата
  - [Postman](#) -> създаване и изпращане на заявки
  - [FireBug](#) -> добавя доста възможности за управление на JS и съдържанието към Firefox



# Дебъгване и намиране на грешки

- Интернет браузър
  - Съвременните браузъри предоставят богат набор от инструменти за дебъгване
  - F12 инструменти
  - \*[FireBug](#) -> ако използвате Mozilla Firefox
- IDE
  - WebStorm предоставя доста добри възможности за дебъгване на JS код
  - Visual Studio също дава добри възможности за дебъгване на JS код



# Други JS инструменти

- Качествен код -> проверява кода за несъответствия с JS конвенциите
  - [JS Hint](#)
  - [JS Lint](#)
- Online редактори и едитори (playgrounds)
  - [JS Bin](#)
  - [JS Fiddle](#)
  - [CodePen](#)
- Измерване на продуктивността на нашия JS
  - [JS Perf](#)
  - \*Измераната продуктивност зависи от браузъра, интернета и машината, на която се изпълнява

**НЯКОЙ ИМА ЛИ**



**ВЪПРОСИ?**