

# Урок 3.1. Введение в JavaScript

- Введение. Понятие алгоритма, свойства алгоритма, методы записи алгоритма
- Подключение скрипта к странице. Структура кода, строгий режим, переменные и константы

## Алгоритм

- Алгоритм – последовательность чётко определенных действий, выполнение которых ведёт к решению задачи. Алгоритм, записанный на языке машины, есть программа решения задачи.
- Алгоритм – это совокупность действий, приводящих к достижению результата за конечное число шагов.
- Свойства алгоритмов:
  - Дискретность – это разбиение алгоритма на ряд отдельных законченных действий (шагов).
  - Детерминированность - любое действие алгоритма должно быть строго и недвусмысленно определено в каждом случае. Например, алгоритм проезда к другу, если к остановке подходят автобусы разных маршрутов, то в алгоритме должен быть указан конкретный номер маршрута 5. Кроме того, необходимо указать точное количество остановок, которое надо проехать, скажем, три.
  - Конечность – каждое действие в отдельности и алгоритм в целом должны иметь возможность завершения.
  - Массовость – один и тот же алгоритм можно использовать с разными исходными данными.
  - Результативность – алгоритм должен приводить к достоверному решению.
- Способы записи алгоритма:
  - словесная (запись на естественном языке);

- псевдокоды (полуформализованные описания алгоритмов на условном алгоритмическом языке, включающие в себя как элементы языка программирования, так и фразы естественного языка, общепринятые математические обозначения и др.);
- графическая (изображения из графических символов – блок-схема);
- программная (тексты на языках программирования – код программы).

# JavaScript

## Подключение скрипта к странице

- Скрипт можно записать в html двумя способами: либо написать код непосредственно в тег `<script></script>`, либо использовать `<script src="путь-до-скрипта.js"></script>`
- При наличии атрибута `src` у тега `script` содержимое тега игнорируется.
- Написать тег можно в любом месте внутри тега `head` или внутри тега `body`
- Положение скрипта влияет на загрузку страницы. Если разместим в `head`, то скрипт загрузится до того, как будет отрендерена страница (asynс скрипты пока не берём в расчёт). Если разместим тег в конце `body` после всех элементов, то сначала они отрендерятся (и будут доступны для использования в JS), потом загрузится и выполнится скрипт.
- Пример подключения скрипта:
  - Структура файлов

```
project-folder
├── index.html
└── script.js
```

- index.html

```
<!DOCTYPE html>
<html>
<head>
```

```
<meta charset="utf-8">
<title>Заголовок</title>
</head>
<body>
  <script src="script.js"></script>
</body>
</html>
```

## Строгий режим

- С появлением в 2009 стандарта ES5, некоторые существующие возможности JS были изменены (раньше до ES5 добавлялись только новые возможности, старые оставались без изменений и не было проблем с обратной совместимостью)
- По умолчанию эти изменения не применяются, чтобы не иметь проблем с обратной совместимостью.
- Чтобы включить обновлённый режим, необходимо использовать директиву `use strict`.
- Директива - это строка `'use strict';`, располагающаяся на первой строке скрипта:

```
'use strict'; // этот код работает в обновлённом, современном режиме...
```

## Переменные и константы

- Переменные и константы - это именованные места для хранения данных, чтобы потом их переиспользовать
- Основное отличие переменной от константы: в процессе выполнения скрипта заданная константа уже не меняется, а переменная может менять как своё значение, так и свой тип.
- Объявление переменной

```
// 1. Разделяем объявление переменной и присваивание ей значения
let someVariable;
// .. здесь может быть ещё какой-то код
someVariable = 35;
```

```
// 2. Совмещаем объявление и присваивание (инициализируем сразу)  
let someOtherVariable = 'Hello, World!';
```

- **Объявление константы**

```
const someConst = 'Значение, которое всегда будет иметь эта константа';
```