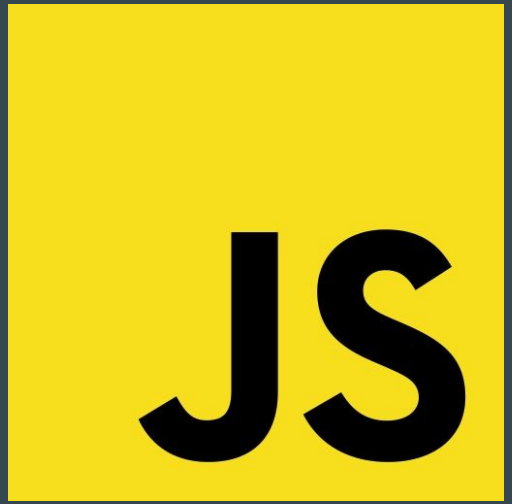


Введение в JavaScript. Переменные и типы

Лекция №1

Немного о себе

- Работаю в компании LAD
- Frontend разработчик
- Участвую в разработке крупных проектов на JavaScript (TypeScript)
- Опыт в коммерческой разработке ~ 1.5 лет
- JavaScript TypeScript React NodeJS
- Проводил летнюю стажировку для студентов в компании LAD

Ход занятия

- Введение в JavaScript
- Переменные
- Типы данных

Что такое JavaScript?

JavaScript язык программирования с динамической типизацией, который позволяет Вам создать динамически обновляемый контент, управляет мультимедиа, анимирует изображения, впрочем, делает всё, что угодно.

Ну или все, что угодно, удивительно, чего можно достичь с помощью нескольких строк JavaScript кода. Данная презентация так же была создана с помощью ПО, написанном на JavaScript.

Преимущества JavaScript?

- Язык браузеров. Включён по умолчанию
- Широкая распространенность
- Полная интеграция с HTML/CSS
- Язык высокого уровня
- Быстрый для пользователя
- Широкая область использования
- Наличие “языков-надстроек” над JavaScript
- Невысокий порог вхождения
- Активное развитие языка
- Большое комьюнити

JavaScript и область применения

HTML и CSS

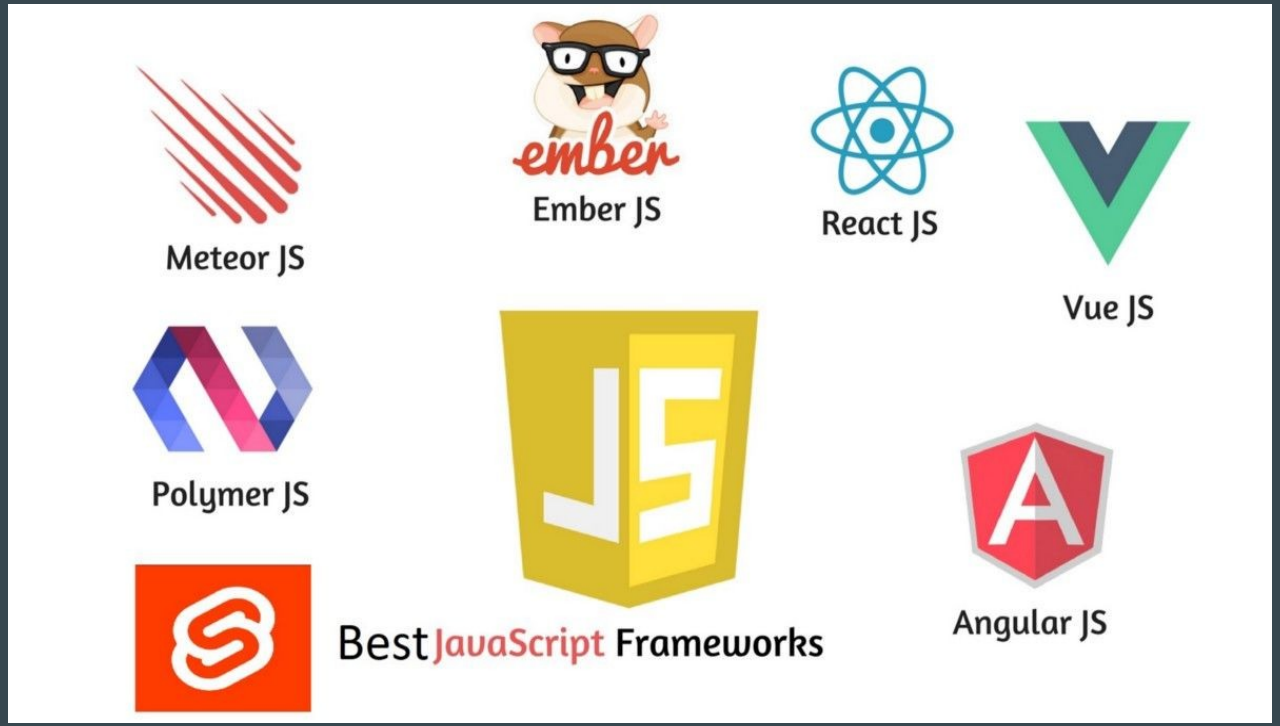
HTML - это язык разметки, который мы используем для визуального и смыслового структурирования нашего web контента, например, определяем параграфы, заголовки, таблицы данных, или вставляем изображения и видео на страницу.

CSS -это язык стилей с помощью которого мы придаем стиль отображения нашего HTML контента, например придаем цвет фону (background) и шрифту, придаем контенту

Использование JavaScript?

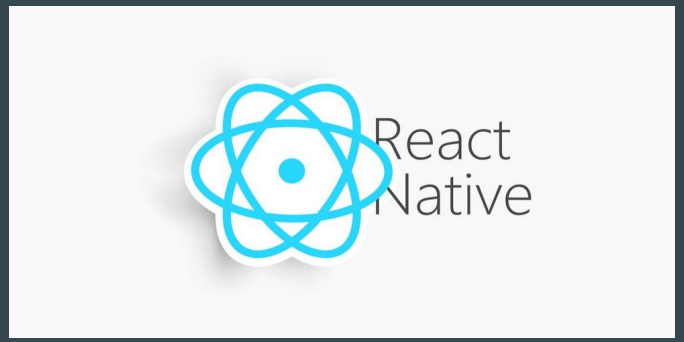
- Frontend => Реализация клиентских приложений. Реализация визуальной части.
- Backend => Реализация серверных приложений. Реализация бизнес-логики приложений.
- Mobile => Реализация приложений для мобильных устройств.
- Desktop => Реализация приложений для настольных ПК.

Инструменты Frontend



Инструменты Backend

Инструменты Mobile

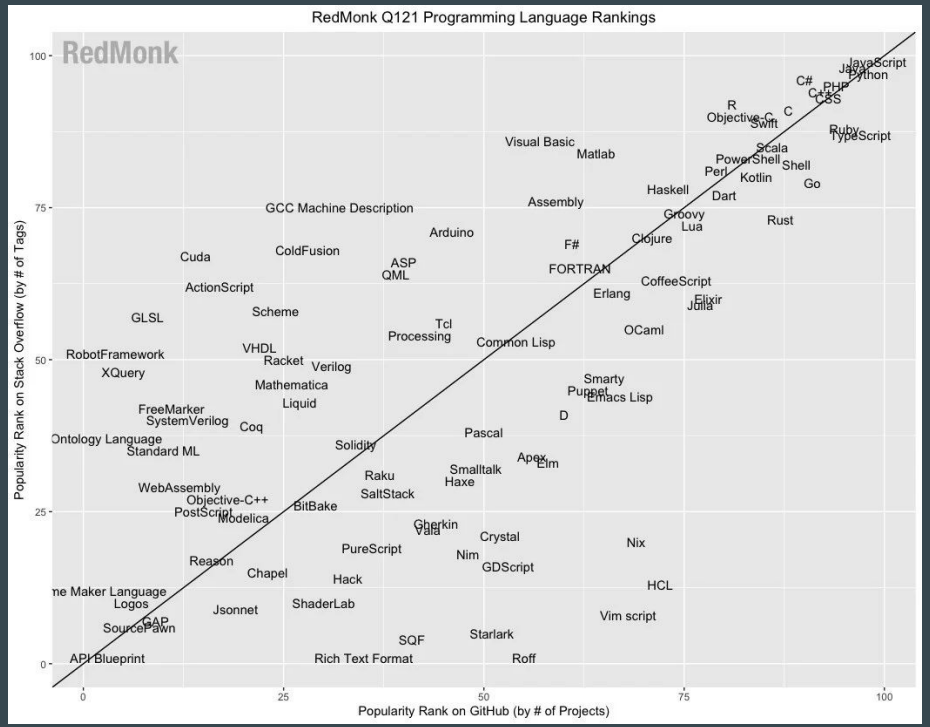


Инструменты Desktop

Недостатки JavaScript

- В JavaScript не поможет компилятор
- Широкая распространенность
- Непривычная объектная модель
- Однопоточность

Топ языков программирования



Топ языков программирования

- JavaScript
- Python
- Java
- PHP
- C#
- C++
- CSS
- TypeScript
- Ruby
- C
- Swift
- R
- Objective-C
- Shell
- Scala
- Go
- PowerShell
- Kotlin
- Rust
- Perl

Стандарты языка
Среда исполнения
…

Что есть JavaScript?

ECMAScript — спецификация скриптового языка программирования

ES5, ES6 и т д

JavaScript — язык программирования, одна из реализаций спецификации ECMAScript (наряду с JScript и ActionScript), их ещё называют диалектами ECMAScript

ECMA-262 — стандарт компании Ecma International, по которому разрабатывается спецификация ECMAScript

Последняя версия: 11-е издание в июне 2020 года

Последний черновик: ECMAScript® 2020 Language Specification

Как запустить JavaScript?

- **Браузер:** через консоль разработчика или онлайн редактор кода

- **NodeJS:** через консоль терминала или через редактор кода

Редакторы кода

Онлайн редакторы кода JavaScript

- JS Bin
- JSFiddle
- CodePen

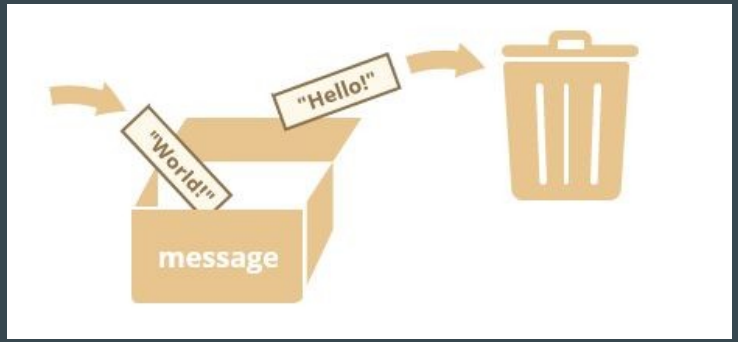
Переменные

...

Переменные

Переменные - это «именованное хранилище» для данных (в памяти).

При этом значение переменных может изменяться по ходу выполнения скрипта.



Имена переменных в JavaScript

- Имя переменной должно содержать только буквы, цифры или символы `$` и `_`
- Первый символ не должен быть цифрой

`userName`	`7a`
`test123`	`user-name`
`$`	`#%()+*`

Влияние регистра

Переменные с именами `apple` и `APPLE` – это две разные переменные.

Нелатинские буквы разрешены, но не рекомендуются

```
let имя = '...';

let我 = '...';
```

* Очень частая ошибка - русская буква 'с' в имени переменной (спасает редактор кода)

Зарезервированные имена

- break
- case
- class
- catch
- const
- continue
- debugger
- default
- delete
- do
- else
- export
- extends
- finally
- for
- function
- if
- import
- in
- instanceof
- let
- new
- return
- super
- switch
- this
- throw
- try
- typeof
- var
- void
- while
- with
- yield

Нотации именования

Нотации именования переменных (виды наименований):

- CamelCase
- snake_case
- kebab-case

Пример CamelCase: `let myVeryLongName;`

Средняя длина имени переменной от 1 до 4 слов.

Правильные имена переменных

Правильно: userName , paymentType

Неправильно: a, b, c, value, _, $

Условия именования:

Предикаты - переменная проверки, проверяет (да) или (нет): isAdmin

Вхождения - имеет что то (да) или (нет): hasChildren

Количество - какое то количество: peopleCount

Правильные имена переменных

Создания переменной в JavaScript

Для создания переменной в JavaScript используйте ключевые слова:

- let - это обычная переменная которую, можно изменить;

```
let userName = “Alex”;
```

- const - это константа, их нельзя изменить. Попытка сделать это приведёт к ошибке;

```
const age = 18;

const COLOR_BLACK = '#000';
```

- var

Что нужно знать про var

- Устаревший способ
- Из-за функциональной области видимости, способствует странному поведению и не очевидным багам если не знать как с ней работать (глобальные переменные).
- Не используется в новых проектах
- С большей долей вероятности вы ее встретите если пойдете работать в компанию где нужно поддерживать старый проект.
- С большей долей вероятности вы его встретите в коде чужих библиотек
- С большей долей вероятности вас спросят про него на собеседовании
- Более детально разберем в следующих лекциях

Хорошая практика

- Всегда использовать `const` если переменная не изменяется
- Не стоит повторно использовать переменные. лучше создать новые
- Избегайте использования `var`

Практика

Придумать наименования переменных для следующих примеров:

- Переменная для “названия нашей планеты”?
- Переменная для “текущее время пользователя”?
- Переменная которая показывает “количества статей”?
- Переменная которая показывает “это оплата наличными деньгами или нет”?
- Три переменные для хранения Ф.И.О

Типы данных

...

Типы данных

JavaScript имеет динамическую типизацию. Переменная в JavaScript может содержать любые данные. В один момент там может быть строка, а в другой – число.

Есть восемь основных типов данных в JavaScript:

- number
- string
- boolean
- null
- undefined
- object
- symbol
- bigint

Числа - number

- number - для любых чисел: целочисленных или чисел с плавающей точкой.

```
const age = 18;

const pi = 3.14;
```

- Infinity - математическая бесконечность (спец-е числовые значения)

```
alert( 1 / 0 );
```

- NaN -означает вычислительную ошибку (спец-е числовые значения)

```
alert( 'Alex' / 0 );
```

Строки - string

string – в JavaScript должна быть заключена в кавычки.

- Двойные кавычки:

```
let userName = "Alex";
```

- Одинарные кавычки:

```
let userName = 'Alex';
```

- Обратные кавычки:

```
let userName = 'Alex';

let helloUser = `Hello ${userName}`; // интерполяция выражений
```

Булевый - boolean

- boolean - может принимать только два значения: true (истина) и false (ложь).

```
let isAdmin = false;

let isOpen = true;
```

Значение null

- null –отдельный тип, специальное значение, которое представляет собой «ничего», «пусто» или «значение неизвестно».

```
let userName = null;
```

Значение undefined

- undefined – отдельный тип, означает, что «значение не было присвоено»

```
let userName;

alert(userName); // выведет "undefined"
```

Объекты - object

- object - все остальные типы выше называются «примитивными». Объекты же используются для хранения коллекций данных или более сложных объектов.

```
let user = {
    name: "Alex",
    age: 25
};
```

Представляет собой коллекцию свойств (переменных), где переменные доступны по ключу имени в объекте `user.name`.

Символы - symbol

- `symbol` – так же примитивный тип как и `number` или `string`. Используется для создания уникальных идентификаторов объектов.

  Грубо говоря его можно задать как скрытое свойство объекта, которое не будет видно в стандартных функциях для работы с объектами и циклах перебора свойств объекта. Редко используются.

Тип - bigint

- bigint – Тип BigInt был добавлен в JavaScript, чтобы дать возможность работать с целыми числами произвольной длины.

  Тип «number» не может содержать числа больше, чем 2^{53} (или меньше, чем -2^{53} для отрицательных). Это техническое ограничение вызвано их внутренним представлением. Используется если нужны действительно гигантские числа, например в криптографии или при использовании метки времени («timestamp») с микросекундами.

Оператор typeof

Оператор `typeof` возвращает тип аргумента. Это полезно, когда мы хотим обрабатывать значения различных типов по-разному или просто хотим сделать проверку.

У него есть два синтаксиса:

- Синтаксис оператора: `typeof x`
- Синтаксис функции: `typeof(x)`

Исключения

- Результатом вызова `typeof null` является `"object"`. Это неверно и это официально признанная ошибка в языке. Ее не могут поправить из-за совместимости, т.к уже очень много кода написано где это используется.
- Вызов `typeof alert` возвращает `"function"`, потому что `alert` является функцией. Но в JavaScript нет специального типа «функция». Функции относятся к объектному типу.