# Веб-программирование

## Основные концепции

1. Структура - HTML  
   HTML - это технология, которая задаёт общую структуру расположения визуальных элементов веб-приложения.
2. Стиль - CSS  
   CSS - задаёт стиль документов веб-приложения.
3. Интерактивность - JavaScript  
   Благодаря своим сценариям обеспечивает максимальную гибкость в плане управления элементами.
4. Мост
5. Сервер
6. Хранение данных
7. Платформа

В результате получаем готовое приложение.

## Интерактивность

Интерактивность можно создать с помощью JS.

DOM - (от англ. Document Object Model — «объектная модель документа») — это не зависящий от платформы и языка программный интерфейс, позволяющий программам и скриптам получить доступ к содержимому HTML, XHTML и XML-документов, а также изменять содержимое, структуру и оформление таких документов.

### jQuery

jQuery - это библиотека написанная на JS.

Вообще jQuery предоставляет нам три основные возможности:

1) упрощенный и четко организованный способ манипулирования элементами DOM (в первую очередь);

2) способ последовательной обработки событий DOM;

3) упрощенный подход к AJAX.

На стадии загрузки

Код jQuery вызывается только в ответ на что-то, что произошло в вашем коде — когда сработал триггер какой-либо из функций, вызванный событием DOM. То есть:

1. Вызвали в коде функцию
2. Элемент DOM вызвал событие
3. Сработал триггер функции

### AngularJS

<https://habrahabr.ru/post/172975/>

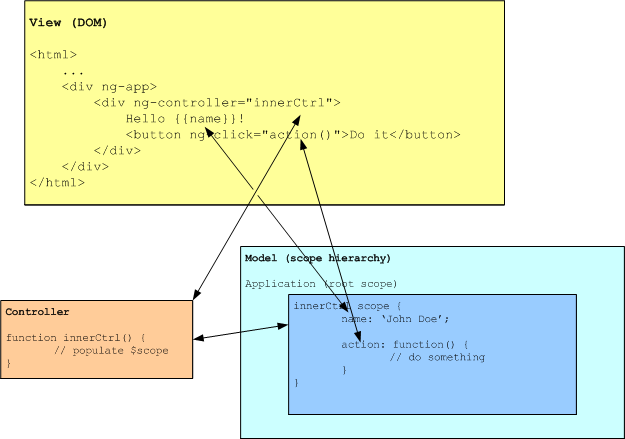
AngularJS - это фреймворк для построения веб-приложений.

Когда ваш код работает с библиотекой, он сам решает, когда вызывать ту или иную функцию.

В случае же фреймворка вы реализуете обработчики событий, и уже фреймворк решает, в какой момент их вызывать.

Что делает AngularJS

Этапы:

Angular на стадии загрузки превращает дерево DOM и код в angular-приложение.  


Html разметка вместе с директивами и фильтрами angular **компилируется** в дерево шаблонов, соотв. области видимости.

Контроллеры присоединяются к шаблонам в нужных местах.

Внутренний цикл обеспечивает привязку данных между моделью и представлением.

Если говорить про общий цикл событий, отрисовку страницы и привязку данных, то вы можете считать, что он выполняется непрерывно все время, вызывая при этом код из ваших контроллеров лишь когда это требуется.

При каждом обновлении модели (неважно, через асинхронный AJAX-запрос или напрямую через изменение данных из контроллера) Angular перезапускает цикл специальной процедуры $digest, которая обновляет привязки данных и сохраняет всю систему в актуальном состоянии.

Прямые манипуляции с DOM не просто перестают быть обязательными; можно сказать даже более, их использование в подходе Angular крайне не рекомендуется.

Дерево DOM (определено) -> в шаблонах

Данные -> в моделях

Функционал -> в контроллерах

нестандартные трансформации -> в собств. фильтрах и директивах

Преимущества AngularJS:

1. Двусторонняя привязка данных:  
   значение из DOM привязывается к данным модели через контроллер. Область видимости контроллера создаёт саму привязку.
2. Внедрение зависимости
3. Доступ к данным

## Мост

### Объекты JavaScript

// создадим объект "карта" номиналом 2 (two)

// червовой масти (hearts)

var cardOne = { "rank":"двойка", "suit":"червей" };

// Выведем номинал cardOne

console.log(cardOne.rank);

//=> двойка

//Выведем масть cardOne

console.log(cardOne.suit);

//=> червей

### JSON

JSON - это технология позволяющая обмениваться информацией между клиентами и серверами используя только объекты JavaScript. Программы могут передавать информацию в виде объектов JSON. Программы, которые принимают, могут легко распарсить этот объект в любой объект любого ЯП. Ну в JavaScript это очень легко.

// создаем строку JSON в единичных кавычках

var jsonString = '{"rank":"десятка", "suit":"червей"}'

// JSON.parse преобразует ее в объект

var card = JSON.parse(jsonString);

console.log(card.rank);

//=> "ten" десятка

console.log(card.suit);

//=> "червей"

console.log(JSON.stringify(card));

//=> {"rank":"ten","suit":"червей"}

### AJAX

AJAX расшифровывается как Asynchronous JavaScript And XML.

Основная идея AJAX — возможность для приложения получать

информацию и отдавать ее другим компьютерам без перезагрузки веб-страницы.

Фраза: “делаем AJAX запрос на сервер и получаем файл JSON”.

Запуск Хрома без ограничений безопасности (чтобы можно было получать локальные файлы):

%userprofile%\AppData\Local\Google\Chrome\Application\chrome.exe

–allow-file-access-from-files