**Вопросы 💎**

1. Напишите JSON к вашим ответам на вопрос 2 из прошлой недели. Например, если вы ответили *Книга (название, автор, год выпуска),* то JSON может выглядеть так (минимум 3 примера):
2. запомнить любимую еду ресторана/доставки

[{

"title": "Салат Цезерь",

"callorite": 44,

"veg": false

"spicy": false

},{

"title": "Том ям",

"callorite": 76,

"veg": false

"spicy": true

},{

"title": "Торт Наполкно",

"callorite": 237,

"veg": true

"spicy": false

}]

1. альбом группы

[{

"executor": "30 Seconds to Mars",

"album ": " A Beautiful Lie",

"year": 2005

"genre": "альтернативный рок"

},{

"executor": "Лазарев Сергей",

"album ": " The Best",

"year": 2015

"genre": "рор"

},{

"executor": "Eminem",

"album ": " Revival",

"year": 2017

"genre": "хип-хоп"

}]

3. книгу и на какой странице остановился пользователь

[{

"title": "Мастер и Маргарита",

"author": "Михаил Булгаков ",

"year": 1937

},{

"title": "Бойцовский клуб",

"author": "Чак Паланик",

"year": 1996

},{

"title": "Гарри Поттер и философский камень",

"author": "Джоан Роулинг",

"year": 1997

}]

1. Самостоятельно разберитесь, что за формат данных XML и чем он отличается от JSON? Приведите пример, как один и тот же объект *собачки с картинки ниже* будет выглядеть в JSON и в XML?

Разница между XML и JSON заключается в том, что XML - это язык мета-языка/разметки, а JSON - это легкий обмен данными. Таким образом, синтаксис XML разработан специально, чтобы не иметь внутренней семантики. Конкретные имена элементов ничего не значат, пока конкретное приложение обработки не обработает их определенным образом. Напротив, синтаксис JSON имеет особую семантику, встроенную в вещи между {} - это объект, вещи между [] - это массив и т.д.

Поэтому анализатор JSON точно знает, что означает каждый документ JSON. Анализатор XML знает только, как отделить разметку от данных. Чтобы разобраться со смыслом XML-документа, вы должны написать дополнительный код.

JSON:

[{

"breed": "Beagle",

"size": "Large",

"colour": "orange",

"age":"6 years"

}]

XML:

<dog>

<breed>Beagle<breed>

<size>Large<size>

<colour>orange<colour>

<age>6 years<age>

</dog>

1. Что такое сериализация и десериализация (парсинг)? В каких ситуациях они нужны?

Сериализация – это процесс где сервер по запросу клиента сначала формирует некоторый набор данных в удобном формате, который затем можно было бы очень просто упаковать в строку JSON. Завершается работа на сервере отправкой JSON данных в качестве результата клиенту.

Десериализация – обратный процусс клиент получает в качестве ответа от сервера строку JSON и распаковывает её, т.е. превращает в JavaScript объект. После этого на клиенте выполняются дальнейшие с ними действия, например, выводятся на страницу.

1. Можно ли обработать ответ от сервера одновременно и как текст, и как JSON?
2. В чем особенность асинхронных запросов?

В JS у нас есть операции, которые выполняются в фоновом/активном режиме, и поэтому наше веб-приложение не зависает каждый раз, когда оно ожидает пользовательское событие.

Тем не менее, иногда все должно проходить по порядку, иначе это вызовет хаос и неожиданные результаты. По этой причине мы можем использовать асинхронные вызовы, чтобы все работало как нужно.

1. В чем преимущество AJAX-запросов перед старым способом работы с сервером через <form action="имя скрипта на сервере">?

AJAX- это асинхронный запрос, мы остаемся на том же сайте на котором и были, form action – формовые запросы, отправляется запрос по клику на кнопку (обновляет страницу)

1. Напишите, как будет выглядеть fetch для получения данных вашего пользователя на github? Адрес URL для запроса должен выглядеть так:

'<https://api.github.com/users/сюда> подставьте свой логин с github'

fetch('https://api.github.com/users/LNad668')

.then(Response => Response.json())

.then(user => console.log(user))

.catch(error => console.log(error));

1. Самостоятельно разберитесь, что такое SPA?

SPA – это web-приложение, размещенное на одной странице, которая для обеспечения работы загружает все javascript-файлы (модули, виджиты, контролы и т.д.) , а также файлы CSS вместе с загрузкой самой страницы.”