

1. markup & markdown

- Markup - это язык разметки. К примеру, xml, yaml, json и так далее
- Markdown - это один из языков разметки. Я его юзал в лабе, могу показать.

2. protobuf?

Отличие PROTOBUF от других форматов(xml, json) в том, что в нём данные хранятся в виде набора байт, а не в виде текста.

Таким образом, сокращаются издержки на передачу данных в сравнении с текстовыми форматами.

3. CSV vs TSV

- CSV использует escape-синтаксис, что хорошо подходит для документов, отредактированных человеком, особенно для электронных таблиц НО: отсюда вытекает недостаток, для парсинга csv требуется читать escape-синтаксис
- Напротив, читать TSV просто. Никакого специального парсера не требуется. И никакое процессорное время не используется для синтаксического анализа escape-синтаксиса.
- Вывод: TSV подходит для больших наборов табличных данных(ml и анализ данных) а CSV больше подходит для наглядного представления данных для человека

TSV uses tab CSV uses comma Вот табличка:

Field-1	Field-2	Field-3
abc	hello, world!	def
ghi	Say "hello, world!"	jkl
Вот она в TSV:		

Field-1 Field-2 Field-3 abc hello, world! def ghi Say "hello, world!" jkl Вот она в CSV:

```
"Field-1","Field-2","Field-3" "abc","hello, world!","def" "ghi","Say ""hello, world!""","jkl"
```

4. Чем обусловлено постоянное появление новых форматов представления данных?

Причины:

1. Различные форматы данных используются для хранения различной информации, спрос порождает предложение. По сути разработчики придумывают разные стандарты представления информации, которые им удобны. И каждый разработчик, очевидно, видит удобство по-своему. Отсюда и 100500 разных форматов.

Рассмотрим пример json и yaml:

2. Был json и xml, некие разработчики "устали" от зоопарка различных форматов и конфигов. Им захотелось упростить себе жизнь, поэтому они создали единый понятный(для них!) формат - YAML. Его цель - быть лёгким в реализации и использовании, быть понятным человеку, быть переносимым между языками

программирования и так далее. История аналогична языку python, кому-то что-то не понравилось, было что-то неудобно и не круто. Поэтому решили разработать свой продукт, [для своего же удобства](#)

5. < и > в XML

< - это < ; > - это > ;

Угловые скобки могут представляться в явном виде только в качестве разделителей разметки, или их можно пихнуть в комментарий

[если остались вопросы, то вам сюда](#)

6. Что такое сериализация данных?

Процесс перевода структуры данных в последовательность битиков для передачи информации по сети и для сохранения её в файлы.

7. Комментарии в YAML?

Вот так: # this is single line comment.

Многострочные комментарии в YAML не поддерживаются, поэтому если хотим закомментировать много, то просто каждый раз ставим #

8. Оформление в md

- Полужирный: **текст**
- Курсив: *текст*
- Зачёркнутый текст: ~~текст~~

```
function test() {  
  console.log("смотри светится");  
}
```

заголовок первого уровня

заголовок второго уровня

заголовок третьего уровня

заголовок четвёртого уровня

заголовок пятого уровня

заголовок шестого уровня

а дальше нельзя, пшёл нафиг

9. Какие форматы обмена данных используются в Viber, WhatsApp, Telegram?

я чё ебу блять, я чё Дуров или кто чтоб такой информацией владеть. по сути там просто есть какой-нибудь удобный им формат, которым они и пользуются. Вероятно, json.

10. SVG - это?

Scalable Vector Graphics

- короче это векторная графика, так можно рисовать треугольники квадраты и всякие приколы

11. Пример ссылки на HTML

```
<a href="https://github.com/BorisDvorkin/Informatics/tree/main/lab4">my_git</a>
```

12. Какие две структуры может представлять собой в закодированном виде JSON-текст?

1. Набор пар ключ: значение (хэш-таблица, словарь):

```
{
  "firstName": "Иван",
  "lastName": "Иванов",
  "address": {
    "streetAddress": "Московское ш., 101, кв.101",
    "city": "Ленинград",
    "postalCode": 101101
  }
}
```

2. Упорядоченный набор значений(массив, вектор, список):

```
{
  "phoneNumbers": [
    "812 123-1234",
    "916 123-4567"
  ]
}
```