

Раздел 0x03

Структуры данных



Ранее в сериале...



Ранее в сериале...

- Перенесли разметку в JS



Ранее в сериале...

- Перенесли разметку в JS
- Настроили окружение



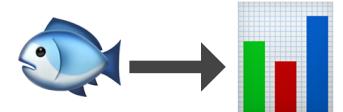
Ранее в сериале...

- Перенесли разметку в JS
- Настроили окружение
- Разбили большой файл на модули



Задача

заменить рыбу в шаблоне настоящими данными



План лекции



План лекции

– Рыба 



План лекции

- Рыба 
- Шаблоны и данные



План лекции

- Рыба 
- Шаблоны и данные
- Проектирование структуры



План лекции

- Рыба 
- Шаблоны и данные
- Проектирование структуры
- Реализация структур данных в JS



Рыба 



Рыба

Продукция Hitachi

Secure | https://www.artlebedev.ru/backstage/fish/hitachi/

	HITACHI Touch the Future	Гарантия и сервис	Техническая поддержка	Новости
О компании	Продукция			
Однажды в студеную зимнюю пору я из лесу вышел, был жаркий июль. Гляжу - поднимается медленно в гору девица, сжимая в зубах ридикюль.	Мониторы ЭЛТ 15" 17" 21" TFT 14" 15" 18" GPP 28" 42" LCD-проекторы Мобильные SVGA XGA Портативные SVGA XGA Другое Экскаваторы MPEG-камеры Handheld PC Безопасность Каталоги	Гляжу - поднимается медленно в гору белогвардейский военный патруль. Шайтан-байтан сидеть на стена. Шайтан-байтан свалиться со сна. Весь королевский илык, весь королевский арык не могут шайтана, не могут байтана, не могут байтана, шайтана-байтана, байтана-шайтана тык-тык.	Если ваш экскаватор HITACHI вдруг стал плохо копать, просто смахните с него пыль!	Шайтан-байтан сидеть на стена. Шайтан-байтан свалиться со сна. Весь королевский илык, весь королевский арык не могут шайтана, не могут байтана, не могут байтана, шайтана-байтана, байтана-шайтана тык-тык.



Рыба

Яндекс Почта  **Почта** *Посылают все!*

О почте Помощь Доставить Адресная книга Настройка Выйти из системы

Яндекс: МОРЕ новых проектов

[Реклама на Яндексе](#)

ivanov@yandex.ru

Входящие 10 (4) Глядеть Тема Рассказ о том, как Иван Васильевич утратил некоторые физические характеристики

Глядеть Дата 12.06.00

Гремячие 43 (21) От кого Roma Voroneshski <croma@design.ru> 

Пыхтящие 4 (3) Кому Ваня Иванов <ivanov@yandex.ru>

Черновики 2 (0)

Отправленные 14 (0)

Корзина 27 (0)



папка...  

Дорогой друг!

Вот ты уже и совсем большой. Пришла пора тебе узнать о том, как Иван Васильевич утратил некоторые физические характеристики, такие, как, например, скорость и плотность.

День был темным. Иван Васильевич встал с кровати, потер затылок напильником и задумался. "То ли сходить в магазин, - думал Иван Васильевич, - то ли вернуться из него." В нерешительности Иван Васильевич прошелся по комнате, подобрав валяющееся на полу зубило, постучал им по нижней челюсти и, не прийдя ни к какому решению, остановился.

"Творческий подход, - внезапно заговорило радио на кухне, - важен всегда и везде. Пишешь ли ты рыбу для Яндекса, ловишь ли ты ее себе на ужин, - неважно. Главное - все, что ты делаешь, ты должен делать с любовью. Хотя бы с любовью к своей зарплате."

Иван Васильевич прислушался.

"А сейчас в эфире сегодняшние афоризмы..." Иван Васильевич вздохнул, взял авоську и вышел из квартиры.

artlebedev.ru/kovodstvo/sections/66/



Шаблон и данные

TASKMANAGER + ADD NEW TASK

ALL 13 OVERDUE 0 TODAY 0 FAVORITES 1 REPEATING 1 TAGS 1 ARCHIVE 115

SORT BY DEFAULT SORT BY DATE up SORT BY DATE down

Example default task with default color.

23 SEPTEMBER
11:15 PM
#todo #personal #important

Example default task with custom color.

23 SEPTEMBER
11:15 PM
#todo #personal #important

Example default task with custom color and without date.

#todo #personal #important

Example default task with custom color and without hashtags.

23 SEPTEMBER
11:15 PM

Example default task without date and hashtags.

It is example of repeating task. It marks by wave.

23 SEPTEMBER
11:15 PM
#todo #personal #important

This is card with missing deadline. Deadline always marked by red line.

23 SEPTEMBER
11:15 PM
#todo #personal #important

LOAD MORE



Шаблон и данные



Шаблон и данные

- **данные** – информация, которую вводит пользователь, присыпает сервер, которая генерируется компьютером



Шаблон и данные

- **данные** – информация, которую вводит пользователь, присыпает сервер, которая генерируется компьютером
- **шаблон** – оболочка для данных, разметка, способ отобразить информацию



Зачем отделять данные от шаблона

данные абстрактны и подходят под разные способы отображения: информация не изменится, если её показывать по-разному



Зачем отделять данные от шаблона

данные абстрактны и подходят под разные способы отображения: информация не изменится, если её показывать по-разному

```
const products = [
```



Зачем отделять данные от шаблона

данные абстрактны и подходят под разные способы отображения: информация не изменится, если её показывать по-разному

```
const products = [  
  {name: `Утюг`},
```



Зачем отделять данные от шаблона

данные абстрактны и подходят под разные способы отображения: информация не изменится, если её показывать по-разному

```
const products = [  
  {name: `Утюг`},  
  {name: `Чайник`},
```



Зачем отделять данные от шаблона

данные абстрактны и подходят под разные способы отображения: информация не изменится, если её показывать по-разному

```
const products = [  
  {name: `Утюг`},  
  {name: `Чайник`},  
  {name: `Пылесос`},
```



Зачем отделять данные от шаблона

данные абстрактны и подходят под разные способы отображения: информация не изменится, если её показывать по-разному

```
const products = [  
  {name: `Утюг`},  
  {name: `Чайник`},  
  {name: `Пылесос`},  
  {name: `Стиральная машинка`},
```



Зачем отделять данные от шаблона

данные абстрактны и подходят под разные способы отображения: информация не изменится, если её показывать по-разному

```
const products = [  
  {name: `Утюг`},  
  {name: `Чайник`},  
  {name: `Пылесос`},  
  {name: `Стиральная машинка`},  
  {name: `Кухонный комбайн`},
```



Зачем отделять данные от шаблона

данные абстрактны и подходят под разные способы отображения: информация не изменится, если её показывать по-разному

```
const products = [  
  {name: `Утюг`},  
  {name: `Чайник`},  
  {name: `Пылесос`},  
  {name: `Стиральная машинка`},  
  {name: `Кухонный комбайн`},  
  {name: `и А-а-а-автомобиль`}]
```



Зачем отделять данные от шаблона

данные абстрактны и подходят под разные способы отображения: информация не изменится, если её показывать по-разному

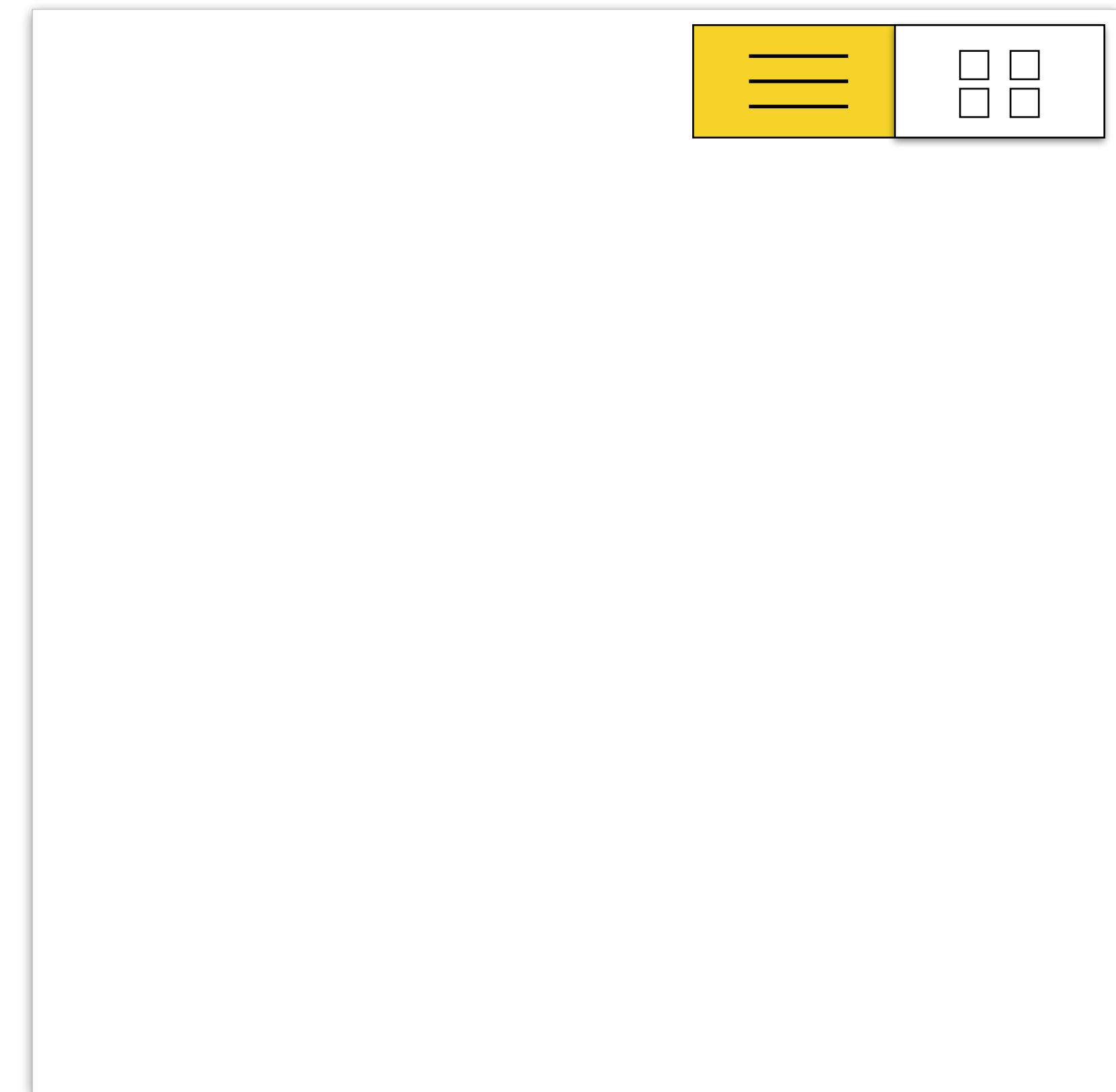
```
const products = [  
  {name: `Утюг`},  
  {name: `Чайник`},  
  {name: `Пылесос`},  
  {name: `Стиральная машинка`},  
  {name: `Кухонный комбайн`},  
  {name: `и А-а-а-автомобиль`}  
];
```



Зачем отделять данные от шаблона

данные абстрактны и подходят под разные способы отображения: информация не изменится, если её показывать по-разному

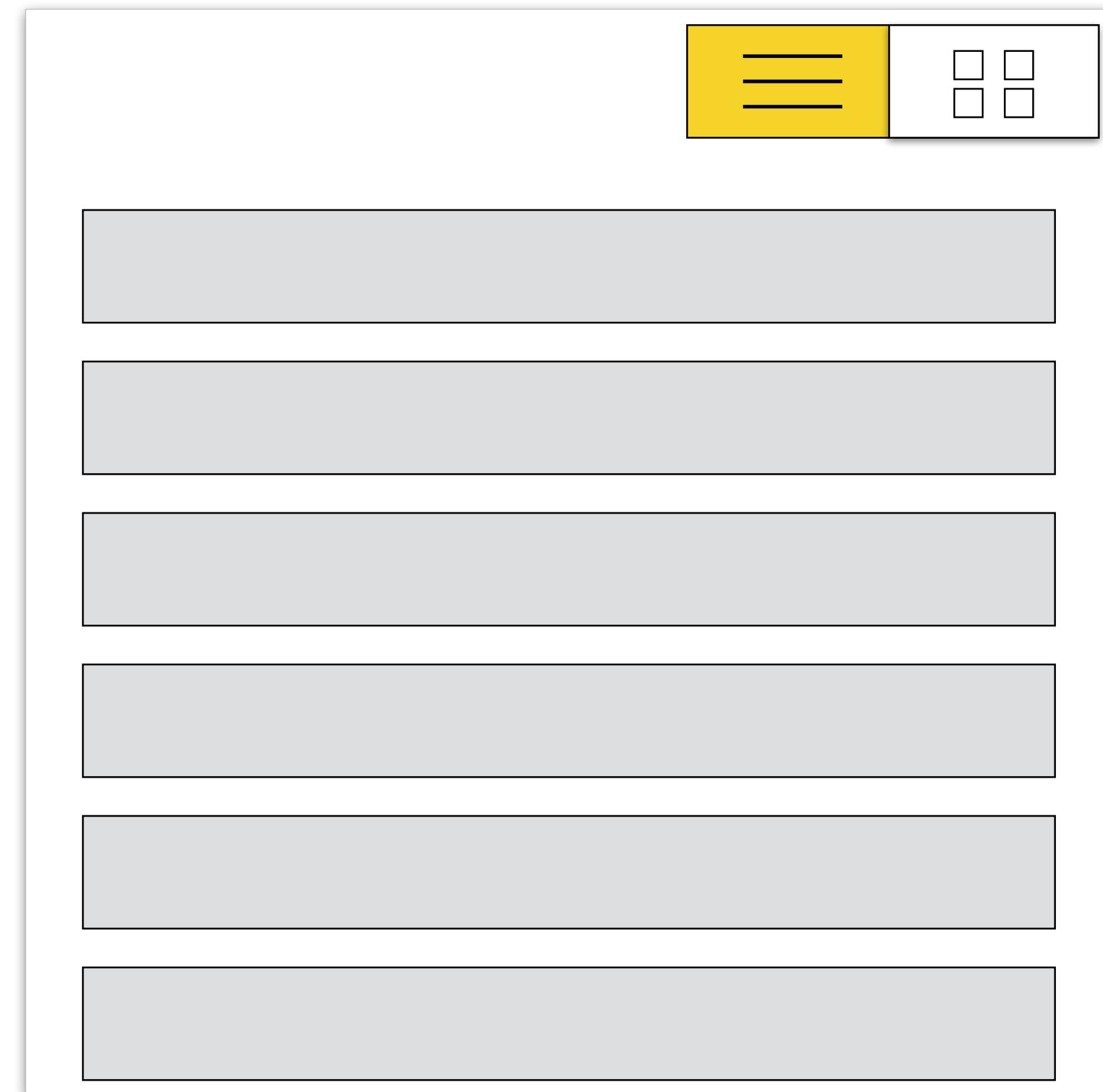
```
const products = [  
  {name: `Утюг`},  
  {name: `Чайник`},  
  {name: `Пылесос`},  
  {name: `Стиральная машинка`},  
  {name: `Кухонный комбайн`},  
  {name: `и А-а-а-автомобиль`}  
];
```



Зачем отделять данные от шаблона

данные абстрактны и подходят под разные способы отображения: информация не изменится, если её показывать по-разному

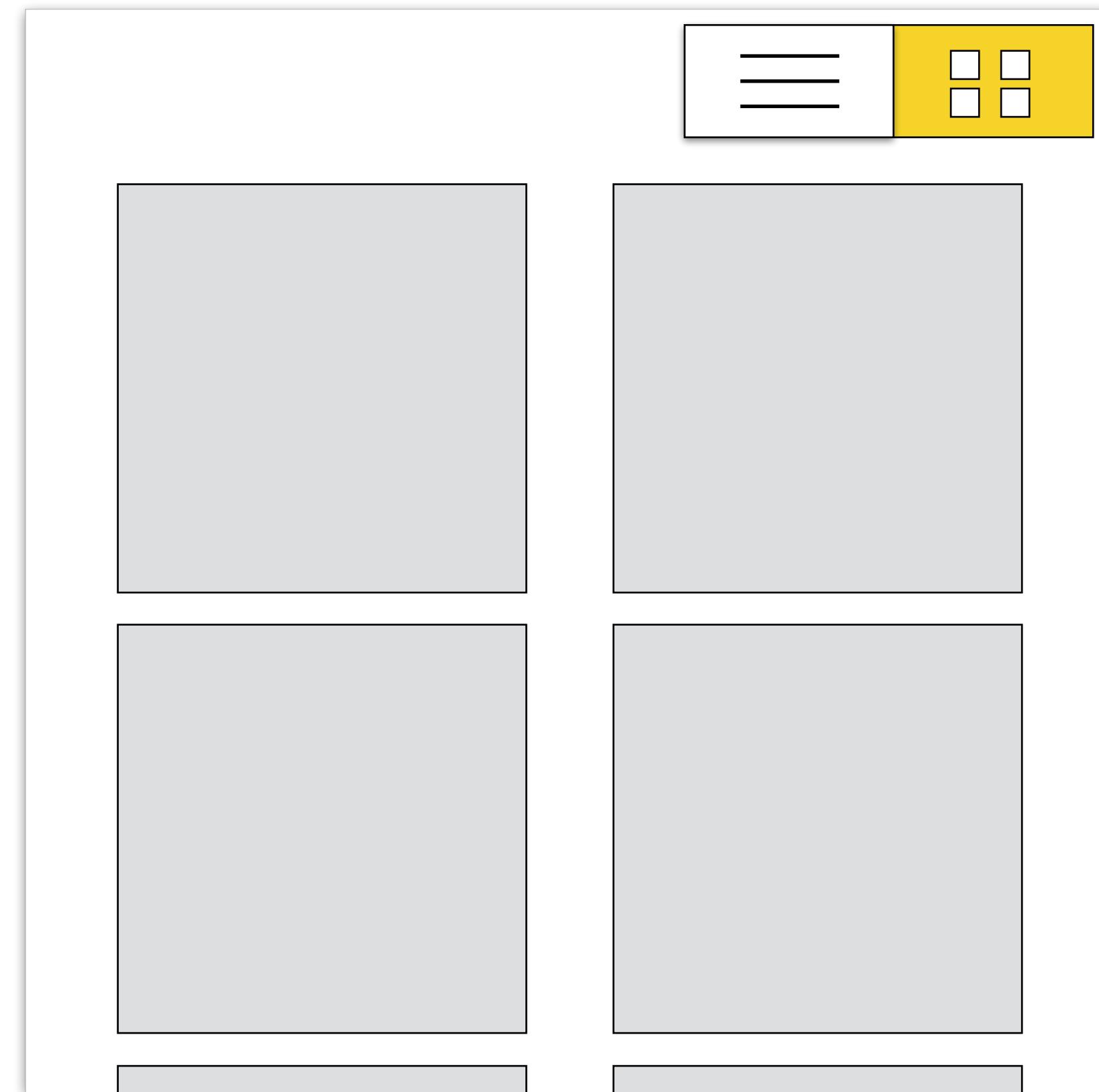
```
const products = [  
  {name: `Утюг`},  
  {name: `Чайник`},  
  {name: `Пылесос`},  
  {name: `Стиральная машинка`},  
  {name: `Кухонный комбайн`},  
  {name: `и А-а-а-автомобиль`}  
];
```



Зачем отделять данные от шаблона

данные абстрактны и подходят под разные способы отображения: информация не изменится, если её показывать по-разному

```
const products = [  
  {name: `Утюг`},  
  {name: `Чайник`},  
  {name: `Пылесос`},  
  {name: `Стиральная машинка`},  
  {name: `Кухонный комбайн`},  
  {name: `и А-а-а-автомобиль`}  
];
```



Шаблон и данные



Шаблон и данные

- Данные не должны повторять шаблон



Шаблон и данные

- Данные не должны повторять шаблон
- Данные должны описывать сущности, которыми мы оперируем.



Шаблон и данные

- Данные не должны повторять шаблон
- Данные должны описывать сущности, которыми мы оперируем.
- Простой способ отделить шаблон от данных – попробовать заменить их: поменять способ отображения или отображаемую информацию



Шаблон и данные

- Данные не должны повторять шаблон
- Данные должны описывать сущности, которыми мы оперируем.
- Простой способ отделить шаблон от данных – попробовать заменить их: поменять способ отображения или отображаемую информацию

```
const header = {  
    logo: {  
        src: 'logo.png',  
        width: 100,  
        height: 30  
    }  
};
```



Шаблон и данные

- Данные не должны повторять шаблон
- Данные должны описывать сущности, которыми мы оперируем.
- Простой способ отделить шаблон от данных – попробовать заменить их: поменять способ отображения или отображаемую информацию

```
const header = {  
    logo: {  
        src: 'logo.png',  
        width: 100,  
        height: 50  
    }  
}
```





Проектирование структуры





Проектирование структуры

TASKMANAGER + ADD NEW TASK

ALL 13 OVERDUE 0 TODAY 0 FAVORITES 1 REPEATING 1 TAGS 1 ARCHIVE 115

SORT BY DEFAULT SORT BY DATE up SORT BY DATE down

Example default task with default color.

23 SEPTEMBER
11:15 PM
#todo #personal #important

Example default task with custom color.

23 SEPTEMBER
11:15 PM
#todo #personal #important

Example default task with custom color and without date.

#todo #personal #important

Example default task with custom color and without hashtags.

23 SEPTEMBER
11:15 PM

Example default task without date and hashtags.

It is example of repeating task. It marks by wave.

23 SEPTEMBER
11:15 PM
#todo #personal #important

This is card with missing deadline.

This is card with missing deadline. Deadline always marked by red line.

23 SEPTEMBER
11:15 PM
#todo #personal #important

LOAD MORE





Проектирование структуры





Проектирование структуры

- выделить отдельные сущности





Проектирование структуры

- выделить отдельные сущности
- выявить свойства и методы сущностей и подобрать оптимальную структуру для каждой из них





Проектирование структуры

- выделить отдельные сущности
- выявить свойства и методы сущностей и подобрать оптимальную структуру для каждой из них
- определиться, как сущности связаны между собой





Проектирование структуры





Проектирование структуры

– от общего к частному





Проектирование структуры

- от общего к частному
- от частного к общему





Проектирование структуры

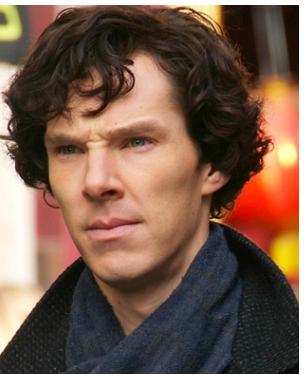
- от общего к частному
- от частного к общему





Проектирование структуры

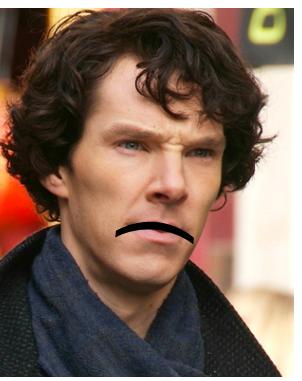
- от общего к частному
- от частного к общему





Проектирование структуры

- от общего к частному
- от частного к общему



1. Выделяем сущности





Проектирование структуры

TASKMANAGER + ADD NEW TASK

ALL 13 OVERDUE 0 TODAY 0 FAVORITES 1 REPEATING 1 TAGS 1 ARCHIVE 115

SORT BY DEFAULT SORT BY DATE up SORT BY DATE down

Example default task with default color.

23 SEPTEMBER
11:15 PM
#todo #personal #important

Example default task with custom color.

23 SEPTEMBER
11:15 PM
#todo #personal #important

Example default task with custom color and without date.

#todo #personal #important

Example default task with custom color and without hashtags.

23 SEPTEMBER
11:15 PM

Example default task without date and hashtags.

It is example of repeating task. It marks by wave.

23 SEPTEMBER
11:15 PM
#todo #personal #important

This is card with missing deadline.

This is card with missing deadline. Deadline always marked by red line.

23 SEPTEMBER
11:15 PM
#todo #personal #important

LOAD MORE





Проектирование структуры

TASKMANAGER + ADD NEW TASK

ALL 13 OVERDUE 0 TODAY 0 FAVORITES 1 REPEATING 1 TAGS 1 ARCHIVE 115

SORT BY DEFAULT SORT BY DATE up SORT BY DATE down

Example default task with default color.

23 SEPTEMBER 11:15 PM #todo #personal #important

Example default task with custom color.

23 SEPTEMBER 11:15 PM #todo #personal #important

Example default task with custom color and without date.

#todo #personal #important

Example default task with custom color and without hashtags.

23 SEPTEMBER 11:15 PM

It is example of repeating task. It marks by wave.

23 SEPTEMBER 11:15 PM #todo #personal #important

This is card with missing deadline. Deadline always marked by red line.

23 SEPTEMBER 11:15 PM #todo #personal #important

Список задач

LOAD MORE





Проектирование структуры

TASKMANAGER + ADD NEW TASK

ALL 13 OVERDUE 0 TODAY 0 FAVORITES 1 REPEATING 1 TAGS 1 ARCHIVE 115

SORT BY DEFAULT SORT BY DATE up SORT BY DATE down

Example default task with default color.

Задача

23 SEPTEMBER
11:15 PM
#todo #personal #important

Example default task with custom color.

23 SEPTEMBER
11:15 PM
#todo #personal #important

Example default task with custom color and without date.

#todo #personal #important

Example default task with custom color and without hashtags.

23 SEPTEMBER
11:15 PM

Список задач

Example default task without date and hashtags.

It is example of repeating task. It marks by wave.

23 SEPTEMBER
11:15 PM
#todo #personal #important

This is card with missing deadline.

This is card with missing deadline. Deadline always marked by red line.

23 SEPTEMBER
11:15 PM
#todo #personal #important

LOAD MORE





Проектирование структуры

TASKMANAGER + ADD NEW TASK

ALL 13 OVERDUE 0 TODAY 0 PENDING 1 UPDATING 1 TAGS 1 ARCHIVE 115

SORT BY DEFAULT SORT BY DATE up SORT BY DATE down

Список фильтров

Example default task with default color.

Задача

23 SEPTEMBER 11:15 PM #todo #personal #important

Example default task with custom color.

23 SEPTEMBER 11:15 PM #todo #personal #important

Example default task with custom color and without date.

#todo #personal #important

Example default task with custom color and without hashtags.

23 SEPTEMBER 11:15 PM

Список задач

Example default task without date and hashtags.

23 SEPTEMBER 11:15 PM #todo #personal #important

It is example of repeating task. It marks by wave.

23 SEPTEMBER 11:15 PM #todo #personal #important

This is card with missing deadline.

This is card with missing deadline. Deadline always marked by red line.

23 SEPTEMBER 11:15 PM #todo #personal #important

LOAD MORE





Проектирование структуры

The screenshot shows a task management application interface with the following elements:

- Top Bar:** Includes "TASKMANAGER", "+ ADD NEW TASK", "Фильтр" (Filter), "ALL 13", "TODAY 0", "PAST DUE 0", "COMPLETED 1", "PENDING 1", "TAGS 1", and "ARCHIVE 115".
- Filter Section:** Contains buttons for "SORT BY DEFAULT", "SORT BY DATE up", and "SORT BY DATE down".
- Task List Header:** "Список задач" (List of tasks).
- Task Cards:** Eight cards are displayed, each with a colored bar at the top and a "LOAD MORE" button at the bottom.

 - Card 1:** Example default task with default color. Contains "Задача" (Task), "23 SEPTEMBER 11:15 PM", and "#todo #personal #important".
 - Card 2:** Example default task with custom color. Contains "23 SEPTEMBER 11:15 PM" and "#todo #personal #important".
 - Card 3:** Example default task with custom color and without date. Contains "#todo #personal #important".
 - Card 4:** Example default task with custom color and without hashtags. Contains "23 SEPTEMBER 11:15 PM".
 - Card 5:** Example default task without date and hashtags. Contains "23 SEPTEMBER 11:15 PM" and "#todo #personal #important".
 - Card 6:** It is example of repeating task. It marks by wave. Contains "23 SEPTEMBER 11:15 PM" and "#todo #personal #important".
 - Card 7:** This is card with missing deadline. Contains "This is card with missing deadline. Deadline always marked by red line.".
 - Card 8:** This is card with missing deadline. Contains "23 SEPTEMBER 11:15 PM" and "#todo #personal #important".



2. Выбираем подходящие структуры





Данные



*“Bad programmers worry about the code. Good
programmers worry about data structures and their
relationships.”*

- Линус Торвальдс, создатель Linux и git





Структуры данных —

способ организовать информацию в компьютере
так, чтобы её можно было удобно использовать.

https://en.wikipedia.org/wiki/Data_structure





Список —

(последовательность) упорядоченный набор значений, в котором одно значение может встречаться более одного раза





Список

The wolf. The wolf.
The Lumberjack. The
woman goes to the
lodge.
The man, who is
difficult to find.





Список

- порядок элементов списка имеет значение





Список

- порядок элементов списка имеет значение
- один и тот же элемент может встречаться несколько раз





Список

- порядок элементов списка имеет значение
- один и тот же элемент может встречаться несколько раз
- элементы списка можно добавлять/удалять/изменять





Список

- порядок элементов списка имеет значение
- один и тот же элемент может встречаться несколько раз
- элементы списка можно добавлять/удалять/изменять
- доступ к элементам производится по ключу
(или порядковому номеру)



Array

В JavaScript массив работает сразу как две структуры данных – массив и список





Множество –

набор уникальных значений, без определенного
порядка



Set

Набор уникальных элементов. Тип элементов может быть любым



WeakSet

Вид Set, предназначенный для более эффективной работы с памятью

https://developer.mozilla.org/en/docs/Web/JavaScript/Reference/Global_Objects/WeakSet





Словарь —

(ассоциативный массив) структура данных, которая используется для сопоставления одних значений другим. Представляет собой набор пар «ключ, значение»





Словарь –





Словарь –

– порядок элементов не имеет значения





Словарь –

- порядок элементов не имеет значения
- одно и то же значение может встречаться сколько угодно раз





Словарь –

- порядок элементов не имеет значения
- одно и то же значение может встречаться сколько угодно раз
- один и тот же ключ может встречаться
только один раз





Словарь –

- порядок элементов не имеет значения
- одно и то же значение может встречаться сколько угодно раз
- один и тот же ключ может встречаться
только один раз
- элементы можно добавлять/удалять/изменять





Словарь –

- порядок элементов не имеет значения
- одно и то же значение может встречаться сколько угодно раз
- один и тот же ключ может встречаться
только один раз
- элементы можно добавлять/удалять/изменять
- доступ к элементам производится по ключу



Object

Словарь вида ключ–значение, где ключами могут быть только строки, а значениями: данные любого типа



Мар

Словарь вида ключ–значение, где и ключ и значение может быть любого типа



WeakMap

Вид Map, предназначенный для более эффективной работы с памятью

https://developer.mozilla.org/en/docs/Web/JavaScript/Reference/Global_Objects/WeakMap



3. Определяем взаимодействие между структурами

сложное взаимодействие разберем на лекции 7



Разбор домашнего задания

интерактивная демонстрация



Рассмотренные критерии



Рассмотренные критерии

- **Б8.** Массивы названы существительными во множественном числе.



Рассмотренные критерии

- **Б8.** Массивы названы существительными во множественном числе.
- **Б9.** В названии переменных не используется тип данных.



Рассмотренные критерии

- **Б8.** Массивы названы существительными во множественном числе.
- **Б9.** В названии переменных не используется тип данных.
- **Б32.** Своевременный выход из цикла: цикл не работает дольше, чем нужно.



Рассмотренные критерии

- **Б8.** Массивы названы существительными во множественном числе.
- **Б9.** В названии переменных не используется тип данных.
- **Б32.** Своевременный выход из цикла: цикл не работает дольше, чем нужно.
- **Д20.** Для итерирования по массивам и структурам данных, по которому можно итерироваться, (`Iterable`) используется конструкция `for..of`. (вместо цикла `for`, а не вместо метода `.forEach`, `.filter`, `.map`, etc.)





```
print("- C ya, pals!")
```