 PLAN

Наш заказчик уехал в отпуск, поэтому мы будем разбираться в важной теме «замыкания» на небольших «домашних» примерах.

Но эти знания так же прийдется использовать и на основном проекте. Тем более заказчик добавляет новую feature к таблице с фильмами. А так же хочет использовать нашу замечательную функцию сортировки по столбцам на другом своем проекте.

Описание релиза:

1. Написать универсальную библиотеку, для сортировки произвольного массива объектов по произвольному полю. И отрефакторить существующий код проекта.
2. Реализовать функцию поиска по точному названию фильма в таблице, выводящую одну запись

Функциональные требования:

* Библиотека принимает на вход три параметра: массив объектов, поле для сортировки и порядок сортировки. Основное отличие от предыдущей реализации в том, что мы не знаем заранее структуру объектов массива
* Библиотека должна возвращать функцию, которую мы можем использовать как обработчик события при нажатии на произвольную колонку в произвольной таблице
* Функцию поиска реализовать через Array.find(), обернутую в функцию, принимающую в качестве аргументов массив фильмов и имя для поиска

Нефункциональные требования:

* По возможности использовать замыкания, там, где это возможно
* Модуль сортировки вынести в отдельный файл sort.js и подключить к проекту

Цели и задачи обучения:

* Изучить тему замыканий на небольших практических примерах, рассмотрев основные аспекты их использования
* Применить замыкания в рамках основного проекта, уметь объяснить причину их использования
* Научиться создавать модули на основе замыканий

@ToDo

Здесь на семинарном занятии требуется:

- наглядно продемонстировать замыкание и понятие Scope функции

- привести примеры использования замыканий (напр. <https://learn.javascript.ru/closures> )

- привести классический пример алгоритма «счетчика» на замыканиях – и акцентировать внимание слушателя на том, что основная идея, это возвращение функции с её сохраненным Scope. Таким образом ее можно использовать впоследствие (например для обработки событий). При условии что исходный Scope заранее не известен, и формируется динамически в RunTime

- акцентировать внимание и перенести в домашнее задание, что по принципу счетчика может быть реализована библиотека сортировки

- акцентировать внимание, так же, что другой причиной использование замыканий является инкапсулирование области видимости переменных внутри функции – обертки. И что этот подход удобно использовать при модульной архитектуре приложения. На семинаре придумать и привести пример конфликта переменных, не обернутых в замыкание (при подключении нескольких модулей). И предложить оформить нашу функцию сортировки – как отдельный модуль (задача со \*, со ссылками на необходимую информацию)

- на семинаре привести пример другого замыкания в контексте реализации поиска фильма. Код может быть примерно такой:

*/\*\* search.js поиск объекта по полю\*/*

function search(n, films) {

let result = films.find( function(el, index){

return el.name == n;

});

return result;

}

* Добавить в чек-лист вопрос, почему внутри коллбека find виден аргумент функции search – n
* Здесь же произойдет знакомство с аспектом, функция – это тоже объект. И что функции могут передаваться как аргументы в другие функции
* Предложить самостоятельно изучить в домашнем задании другие встроенные функции работы с массивами и объектами. Потому что они все завязаны на коллбеки

 DO

Актуализация опыта и приобретаемые навыки:

На прошлых занятиях мы научились с вами делать что-то полезное. Это позволит нам сделать это и то-то. Но для того, чтобы научиться делать вот это и вот это, нам необходимо отработать некоторые шаги на практике и усвоить/повторить некоторый теоретический материал:

* **Повторить:**
  + Тему, которую изучали когда-то давно……
  + Аспект, который знаем, но давно не использовали …..
* **Прослушать лекции:**
  + Как правило последнюю/последние лекции текущего блока
* **Изучить самостоятельно:**
  + Отсылка на литературу из предыдущего **ACT**
  + Ссылка на что-то полезное, если нет в предыдущем **ACT**

<https://www.autoitscript.com/autoit3/docs/appendix/fonts.htm>

Ход работы

* ***Шаг 1.*** *Съешь же ещё этих мягких французских булок, да выпей чаю*

Съешь же ещё этих мягких французских булок, да выпей чаю Съешь же ещё этих мягких французских булок, да выпей чаю Съешь же ещё этих мягких французских булок, да выпей чаю Съешь же ещё этих мягких французских булок, да выпей чаю

Съешь же ещё этих мягких французских булок, да выпей чаю Съешь же ещё этих мягких французских булок, да выпей чаю Съешь же ещё этих мягких французских булок, да выпей чаю Съешь же ещё этих мягких французских булок, да выпей чаю

* ***Шаг 2.*** *Съешь же ещё этих мягких французских булок, да выпей чаю*

Съешь же ещё этих мягких французских булок, да выпей чаю Съешь же ещё этих мягких французских булок, да выпей чаю Съешь же ещё этих мягких французских булок, да выпей чаю Съешь же ещё этих мягких французских булок, да выпей чаю

* ***Шаг 3.*** *Съешь же ещё этих мягких французских булок, да выпей чаю*

Съешь же ещё этих мягких французских булок, да выпей чаю Съешь же ещё этих мягких французских булок, да выпей чаю Съешь же ещё этих мягких

французских булок, да выпей чаю Съешь же ещё этих мягких французских булок, да выпей чаю



*Рисунок 1. IDE VsCode*

Съешь же ещё этих мягких французских булок, да выпей чаю Съешь же ещё этих мягких французских булок, да выпей чаю Съешь же ещё этих мягких французских булок, да выпей чаю Съешь же ещё этих мягких французских булок, да выпей чаю

* ***Шаг 4.*** *Съешь же ещё этих мягких французских булок, да выпей чаю*

Съешь же ещё этих мягких французских булок, да выпей чаю Съешь же ещё этих мягких французских булок, да выпей чаю Съешь же ещё этих мягких



*Рисунок 2. Инспектор Google Chrome*

Съешь же ещё этих мягких французских булок, да выпей чаю Съешь же ещё этих мягких французских булок, да выпей чаю Съешь же ещё этих мягких французских булок, да выпей чаю Съешь же ещё этих мягких французских булок, да выпей чаю

* ***Шаг 5.*** *Съешь же ещё этих мягких французских булок, да выпей чаю*

Съешь же ещё этих мягких французских булок, да выпей чаю Съешь же ещё этих мягких французских булок, да выпей чаю Съешь же ещё этих мягких Съешь же ещё этих мягких французских булок, да выпей чаю Съешь же ещё этих мягких французских булок, да выпей чаю

Съешь же ещё этих мягких французских булок, да выпей чаю Съешь же ещё этих мягких французских булок, да выпей чаю

* ***Шаг 6.*** *Пример листинга кода - html*

Код должен выглядеть вот таким образом. Обязательно указание имени файла или другой системной информации в начале файла. Курсивом, в виде комментария соответствующего языка

*<!--index.html html-разметка навигации-->*

<div class="menu-wrap">

<nav class="block01\_\_nav text-right">

<ul>

<li><a href="#">Заказать билет</a></li>

<li><a href="#">Преимущества</a></li>

<li><a href="#">Фильмы</a></li>

<li><a href="#">Контакты</a></li>

</ul>

<div class="menu-close"></div>

</nav>

</div>

мягких Съешь же ещё этих мягких французских булок, да выпей чаю Съешь же ещё этих мягких французских булок, да выпей чаю

* ***Шаг 7.*** *Пример листинга кода – командная строка*

Код должен выглядеть вот таким образом. Обязательно указание имени файла или другой системной информации в начале файла. Курсивом, в виде комментария соответствующего языка

*#!/bin/bash вызов сборщика webpack*

#> npm webpack dev --watch

* ***Шаг 8.*** *Пример листинга кода - JavaScript*

Код должен выглядеть вот таким образом. Обязательно указание имени файла или другой системной информации в начале файла. Курсивом, в виде комментария соответствующего языка

*/\*\* main.js точка входа для сборки Vue приложения \*/*

window.app = new Vue({

el: '#app',

store\_modules: {userData, menuData, roleData, notificationData}

components: {

Page

},

data () {

return {

}

},

mounted () {

this.$store.dispatch('userData/current');

this.$store.dispatch('menuData/load');

this.$store.dispatch('notificationData/load');

},

render: h => h(Page)

});

* ***Шаг 9.*** *Пример листинга кода – стилевой файл*

*/\*\* main.css стили кнопок \*/*

.simple-text {

font-size: .875em;

font-weight: 100;

}

.button1 {

display: inline-block;

background: #90d23d;

color: #ffffff;

padding: 1.5rem 1.5vw;

border-radius: 12px;

font-size: 1.125em;

transition: .2s;

border: 2px solid #90d23d;

cursor: pointer;

}

.button1:hover {

color: #90d23d;

background: transparent;

}

* ***Шаг 10.*** *Пример акцентов*

Съешь же ещё этих мягких французских булок, да выпей чаю Съешь же ещё этих мягких французских булок, да выпей чаю Съешь же ещё этих мягких французских булок, да выпей чаю Съешь же ещё этих мягких французских булок, да выпей чаю



Запомнить!

Съешь же ещё этих мягких французских булок, да выпей чаю Съешь же ещё этих мягких французских булок, да выпей чаю Съешь же ещё этих мягких французских булок, да выпей чаю Съешь же ещё этих мягких французских булок, да выпей чаю



Важно!

Съешь же ещё этих мягких французских булок, да выпей чаю Съешь же ещё этих мягких французских булок, да выпей чаю Съешь же ещё этих мягких французских булок, да выпей чаю Съешь же ещё этих мягких французских булок, да выпей чаю



Интересно!

Съешь же ещё этих мягких французских булок, да выпей чаю Съешь же ещё этих мягких французских булок, да выпей чаю Съешь же ещё этих мягких французских булок, да выпей чаю Съешь же ещё этих мягких французских булок, да выпей чаю

Съешь же ещё этих мягких французских булок, да выпей чаю Съешь же ещё этих мягких французских булок, да выпей чаю Съешь же ещё этих мягких французских булок, да выпей чаю Съешь же ещё этих мягких французских булок, да выпей чаю