Week 22

1. Найдите еще три библиотеки (модуля), которые могут быть вам полезны в разработке ( \*Например, js-datepicker - для ввода дат, ChartJS - для работы с графиками и datatable для простых и удобных таблиц с пейджингом и сортировкой.\*).  
     
   validator.js - валидатор  
   [tinycolor.js](https://raw.githubusercontent.com/bgrins/TinyColor/master/tinycolor.js)- TinyColor is a small, fast library for color manipulation and conversion in JavaScript. It allows many forms of input, while providing color conversions and other color utility functions. It has no dependencies.  
   Autosize.js — плагин автоматически регулирует высоту текстовой области, добавляя ей свойство автоматической регулировки высоты. Пользователю удобней просматривать набранную информацию.  
   [Lettering.js](http://letteringjs.com/) — дает полный контроль над текстом. Используется для анимации. С помощью плагина слово разбивается на буквы и с каждой можно работать отдельно.  
   [Barba.js](http://barbajs.org/) — библиотека делает «приложение одной страницы». Компоненты сайта загружаются на странице единоразово. Иными словами — страница без перезагрузки. Переходя между вкладками ничего не прогружается и создается впечатление, что находишься на одной странице. Как будто ты в приложении.

2. Какие 2 обязательных шага нужно сделать до начала обращения к методам модулей?  
Создать новую папку или репозиторий, зайти туда. И сделать npm init. В результате получаем файл Package.json с данными о проекте. После этого можно устанавливать библиотеки.

3. Как узнать, установлен ли у тебя менеджер пакетов \*\*npm\*\*?  
  
В терминале в нужной папке набрать npm -v

4. Зачем нужен блок `finally`?   
  
Секцию `finally` часто используют, когда начали что-то делать и хотим завершить это вне зависимости от того, будет ошибка или нет.

Секция `finally` подходит для завершения измерений несмотря ни на что.

5. Есть следующий код:

let user = undefined;

alert(`Привет, ${user.name}`);

Как сделать так, чтобы при обращении к нему выводилось внятное сообщение об ошибке "Имя пользователя не заполнено"?  
  
let user = undefined;

try {

alert(`Привет, ${user.name}`);}

catch(e) {alert('Имя пользователя не заполнено')};

6. Как сгенерировать собственную ошибку, например, в случае некорректного формата данных?

let a = 6;

let b = -4;

try {

if (b<=0) {

throw new Error("Введите положительное число");

let result = a/b;

}

console.log(result);

}

catch(error) {

console.log("Неверный формат введенных данных. Подробности: " + error.message);

}

7. Какую команду надо ввести, чтобы сгенерировался файл package.json?

Npm init|  
после ответов на вопросы - ls

8. Приведите пример захвата ошибки в случае некорректного преобразования данных

console.log(parseInt('ыыыы'));  
  
 try {

let x= ыыыы;

result = parseInt('x');

if (result = NaN) {

throw new Error('Неверный формат данных');}

}

catch(error) {

console.log('Ошибка: получены некорректные данные'); /\*error.message\*/ //error.name //error.stack);

}

console.log(parseInt('ыыыы'));

9. Изучите документацию к библиотеке moment [https://momentjs.com/](https://momentjs.com/) и скажите, как вывести название дня недели по дате?  
  
Возможно, так:  
moment().day(Number|String);

moment().day(); // Number

moment().days(Number|String);

moment().days(); // Number

10. Что такое сборщик и зачем он нужен? Какие еще бывают сборщики помимо \*browserify\*?  
  
Суть работы сборщиков (бандлеров) заключается в том, что они берут JavaScript-код, содержащийся во множестве файлов, и упаковывают его в один или несколько файлов, определённым образом упорядочивая и подготавливая к работе в браузерах. Более того, благодаря различным подключаемым модулям (плагинам) и загрузчикам, код можно минифицировать, можно упаковывать, помимо кода, и другие ресурсы (вроде CSS-файлов и изображений). Бандлеры позволяют использовать препроцессоры, умеют разделять код на фрагменты, загружающиеся тогда, когда в них возникнет необходимость. Но их возможности этим не ограничиваются. Фактически, речь идёт о том, что они помогают организовывать процесс разработки.

WEBPACK, Parcel