Тестовые задания по frontend

## Задание 1. Основы JS

1.1 Что выведет в консоль следующий код и почему?

setTimeout(() => console.log('setTimeout 1'), 0);

Promise.resolve().then(() => console.log('Promise 1'));

Promise.resolve().then(() => setTimeout(() => console.log('setTimeout 2'), 0));

Promise.resolve().then(() => console.log('Promise 2'));

setTimeout(() => console.log('setTimeout 3'), 0);

console.log('final');

**Ответ:**  
- Первоначально выполнится последовательно синхронный код – ‘final’

- После будет выполняться асинхронный код, а именно сначала микротаски (promises), а после макротаски (WebApi, setTimeout).

- Получается сначала выполнятся 'Promise 1', в промисе который на 3 строке есть макротаска, значит это действие откладывается, и после выполняется 'Promise 2'

- Теперь выполняются макротаски 'setTimeout 1' и 'setTimeout 3', а так как 'setTimeout 2' в промисе, то выполнится последний

В консоль выведется:

'final'

'Promise 1'

'Promise 2'

'setTimeout 1'

'setTimeout 3'

'setTimeout 2'

1.2 Почему в консоль выводится не то, что мы хотим, и как исправить без удаления setTimeout?

for (var i = 0; i < 5; i++) {

setTimeout(function() {

console.log(i);

});

}

**Ответ:**

В консоль выводится 5 раз число 5. Такое происходит потому что в цикле у нас переменная определена в глобальной области видимости через var и имеется setTimeout, и поэтому в константу сразу записывается конечный результат, а не выводится список чисел от 0 до 4.  
Если мы хотим видеть числа от 0 до 4 есть 2 варианта решения:  
1. заменить var на let

2. убрать setTimeout, если нам нету необходимости выводить через какой то промежуток времени

1.3 Опишите как работает следующий код:

var a = 15;

function foo() {

var a = 25;

var result = function () {

console.log(this.a);

}

result.a = 35;

return result;

}

var bar = foo().bind({ a: 45 });

var buzz = foo().bind({ a: 55 }).bind({ a: 65 });

bar();

buzz();

Что будет, если в console.log заменить this.a на a? Почему? Как доработать функцию, чтобы выводилось 15 из первой строки?

**Ответ:**

- Если заменить this.a на a, то в консоли мы увидим два раз 25. Потому что перестали ссылаться на контекст, который передаётся через bind

- удалить строку с переменной а=25, и удалить this. Тогда как внутри функции переменной а не будет, а она есть в области видимости выше, то в логах увидим 15

- var a = 15;

function foo() {

var result = function () {

console.log(a);

}

result.a = 35;

return result;

}

var bar = foo().bind({ a: 45 });

var buzz = foo().bind({ a: 55 }).bind({ a: 65 });

bar();

buzz();

## Задание 2. Рецензирование и рефакторинг

Прорецензируйте представленный код (можно написать комментарии в коде), напишите свой вариант реализации.

let array = [-2, 4, -10, 8];

let count = 0;

for (let i = 0; i < array.length; i++) {

if (!String(array[i]).startsWith("-")) {

count += array[i]

}

}

console.log(count);

**Ответ:**

- Не понятно зачем каждое число из массива переводить в строку, для того чтоб проверить на отрицательность. Используя перебор массива через «for», кажется что ресурсы по памяти важны, но после манипуляции с данными, сводится на нет.

- мои 2 варианта реализации:

let array = [-2, 4, -10, 8];

let count = 0;

for (let i = 0; i < array.length; i++) {

if (array[i]>0) {

count += array[i]

}

}

console.log(count);

let array = [-2, 4, -10, 8];

let count = array.reduce((a,b)=>a+(b>0 &&b),0)

console.log(count);

## Задание 3. Разработка сайта

Сервис https://jsonplaceholder.typicode.com предоставляет некоторое API. Требуется разработать сайт, на главной странице которого расположить все фото, предоставляемые API, в виде сетки. Под каждым фото должна быть ссылка на автора, которая ведет в его профиль, где можно посмотреть альбомы автора и контакты. Должна быть возможность открыть альбом и посмотреть фото внутри него.

Дизайн интерфейса на Ваше усмотрение.

Хостить сайт не нужно, готовое решение можно предоставить в виде архива либо ссылкой на Git.

Требования:

* поддержать работу в браузерах IE11, Chrome, Opera, Firefox;
* использовать ui-kit (любой на выбор);
* использовать React (предпочтительно), либо Vue/Angular;
* использовать Redux (предпочтительно), либо MobX.

Будет плюсом:

* эстетика и удобство использования;
* адаптивная верстка;
* использование системы контроля версий.