Junior

Загальні:

1.Які методи HTTP-запитів ви знаєте?

2.Які версії HTTP-протоколу вам відомі?

3.Які знаєте коди відповіді (стану) HTTP?

4.Що таке Cross-Origin Resource Sharing? Як усунути проблеми з CORS?

5.Що таке cookie?

6.Який максимальний розмір cookie?

7.Що означає директива use strict?

8.Чим JS відрізняється під час роботи на front-end і back-end?

9.Що таке статична і динамічна типізації?

10.Як клієнт взаємодіє із сервером?

11.Що таке REST?

12.Поясніть поняття мутабельність / іммутабельність? Які типи є мутабельними й навпаки?

13.Як шукати помилки в коді? Чи використовуєте дебагер?

14.Яких відомих людей зі світу JS знаєте?

JS Core

15.Які існують типи даних у JS?

16.Як перевірити, чи об’єкт є масивом?

17.Як перевірити, чи число є скінченним?

18.Як перевірити, що змінна рівна NaN?

19.Чим відрізняється поведінка isNaN() та Number.isNaN()?

20.Порівняйте ключові слова var, let, const.

21.Що таке область видимості?

22.Що таке деструктуризація?

23.Для чого призначені методи setTimeout і setInterval?

24.Порівняйте підходи роботи з асинхронним кодом: сallbacks vs promises vs async/await.

25.Чи можна записувати нові властивості / функції в прототипи стандартних класів (Array, Object тощо)? Чому ні? У яких випадках це робити можна? Як убезпечити себе, якщо потрібно розширити прототип?

26.Назвіть методи масивів, які пам’ятаєте, і скажіть, для чого вони потрібні.

27.Які перебираючі методи масиву знаєте? У чому їхня відмінність?

28.Як працюють оператори присвоєння / порівняння / рядкові / арифметичні / бітові тощо?

29.Опишіть призначення і принципи роботи з колекціями Map і Set.

30.Що означає глибока (deep) та поверхнева (shallow) копія об’єкта? Як зробити кожну з них?

Функції:

31.Яка різниця між декларацією функції (function declaration) та функціональним виразом (function expression)?

32.Що таке анонімна функція?

33.Розкажіть про стрілкові функції (arrow function). В чому полягають відмінності стрілкових функцій від звичайних?

34.Що таке і для чого використовують IIFE (Immediately Invoked Function Expression)?

35.Що таке hoisting, як він працює для змінних і функцій?

36.Що таке замикання (closure) і які сценарії його використання?

37.Як ви розумієте замикання? Що буде виведено в консолі в цьому випадку?

var f = function() {

console.log(1);

}

var execute = function(f) {

setTimeout(f, 1000);

}

execute(f); // что выведет в консоль и почему - ничего, потому что f не возвращает никакого значения.

f = function() {

console.log(2);

}

38.Що таке рекурсія? Это когда функция упрощается(путем разделения на несколько аналогичных но более простых) и вызывает саму себя внутри

39.Що означає ключове слово this? - что свойство нужно искать в этом объекте

40.Що таке втрата контексту, коли відбувається і як їй запобігти?

41.Методи функцій bind / call / apply — навіщо і в чому різниця?

Front-end

42.Що таке DOM? Документ Обжект Модэл (объектная модель документа)

43.Порівняйте атрибути підключення скрипту async і defer в HTML-документі.

44.Яка різниця між властивостями HTML-елементів innerHTML і innerText? ИннерХТМЛ позволяет получать и записывать ХТМЛ теги, а текст только текст.

45.Опишіть процес спливання (bubbling) подій у DOM.

46.Як зупинити спливання (bubbling) події?

47.Як зупинити дефолтну обробку події? Функцией отмены стандартного поведения event.preventDefault();

48.Чому дорівнює this в обробнику подій (event handler)?

49.Що таке LocalStorage і SessionStorage? Який максимальний розмір LocalStorage?

50.Як отримати висоту блоку? Його положення щодо меж документа?

51.Що таке webpack?

52.Чим відрізняється dev-збірка від prod?

Верстка

53.Що таке блокова модель CSS?

54.Які способи центрування блокового контенту по горизонталі та вертикалі знаєте?

55.Які підходи у верстці вам відомі (float, flex, grid, etc)?

56.Як зробити додаток responsive?

57.Які є принципи семантичної верстки?

58.Навіщо потрібні префікси для деяких CSS-властивостей (-webkit-, -moz- тощо)?

59.Як спростити написання кросбраузерних стилів?

60.Практичне завдання: прокоментувати та виправити приклад поганого CSS або HTML.

61.Що таке CSS-препроцесори? З якими працювали? Що нового вони приносять у стандартний CSS?

Angular

62.Перерахуйте основні компоненти фреймворку (модуль, роут, директива тощо).

63.У чому різниця між компонентом і директивою?

64.Розкажіть про життєвий цикл компонента.

65.Перерахуйте часто використовувані хуки життєвого циклу компонента та розкажіть, для чого вони потрібні?

66.У чому різниця між конструктором і ngOnInit-хуком?

67.Як захистити роут від несанкціонованого доступу? Які механізми надає для цього фреймворк?

68.Що таке Lazy loading, як і для чого використовується?

69.Яке призначення RouterOutlet?

70.Як компоненти можуть взаємодіяти один з одним?

71.Як створити two-way binding властивість для компонента?

72.Які типи форм є у фреймворку? У яких випадках і що краще використовувати?

73.Які стани є у форми і як це можна застосувати?

74.Навіщо потрібні сервіси? Як з ними працювати?

75.Що таке singleton-сервіси? Яке їхнє призначення? Спосіб створення?

76.Які є способи оголошення сервісів?

77.Для чого потрібні модулі? Скільки їх має бути в проєкті?

78.Навіщо потрібні загальні модулі (shared)?

79.Які переваги типізації в TypeScript?

80.Які можливості TypeScript можна використовувати для типізації (тут мають на увазі інтерфейси, типи, enum тощо)?

81.Яка різниця між інтерфейсом і класом?

82.У чому різниця між інтерфейсом і абстрактним класом?

83.Яка різниця між інтерфейсом і типом?

84.Що таке RxJS? Як він використовується у фреймворку? Які компоненти фреймворку тісно пов’язані з ним?

85.Чим відрізняються Observable і Promise?

86.Для чого потрібні Subjects? Які типи Subjects існують?

87.Як зробити кілька послідовних запитів до API за допомогою HTTP-сервісу і RxJS?

88.Яка різниця між switchMap, concatMap, mergeMap?

89.Як можна конфігурувати Angular-застосунок?

90.Навіщо потрібні environment-файли? Коли їх краще не використовувати?

91.У чому різниця між «розумним» (smart) і «дурним» (dumb) компонентами? У яких випадках застосовується кожен з них?

92.У чому різниця між NgForm, FormGroup і FormControl і як їх застосовують для побудови форм?

93.Навіщо потрібен і як працює async pipe?

94.Як стежити за розвитком фреймворку? Яких відомих людей, пов’язаних з Angular, знаєте / читаєте?

React

95.Чи працювали з класовими компонентами? У чому їхня особливість?

96.Які дані краще зберігати в стані компонента, а які передавати через пропси? Наведіть приклад.

97.Чи ознайомлені з хуками? У чому їхні переваги? Чи доводилося робити свої і з якою метою?

98.Чи ознайомлені з фрагментами та порталами? Навіщо вони потрібні?

99.Коли й для чого використовують рефи?

100.Які ви знаєте методи життєвого циклу компонента?

101.В якому методі життєвого циклу компонента краще робити запити на сервер? Чому?

102.В якому методі життєвого циклу компонента краще робити підписку і відписку від лістенера? Чому? Навіщо відписуватися?

103.Чи був досвід роботи з контекстом? Коли його варто використовувати?

104.У чому особливість PureComponent?

105.Чи працював з мемоізованими селекторами (memoized selectors)? Для чого їх використовують і який принцип роботи?

106.У чому бачите переваги бібліотеки React?

107.Чому бібліотека React швидка? Що таке Virtual DOM і Shadow DOM?

108.Навіщо в списках ключі? Чи можна робити ключами індекси елементів масиву? Коли це виправдано?

109.В чому основна ідея Redux?

110.Робота зі стилями в React.

111.React — це бібліотека чи фреймворк? Яка різниця між цими двома поняттями.

112.Чи можна використовувати jQuery разом з React? Чому так / ні?

113.Що таке codemod?

114.Чи доводилося налаштовувати проєкт React з нуля? За допомогою яких інструментів ви це робили?

115.Перерахуйте всі бібліотеки, які використовували у зв’язці з React.

116.Що найскладніше доводилося реалізовувати за допомогою React?

Back-end

117.Що таке REPL?

118.Що таке streams в Node.js?

119.Що таке middleware?

120.Для чого використовують функцію setImmediate?

121.Навіщо потрібен app.param() в express?

122.Що таке token based authentication?

Бази даних

123.Напишіть простий запит для обчислення трьох авторів, у яких найбільше книг.

124.Напишіть запит, який вибирає останні три коментарі для конкретного користувача для двох таблиць: коментарі та користувачі.

125.Спроєктуйте просту схему бази даних для бібліотеки.

126.Для чого використовують SQL-оператор HAVING?

127.Навіщо використовують SQL-оператор LEFT JOIN?

128.Чим відрізняється embed- від reference-зв’язку в MongoDB?

129.В одному проєкті програмісти зберігають дані в MongoDB-колекції коментарів, використовуючи такі типи даних (дивіться нижче). Що поганого в цьому рішенні?

id: ObjectID

text: string

author\_id: string

created\_at: Date

130.У проєкті знадобилося внести зміни в структуру таблиць, додати кілька полів і індекси. Як програмісти будуть робити це на продакшені?

Інструменти

131.Щоразу, коли ви робите pull, чомусь трапляється конфлікт в останньому рядку в усіх файлах, які ви редагували. Що відбувається?

132.Що робить команда git fetch?

133.Які git hygiene підходи ви знаєте?

134.Що таке CI/CD? Для чого це потрібно?

Практичні завдання

135.Розкажіть, які є способи копіювання простого об’єкта типу obj = {a: 1, b: 2, c: 3}

136.Напишіть deep clone для об’єкта.

137.Назвіть різні способи, як поміняти місцями значення двох змінних.

138.Менеджер попросив у задачі поміняти статуси з «active, inactive» на «active, removed», але в коді фігурують тільки цифри й незрозуміло, який статус відповідає якій цифрі. Як допомогти майбутнім програмістам не лізти в документацію за кодом? Питання ставлять на конкретному прикладі з кодом.

139.Необхідно зробити мініпроєкт — список користувачів з формою створення/редагування користувача:

Для зберігання користувачів використовуйте Firebase (це безкоштовно).

Для стилізації використовуйте Bootstrap.

Мінімальний набір полів користувача:

ім’я;

прізвище;

електронна пошта;

телефон (у форматі +380 (XX) XXX-XX-XX);

дата народження;

буде плюсом — додавання аватара та можливість crop-картинки.

Список користувачів повинен мати можливість фільтрації та пагінацію.

Проєкт має містити README-файл з кроками для запуску.

Middle

Загальні

1.Розкажіть про піраміду тестування.

2.Які типи автоматизованих тестів випадала нагода писати? Які бібліотеки при цьому використовували? Яким інструментам віддаєте перевагу і чому?

3.Що таке unit-тести? Яке місце в піраміді тестування займають unit-тести?

4.Що таке code coverage? Чи обов’язкове 100% покриття тестами коду?

5.Як заборонити браузеру віддавати кеш на HTTP-запит?

6.Що таке XSS (Cross-Site Scripting)?

7.Розкажіть про патерни Observer, Pub / Sub. Яка між ними різниця? Наведіть приклади реалізації цих патернів у відомих фреймворках / бібліотеках / браузерних API.

8.З якою метою може бути використаний event listener події fetch self.addEventListener(’fetch’, event => {}) ?

9.Що таке Event loop і як він працює? Розкажіть про мікрозадачі та макрозадачі.

JS Core

10.Які типи даних бувають у JavaScript? Який буде результат виконання коду?

let firstObj = { name: 'Hello' };

let secondObj = firstObj;

firstObj = { name: 'Bye' };

console.log(secondObj.name);

11.Що таке temporal dead zone?

12.Як працює boxing / unboxing у JavaScript?

13.У чому різниця між оператором in і методом hasOwnProperty?

14.Опишіть, за допомогою чого в JS реалізуються такі ООП-парадигми, як інкапсуляція, поліморфізм, абстракція?

15. Що таке прототип? Як працює прототипне наслідування в JS? Поясніть роботу коду.

function Main () {}

Main.prototype = { protected: true };

const obj = new Main();

Main.prototype = { protected: false };

console.log('Object protection: ', obj.protected);

16.Яка різниця між композицією та наслідуванням?

17. Чому не варто використовувати конструктори типу new String?

18.Розкажіть про базовий пристрій і механізм роботи Event loop.

19. Що таке записи (records) і кортежі (tuples)? Чим вони відрізняються від звичайних об’єктів?

20.Які відмінності в поведінці ES5 функції-конструктора та ES2015 класу?

21.Як реалізувати патерн «Модуль»?

22. Чому typeof null повертає object?

23.Що таке приведення (перетворення) типів у JS?

24.Що таке явне і неявне приведення (перетворення) типів даних у JS? Як відбувається перетворення типів у таких прикладах:

{}+[]+{}+[1]

!!"false" == !!"true"

['x'] == 'x'

25.Що таке Garbage Collector?

26.Опишіть основні принципи роботи «збирача сміття» у JS-рушіях (engines).

27.Опишіть призначення і принципи роботи з колекціями WeakMap і WeakSet? Чим вони відрізняються від колекцій Map і Set відповідно?

28.Чим відрізняється Observable від Promise?

29. Що таке Promise? Назвіть порядок виконання then і catch у ланцюжку.

Promise.resolve(10)

.then(e => console.log(e)) // ??

.then(e => Promise.resolve(e))

.then(console.log) // ??

.then(e => {

if (!e) {

throw 'Error caught';

}

})

.catch(e => {

console.log(e); // ??

return new Error('New error');

})

.then(e => {

console.log(e.message); // ??

})

.catch(e => {

console.log(e.message); // ??

});

30.Розкажіть про послідовне і паралельне виконання асинхронних функцій. У чому різниця між Promise.all () і Promise.allSettled ()?

31.Що таке дескриптори властивостей об’єктів? Розкажіть про їхнє практичне застосування.

32.Назвіть кілька способів створення незмінного об’єкта в JavaScript.

33.Як створити властивість в об’єкта, яку не можна буде змінити?

34.Навіщо потрібен конструктор Proxy? Наведіть приклад використання.

35.Що таке ArrayBuffer? У чому різниця між Uint32Array і Float32Array? Який результат виконання коду?

const uint32Array = new Uint32Array();

Array.isArray(uint32Array);

36.Яким буде результат порівняння?

const url = “HTTPs://xyz.com/path<to>page.html”;

encodeURI(url) == encodeURIComponent(url);

37.Розкажіть про генератори та ітератори.

38.. Поясніть, що робить наведений код:

function \* fn(num) {

for (let i = 0; i < num; i += 1) {

yield console.log(i);

}

}

const loop = fn(5);

loop.next();

loop.next();

39.Розкажіть про тип даних Symbol і його практичне застосування. Як перевести число з 10-розрядної системи в 16(2,8)-розрядну систему числення?

Функції

40.Поясніть, що означає currying. Наведіть приклад використання на практиці.

41.Наведіть приклад функції з мемоізацією. Коли варто застосовувати цю техніку?

42.Що таке чейнінг функцій? Напишіть приклад з використанням цього підходу.

43.У чому різниця між function і arrow function? Яким буде результат виконання коду?

const pluckDeep = key => obj => key.split('.').reduce((accum, key) => accum[key], obj)

const compose = (...fns) => res => fns.reduce((accum, next) => next(accum), res)

const unfold = (f, seed) => {

const go = (f, seed, acc) => {

const res = f(seed)

return res ? go(f, res[1], acc.concat([res[0]])) : acc

}

return go(f, seed, [])

}

Front-end

44.У чому принципова різниця між подіями mouseleave і mouseout?

45.У якому порядку обробляються призначені для користувача події в DOM (click, mouseover тощо)? FIFO чи LIFO?

46.Що таке Event bubbling та Event capturing?

47.Порівняйте методи об’єкта event stopPropagation та stopImmediateProparation.

48.Які є підходи оптимізації продуктивності вебсторінки?

49.Як реалізований механізм same-origin policy в браузері? На які браузерні API він поширюється?

50.Назвіть способи зберігання даних у браузері. Порівняйте їх.

51.Web worker’и. Опишіть особливості передачі даних між worker’ами та основним потоком, між розділеними worker’ами.

51.Що таке Transferable-об’єкти?

52.Розкажіть про способи оптимізації виконання ресурсомістких операцій JS для поліпшення продуктивності рендерингу контенту на сторінці.

53.Чому ResizeObserver викликає події зміни розміру до відтворення елемента, а не після?

54.Розкажіть, як ви розумієте Web Accessibility?

55.Опишіть алгоритм створення функціоналу, що забезпечує читання вмісту .txt-файлу при перетягуванні його з файлової системи у вікно браузера.

56.Що таке Virtual DOM?

Верстка

57.Поясніть різницю між одиницями вимірювання px, em, rem.

58.Для чого потрібні CSS-змінні? Наведіть кілька прикладів використання.

59.Що станеться при додаванні наступного селектора?

\* { box-sizing: border-box; }

60.Як адаптувати сторінку для друку?

61.Опишіть особливості кастомізації стилів стандартних елементів форм.

62.Що таке progressive рендеринг? Які підходи використовуються?

63.Назвіть кілька способів реалізації lazy-loading медіаресурсів на сторінці.

64.Назвіть популярні шаблонізатори для фронтенд-розробки. Опишіть особливості їх використання.

65.Назвіть популярні CSS-методології та їхні відмінності.

66.Як працює CSS Grid?

67.Які формати зображень підтримують анімацію?

68.Як відстежити прогрес / закінчення CSS @keyframes анімацій або плавних переходів, реалізованих за допомогою transition, у JS?

69.Які CSS-властивості можуть бути оброблені безпосередньо через GPU? Що таке композитні шари й чому їх велика кількість може призвести до аварійного завершення роботи браузера на мобільних пристроях?

70.Як перевикористати інлайнові SVG-елементи на сторінці?

71.Опишіть способи оптимізації SVG-файлів.

72.Як реалізувати іконковий шрифт з певного набору SVG-файлів?

73.Що таке хибне жирне або хибне курсивне (Faux) накреслення шрифтів?

74.Що таке #shadow-root в інспекторі HTML-сторінки?

75.Навіщо потрібні Custom Elements?

76.Чому видалення зайвих символів пробілів / символів перенесення в HTML не відбивається на кінцевій продуктивності завантаження сторінки?

77.Що таке контекст відображення canvas? Які існують типи контексту для рендерингу двовимірної та тривимірної графіки?

Angular

78.Як працює Dependency injection? Навіщо це потрібно? Розкажіть про використання кастомних інжекторів.

79.Що таке zone.js? Для чого Angular використовує зони? З якою метою можна використовувати NgZone-сервіс?

80.Як працює Change detection? Як можна оптимізувати компонент за допомогою схем Change detection? Які ще прийоми є для оптимізації рендерингу (пов’язані з Change detection)?

81.Як виконати конфігурацію HTTP-сервісу? Навіщо вона потрібна? Обробка HTTP-помилок?

82.Які є підходи до організації роботи з даними?

83.Як підготувати збірку до деплою?

84.Що таке NgRx? Коли варто використовувати?

85.У яких випадках краще використовувати Renderer-сервіс замість нативних методів? І навпаки?

86.Як працюють і для чого потрібні резолвери? Як отримати дані, завантажені резолверами?

87.Як працюють і навіщо потрібні динамічні компоненти? Наведіть приклади їх доцільного використання.

88.Яка різниця між @ViewChild і @ContentChild?

89.Що робить код і як інакше можна пов’язати клас компонента зі змінною?

@HostBinding('class.valid') isValid;

90.Як можна кешувати дані, використовуючи сервіси або RxJS?

91.Що таке асинхронна валідація форм? Коли застосовується і як реалізується?

92.Навіщо потрібна forRoot-функція модуля?

93.Яка різниця між декларуванням і експортом компонента з модуля?

94.Чому погано «провайдити» сервіс з shared-модуля в lazy-loaded модуль? (Питання про scope модулів.)

95.Що таке ::ng-deep і для чого використовується?

96.Які тести можна запустити для Angular-програми? Які інструменти використовують для тестування Angular-програми?

97.Як протестувати API-сервіс?

React

98.Що таке JSX? Що лежить в його основі?

99.Як працює алгоритм Virtual DOM?

100.Для чого потрібна властивість key під час рендерингу списків?

101.У чому різниця між функціональними та класовими компонентами?

102.Навіщо і коли потрібно передавати props в super () при використанні класових компонентів?

103.Чому потрібно використовувати setState () для оновлення внутрішнього стану компонента?

104.У чому полягає принцип «підйому стану»?

105.Які бібліотеки менеджменту стану React-застосунку ви знаєте? Навіщо вони?

106.Коли варто використовувати Redux? Які є альтернативи?

107.Redux vs Mobx?

108.Розкажіть про базовий принцип роботи React Hooks.

109.У чому різниця між createRef і useRef?

110.Коли варто використовувати React refs? Коли не варто?

111.Які недоліки бібліотеки React бачите?

112.Які патерни використовуєте разом з React?

113.Як ставитесь до типізації разом з React?

114.Як побудувати хорошу архітектуру React-проєкту?

115.Оптимізація React-застосунків? Як виміряти продуктивність програми?

116.Чи можна застосунок на React вбудувати в інший застосунок на React?

Back-end

117.Чому Node.js однопотоковий, а не багатопотоковий?

118.Що таке event driven development?

119.Порівняйте fork() та spawn() методи.

120.Розкажіть про Node.js фреймворки, які використовували. Яка між ними різниця?

121.Опишіть словами код ендпоїнта, який повинен зберегти з клієнта файл розміром 4 гігабайти і покласти його на S3 або інший CDN.

122.Що таке мікросервіси, навіщо їх використовують?

123.У яких випадках ви б обрали моноліт, а в яких — мікросервіси?

124.Як зрозуміти, що застосунок у певний момент працює справно?

125.Як зрозуміти, що застосунок за останні три дні працював справно?

126.Як відбувається перевірка правильності паролю при використанні bcrypt?

127.Що таке JWT?

128.Джуніор надіслав код на рев’ю. Що тут не так? Як виправити?

router.post ( '/ users', async (req, res, next) => {

const user = await db.createUser (req);

if (user) {

return res.json (users);

}

res.json ({error: "can not create user"})

})

Бази даних

129.Що таке Redis і для чого його використовують?

130.Які бази даних використовували? Яка різниця між SQL і NoSQL?

131.Для двох таблиць — коментарі та користувачі — напишіть запит, який вибирає останні три коментарі для кожного користувача.

132.Я як замовник прошу вибрати вас базу даних для нового проєкту. Ваші дії?

Інструменти та інше

133.Для чого потрібен package-lock.json?

134.У чому різниця між npm install і npm ci?

135.Для чого потрібні бандлери?

136.Розкажіть про модульне підключення скриптів. Наведіть приклад використання завантажувальників / бандлерів модулів.

137.У чому полягає різниця між git merge і git rebase?

138.Що таке staging area в git?

139.Опишіть процес code review. Назвіть основні правила, способи вирішення конфліктів і суперечок під час його проведення.

Практичні завдання

140.Напишіть функцію Sleep (ms), яка зупиняє виконання async-функції на заданий проміжок часу.

141.Реалізуйте один з методів масиву (наприклад, splice).

142.Напишіть функцію з RegExp для знаходження всіх HTML-посилань у рядку.

143.Реалізуйте функцію, яка виконає callback для всіх елементів певної гілки DOM-дерева.

144.Реалізуйте таблицю з віртуальним скролом.

145.Реалізуйте функцію перетворення URL query рядка в JSON.

const inData = "user.name.firstname=Bob&user.name.lastname=Smith&user.favoritecolor=Light%20Blue";

function queryObjectify(arg) {

// ??

}

queryObjectify(inData)

/\* Результатом виконання для вхідного рядка, повинен бути наступний об’єкт

{

'user': {

'name': {

'firstname': 'Bob',

'lastname': 'Smith'

},

'favoritecolor': 'Light Blue'

}

};

\*/

146.Реалізуйте функцію знаходження перетину двох масивів.

const first = [1, 2, 3, 4];

const second = [3, 4, 5, 6];

function intersection (a, b) {

// ??

}

intersection(first, second) // -> [3, 4]

147.Реалізуйте функцію / клас для генерації HTML.

const HTMLConstruct = {};

HTMLConstruct.span('foo'); // -> <span>foo</span>

HTMLConstruct.div.span('bar'); // -> <div><span>bar</span></div>

HTMLConstruct.div.p(

HTMLConstruct.span('bar'),

HTMLConstruct.div.span('baz')

); // -> <div><p><span>bar</span><span>baz</span></p></div>

148.Якщо є проєкт з обмеженими термінами та некритичною продуктивністю, чим будете керуватися при виборі бібліотек, підходів? Чи все ж будете звертати увагу на продуктивність? Або навпаки: терміни нелімітовані, продуктивність важлива. Ваші дії?

Senior

Загальні

1.Розкажіть про функціональне програмування.

2.Що таке TDD (Test Driven Development) / BDD (Behavior Driven Development)?

3.Розкажіть детально про роботу HTTPS.

4.Який стек технологій можна обрати для реалізації клону якогось відомого проєкту і чому?

5.Є проєкт на старих технологіях, необхідно в нього вносити зміни. Як це зробити найкраще?

6.Якщо у кандидата є досвід роботи з кількома фреймворками: який будете використовувати для наступного проєкту? Які фактори будуть впливати на вибір?

7.Що таке V8 Engine?

JS Core

8.Реалізація патерну Class Free OOP (HTTPs://observablehq.com/@bratter/class-free-oop).

9.Патерн async disposer (HTTPs://advancedweb.hu/what-is-the-async-disposer-pattern-in-javascript).

10.Використання регулярних виразів. Коли прийнятно / неприйнятно? Як вони працюють? Як можна зробити читабельним код?

Front-end

11.Як браузер визначає, чи ми можемо спілкуватися між вкладками?

12.Що таке Content Security Policy?

13.Як уникнути завантаження кешованих файлів скриптів і стилів?

14.Що таке requestAnimationFrame?

15.Розкажіть про мікросервісну архітектуру Front-end App.

16.Що таке Shadow DOM?

17.Порівняйте nextElementSibling та nextSibling.

18.Які знаєте метрики вебсайту?

Angular

19.Як проводиться конфігурація NgZone-модуля? Коли це необхідно?

20.Що дратує у фреймворку? Що б ви змінили?

21.Якби ви вирішували, що додати в наступному релізі фреймворку, яка б фіча це була?

22.Чи писали кастомні декоратори? Якщо так, то навіщо?

23.Зробіть рев’ю коду та дати зауваження щодо архітектури.

24.Розкажіть, як би ви робили такі фічі. Опишіть архітектуру фічі в застосунку.

Back-end

25.Порівняйте Common.js з AMD Modules та ES6 Imports.

26.Який фреймворк вибрали б для бекенду, чому?

27.Опишіть своїми словами, як працює OAuth v2.

28.Є проєкт з джерелами витоку пам’яті, як їх виявити, усунути та запобігти цьому в майбутньому?

29.Є проєкт з performance issues, як їх виявити, усунути та запобігти в майбутньому?

Бази даних

30.Які альтернативні види баз даних використовуєте?

31.Що таке RDS і чому він іноді не підходить?

32.Що таке SQL Injections і як їх уникнути?

33.Чому для запитів у БД треба використовувати плейсхолдери?

34.Як спроєктувати кластер на MongoDB?

35.Для чого використовують MongoDB Aggregation framework?

36.Розкажіть про GraphQL.

Інструменти

37.Чи можете описати суть методології git flow двома словами?

38.Що означає вимога робити squash commits під час rebase?

39.Яка ваша думка про альтернативні системи контролю версій (Version Control System)?

40.Які конвенції знаєте і використовуєте для git?

41.Розкажіть про свій досвід використання / впровадження СI/CD.

42.Потрібно налаштувати деплой проєкту на кілька середовищ. Розкажіть, як би ви побудували процес? Які інструменти використали б?

Практичні завдання

43.Реалізуйте асинхронний метод filter для Array (повинні працювати await).

44.Реалізуйте функцію reduce за допомогою рекурсії.

45.Як можна було б зробити toggle-компонент, як в iPhone, без використання JS?