

# Співбесіда з Front-end розробником. 400+ запитань для Junior, Middle і Senior

За [даними](#) DOU, Front-end — одна з найпопулярніших категорій вакансій для фахівців з досвідом від 3 років. Медіанні зарплати у фронтендерів, [за результатами цього річного опитування](#), становили \$2900.

Ми зібрали список запитань для проходження співбесіди на позицію Front-end Developer рівня Junior, Middle та Senior. Для підготовки матеріалу ми поспілкувалися зі спеціалістами компаній, що працюють на українському ринку. Тож цей список містить різні погляди на проведення інтерв'ю. Вважайте його певним орієнтиром у підготовці, але пам'ятайте, що у кожній компанії відштовхуються від власних потреб і специфіки роботи. Попередні статті з рубрики шукайте за тегом [100 техпитань](#).

👉 Триває зарплатне опитування DOU та рейтинг мов програмування. [Встигніть взяти участь!](#)

## 🎓 Junior Front-end Developer

### Загальні питання

1. Чи стежите ви за новинами у світі фронтенду? Як саме?
2. Що відрізняє хороший код від поганого?
3. Які HTTP-методи та статуси знаєте?
4. У чому небезпека виконання коду, що надходить від користувача?
5. Що таке ООП?
6. Яка різниця між класом і абстрактним класом?

## Основи HTML/CSS

7. Що таке DOCTYPE і навіщо його використовують?
8. Які основні теги структури HTML-сторінки?
9. Що таке семантичні теги та навіщо вони потрібні? Наведіть приклади таких тегів.
10. Що можете сказати про тег `<br>`?
11. Які теги з найсвіжіших оновлень ви знаєте?
12. У чому різниця між `<div>` і `<span>`.
13. Що таке data-атрибути, в яких випадках їх використовуєте?
14. Як використовують атрибути `class` та `id`? Яка різниця між класами та ідентифікаторами?
15. Що таке внутрішні та зовнішні гіперпосилання і які вони мають атрибути?
16. Що ви знаєте про SVG? Які є варіанти додавання SVG на сторінки сайту? Чим вони відрізняються?
17. Як створити форму в HTML і забезпечити її валідацію?
18. Які є типи списків у HTML? Що таке `ul` / `ol` / `dl`?

19. Що таке `iframe` і для чого його використовують?
20. Які методи відправки даних форми ви знаєте?
21. Які основні відмінності між блоковими та вбудованими елементами в HTML? Як вони впливають на структуру сторінки?
22. Які CSS-методології ви знаєте та які використовуєте самі? Обґрунтуйте вибір нинішньої методології.
23. Що таке CSS-селектори? Наведіть приклади.
24. Як працює наслідування в CSS?
25. Що таке «модель коробки» в CSS і як її можна змінити?
26. У чому різниця між `margin` і `padding`?
27. Що таке Flexbox? Як реалізувати верстку за допомогою Flexbox чи Grid і який підхід найбільш доречний в якому випадку?
28. Що таке медіазапити та як їх використовувати?
29. Як можна приховати елемент за допомогою CSS? Чи однаково будуть працювати `opacity: 0`; `display: none`; `visibility: hidden`?
30. Як можна зробити лого посередині екрана?
31. Як розмістити хрестик для закриття справа зверху елемента?
32. Як зробити останній елемент списку червоним, незалежно від довжини списку без JS?
33. Які є можливості задати колір?
34. Що таке `z-index`? Чи кожен елемент його має?
35. Які переваги і недоліки використання CSS-препроцесорів?

36. Які методи центрування елемента в CSS?
37. Які методи вертикального центрування в CSS ви знаєте?
38. Для чого існують різні рівні заголовків, якщо розмір можна встановити стилями?
39. Які є варіанти позиціонування елемента на сторінці?
40. Які бувають види display? Який використовуєте найчастіше? У чому особливість inline-block?
41. У чому відмінність між position: relative та position: absolute ?
42. Які одиниці розмірів є в CSS?
43. Які ризики використання від'ємних значень для відступів?
44. Що таке псевдокласи та псевдоелементи? В чому між ними різниця? Як ви використовуєте псевдокласи та псевдоелементи в CSS?
45. Що таке адаптивний дизайн?
46. Яка різниця між картинками, доданими через HTML-тег <img>, і CSS-властивістю background-image?
47. Як ви вирішуєте проблеми з кросбраузерною сумісністю в розробці вебсайтів, зокрема в контексті різних версій браузерів?
48. Як із врахуванням HTML і CSS забезпечити доступність сайту?
49. Як реалізувати responsive design, використовуючи CSS?
50. Поясніть концепцію CSS-змінних.

- 51. Які властивості CSS впливають на швидкодію?
- 52. Що зазвичай найбільше знижує показники швидкодії сайту?
- 53. Як зробити дві колонки на CSS так, щоб одна розтягувалася, а інша — ні?
- 54. Які плюси та мінуси CSS-анімацій порівняно з JS-анімаціями?
- 55. Як можна перебити important або досягти схожого ефекту без important?
- 56. Чи валідний цей HTML? виправте всі помилки. Розкажіть про валідність HTML загалом.

```
<ul>  
<a href="/some-url"><li>Some Link in Menu</a>  
</li>  
<a href="/some-url">  
<li>Some Link in Menu</li></a>  
</ul>
```

- 
- 57. Що таке mobile-first підхід, які його переваги та недоліки?
  - 58. Поясніть, як обчислюється специфічність у CSS-селекторах.
  - 59. Для чого використовують пре-/постпроцесори?
  - 60. Назвіть типи інпутів у формах. Як відправити форму на сервер (без JS)?

## Web Performance

- 61. Що ви знаєте про Core Web Vitals?

- 62. Чим відрізняються лабораторні та польові заміри перформансу?
- 63. Які формати картинок знаєте, чим вони відрізняються?
- 64. Ваш сайт довго завантажується на мобільному пристрої. Розкажіть крок за кроком, як будете шукати причину.
- 65. Які інструменти використовуєте для оптимізації швидкості завантаження вебсторінки у роботі з HTML і CSS?
- 66. Як ви оптимізуєте роботу зображень на вебсторінці для покращення продуктивності?
- 67. Які методи оптимізації вебсторінок ви знаєте?

## Основи JavaScript

- 68. Перерахуйте типи даних у JS.
- 69. Чим відрізняється масив та об'єкт? У яких випадках що використовуєте?
- 70. Events в JS. Розкажіть про event propagation, bubbling, delegation, preventing тощо.
- 71. У чому різниця між var, let і const. У яких випадках що потрібно використовувати?
- 72. Чи можливо перевизначити const?
- 73. Чи відомі принципи DRY, KISS, YAGNI ?
- 74. У чому відмінності LocalStorage від SessionStorage? Як ви використовуєте LocalStorage та SessionStorage в JavaScript? У яких випадках вважаєте їхнє використання доцільним?

75. Що таке замикання (closure) в JavaScript і як воно працює?
76. Що таке область видимості змінної?
77. Що таке анонімна функція?
78. Що таке стрілкова функція і як вона відрізняється від звичайної функції?
79. Які методи перебору масивів ви знаєте?
80. Що таке JSON і як працювати з ним у JavaScript?
81. Як виконується асинхронний код у JavaScript?
82. Як ви працюєте з асинхронним кодом у JavaScript без використання `async/await`? Які інші підходи ви розглядаєте для роботи з асинхронністю?
83. Різниця між `==` і `===`.
84. Що таке прототип у JavaScript?
85. У чому різниця між `null` та `undefined`?
86. Що таке контекст? Як можна змінювати контекст виконання функції?
87. Що таке Promise? У яких випадках ви ним користуєтесь?
88. Які особливості об'єктноорієнтованого програмування у JavaScript?
89. Як з JS-коду звернутись до HTML-елемента та змінити його текст?
90. Що таке деструктуризація?
91. Що буде, якщо звернутись до елемента, якого немає за індексом (`const arr = ['a', 'b']; console.log(arr[20])`)?
92. У чому різниця між `for..of` та `for...in`?

# TypeScript

93. Які основні відмінності між TypeScript і JavaScript?
94. Які переваги та недоліки використання TypeScript?
95. Що таке типи даних у TypeScript і як їх використовують?
96. Розкажіть про інтерфейси в TypeScript і наведіть приклади їх використання.
97. Як працюють Generics у TypeScript і чому вони корисні?
98. Як можна використовувати Enums у TypeScript і які їхні переваги?
99. Як TypeScript покращує безпеку типів проти JavaScript?
100. Які можливості для рефакторингу та підтримки коду надає TypeScript?
101. Що таке Decorators у TypeScript і в яких випадках їх використовують?
102. Як у TypeScript використовують простори імен і модулі?
103. Які стратегії міграції з JavaScript на TypeScript ви знаєте?

## API браузера

104. Що таке DOM?
105. Як JavaScript взаємодіє з DOM?
106. Що таке AJAX?



- 107. Як виконати HTTP-запит за допомогою JavaScript?
- 108. Що таке JSONP і як його використовують?
- 109. Які є способи зберігання даних у браузері?
- 110. Що таке обробка подій у JavaScript?
- 111. Що таке event bubbling та event capturing у контексті обробки подій у DOM?  
Як використовуєте ці концепції для ефективною обробки подій?
- 112. Що таке делегування подій у DOM,  
як ви використовуєте цей підхід для оптимізації обробки подій, особливо на великих сторінках?

## **Фреймворки та бібліотеки**

- 113. Навіщо використовують фреймворки, які їхні переваги та недоліки?
- 114. Чим варто керуватися у виборі фреймворку?
- 115. Що таке компонентно-орієнтований підхід?
- 116. У чому проблема керування станом і які є підходи до її вирішення?
- 117. Що таке віртуальний DOM?
- 118. Які методи життєвого циклу компонентів ви знаєте?
- 119. Як оптимізувати продуктивність рішень з використанням фреймворків?
- 120. Що таке рендеринг на стороні сервера? Навіщо він потрібен, які бувають варіанти реалізації?
- 121. Як взаємодіяти з API в межах бібліотек/

фреймворків?

122. Які особливості управління даними та маршрутизації?
123. Які основні відмінності між Angular і React?
124. Як ви розрізняєте класові та функціональні компоненти в React? В яких випадках вважаєте за доцільне використовувати той або інший тип компонента?
125. Як можна оптимізувати продуктивність у React-застосунках? Які підходи використовуєте для уникнення надмірного рендерингу та для оптимізації часу завантаження компонентів?
126. Як реалізуєте управління станом у React-застосунках? Які підходи використовуєте для підтримки читабельності та ефективного управління більш складним/глобальним станом?
127. Як вирішуєте питання управління формами в React? Які підходи використовуєте для валідації та обробки форм?
128. Як виконати код функції лише при першому рендері компонента?
129. Як обробляєте події у React?
130. Що таке React-хуки, які вбудовані хуки використовуєте?
131. Яким умовам має відповідати функція, щоб бути хуком?
132. Як ви робите серверний рендеринг у Next.js? Які переваги цього підходу бачите, у яких випадках вважаєте його найбільш доцільним?

- 133. Що таке компоненти у Vue.js?
- 134. Як працює директива ngIf в Angular?
- 135. Є завдання порахувати коефіцієнт, який буде потрібен у компоненті і який залежить тільки від декількох констант у проєкті. Де його найкраще порахувати?

## Основи мережі та протоколи

- 136. Що таке HTTP і HTTPS, яка між ними різниця?
- 137. Які основні методи HTTP-запитів (GET, POST, PUT, DELETE) і їхні відмінності?
- 138. Що таке статус-коди HTTP, наведіть приклади найбільш поширених.
- 139. Що таке REST API і які його основні принципи?
- 140. Які засоби безпеки використовують під час роботи з вебзапитами?
- 141. Що таке JSON і XML і чим вони відрізняються у використанні?
- 142. Як працює механізм CORS (Cross-Origin Resource Sharing)?
- 143. Що таке WebSocket і в чому його особливості порівняно з традиційними HTTP-запитами?
- 144. Як функціонує кешування вебресурсів?
- 145. Що таке SSL/TLS і як це використовують для реалізації безпеки?

## Робота з бекендом

- 146. Що таке API і як фронтенд взаємодіє з ним?

147. Які є формати обміну даними між фронтом і бекендом?
148. Що таке CRUD-операції і як вони реалізуються на практиці?
149. Як використовувати вебсокети для реалізації реального часу комунікацій?
150. Які засоби аутентифікації та авторизації використовують у сучасних вебзастосунках?
151. Які основні принципи безпечного зберігання та передачі користувацьких даних?
152. Що таке GraphQL і в чому його переваги та недоліки порівняно з REST?
153. Які стратегії кешування даних можуть бути використані у фронтенді?
154. Як можна обробляти великі обсяги даних без негативного впливу на продуктивність фронтенду?
155. Які інструменти і підходи використовують для моніторингу та відлагодження запитів до сервера?

## **Алгоритми та структури даних**

156. Що таке алгоритмічна складність і як її оцінювати?
157. Розкажіть про основні структури даних, такі як масиви, списки, стеки та черги.
158. Які алгоритми сортування ви знаєте і як вони працюють?
159. Як реалізувати пошук у ширину та пошук

у глибину на графі й деце потрібно на фронтенді?

- 160. Що таке рекурсія, наведіть приклади її використання.
- 161. Як можна оптимізувати алгоритми за допомогою мемоізації?
- 162. Які алгоритми пошуку ви знаєте і як вони відрізняються один від одного?
- 163. Як працює хеш-таблиця та які її переваги та недоліки?
- 164. Розкажіть про алгоритми роботи з деревами, такі як двійкове дерево пошуку.
- 165. Які алгоритми можна використовувати для пошуку найкоротшого шляху в графі?

## **Git і системи контролю версій**

- 166. Що таке система контролю версій і навіщо її використовують?
- 167. Які основні команди Git ви знаєте і для чого їх використовують?
- 168. Яка різниця між git merge та git rebase?
- 169. Що таке commit у Git і як його правильно формувати?
- 170. Як виправити помилки в попередніх комітах?
- 171. Що таке гілки в Git і як їх ефективно використовувати?
- 172. Як створити нову гілку в Git?
- 173. Які стратегії вибору віток використовують у різних розробницьких процесах (наприклад, Git Flow)?

- 174. Як використовувати теги в Git?
- 175. Що таке конфлікт злиття і як його вирішити?
- 176. Які найкращі практики роботи з віддаленими репозиторіями?
- 177. Як використовувати .gitignore файл?

## Тестування

- 178. Чиєю відповідальністю є баг на продакшні — розробника чи тестувальника?
- 179. Як ви тестуєте свій код? Чи писали код, який не можна було перевірити до деплою?
- 180. Що таке AAA в тестуванні?

## Accessibility

- 181. Що таке вебдоступність?
- 182. Як тестувати вебдоступність?
- 183. Назвіть базові техніки створення доступних вебінтерфейсів.
- 184. В яких випадках треба використовувати посилання, а в яких — кнопки?
- 185. Який текст має міститися в атрибуті alt?



## Middle Front-end Developer

### Загальні запитання

- 1. Назвіть принципи ООП.
- 2. Поясніть кожну складову SOLID.

3. Чому треба звертати увагу на складність алгоритму?
4. На які сутності ви розділяєте файлову структуру на початку проєктування?
5. Як ви розумієте, що настав час відокремити певну сутність у проєкті?
6. Поверхово опишіть, як відбувається білд проєкту?
7. Яка ваша улюблена технологія/інструмент/тег у розробці?
8. Що ви робите, щоб код було легше підтримувати?
9. Що таке рефакторинг і як ви його проводите?

## HTML/CSS

10. Що таке мікророзмітка?
11. Що таке SEO? На що має звертати увагу розробник для забезпечення гарних SEO-показників у майбутньому?
12. Які техніки організації іконок на сайті ви знаєте? Назвіть плюси та мінуси кожної.
13. Як можна оптимізувати CSS для кращої продуктивності?
14. Що таке «сучасний CSS», на вашу думку? Що із сучасного CSS ви використовуєте в щоденній роботі?
15. Чим відрізняються змінні в SCSS і CSS Custom Properties? У чому перевага тих чи інших?
16. Як ви вирішуєте конфлікти стилів при роботі з декількома фронтендерами на проєкті?

чи запобігаєте цьому?

17. У чому різниця між transition і animation? В яких випадках що використовуєте?
18. Потрібно додати кастомні стилі для скролбару. Як би ви це зробили?
19. Які методи ізоляції стилів ви знаєте?
20. Навіщо потрібні CSS Logical Properties? Чи використовуєте їх?
21. Розкажіть про свій досвід використання CSS Grid Layout.
22. Як працює CSS-функція clamp()? Назвіть корисні приклади її використання.
23. Як оптимізувати стилі для друку на принтері?
24. Які CSS-фреймворки ви знаєте? З якими з них працювали на проєктах? Які плюси та мінуси вони мали?
25. Як ви працюєте з адаптивними та резиновими макетами? Як розглядаєте проблеми, пов'язані з різними роздільними здатностями екранів і пристроями?
26. Як організовуєте CSS-стилі у великих проєктах? Чи використовуєте які-небудь методології?
27. Як обираєте між використанням і не використанням препроцесорів CSS, таких як Sass чи Less, у своїх проєктах? Які переваги ви бачите у цьому?
28. Як впроваджуєте концепцію «компонентного» підходу у вашому CSS-коді? Які стратегії використовуєте для створення повторно



використовуваних і легко модифікованих компонентів у ваших стилях?

29. Як семантична верстка впливає на доступність сторінки?
30. На що впливає doctype?
31. Які препроцесори CSS ви знаєте і які їхні переваги?
32. Як зробити посилання, з якого можна завантажити прикріплений файл?
33. Чому іноді translate краще за position:absolute під час написання стилів?
34. Назвіть особливості та відмінності flex і grid? Чому віддаєте перевагу?
35. Дублювання HTML/CSS для responsive design: плюси, мінуси, підводні камені.

## Performance

36. Які основні метрики вимірюють продуктивність сайту?
37. Що таке Cumulative Layout Shift і як покращити його значення?
38. Що таке Critical CSS?
39. Як ви розробляєте анімації з погляду продуктивності, зокрема при використанні CSS або JavaScript?
40. Які формати зображень краще використовувати у вебi? Які методи оптимізації зображень для вебу?

41. Як управляєте інтеграцією сторонніх бібліотек або плагінів для зменшення впливу на продуктивність?
42. Як реалізуєте lazy loading для зображень та інших ресурсів для оптимізації часу завантаження?
43. Як використовуєте prefetching і preloading ресурсів для підвищення швидкості переходу між сторінками?
44. У вас є сторінка галереї, на якій потрібно показати велику кількість відео з YouTube.  
Як би ви це реалізували так, щоб сторінка швидко завантажувалась?
45. Вам потрібно відобразити таблицю на кілька тисяч рядків. Як би ви це втілили, щоб сторінка швидко завантажувалась і швидко реагувала на дії користувача?
46. Як ви б організували завантаження великої кількості CSS-стилів на сторінці так, щоб користувач міг якнайшвидше взаємодіяти з нею, при цьому уникнувши зсуву макета?

## JavaScript

47. У чому різниця між LocalStorage, SessionStorage та cookies?
48. Коли використовуєте Map та Set?
49. Які знаєте способи, щоб виявити, чи перебуває елемент у viewport користувача?
50. Що таке Service Workers, у яких випадках

їх використовуєте?

51. Що таке Websockets? Для чого вони потрібні?
52. Як обробляєте помилки в застосунку?
53. Що таке event loop? Як воно працює?
54. У чому різниця між типами та інтерфейсами?
55. Що таке модифікатори доступу?
56. Що таке type guard?
57. Яка мета створення abstract class?
58. Які види об'єктних моделей ви знаєте?
59. Що таке поліфілі й для чого вони потрібні?
60. Для чого використовують анонімні функції?
61. Що таке ES6-модулі?
62. Як використовуєте модульність у JavaScript для організації вашого коду?
63. Які підходи використовуєте для уникнення конфліктів та забезпечення читабельності та масштабованості коду?
64. Як ви оптимізуєте та працюєте з анімаціями в браузері? Які підходи використовуєте для досягнення високої продуктивності та плавності анімацій?
65. Як реалізовуєте lazy loading великих ресурсів, таких як зображення або скрипти, для оптимізації завантаження сторінок? Як це взаємодіє з асинхронним кодом?
66. Як використовуєте інструменти забезпечення якості коду, такі як ESLint? Як визначаєте та дотримуетесь кодового стилю в проєкті?
67. Як ви працюєте з архітектурою та патернами

проєктування в JavaScript? Які конкретні патерни використовуєте для структурування коду та полегшення його розширення?

- 68. Як би ви реалізовували відсотковий прелоадер на сайті, де треба завантажувати картинки, 3D-об'єкти та додаткові JSON-файли?
- 69. Чи є у вас досвід з PWA?
- 70. Що таке функції-генератори і яке їх практичне застосування?
- 71. Розкажіть про всі способи використання оператора '...'

## **Фреймворки та бібліотеки**

- 72. Розкажіть про управління станом у Vue.js.
- 73. Як реалізувати маршрутизацію (routing) в React?
- 74. Як оптимізувати рендеринг компонентів у React?
- 75. Як оптимізувати продуктивність у великих React-застосунках?
- 76. З якими проблемами в React-компонентах часто стикались?
- 77. Як ви управляєте станами в React APP?  
Як передавати стани між кількома компонентами?
- 78. Які підходи використовуєте для lazy loading компонентів, управління станом і кешування даних?
- 79. Що таке side effects? З якими інструментами є досвід?

80. Як вирішуєте проблеми маршрутизації в складних React-застосунках? Які підходи використовуєте для організації маршрутів та управління структурою застосунку?
81. Як обробляєте анімації в React?  
Як використовуєте CSS-анімації та бібліотеки для створення плавних і високопродуктивних анімацій?
82. Як використовуєте архітектурні патерни (наприклад, Flux або Redux) для ефективного управління станом у React-застосунку? Які переваги та обмеження бачите в застосуванні цих патернів?
83. Як оптимізуєте та масштабуєте Next.js застосунки для високого трафіку? Як вирішуєте питання кешування, CDN, інших аспектів продуктивності?
84. Що таке ре-рендер компонента, коли він відбувається чи можна вплинути на цей процес?
85. Що таке React Context, коли його використовуєте?
86. Що таке мемоїзація в React та коли вона потрібна?
87. Як виконати код перед видаленням компонента з дерева?
88. У яких випадках використовуєте `useReducer`?
89. Які UI-бібліотеки знаєте, у чому їхні переваги та недоліки?

## Тести

90. Для чого впроваджують тестування?
91. Чи є випадки, коли тестування не є доцільним?
92. Які види тестування знаєте?
93. У чому різниця між e2e та unit-тестами?
94. Які інструменти для автоматизації тестування знаєте?
95. Як тестувати RESTful API?
96. Що таке TDD?
97. Як обираєте, чи покривати функціонал тестами?
98. Як виявляєте та ізолюєте помилки в коді? Які інструменти для відладки використовуєте?

## **Accessibility**

99. Які рівні доступності знаєте? Чим вони відрізняються?
100. Опишіть особливості імплементації навігації по сайту з клавіатури.
101. Для чого використовують HTML5 Landmarks?
102. Які проблеми з вебдоступністю трапляються в SPA-застосунках найчастіше?
103. Дизайнер консультується з вами щодо дизайну майбутнього проєкту. Які поради в контексті доступності ви йому дасте?



## **Senior Front-end Developer**

### **Загальні запитання**

1. Якими необхідними знаннями та навичками

- потрібно володіти Senior-розробнику?
2. Якою має бути робота та комунікації в команді розробників?
  3. Які методи оцінки та планування проєктів використовуєте?
  4. На що звертаєте увагу під час вибору технологій на проєкті?
  5. Як підходите до менторства та розвитку менш досвідчених розробників?
  6. Як впроваджуєте нові технології або практики в команду?
  7. Як вирішуєте конфлікти та проблеми в команді?
  8. Як буде виглядати фронтенд-розробка за два-три роки?
  9. Які з нових фреймворків ви пробували використовувати?
  10. Які маєте пет-проєкти, можливо, займаєтеся опенсорсом?
  11. Чи виступаєте на конференціях/мітапах?
  12. Чи ділитеся десь регулярно професійними думками, в яких спільнотах берете участь?
  13. Які порекомендуєте подкасти/канали/авторів?
  14. Які нові інструменти ви почали використовувати і можете порекомендувати?

## **HTML/CSS**

15. Які ви знаєте системи іменування класів?  
Чи взагалі використовуєте якісь?

16. Як збираєте CSS?
17. Чи доводилось працювати з Web Components?  
Що таке Shadow DOM?
18. Чому деякі теги не мають псевдоелементів,  
наприклад `img`?
19. Чи є місце для CSS-властивості `float` у сучасному  
вебі?
20. Що таке адаптивна типографіка? Які техніки  
ви знаєте та використовуєте на своїх проєктах?
21. Розкажіть про ваше бачення сучасного  
адаптивного вебдизайну: які існують  
альтернативи для `media queries`. Обґрунтуйте  
вибір одиниць виміру для брейкпоінтів  
та їх значення.
22. У чому відмінність Node Sass від Dart Sass та які  
особливості міграції?
23. Назвіть відмінності між `@import`, `@use` та `@forward`  
в Sass.
24. Чи створюєте ви стайлгайди для своїх проєктів?  
Які техніки або інструменти використовуєте для  
цього?
25. Які є альтернативи для `overflow: hidden`?
26. Дизайнер наполягає на кастомізації стилів для  
всіх елементів складної форми. Які ваші дії?
27. Які є кейси використання `@media` в CSS, окрім  
`screen`?
28. Як вирішуєте проблеми з високим обсягом стилів  
і підтримкою великих кодових баз у великих  
проєктах? Які підходи та методології



використовуєте для організації CSS-коду на рівні архітектури?

29. Як оптимізуєте завантаження сторінок, зокрема використання Critical CSS та інших стратегій для зменшення часу до першого байту та часу до відображення?
30. Розкажіть про ваш досвід використання препроцесорів CSS (наприклад, Sass або Less) та інших інструментів для автоматизації завдань під час розробки великих проєктів.
31. Як вирішуєте завдання кросбраузерної сумісності на етапі розробки та як уникнути непорозумінь між розробниками та QA-командою під час тестування на різних браузерах?
32. Як реалізуєте server-side rendering (SSR) або static site generation (SSG) із CSS-in-JS, перед якими викликами ви можете постати в цьому контексті?
33. Яку нову фічу в CSS (з тих, що зараз перебувають на стадії розробки або імплементації в деяких браузерах) ви очікуєте найбільше? Чому саме цю?
34. Уявіть, що ви дізналися про свій майбутній проєкт. Це великий публічний мультимовний сайт з арабською та японською мовами включно. На проєкті вам будуть допомагати два джуніор-фронтендери. Опишіть, як ви сплануєте CSS-архітектуру та підготуєте своїх джунів для такого проєкту.
35. Вам потрібно зробити галерею картинок, які мають плавно гортатись при скролі вбік усередині

галереї. Як би ви це реалізували? Чи можливо це зробити без JS?

## Performance

- 36. На умовному сайті є кастомний декоративний шрифт, який негативно впливає на продуктивність. Назвіть усі можливі варіанти його оптимізації.
- 37. Уявіть, що є старий проєкт на підтримці (не JS-стек, бек на PHP-фреймворку), який перекинули на вас з іншої команди. Завдання: максимально оптимізувати фронтенд-проєкту, вклавшись у бюджет і 16 годин. Опишіть вашу стратегію оптимізації та кроки в межах означених обмежень.
- 38. У чому особливість підготовки та під'єднання фронтенд-асетів залежно від використання HTTP1 або HTTP2 на сервері?
- 39. Чи ознайомлені ви зі складанням бюджету продуктивності (Performance Budget)? Як ви розумієте це поняття?
- 40. Що таке DOM Reflow, Repaint, Composite? Як це працює?

## JavaScript

- 41. Як паралельно пройтись по масиву Promises? А послідовно?
- 42. Що таке Symbol? Яке його практичне

застосування?

43. Наведіть приклади реалізації патерну Observer у браузері.
44. Наведіть приклади сценаріїв, які можуть призвести до витоку пам'яті у клієнтському коді.
45. У чому різниця між Websockets і Server Send Event?
46. Розкажіть про Service Workers.
47. Як працювати з WebSockets для реалізації двостороннього зв'язку?
48. Якщо якась анімація на сайті повільна (низький FPS), як дізнатися причину? Які причини бувають?
49. Які способи організації коду/патерни використовуєте і чому?
50. Які є варіанти зберігання даних на клієнті?
51. Як би ви реалізували кешування на клієнті? Коли це доречно?
52. Як ви реалізуєте аутентифікацію та авторизацію (якщо є) у вашому застосунку? Як ви дбаєте про безпеку та управління правами доступу на клієнтській та серверній стороні?
53. Як ви використовуєте code splitting та lazy loading у коді для оптимізації завантаження сторінок та ресурсів? Як це впливає на продуктивність застосунку?
54. Як ви реалізуєте автоматичне тестування під час розробки? Які інструменти використовуєте для unit-тестування, інтеграційного тестування та тестування взаємодії?

55. Є завдання передавати дані між двома різними вебсайтами. Які є способи це робити? А якщо тільки client side? Якщо ми маємо доступ до коду обох сайтів? Якщо один показується на іншому через iframe? Якщо ж ми не маємо доступу до одного з них?

## Фреймворки та бібліотеки

- 56. Що нового з'явилось у 18-й версії React?
- 57. Як тримати посилання на функцію з актуальними параметрами, але не викликати useEffect зайвий раз?
- 58. У чому різниця між useEffect та useLayoutEffect, у яких випадках що використовуєте?
- 59. Розкажіть про хук useTransition.
- 60. Які патерни з React ви знаєте?
- 61. Наведіть приклад React-патерну Compound components.
- 62. Чи потрібні бібліотеки для state management, якщо є бібліотеки для кешування (React Query/SWR) і React Context?
- 63. Як обробляєте помилки у React-компонентах?
- 64. Поясніть, що відбувається в цьому коді та в яких сценаріях він може бути корисним:

```
const useHandler = (f) => {  
  const ref = React.useRef(f);  
  ref.current = f;  
  return React.useCallback((...args) => ref.c
```

};

65. Як управляти глобальним станом у застосунках на Angular через ngrx/store?
66. Як ви підтримуєте консистентність дизайну та стилізації великих кодових баз React-проектів? Які підходи або інструменти використовуєте для підтримання єдності інтерфейсу користувача?
67. Як ви структуруєте та організовуєте компоненти великих React-застосунків? Які підходи використовуєте для логічного розділення компонентів, повторного використання та підтримки коду?
68. Як управляєте формами в React, зокрема використовуючи бібліотеки для форм, такі як Formik чи React Hook Form? Як обробляєте валідацію, маскування та інші аспекти форм?
69. У вас є завдання відрендерити великий список (100к + items). Як би виконали це завдання?
70. Оптимізація React-компонентів: що варто робити, а що ні? Чи варто і коли варто використовувати useMemo та useCallback?
71. Як збираєте React-застосунок?
72. Які інструменти використовували для state management і чому обирали саме їх?
73. Чи використовуєте інкрементальну перевірку (incremental static regeneration) в Next.js для оновлення статичних сторінок на основі часових інтервалів або змін у даних? Як це впливає

на продуктивність і кешування?

## Тестування

- 74. Чи є у вас досвід покриття великих систем тестами?
- 75. Чи є у вас досвід кастомізації тестових утиліт під потреби проєкту.
- 76. Як організовувати тестові дані та робити їх відповідними до різних сценаріїв тестування?
- 77. Як інтегрувати процеси тестування в системи неперервної інтеграції та розгортання?
- 78. Як варто автоматизовувати тестування? Які інструменти та мови скриптів використовуєте?
- 79. Як забезпечити консистентність результатів тестування в різних середовищах (розробка, тестування, продакшн)?
- 80. Як визначаєте, коли програмний продукт готовий до випуску? Які критерії «зеленого світла» для релізу використовуєте?
- 81. Чи знаєте підхід chaos engineering, якщо так, то як його можна застосувати на фронтенді?

## Алгоритми і структури даних

- 82. З якими нетривіальними структурами даних ви працювали або які знаєте?
- 83. Розкажіть про CRDT.
- 84. Які алгоритми оптимізації використовували для підвищення продуктивності великих застосунків?

85. Які алгоритми сортування вважаєте найбільш ефективними для обробки великих наборів даних?
86. Як використовуєте динамічне програмування в реальних проектах? Наведіть приклад.
87. Розкажіть про практичне використання структур даних, таких як AVL-дерева або червоно-чорні дерева.
88. Як можна реалізувати «глибоку заморозку об'єкта у JavaScript»?
89. Як можна реалізувати серіалізацію об'єкта, який містить циклічні посилання?
90. Як можна реалізувати Virtual DOM аналогічно реалізації в React?

## Архітектура

91. За якими характеристиками оберете фреймворк для нового проєкту?
92. За якими характеристиками оберете UI-бібліотеку для нового проєкту?
93. Що таке сигнали, як вони реалізовані у різних фреймворках?
94. Які переваги та недоліки монорепозіторіїв проти використання різних репозіторіїв?
95. Опишіть CI/CD-процес у нинішньому проєкті.
96. Що таке мікросервісна архітектура і її переваги порівняно з монолітною?
97. Як ви б дизайнили систему для високої

доступності та масштабування?

- 98. Які патерни проектування ви знаєте і де їх застосовувати?
- 99. Які є стратегії масштабування вебзастосунків?
- 100. Розкажіть про Single Page Application, його переваги та недоліки.
- 101. Які архітектурні патерни вебзастосунків використовували та які їхні переваги?
- 102. Як варто розробляти масштабовані та високопродуктивні фронтенд-системи?
- 103. Чи ознайомлені ви з архітектурою мікрофронтендів? Можливо, знаєте конкретні реалізації?
- 104. Як підтримувати баланс між готовими рішеннями та розробкою власних рішень у проєктах?  
Як варто ухвалювати рішення щодо вибору?

## **Безпека**

- 105. Як подбати про безпеку вебзастосунків?
- 106. Поясніть принципи Cross-Site Scripting (XSS) і Cross-Site Request Forgery (CSRF) атаки та способи захисту від них.
- 107. Які найкращі практики для забезпечення безпеки API?
- 108. Розкажіть про захист від SQL-ін'єкцій.
- 109. Що таке CORS і як його правильно налаштувати?
- 110. Як керувати секретами у проєктах?
- 111. Якби вас попросили скласти чекліст з безпеки



для проєкту, то як би він був?

## Accessibility

112. Що таке ARIA, які її основні ролі, стани та властивості?
113. Опишіть створення доступного модального вікна з нуля. Які є варіанти і на що треба звернути увагу?
114. Чи знаєте ви особливості законодавства щодо вебдоступності в країнах основних клієнтів (Штати, ЄС, УК, Канада, Україна)?
115. Уявіть, що я — клієнт, який нічого не знає про доступність і ніколи про це не замислювався. Переконайте мене, чому доступність — це важливо і чому я маю витратити на це додаткові гроші.

Дякуємо за технічну рецензію і запитання [В'ячеславу Колдовському](#) (Competence Manager у SoftServe, автор проєкту Programming Mentor) і [Вадиму Олійнику](#) (Front-end Tech Lead), а також за надані запитання [Юрію Артюху](#) (СТО у фронтенд-агенції Coderiver), [Вадиму Олійнику](#) (Front-end Tech Lead), [Олександрю Микуличу](#) (Front-end Guild Master в SKELAR), [Сергію Коржу](#) (Technical Lead у Namecheap, Inc), [Олександрю Смолянину](#) (Front-end Director в Digital Silk), [Лілії Карпенко](#) (Soft Skills Trainer, SSE, Engineering Manager, Team Lead), [Тетяні Мельник](#) (Front-end/Tech Lead

в Projector Institute), [Віктору Галочці](#) (Front-end Lead у Lift (Genesis Tech)).

Читайте також: [Front-end Developer: хто він та скільки заробляє. Аналітика DOU](#)



Все про українське IT в телеграмі — підписуйтеся на [канал DOU](#)