**  
  
Минимальные Требования Для Стажёров(JS, Node)**

**Basic CS principles**

1. Version control (Git or SVN)   
   [Система контроля версий](https://git-scm.com/book/ru/v2)
2. Command line basics  
   [Bash: Основы командной строки](https://ru.hexlet.io/courses/bash)
3. OOP basics  
   [Основные принципы ООП](https://training.by/News/Items/275?lang=ru)
4. SOLID  
   [SOLID](https://web-creator.ru/articles/solid)

**HTTP**

1. HTTP basics  
   [HTTP](https://developer.mozilla.org/ru/docs/Web/HTTP)
2. HTTP request methods  
   [Методы HTTP запроса](https://developer.mozilla.org/ru/docs/Web/HTTP/Methods)
3. Response codes  
   [Коды ответа](https://developer.mozilla.org/ru/docs/Web/HTTP/Status)
4. HTTP caching  
   [HTTP-кеширование](https://developer.mozilla.org/ru/docs/Web/HTTP/Caching)
5. HTTPS  
   [HTTPS](https://developer.mozilla.org/ru/docs/Glossary/https)
6. CORS  
   [CORS](https://developer.mozilla.org/ru/docs/Web/HTTP/CORS)
7. WebSocket  
   [WebSocket](https://developer.mozilla.org/ru/docs/Web/API/WebSocket)
8. HTTP-cookie  
   [HTTP-](https://developer.mozilla.org/ru/docs/Web/HTTP/Cookies)куки

**HTML**

1. HTML basics  
   [Основы HTML](https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/HTML_basics)
2. Meta tags  
   [Мета-теги](https://wm-school.ru/html/html_meta_tags.html)
3. Semantic layout  
   [Семантическая верстка](https://htmlacademy.ru/blog/html/semantics#:~:text=%D0%A1%D0%B5%D0%BC%D0%B0%D0%BD%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F%20%D0%B2%D1%91%D1%80%D1%81%D1%82%D0%BA%D0%B0%20%E2%80%94%20%D0%BF%D0%BE%D0%B4%D1%85%D0%BE%D0%B4%20%D0%BA%20%D1%80%D0%B0%D0%B7%D0%BC%D0%B5%D1%82%D0%BA%D0%B5,%D0%B2%D1%8B%D1%81%D1%82%D1%80%D0%BE%D0%B8%D1%82%D1%8C%20%D0%B2%20%D0%B3%D0%BE%D0%BB%D0%BE%D0%B2%D0%B5%20%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D1%83%20%D0%B4%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0.)

**CSS**

1. CSS basics  
   [Основы CSS](https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/CSS_basics)
2. Pseudo-classes  
   [Псевдоклассы](https://developer.mozilla.org/ru/docs/Web/CSS/Pseudo-classes)
3. Pseudo-elements  
   [Псевдоэлементы](https://developer.mozilla.org/ru/docs/Web/CSS/Pseudo-elements)
4. CSS preprocessor  
   [CSS препроцессор](https://developer.mozilla.org/ru/docs/Glossary/CSS_preprocessor)

**JS CORE**

1. Data types  
   [Типы данных JavaScript и структуры данных - JavaScript](https://developer.mozilla.org/ru/docs/Web/JavaScript/Data_structures)

[Типы данных](https://learn.javascript.ru/data-types)

1. Ways to declare a variable and differences between them  
   [Var, let или const.   
   Проблемы областей видимости](https://habr.com/ru/company/ruvds/blog/420359/)
2. Types casting  
   [Преобразование типов для примитивов](https://learn.javascript.ru/types-conversion)
3. Comparisons  
   [Операторы сравнения](https://learn.javascript.ru/comparison)
4. Conditional and logical operators  
   [Логический оператор](https://learn.javascript.ru/logical-ops)
5. Conditional branching: if, “?”  
   [Условное ветвление: if, '?'](https://learn.javascript.ru/ifelse)
6. Сycles while and for  
   [Циклы while и for](https://learn.javascript.ru/while-for)
7. Design “switch”  
   [Конструкция “switch”](https://learn.javascript.ru/switch)
8. Functions and ways to declare it.Functional Expression  
   [Функции](https://learn.javascript.ru/es-function)
9. Error handling  
   t[ry…catch - JS](https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Statements/try...catch)
10. Strict mode  
    Строгий режим [- JavaScript | MDN](https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Strict_mode)
11. Arrays and array methods  
    [Массивы - JavaScript | MDN](https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Global_Objects/Array)
12. Objects and methods of object  
    [Объекты- JavaScript | MDN](https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Global_Objects/Object)
13. Garbage collection  
    [Сборка мусора](https://learn.javascript.ru/garbage-collection)
14. This  
    [This](https://learn.javascript.ru/object-methods#klyuchevoe-slovo-this-v-metodah)
15. [Constructor,“new”operator](https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Operators/new)  
    [Оператор new](https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Operators/new)
16. Named Function Expression  
    [Именованное функциональное выражение](https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Operators/function#%D0%B8%D0%BC%D0%B5%D0%BD%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5_%D0%B2%D1%8B%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5)
17. Prototype: inheritance,methods  
    [Прототипы](https://developer.mozilla.org/ru/docs/Learn/JavaScript/Objects/Object_prototypes)
18. Classes, class based syntax,three pillars of OOP, class checking  
    [Классы - JavaScript | MDN](https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Classes)
19. Ways of handling asynchronous operations: callbacks,promises,async functions. Concepts of callback hell,promisification,eventloop  
    [Асинхронный JavaScript](https://developer.mozilla.org/ru/docs/Learn/JavaScript/Asynchronous/Introducing)
20. Generators concept  
    [Генераторы](https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Statements/function*)
21. Modules, import/export  
    [Модули](https://learn.javascript.ru/modules)

[Импорт](https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Statements/import)

1. Network requests:fetch, xhr  
   [FetchAPI - Интерфейсы веб API | MDN](https://developer.mozilla.org/ru/docs/Web/API/Fetch_API)

[XML HttpRequest](https://developer.mozilla.org/ru/docs/Web/API/XMLHttpRequest)

1. Closure  
   [Замыкания в JavaScript](https://developer.mozilla.org/ru/docs/Web/JavaScript/Closures)
2. Recursion  
   [Как работает рекурсия](https://habr.com/ru/post/337030/)

**Node**

1. Event loop  
   [Event loop](https://nodejs.org/en/docs/guides/event-loop-timers-and-nexttick)
2. Event emitter  
   [Event emitter](https://nodejs.dev/en/learn/the-nodejs-event-emitter/)
3. Streams  
   [Streams](https://habr.com/ru/post/479048/)
4. Process  
   [Process](https://nodejs.org/api/process.html)
5. JWT  
   [JWT](https://jwt.io/introduction)
6. Auth 2.0  
   [Auth 2.0](https://oauth.net/2/)

**Typescript**

1. Everyday Types  
   [Типы](https://www.typescriptlang.org/docs/handbook/2/everyday-types.html)
2. Generics  
   [Дженерики](https://igorfonin.gitbook.io/typescript-book-ru/typescript-type-system/generices)
3. Classes  
   [Классы](https://www.typescriptlang.org/docs/handbook/2/classes.html)
4. Utility Types  
   [Типы утилит](https://www.typescriptlang.org/docs/handbook/utility-types.html)

**Codewars**

Количество решенных задач на ресурсе [codewars](https://www.codewars.com/kata/latest?beta=false):от 50-ти (уровень 6, 5, 4).