
Оглавление

| | |
|--|-----------|
| ОТЗЫВЫ | 10 |
| ВВЕДЕНИЕ | 12 |
| От издательства..... | 17 |
| ГЛАВА 1. ЗНАКОМСТВО С TYPESCRIPT..... | 18 |
| Правило 1. TypeScript и JavaScript взаимосвязаны..... | 18 |
| Правило 2. Выбирайте нужные опции в TypeScript | 25 |
| Правило 3. Генерация кода не зависит от типов | 28 |
| Правило 4. Привыкайте к структурной типизации..... | 35 |
| Правило 5. Ограничьте применение типов any | 40 |
| ГЛАВА 2. СИСТЕМА ТИПОВ В TYPESCRIPT | 44 |
| Правило 6. Используйте редактор для работы с системой типов | 44 |
| Правило 7. Воспринимайте типы как наборы значений..... | 48 |
| Правило 8. Правильно выражайте отношение символа к пространству типов или пространству значений | 56 |
| Правило 9. Объявление типа лучше его утверждения | 62 |
| Правило 10. Избегайте оберточных типов (String, Number, Boolean, Symbol, BigInt)..... | 66 |
| Правило 11. Проверяйте пределы исключительных свойств типа | 69 |
| Правило 12. По возможности применяйте типы ко всему выражению функции | 73 |

| | |
|--|-----|
| Правило 13. Знайте разницу между <code>type</code> и <code>interface</code> | 76 |
| Правило 14. Операции типов и обобщения сокращают повторы..... | 81 |
| Правило 15. Используйте сигнатуры индексов для динамических данных..... | 90 |
| Правило 16. В качестве сигнатур индексов используйте массивы, кортежи и <code>ArrayLike</code> , но не <code>number</code> | 94 |
| Правило 17. Используйте <code>readonly</code> против ошибок, связанных с изменяемостью | 98 |
| Правило 18. Используйте отображенные типы для синхронизации значений..... | 105 |
| ГЛАВА 3. ВЫВОД ТИПОВ | 109 |
| Правило 19. Не засоряйте код ненужными аннотациями типов | 109 |
| Правило 20. Для разных типов — разные переменные..... | 117 |
| Правило 21. Контролируйте расширение типов..... | 119 |
| Правило 22. Старайтесь сужать типы..... | 123 |
| Правило 23. Создавайте объекты целиком..... | 127 |
| Правило 24. Применяйте псевдонимы согласованно | 130 |
| Правило 25. Для асинхронного кода используйте функции <code>async</code> вместо обратных вызовов..... | 134 |
| Правило 26. Используйте контекст при выводе типов | 139 |
| Правило 27. Используйте функциональные конструкции и библиотеки для содействия движению типов | 144 |
| ГЛАВА 4. ПРОЕКТИРОВАНИЕ ТИПОВ | 149 |
| Правило 28. Используйте типы, имеющие допустимые состояния..... | 149 |
| Правило 29. Будьте либеральны в том, что берете, но консервативны в том, что даете..... | 155 |
| Правило 30. Не повторяйте информацию типа в документации..... | 159 |
| Правило 31. Смещайте нулевые значения на периферию типов..... | 161 |
| Правило 32. Предпочитайте объединения интерфейсов интерфейсам объединений..... | 165 |

| | |
|--|------------|
| Правило 33. Используйте более точные альтернативы типов <code>string</code> | 169 |
| Правило 34. Лучше сделать тип незавершенным, чем ошибочным | 174 |
| Правило 35. Генерируйте типы на основе API и спецификаций, а не данных..... | 178 |
| Правило 36. Именуйте типы согласно области их применения..... | 183 |
| Правило 37. Рассмотрите использование маркировок для номинального типизирования | 186 |
| ГЛАВА 5. ЭФФЕКТИВНОЕ ПРИМЕНЕНИЕ ANY | 190 |
| Правило 38. Используйте максимально узкий диапазон для типов <code>any</code> | 190 |
| Правило 39. Используйте более точные варианты <code>any</code> | 193 |
| Правило 40. Скрывайте небезопасные утверждения типов в грамотно типизированных функциях | 195 |
| Правило 41. Распознавайте изменяющиеся <code>any</code> | 197 |
| Правило 42. Используйте <code>unknown</code> вместо <code>any</code> для значений с неизвестным типом | 201 |
| Правило 43. Используйте типобезопасные подходы вместо обезьяньего патча..... | 205 |
| Правило 44. Отслеживайте зону охвата типов для сохранения типобезопасности | 207 |
| ГЛАВА 6. ДЕКЛАРАЦИИ ТИПОВ И @TYPES..... | 211 |
| Правило 45. Размещайте TypeScript и <code>@types</code> в <code>devDependencies</code> | 211 |
| Правило 46. Проверяйте совместимость трех версий, задействованных в декларациях типов | 213 |
| Правило 47. Экспортируйте все типы, появляющиеся в публичных API..... | 218 |
| Правило 48. Используйте TSDoc для комментариев в API | 219 |
| Правило 49. Определяйте тип <code>this</code> в обратных вызовах | 222 |
| Правило 50. Лучше условные типы, чем перегруженные декларации | 226 |
| Правило 51. Зеркалируйте типы для разрыва зависимостей..... | 229 |
| Правило 52. Тестируйте типы с осторожностью..... | 231 |

| | |
|--|-----|
| ГЛАВА 7. НАПИСАНИЕ И ЗАПУСК КОДА | 237 |
| Правило 53. Используйте возможности ECMAScript, а не TypeScript..... | 237 |
| Правило 54. Проводите итерацию по объектам..... | 243 |
| Правило 55. Иерархия DOM — это важно..... | 246 |
| Правило 56. Не полагайтесь на private при скрытии информации..... | 251 |
| Правило 57. Используйте карты кода для отладки..... | 254 |
| ГЛАВА 8. ПЕРЕНОС ДАННЫХ В TYPESCRIPT | 259 |
| Правило 58. Пишите современный JavaScript..... | 260 |
| Правило 59. Используйте // @ts-check и JSDoc для экспериментов в TypeScript..... | 269 |
| Правило 60. Используйте allowJs для совмещения TypeScript и JavaScript..... | 274 |
| Правило 61. Конвертируйте модуль за модулем, восходя по графу зависимостей..... | 276 |
| Правило 62. Не считайте миграцию завершенной, пока не включите noImplicitAny..... | 281 |
| ОБ АВТОРЕ | 284 |
| ОБ ОБЛОЖКЕ | 285 |