

УДК 004.43  
ББК 32.973.3  
Р64

Р64 Натан Розенталс  
Изучаем TypeScript 3. – М.: ДМК Пресс, 2019. – 624 с.

**ISBN 978-5-97060-757-2**

TypeScript – это и язык, и набор инструментов для генерации кода JavaScript. Язык TypeScript и его компилятор завоевали прочные позиции в сообществе разработчиков на JavaScript и продолжают набирать силу, имея богатый инструментарий разработки. Многие масштабные проекты на JavaScript, в том числе проекты Adobe, Mozilla и Asana, приняли решение перевести свою кодовую базу с JavaScript на TypeScript.

Эта книга представляет собой руководство по TypeScript, которое начинается с базовых понятий, а затем представляет более продвинутые возможности языка. Подробно рассказано об использовании TypeScript со множеством современных фреймворков, применяются методы разработки через тестирование, дано много стандартных шаблонов проектирования. Итогом изучения будет полностью готовое к использованию приложение на TypeScript.

Издание будет полезно всем разработчикам приложений.

УДК 004.43  
ББК 32.973.3

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельцев авторских прав

Материал, изложенный в данной книге, многократно проверен. Но, поскольку вероятность технических ошибок все равно существует, издательство не может гарантировать абсолютную точность и правильность приводимых сведений. В связи с этим издательство не несет ответственности за возможные ошибки, связанные с использованием книги.

ISBN 978-5-97060-757-2 (рус.) Copyright © 2019 Packt. All rights reserved.  
ISBN 978-1-78953-670-6 (анг.) © Оформление, издание, ДМК Пресс, 2019

# Оглавление

<b>Предисловие</b>	19
<b>Глава 1 Инструменты TypeScript и параметры фреймворков</b>	25
<b>Что такое TypeScript?</b>	27
JavaScript и ECMAScript	28
Преимущества TypeScript	29
Компиляция	29
Сильная типизация	30
Синтаксический сахар	31
Определение типов для популярных библиотек JavaScript	32
DefinitelyTyped	33
Инкапсуляция	33
Классы TypeScript генерируют замыкания	35
Методы доступа public и private	35
<b>Интегрированные среды разработки TypeScript</b>	37
Компиляция на основе Node	38
Создание файла tsconfig.json	39
Локализованные сообщения	40
Visual Studio Code	41
Установка VSCode	41
Изучение VSCode	41
Создание файла tasks.json	41
Сборка проекта	44
Создание файла launch.json	44
Установка точек останова	44
Отладка веб-страниц	45
Microsoft Visual Studio	48
Создание проекта в Visual Studio	48
Настройки проекта по умолчанию	50
Отладка в Visual Studio	53
WebStorm	54
Создание проекта в WebStorm	55
Файлы по умолчанию	55
Создание простого HTML-приложения	56
Запуск веб-страницы в Chrome	57
Другие редакторы	59
Использование --watch и Grunt	59
<b>Резюме</b>	62
<b>Глава 2 Типы, переменные и методы функций</b>	63
<b>Базовые типы</b>	64
Типизация в JavaScript	64
Типизация в TypeScript	65
Синтаксис типов	66
Типизация с поддержкой вывода типов	69
Утиная типизация	70

## Оглавление

---

Шаблонные строки	71
Массивы	72
for...in и for...of	73
Тип any	74
Явное приведение типов	75
Перечисления	76
Const enum	77
Строковые перечисления	78
Реализация перечислений	78
Const	80
Ключевое слово let	80
Определенное присваивание	83
Типы свойств с точечной нотацией	84
Числовые разделители	85
<b>Функции</b>	85
Типы возвращаемого значения	85
Анонимные функции	86
Необязательные параметры	87
Параметры по умолчанию	88
Оставшиеся параметры	89
Функции обратного вызова	91
Сигнатуры функций	93
Переопределение функций	95
Try...catch	96
<b>Расширенные типы</b>	97
Объединенные типы	98
Охранники типов	98
Псевдонимы типов	100
Null и undefined	101
Нулевые операнды	103
Never	104
Unknown	105
Object rest and spread	106
Приоритет распространения	107
Использование операторов остатка и распространения с массивами	108
Кортежи	109
Деконструкция кортежей	110
Необязательные элементы кортежа	111
Кортежи и синтаксис оператора остатка	111
Bigint	112
<b>Резюме</b>	116
<b>Глава 3 Интерфейсы, классы и наследование</b>	117
<b>Интерфейсы</b>	118
Необязательные свойства	119
Компиляция интерфейса	120

---

Слабые типы	120
Вывод типов с помощью оператора in	121
<b>Классы</b>	122
Свойства класса	124
Реализация интерфейсов	124
Конструкторы классов	126
Функции класса	127
Определения функций интерфейса	130
Модификаторы класса	131
Модификаторы доступа конструктора	132
Свойство readonly	134
Методы доступа к свойствам класса	134
Статические функции	136
Статические свойства	136
Пространства имен	137
<b>Наследование</b>	138
Наследование интерфейса	139
Наследование классов	139
Ключевое слово super	140
Переопределение функции	141
Члены класса protected	142
Абстрактные классы	143
Замыкания JavaScript	146
instanceof	148
<b>Использование интерфейсов, классов и наследования – шаблон проектирования Factory</b>	150
Бизнес-требования	150
Что делает шаблон проектирования Factory	150
Интерфейс IPerson	151
Класс Person	151
Классы специалистов	152
Класс Factory	153
Использование класса Factory	154
<b>Резюме</b>	155
<b>Глава 4 Декораторы, обобщения и асинхронные функции</b>	156
<b>Декораторы</b>	157
Синтаксис декораторов	158
Несколько декораторов	159
Фабрика декораторов	159
Параметры декораторов класса	160
Декораторы свойств	162
Декораторы статических свойств	163
Декораторы методов	164
Использование декораторов методов	165
Декораторы параметров	167

---

Метаданные декораторов	168
Использование метаданных декораторов	170
<b>Обобщения</b>	171
Синтаксис обобщений	172
Создание экземпляра обобщенного класса	172
Использование типа T	174
Ограничение типа T	176
Обобщенные интерфейсы	178
Создание новых объектов в обобщениях	179
<b>Расширенные типы с обобщениями</b>	181
Условные типы	182
Распределенные условные типы	185
Выведение условных типов	187
keyof	189
keyof с числом	190
Отображаемые типы	192
Partial, Readonly, Record и Pick	193
<b>Асинхронное программирование</b>	195
Промисы	195
Синтаксис промисов	197
Использование промисов	199
Механизм обратного вызова в сравнении с синтаксисом промиса	200
Возвращение значений из промисов	201
async и await	203
awaitError	204
Синтаксис промиса в сравнении с синтаксисом async await	205
awaitMessage	206
<b>Резюме</b>	207
<b>Глава 5 Файлы объявлений и строгие опции компилятора</b>	208
<b>Глобальные переменные</b>	209
<b>Использование блоков кода JavaScript в HTML</b>	211
Структурированные данные	212
<b>Пишем свой файл объявлений</b>	214
Ключевое слово module	216
Интерфейсы	218
Объединенные типы	220
<b>Слияние модулей</b>	221
<b>Справочник синтаксиса объявлений</b>	222
Переопределение функций	222
Синтаксис JavaScript	222
Синтаксис файла объявлений	223
Вложенные пространства имен	223
Синтаксис JavaScript	223
Синтаксис файла объявлений	223

---

Классы	223
Синтаксис JavaScript	223
Синтаксис файла объявлений	223
Пространства имен классов	224
Синтаксис JavaScript	224
Синтаксис файла объявлений	224
Перегрузки конструктора класса	224
Синтаксис JavaScript	224
Синтаксис файла объявлений	224
Свойства класса	224
Синтаксис JavaScript	225
Синтаксис файла объявлений	225
Функции класса	225
Синтаксис JavaScript	225
Синтаксис файла объявлений	225
Статические свойства и функции	225
Синтаксис JavaScript	225
Синтаксис файла объявлений	226
Глобальные функции	226
Синтаксис JavaScript	226
Синтаксис файла объявлений	226
Сигнатуры функций	226
Синтаксис JavaScript	226
Синтаксис файла объявлений	226
Необязательные свойства	226
Синтаксис JavaScript	227
Синтаксис файла объявлений	227
Слияние функций и модулей	227
Синтаксис JavaScript	227
Синтаксис файла объявлений	227
<b>Строгие опции компилятора</b>	227
noImplicitAny	228
strictNullChecks	229
strictPropertyInitialization	230
noUnusedLocals и noUnusedParameters	231
noImplicitReturns	232
noFallthroughCasesInSwitch	233
strictBindCallApply	234
<b>Резюме</b>	236
<b>Глава 6 Сторонние библиотеки</b>	237
<b>Использование файлов определений</b>	237
<b>Использование NuGet</b>	238
Использование диспетчера расширений NuGet	238
Установка файлов объявлений	240
Использование консоли диспетчера пакетов	241

---

Установка пакетов	241
Поиск имен пакетов	241
Установка конкретной версии	242
<b>Использование npm и @types</b>	242
<b>Использование сторонних библиотек</b>	243
Выбор фреймворка JavaScript	244
<b>Backbone</b>	245
Использование наследования с Backbone	245
Использование интерфейсов	248
Использование синтаксиса обобщений	248
Использование ECMAScript 5	249
Совместимость Backbone с TypeScript	250
<b>Angular 1</b>	250
Классы Angular и \$scope	252
Совместимость Angular 1 с TypeScript	254
<b>Наследование – Angular 1 против Backbone</b>	254
<b>ExtJS</b>	255
Создание классов в ExtJS	256
Использование приведения типов	257
Компилятор TypeScript специально для ExtJS	258
<b>Резюме</b>	259
<b>Глава 7 Фреймворки, совместимые с TypeScript</b>	260
<b>Что такое MVC?</b>	261
Модель	261
Представление	262
Контроллер	263
Резюмируя	264
Преимущества использования MVC	265
Пример приложения	265
<b>Использование Backbone</b>	267
Производительность визуализации	267
Настройка Backbone	269
Структура Backbone	269
Модели Backbone	270
Класс ItemView	273
Класс ItemCollectionView	275
Приложение Backbone	277
Формы	278
Резюмируя	282
<b>Использование Aurelia</b>	282
Настройка Aurelia	282
Контроллеры и модели Aurelia	283
Представления Aurelia	284
Начальная загрузка приложения	285

## Оглавление

---

События	286
Формы	287
Резюмируя	288
<b>Angular</b>	288
Установка Angular	288
Модели Angular	289
Представления Angular	290
События	292
Формы	293
Формы шаблонов	293
Ограничения форм шаблонов	295
Реактивные формы	295
Использование реактивных форм	296
Резюмируя	298
<b>Использование React</b>	299
Настройка React	299
Настройка webpack	301
ItemView	304
CollectionView	306
Начальная загрузка	309
Формы	311
Изменение состояния	312
Свойства состояния	313
Возможности TypeScript в React	316
Синтаксис оставшихся параметров	316
Свойства по умолчанию	317
Резюмируя	318
<b>Сравнение производительности</b>	318
<b>Резюме</b>	321
<b>Глава 8 Разработка через тестирование</b>	322
<b>Разработка через тестирование</b>	323
<b>Модульные, интеграционные и приемочные тесты</b>	325
Модульные тесты	325
Интеграционные тесты	325
Приемочные тесты	326
<b>Фреймворки для модульного тестирования</b>	326
<b>Jasmine</b>	327
Простой тест	328
Репортеры	331
Сопоставители	332
Запуск и завершение теста	334
Принудительные тесты	335
Пропуск тестов	336
Тесты, управляемые данными	337
Использование шпионов	340



---

Служба за функциями обратного вызова	341
Использование шпионов в качестве фальшивок	342
Асинхронное тестирование	343
Использование функции done()	345
Использование async await	347
HTML-тесты	348
Фикстуры	350
События DOM	351
<b>Библиотеки для модульного тестирования</b>	352
Testem	353
Karma	354
Тестирование в режиме headless	356
Protractor	357
Selenium	357
Поиск элементов страницы	359
<b>Использование непрерывной интеграции</b>	361
Преимущества непрерывной интеграции	361
Выбор сервера сборки	362
Team Foundation Server	363
Jenkins	363
TeamCity	363
Отчеты об интеграционных тестах	363
<b>Резюме</b>	365
<b>Глава 9 Тестирование фреймворков, совместимых с Typescript</b>	366
<b>Тестирование нашего приложения</b>	366
<b>Тестирование Backbone</b>	367
Настройка теста	367
Тесты моделей	368
Тесты сложных моделей	371
Тесты визуализации	372
Тесты событий DOM	374
Тесты представления коллекции	375
Тесты формы	377
Резюмируя	379
<b>Тестирование Aurelia</b>	379
Настройка	379
Модульные тесты	380
Тесты визуализации	381
События DOM	385
Резюмируя	386
<b>Тестирование Angular</b>	386
Настройка	386
Тесты моделей	389
Тесты визуализации	390
Тесты форм	393

---

Резюмируя	395
<b>Тестирование с React</b>	396
Несколько точек входа	396
Использование Jest	397
Тесты начального состояния	400
Элемент input	402
Отправка формы	403
Подводя итоги	405
<b>Резюме</b>	405
<b>Глава 10 Модуляризация</b>	407
<b>Основы</b>	408
Экспорт модулей	410
Импорт модулей	411
Переименование модулей	411
Экспорт по умолчанию	412
Экспорт переменных	413
Типы импорта	414
<b>Асинхронное определение модуля</b>	415
Компиляция	416
Установка модуля AMD	417
Настройка Require	418
Настройка браузера	419
Зависимости модуля AMD	420
Начальная загрузка Require	423
Исправление ошибок конфигурации	424
Неправильные зависимости	425
Ошибки 404	425
<b>Загрузка модулей с помощью SystemJS</b>	426
Установка SystemJS	426
Конфигурация браузера	426
Зависимости модулей	429
Начальная загрузка Jasmine	431
<b>Использование Express с Node</b>	432
Установка Express	432
Использование модулей с Express	434
Маршрутизация	435
Шаблонизаторы	437
Использование Handlebars	438
События POST	441
Перенаправление HTTP-запросов	445
<b>Функции Lambda</b>	447
Архитектура функции Lambda	447
Настройка AWS	449
Serverless	451
Настройка Serverless	452

---

Развертывание	453
Функции Lambda в TypeScript	456
Node-модули функции Lambda	458
Логирование	460
Тестирование REST	461
<b>Резюме</b>	464
<b>Глава 11 Объектно-ориентированное программирование</b>	465
<b>Принципы объектно-ориентированного программирования</b>	466
Программирование в соответствии с интерфейсом	466
Принципы SOLID	467
Единственная ответственность	467
Открытость/закрытость	467
Принцип подстановки Барбары Лисков	468
Разделение интерфейса	468
Инверсия зависимостей	468
<b>Проектирование пользовательского интерфейса</b>	468
Концептуальный дизайн	469
Настройка Angular	471
Использование Bootstrap и Font-Awesome	472
Создание боковой панели	473
Создание наложения	477
Координация переходов	479
<b>Шаблон State</b>	480
Интерфейс шаблона State	481
Конкретные состояния	482
<b>Шаблон Mediator</b>	483
<b>Модульный код</b>	484
Компонент Navbar	484
Компонент SideNav	486
Компонент RightScreen	487
Дочерние компоненты	490
Реализация интерфейса посредника	490
Класс Mediator	491
Использование Mediator	495
Реагирование на события DOM	496
<b>Резюме</b>	498
<b>Глава 12 Внедрение зависимости</b>	499
<b>Отправка почты</b>	500
Использование nodemailer	501
Использование локального SMTP-сервера	502
Служебный класс	502
Настройки конфигурации	505
Зависимость объектов	507
Service Location	507

---

Антишаблон Service Location	510
Внедрение зависимости	510
<b>Создание инжектора зависимостей</b>	511
Разрешение интерфейса	511
Разрешение enum	511
Разрешение класса	512
Внедрение конструктора	515
<b>Внедрение декораторов</b>	516
Использование определения класса	517
Синтаксический анализ параметров конструктора	518
Поиск типов параметров	520
Внедрение свойств	520
Использование внедрения зависимости	521
Рекурсивное внедрение	522
<b>Резюме</b>	524
<b>Глава 13 Создание приложений</b>	525
<b>Интеграция Node и Angular</b>	526
Сервер Express	528
Конфигурация сервера	530
Ведение журнала сервера	531
<b>Опыт взаимодействия</b>	535
Использование Brackets	536
Использование Emmet	537
Создание панели входа	540
<b>Аутентификация</b>	542
Маршрутизация в Angular	543
Использование HTML-кода, предназначенного для пользовательского интерфейса	546
Стражи аутентификации	547
Связывание формы входа	549
Использование HttpClient	551
Использование Observable	553
Использование JWT-токенов	557
Верификация токенов	560
Использование Observables в гнев – of, pipe и map	561
<b>Внешняя аутентификация</b>	565
Получение API-ключа Google	565
Настройка социального логина	567
Использование данных пользователя Google	568
<b>Резюме</b>	570
<b>Глава 14 Переходим к практике</b>	572
<b>Приложение Board Sales</b>	573
<b>API на основе базы данных</b>	574
Структура базы данных	575

---

## Оглавление

---

Конечные точки API	577
Параметризованные конечные точки API	581
Службы REST в Angular	584
Спецификация OpenAPI	587
<b>Приложение BoardSales</b>	588
Компонент BoardList	588
Визуализация данных REST	591
concatMap	594
forkJoin	599
Модульное тестирование Observables	602
<b>Шаблон проектирования Domain Events</b>	605
Вызов и использование событий предметной области	608
Фильтрация данных	612
<b>Резюме</b>	620
<b>Указатель</b>	621