



TYPESCRIPT NA POWAŻNIE

Michał Miszczyszyn



TypeScript na poważnie

Michał Miszczyszyn

26.0.0-6-g4ac076e

Dla mojej żony, Martyny

Projekt okładki: Ewelina Sygut-Pawłowska
Redaktor merytoryczny: Michał Miszczyszyn
Redaktorka prowadząca: Martyna Wygonna-Miszczyszyn
Sugestie merytoryczne: Bartosz Cytrowski, Michał Michalczuk

Książka, którą nabyłeś, jest dziełem twórcy i wydawcy. Prosimy, abyś przestrzegał praw, jakie im przysługują. Jej zawartość możesz udostępnić nieodpłatnie osobom bliskim lub osobiście znanym. Ale nie publikuj jej w internecie. Jeśli cytujesz jej fragmenty, nie zmieniaj ich treści i koniecznie zaznacz, czyje to dzieło. A kopiując ją, rób to jedynie na użytek osobisty.

Szanujemy cudzą własność i prawo!

Polska Izba Książki

Więcej o prawie autorskim na www.legalnakultura.pl

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

TypeScript na poważnie

<https://typescriptnapowaznie.pl>

Copyright © Michał Miszczyszyn

ISBN e-book EPUB: 978-83-957363-0-8

ISBN e-book MOBI: 978-83-957363-1-5

ISBN e-book PDF: 978-83-957363-2-2

ISBN druk: 978-83-957363-3-9

Wydanie I

Gdańsk 2020

Wydawnictwo Type of Web

email: hi@typeofweb.com

www: <https://typeofweb.com>

Spis treści

Przedmowa	3
1 Wstęp: Po co TypeScript?	5
1.1 Czym jest TypeScript?	5
1.2 Statyczne typowanie	6
1.3 Inferencja typów	7
1.4 Dokumentacja i kontrakty	8
1.5 Pewność i niepewność.	9
1.6 Pomoc narzędzi	10
1.7 Krótko mówiąc	11
1.8 Zanim przejdiesz dalej.	12
1.9 Jak poprawnie odmieniać słowo TypeScript?	12
1.10 Konwencje używane w książce	14
2 TypeScript w 10 minut	15
2.1 Pierwsze linijki TypeScripta	15
2.2 TypeScript a JavaScript	16
2.3 Instalacja kompilatora	16
2.4 Kompilacja pliku	17
2.5 Dodajmy typy	17
2.6 Strict.	19
2.7 Piaskownica	20
2.8 Edytor	20
2.9 Przykładowy projekt.	21
2.10 Node.js	22
2.11 tsconfig.json.	23

Spis treści

2.12	Podsumowanie	24
3	Dygresja: Wprowadzenie do przydatnych elementów z ES6+	25
3.1	ECMAScript	26
3.2	Klasy.	26
3.3	Moduły	27
3.4	Funkcje strzałkowe	28
3.5	let i const.	29
3.6	Destrukturyzacja i skrócony zapis obiektów.	33
3.7	Domyślna wartość parametrów.	36
3.8	Rest	38
3.9	Spread.	38
3.10	Pozostałe nowości	40
3.11	Niestandardowe elementy języka.	40
3.12	Operator import(...)	42
3.13	Prywatne pola w klasach ze znacznikiem #	43
4	Typowanie statyczne i typowanie silne	45
4.1	Systemy typów	45
4.2	Dynamiczne typowanie.	46
4.3	Styczne typowanie	47
4.4	Słabe typowanie	48
4.5	Silne typowanie.	49
4.6	Typy silne, słabe, statyczne, dynamiczne...	51
4.7	ts-ignore	51
4.8	Typowanie strukturalne, nominalne i duck typing.	51
4.9	Podsumowanie	55
5	Typy elementarne	57
5.1	Podstawy	57
5.2	Słowo kluczowe type	68
5.3	Podsumowanie	69
6	Funkcje	71
6.1	Argumenty funkcji.	71
6.2	Typ zwracany	72
6.3	Wyrażenia funkcji	72

6.4	Typ funkcji.	73
6.5	Inferencja typu argumentów	73
6.6	Parametry opcjonalne	74
6.7	Parametry domyślne.	75
6.8	Funkcje wariadyczne	75
6.9	Przeładowywanie funkcji.	76
6.10	this.	79
6.11	Podsumowanie	80
7	Klasy i interfejsy	81
7.1	Klasa.	81
7.2	Modyfikatory public, private i protected	85
7.3	Klasa abstrakcyjna	89
7.4	Interfejs.	90
7.5	Typ statyczny i typ instancji w klasach	97
7.6	Pola prywatne ES.	99
8	Typy generyczne	103
8.1	Funkcje generyczne	103
8.2	Inferencja w generykach	104
8.3	Generyczne typy	105
8.4	Inne generyki	105
8.5	Ograniczenia generyków	107
8.6	Generyki wielu typów	108
8.7	Podsumowanie	109
9	Inferencja typów i const	111
9.1	Przykład	111
9.2	Wnioskowanie typów argumentów	112
9.3	Typ wspólny	112
9.4	Inferencja czasem zawodzi.	113
9.5	Inferencja kontekstowa.	115
9.6	Cementowanie typów	117
9.7	Inferencja przy const i let.	118
9.8	Podsumowanie	119
10	Kompatybilność typów	121

Spis treści

10.1	Kompatybilność: podtyp a przypisywanie	122
10.2	Kompatybilność strukturalna	122
10.3	Klasy z polami publicznymi	122
10.4	Klasy z polami prywatnymi	124
10.5	Kompatybilność podtypów.	125
10.6	Przypisywanie literałów obiektów	126
10.7	Kompatybilność funkcji wariadycznych	127
10.8	Kompatybilność argumentów funkcji	129
10.9	Kompatybilność metod w obiektach	130
10.10	Kompatybilność typu zwracanego przez funkcje	131
10.11	Argumenty opcjonalne i rest.	131
10.12	Typy kowariantne, kontrawariantne, biwariantne i in- variantne	133
10.13	Powtórzenie	136
10.14	Więcej o kowariancji i kontrawariancji	137
10.15	Dla dociekliwych	139
10.16	Kowariancja i kontrawariancja przez analogię	140
10.17	Wariancja a mutowalność	140
10.18	Inferencja typów a wariancja	142
10.19	bivarianceHack	144
10.20	Dlaczego metody są biwariantne	145
11	Enumy	149
11.1	Enumy numeryczne	149
11.2	Enumy z ciągami znaków	153
11.3	Różnice pomiędzy enumami.	153
11.4	Enumy są typowane nominalnie	155
11.5	Test wyczerpania.	155
11.6	Test wyczerpania z liczbami.	156
11.7	Test wyczerpania bez noImplicitReturns.	157
11.8	Kompatybilność obiektów i enumów	159
11.9	const enum	159
11.10	Enum a literał stringa	161
11.11	Podsumowanie	161
12	Typy zaawansowane	163

12.1	Unique Symbol	163
12.2	typ i interfejs	165
12.3	Łączenie deklaracji.	166
12.4	Aliasy typów.	167
12.5	Część wspólna i suma typów (unia)	168
12.6	Index signature	172
12.7	Literal type.	174
12.8	As const – niemutowalne typy danych	175
12.9	Type guards	177
12.10	Pobieranie typu wartości	184
12.11	Przeładowywanie funkcji literałami.	186
12.12	Typy rekurencyjne	186
12.13	Algebraiczne typy danych	188
12.14	Wyłuskiwanie typu	192
12.15	Mapped types	192
12.16	async i Promise	198
13	Typy warunkowe	199
13.1	Co to są typy warunkowe?	199
13.2	Przykładowe użycie	200
13.3	Typy warunkowe na unii	200
13.4	Zagnieżdżanie.	201
13.5	Warunkowe typy dystrybutywne	202
13.6	Przykład użycia	203
13.7	Opóźnione warunki	207
13.8	Kompatybilność typów warunkowych	208
13.9	infer	209
13.10	Podsumowanie	213
14	Unie w praktyce	215
14.1	Problem.	215
14.2	Pierwsze podejście	216
14.3	Podejście drugie.	219
14.4	Podejście trzecie	221
14.5	Co tu się stało?	224
14.6	Finalizacja	225

Spis treści

14.7	Podsumowanie	226
14.8	Ale rzutowanie?	226
15	Bezpieczna praca z danymi	229
15.1	Dlaczego typy znikają po kompilacji.	230
15.2	Dane z zewnątrz i rzutowanie.	230
15.3	Generowanie typów z JSON	231
15.4	Walidacja	232
15.5	Zła walidacja.	234
15.6	Kiedy walidować	235
15.7	Biblioteki do walidacji.	235
15.8	io-ts i zod.	236
15.9	Unie w praktyce.	240
15.10	Podsumowanie	242
16	Typy nominalne	243
16.1	Czy typy pomagają?	244
16.2	Problemy z typowaniem strukturalnym	245
16.3	Typy nominalne w TS	247
16.4	Zestawienie rozwiązań	255
16.5	Biblioteki	257
16.6	Typy nominalne a walidacja w io-ts	258
16.7	Podsumowanie	259
17	Własna walidacja, typy warunkowe i testowanie typów	261
17.1	Tworzenie własnego walidatora	261
17.2	Przechowywanie informacji o typach	262
17.3	Walidator z konfiguracją	264
17.4	Walidacja obiektów.	265
17.5	Nasz własny walidator	267
17.6	Testowanie typów	268
17.7	dtslint	268
17.8	Podsumowanie	270
18	Implementacja systemu jednostek w TS	271
18.1	Koncepcja	274
18.2	Liczby Peano.	279

18.3	Implementacja	284
18.4	Biblioteki	287
19	Migracja z JavaScriptu	291
19.1	Konfiguracja dla łagodnego przejścia	293
19.2	Sprawdzanie poprawności plików JS.	298
20	Pliki definicji typów .d.ts	307
20.1	Tworzenie bibliotek w TypeScriptie	309
20.2	Dodawanie typów do istniejących bibliotek w JS	314
21	Popularne problemy z TS	321
21.1	Poprawne typowanie Array#filter	321
21.2	Typ Object.keys() jest niepoprawny	325
21.3	Wszystkie elementy tablicy są zdefiniowane	327
21.4	Funkcja przyjmuje obiekty, które mają więcej pól, niż powinny	329
21.5	Otagowane unie nie zawsze działają poprawnie	331
Dodatek:	Przydatne typy	335
21.6	Typy wbudowane w TypeScript.	335
21.7	Inne przydatne typy	336

Errata

Dołożyłem wszelkich starań, aby w niniejszej książce znalazły się wyłącznie informacje rzetelne i sprawdzone. Przetestowałem każdy fragment kodu i każdy przykład. Pomimo tego, może się tak zdarzyć, że gdzieś wkradły się błędy.

Jeśli znajdziesz jakiegokolwiek pomyłki, to proszę, koniecznie mi to zgłoś! Możesz to zrobić poprzez stronę typeofweb.com/errata. Pod tym adresem znajdziesz również wypisane poprawki, które do książki zostały wprowadzone już po premierze, aby łatwiej Ci było się z nimi zapoznać.