

Epilog: lekcja 50

Witamy na ostatniej lekcji. Kiedy wyruszyliśmy w tę podróż, skupiliśmy się na konkretnym zagadnieniu: w jaki sposób możemy używać TypeScript jako rozszerzenia JavaScript, języka programowania, który napędza sieć? Podążanie tą drogą zaprowadziło nas w głąb sfer systemów typów, ucząc się, jak możemy modelować dane za pomocą typów sum i przecięć, modelować zachowanie za pomocą typów ogólnych i warunkowych oraz próbować maksymalnie ograniczyć obsługę typów. Na koniec mieliśmy do dyspozycji arsenal narzędzi, które pomagały nam używać jak najmniejszej ilości kodu do definiowania najbardziej złożonych scenariuszy JavaScript. Jednak to tylko część tego, co ma do zaoferowania TypeScript. Prawdopodobnie najważniejsza część, ale nigdy nie wiesz, gdzie skończysz ze swoimi nowo odkrytymi umiejętnościami TypeScript. W tej ostatniej lekcji chcę cię przygotować na nieznaną. Nie typ nieznaną, ale to, co nas czeka: przyszłość, której nie da się opisać w książce, która ma być ponadczasowa. Więc dokąd się stąd wybierasz?

Słuchanie

Nalegam, abyś zwrócił się do zespołu TypeScript. Pracują na otwartej przestrzeni. Możesz zobaczyć komunikację zespołu na GitHub, znaleźć mapy drogowe, nadchodzące funkcje i ich ogólne plany. Wiemy, że nawet jeśli TypeScript obejmuje już szeroki zakres scenariuszy JavaScript, nadal istnieją sytuacje, w których potrzebujemy rzutowania typu lub, co gorsza: any! To ma się zmienić. A zespół TypeScript bardzo głośno mówi o swoich planach. Szczególnie dobrą lekturą jest mapa drogowa. Powinieneś także uważać na TC39, komitet, który standaryzuje ECMAScript i który ściśle współpracuje z zespołem TypeScript. Członkostwo nawet w niewielkim stopniu się pokrywa. Nowe funkcje językowe są określane przez TC39 i implementowane w TypeScript, gdy osiągną określony poziom dojrzałości i są gotowe do zaimplementowania w silnikach JavaScript. Ich repo GitHub jest doskonałym źródłem dyskusji i nowych funkcji.

Uczenie się

Omówiliśmy dużą i znaczącą część TypeScript, ale to nie wszystko. Istnieją funkcje, które sięgają głęboko do języków programowania zorientowanych obiektowo. Inne konstrukcje językowe po prostu ułatwiają pisanie niektórych formularzy. Istnieją eksperymentalne funkcje wymagane przez niektóre frameworki. Oficjalny podręcznik i dokumentacja TypeScript są doskonałymi źródłami tego wszystkiego. Znajdziesz zestawy startowe dla języka TypeScript z ulubionym frameworkiem. Otrzymasz również dobry przegląd funkcji językowych, których jeszcze nie widziałeś i które mogą Ci się przydać.

Czytaj

Spółeczność TypeScript jest bardzo aktywna w dostarczaniu nowych narzędzi, pisaniu artykułów i pokazywaniu programistom, co można zrobić z mocą tego języka. Jest nowe środowisko wykonawcze podobne do Node.js o nazwie Deno, opracowane przez oryginalnych twórców Node.js, które od razu obsługuje TypeScript. Razem z ludźmi z menedżera pakietów Pika upewniają się, że otrzymujesz pliki deklaracji typu przez HTTP po zaimportowaniu pakietu Deno z adresu URL.

Następnie jest mnóstwo blogów w języku TypeScript. Marius Schulz od lat pisze o TypeScript i jest kuratorem wspaniałego biuletynu zatytułowanego TypeScript Weekly.