

TypeScriptとは、AltJS (Alternative JavaScript) の一つで、2012年頃にマイクロソフトが開発したJavaScriptを進化させたプログラミング言語です。

大人数での開発でもエラーを防ぐように設計されており、JavaScriptの持つ問題点を解消し、コンパイルするとJavaScriptのコードに変換されます。そのため、TypeScriptからJavaScriptライブラリが使用できるなど、高い互換性を持ちます。

Googleの社内標準言語として採用されたTypeScriptには、JavaScriptにはないメリットや特徴があります。今回はTypeScriptについて、どのような言語なのか詳しく解説していきます。

TypeScriptとは？

TypeScriptはJavaScriptを拡張して作られたオープンソースのプログラミング言語で、2012年頃にマイクロソフトが開発しました。2017年2月にリリースされ、Googleの標準開発言語に承認された2.2以降は、2ヶ月に1回のサイクルでリリースが行われています。

TypeScriptが開発された背景には、JavaScriptを使用した大規模アプリケーションを開発する際に、複雑なJavaScriptのコードを扱うため、コンポーネント群の開発を用意にする独自ツールが必要となったことが挙げられます。その際、JavaScriptの欠点を補い、使いにくさを改善して、大規模開発においてエラーが起きにくい言語として設計されました。

TypeScriptはGoogleが開発するフレームワークにも導入されており、将来性がある言語の一つといえます。

TypeScriptの特徴

TypeScriptは、JavaScriptに省略可能な静的型付けとクラスベースオブジェクト指向を加えた、JavaScriptの厳密なスーパーセット（上位互換）となっています。JavaScriptにはない型定義が使えるほか、JavaやC#と同様にインターフェースとクラスが使用できます。

TypeScriptの特徴をさらに詳しく見ていきましょう。

TypeScriptはJavaScriptとの互換性が高い

TypeScriptは、他のAltJS同様に、記述したコードをコンパイルするとJavaScriptのコードに変換するため、JavaScriptとはライブラリ群を含め100%の互換性を持っています。

TypeScriptで拡張された機能は、コンパイラを通すことでJavaScriptのコードに変換されます。現行のJavaScriptで記述されたプログラムはすべて、TypeScriptのプログラムとして実行が可能

です。

TypeScriptは型定義が可能

TypeScriptはJavaScriptの拡張として、変数の型定義が可能です。型定義ができることで、変数に定義した型と割り当てた値の型が異なる場合にコンパイルエラーが発生するため、バグチェックに有効です。

また、変数の宣言時に値を渡すことで、型指定を指定なくても自動的に型が宣言されていたように振る舞う「型の推論」が可能です。

さらにJavaScriptと異なる特徴的な機能として、ジェネリックが使用できます。ジェネリックとは型を抽象化し、実際に使用されるまで型が確定しないクラスや関数を扱う型引数のことで、同じような機能を持っている複数の関数を一つにまとめ、コードの冗長化を省きます。

TypeScriptとJavaScriptの違いとは？



TypeScriptでは、JavaScriptと同じ構文やライブラリが使用できます。

TypeScriptとJavaScriptの違いとして特徴的なのは、型付けの違いです。JavaScriptが実行時にデータ型を決める動的型付けであるのに対し、TypeScriptはあらかじめ型を定義する静的型付けになっています。

JavaScriptは動的型付けであるため、実行するまでエラーの存在に気付きにくく、バグが発生しやすい問題点を持っています。そこで、TypeScriptは静的型付けを採用し、コンパイル時にバグの存在に気付きやすくなるように設計されています。

また、JavaScriptで開発中に多くのエンジニアが直面する「Uncaught TypeError: Cannot read property *** of undefined」を避けるため、コンパイラオプションでnull や undefined である可能性を持った変数をチェックできます。

このように言語自体がエラーを未然に防ぎ、バグを発見しやすいように設計されています。加えてTypeScriptはJavaScriptよりもコードが簡潔で可読性が高く、Microsoft Visual Studioなどの統合開発環境を活用すると強力な入力補完が利用でき、開発効率が格段に向上します。

簡潔なコードと開発環境のサポートにより、JavaScriptと比較して、複数人のエンジニアが参画する大人数での開発に適しています。

TypeScriptの5つのメリット

JavaScriptの開発を効率化し、機能を拡張するAltJS の中で、特にTypeScriptを開発に採用する理由は、TypeScriptが持つ5つのメリットがあります。

（1）コードが短く読みやすい

TypeScriptはクラスベースオブジェクト指向の言語として開発されています。JavaScriptでは使えなかったクラスの生成が可能なので、JavaScriptでは冗長になりがちなコードの記述を大幅に簡略化できるようになりました。

また、生成したクラス単位でfunctionを管理することによって、大人数が開発に参画した場合でも、コーディングの効率化や統一化が容易です。

（2）JavaScriptとの互換性が高い

前項で解説したとおり、TypeScriptはJavaScriptの厳密なスーパーセット（上位互換）になります。そのため、JavaScriptの開発環境や実行環境があれば、すぐにTypeScriptを導入できます。

また、JavaScriptで作成したファイルをTypeScriptから呼び出したり、JavaScriptに向けて開発されたライブラリやフレームワークをTypeScriptで使用できます。

（3）エラーを防止できる

TypeScriptは省略可能な静的型付けの言語であるため、変数の型をコード内で宣言できます。型と値が異なる場合はコンパイルエラーとなるため、プログラム全体のエラーを防止できます。

また、JavaScriptはundefinedが発生しやすく、実行時エラーの原因となります。TypeScriptはコンパイラオプションを使い、null や undefined である可能性を持った変数をチェックすることで、コードの問題を発見し、修正しやすくなっています。

（４）動作が早い

TypeScriptは静的型付けの言語なのでコード内で型を宣言できるため、実行時に型を決める動的型付けのJavaScriptよりも動作が早くなります。

また、TypeScriptは定期的にリリースされ、その中で速度の改善も行われています。TypeScript3.9では大規模な速度の改善により、特定のパッケージで編集・コンパイルが極端に遅い問題を解決するなど、こまめなアップデートが行われています。

（５）JavaScriptの知識があれば覚えやすい

TypeScriptはJavaScriptをベースに、機能を拡張しています。そのため、JavaScriptと記述法が近いので、JavaScriptを習得している人にとってTypeScriptは覚えやすい言語といえます。

また、JavaScriptのライブラリやフレームワークを使用する時には、JavaScriptの知識が必要とされることもあります。TypeScriptを学んで使いこなすにはJavaScriptの知識があるとスムーズにスキルアップが可能です。

TypeScriptとJavaScriptフレームワークの関係性

TypeScriptでは、JavaScript向けのフレームワークも使用できます。フレームワークを使用することで、複雑な機能の追加の実装が容易になり、アプリケーション開発の効率が向上します。

JavaScriptには複数のフレームワークがあり、それぞれに特徴やTypeScriptとの関係性が異なります。以下にいくつか紹介するフレームワークは、いずれもオープンソースになっており、githubにリポジトリが公開されています。

AngularJS

Googleを始めとする複数の企業や個人を含むコミュニティによって開発されている、フロントエンドWebアプリケーションフレームワークです。

シングルページアプリケーションの開発が可能で、一つのフレームワーク内ですべての機能が完結しているフルスタックフレームワークなのが特徴として挙げられます。

JavaScript用のフレームワークとして開発されましたが、バージョン2.0以降ではTypeScriptでの開発が推奨されています。

React

Facebookを中心としたコミュニティが開発している、ユーザーインターフェース構築に適したフレームワークです。シングルページアプリケーションやモバイルアプリケーションの開発ベースとして使用できます。ルーティングやAPIとの対話には追加ライブラリを使用します。

JavaScript向けのフレームワークで、JavaScriptでの開発が推奨されていますが、拡張機能を利用することでTypeScriptも使用できます。

Vue.js

GoogleでAngularJSの開発に携わったEvan You氏が開発している、プログレッシブフレームワークです。シンプルで自由度が高く、軽量で動作が早いのが特徴で、WebUI開発において必要な部分だけを採用して他のライブラリと自由に組み合わせることが可能です。

コアライブラリとサポータライブラリやパッケージが分かれているため、部分的に必要な場合はコンパクトに導入できるのがメリットといえます。

Vue3.0はTypeScriptで記述されており、拡張機能を使用せずにTypeScriptでの開発が可能です。

なお、ここでご紹介した3つのフレームワークについては、以下の記事で詳しく紹介していますので、あわせてご確認ください。

【Typescript初心者向け】JavaScriptフレームワークとの関係について

TypeScriptの学習方法



TypeScriptはC系言語と比較すると、プログラミング言語としての難易度は決して高い言語ではありません。そのため、すでにある程度JavaScriptの知識がある場合は、Webサイトや動画、参考書などで独学することができます。

JavaScriptを習得していない、もしくは、TypeScriptの特徴や開発手法を短期間でしっかり学びたい場合は、スクールやオンライン講座の受講がおすすめです。

どちらの場合でもTypeScriptは実際に動作するコードを書いて、コードが実行環境でどのような動きをするのか理解を深めながら学習することが大切です。TypeScriptのIDE（統合開発環境）は強力な入力サポートを持つので、学習の際にはMicrosoft Visual Studioなどを導入するのがおすすめです。

TypeScriptの学習の前に、JavaScriptの基礎を身に付けておくことで、より学習効率が向上します。

TypeScriptは将来性があり人気の高いプログラミング言語

TypeScriptは静的型付けを導入することでJavaScriptの動作の重さやコードの冗長化など、大規模開発では使いにくい部分を改善した言語です。シンプルなコードで高い可読性を持ち、大人数での開発でもエラーが起きにくい設計になっています。

JavaScriptとはスーパーセットの関係にあり、100%の互換性を持っているので、相互にコードを呼び出したり、JavaScriptのライブラリやフレームワークを使用できます。

Googleの標準開発言語に承認され、年々Webアプリケーション開発の現場で人気が高まっているTypeScriptを、ぜひ習得してみてはいかがでしょうか。