

バージョニングポリシー

React はセマンティック バージョニング ([semantic versioning; semver](#)) の原則に従います。
すなわちバージョン番号は **x.y.z** になります。

- **破壊的変更**をする時、**x** の番号を変更することで**メジャーリリース**をします。(例 15.6.2 から 16.0.0)
 - **新機能追加**をする時、**y** の番号を変更することで**マイナーリリース**をします。(例 15.6.2 から 15.7.0)
 - **バグ修正**をする時、**z** の番号を変更することで**パッチリリース**をします。(例 15.6.2 から 15.6.3)
- メジャーリリースには新機能を含むことができ、全てのリリースにバグ修正を含められます。

破壊的変更

破壊的変更は誰にとっても不便なので、私たちはメジャーリリースを最小限にするようにしています。例えば React 15 は 2016 年 4 月にリリースされており、React 16 は 2017 年の 9 月にリリースされています。そして React 17 は 2019 年までリリースが見込まれていません。
その代わり、新機能のリリースをマイナーバージョンでしています。つまりマイナーリリースは控えめな名前にも関わらず、メジャーリリースよりしばしば興味深く魅力的です。

安定性への取り組み

React が徐々に変化していく中で、私たちは新機能を取り入れるために必要な労力を最小限にするようにしています。別のパッケージに入れることはあっても、可能な限り古い API が動作するように保ちます。例えばミックスインは長年推奨されていませんが `create-react-class` を通じて今日までサポートされており、多くのレガシーコードが安定してそれらを継続使用しています。
100 万人を超える開発者が React を使用し、合わせると何百万ものコンポーネントを管理しています。Facebook のコードベースだけでも 5 万以上の React コンポーネントがあります。なので React はできるだけ簡単に新バージョンにアップグレードできるようにする必要があります。もし移行手段なしに React へ大きな変更を行えば、開発者は古いバージョンにとどまるでしょう。私たちはアップグレード方法を Facebook 自体でテストしています。10 人以下の私たちのチームが単独で 5 万以上のコンポーネントをアップデートできるなら、React を使用している全ての人にとって管理しやすいアップグレードであると見込めます。多くの場合、私たちはコンポーネントの構文をアップグレードするための[自動化スクリプト](#)を書き、オープンソースのリリースに含め誰でも使用できるようにしています。

警告による段階的アップグレード

React の開発ビルドは多くの有益な警告を含みます。可能な限り、私たちは将来の破壊的変更に備える警告を追加します。最新のリリースでもしあなたのアプリが警告を出さないのであれば、次期メジャーリリースとの互換性があるでしょう。これによりアプリを 1 つのコンポーネントずつアップグレードすることが可能になります。
開発時の警告はあなたのアプリの実行に影響しません。なので開発ビルドと本番ビルドでアプリの動作は同じであると確信できます。違いは、本番ビルドは警告をロギングしないこと、そして本番ビルドはより効率的に動作すること、の 2 点のみです(万一そうならないことに気づいた場合は、issue を作成してください)。

何を破壊的変更とみなすのか？

通常、私たちは下記の変更ではメジャー番号を**上げません**。

- **開発時の警告**。これらは本番環境の動作に影響を与えないので、新しい警告の追加や既存の警告の修正はメジャーバージョンの間で行います。これにより次期の破壊的変更を確実に警告することができるのです。
- **unstable_ から始まる API**。これらは、API をまだ信頼することができない、実験的な機能として提供されます。unstable_ という接頭語をつけてリリースすることで、より速く開発サイクルを進め、より早く安定した API にすることができます。
- **React のアルファバージョンとカナリア (canary) バージョン**。新機能を早くテストするために React のアルファバージョンを提供しますが、アルファで学んだことを基に柔軟に変更を加える必要があります。もしこれらのバージョンを使用する場合は、安定板のリリース前に API が変わる可能性に注意してください。
- **ドキュメント化されていない API と内部データ構造**。もし内部プロパティである `__SECRET_INTERNALS_DO_NOT_USE_OR_YOU_WILL_BE_FIRED` や `__reactInternalInstance$uk43rzhitjg` などにアクセスしたいのであれば保証はされません。自己責任で行ってください。

このポリシーはみなさんの頭痛の種とならないよう、実用的に構成されています。上記の全ての変更のためにメジャーバージョンを上げると、より多くメジャーリリースが必要になり、最終的により多くのバージョニングの問題をコミュニティに対して引き起こすことになります。それは React の改善を私たちが望むほど早くできないことも意味します。
それでも、上記のリストのような変更がコミュニティ内で広域に渡る問題を引き起こすと予想される場合は、私たちは段階的な移行手段を提供するように最善を尽くします。

このページを編集する