Build Insiderオピニオン：小野将之（3）

Swiftの開発体制、swift.org／Swift Evolutionリポジトリとは？

―― Swift 3へのロードマップ（後編） ――

次期Swiftに搭載予定の新機能といった最新情報はどこで入手できるのか。Swiftについての情報を常にキャッチアップするために見ておくべきサイトを紹介する。

小野 将之

2016/08/xx

############################################################

　前編ではSwift 3.0のリリース予定内容について紹介した。後編ではそれらの情報をどのように得るのかを紹介する。

# ■Swift 3の開発がどのように進められているか

　以前はSwiftの開発が内部的にどのように進められているかはアップルのコアメンバー以外にはほぼブラックボックスの状態になっていた。オープンソース化以降は開発に関するあらゆる情報が公開され、体系的にまとめられ、オープンかつ統制の取れた開発が行われるようになっている。「そうはいっても、どこをどう見ればよいの？」という疑問が生じるかもしれない。基本的には◆swift.org◇https://swift.org◆と◆Swift Evolutionリポジトリ◇https://github.com/apple/swift-evolution◆の2つに目を通せばよい。

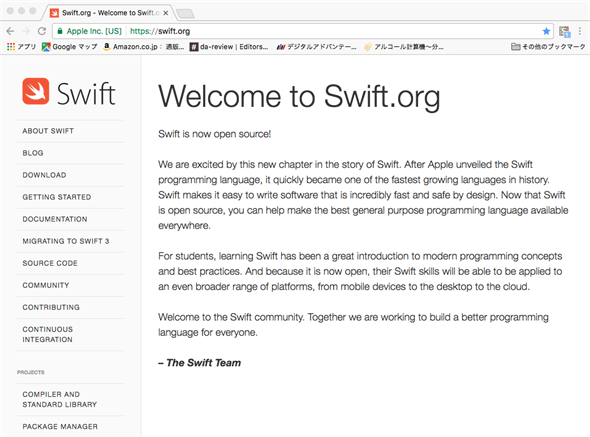
　そこで、今回はこの2つのサイト／リポジトリにどんな情報がまとめられているか、その開発体制はどんなものか、Swiftに貢献するにはどうしたらよいかを見てみよう。

## ●swift.orgとは

　◆swift.org◇https://swift.org◆は、2015年12月3日のSwiftオープンソース化と同時に公開され、Swiftに関する情報のハブ的役割を担っている。

□------------------------------

★画像【01.png】★



□□----------------------------

swift.org

□□□--------------------------

　ここにはあらゆる情報が網羅されているが、特に重要なのは以下のページだ。これらのページにはまず概要が記載され、さらに詳しい内容については別ページへのリンクがあるという構造になっている（かっこ内はswift.org公式ページの左側にあるナビゲーションペーンの項目名）。

・ ①◆▲Swiftとは▲（「ABOUT SWIFT」）◇https://swift.org/about/◆

・ ②◆▲Swiftブログ▲（「BLOG」）◇https://swift.org/blog/◆

・ ③◆▲Swiftでの開発を始めるに当たって▲（「GETTING STARTED」）◇https://swift.org/getting-started/◆

・ ④◆ドキュメント▲（「DOCUMENTATION」）◇https://swift.org/documentation/◆

・ ⑤◆▲コミュニティガイドライン▲（「COMMUNITY」）◇https://swift.org/community/◆

・ ⑥◆▲Swiftへのコントリビューションガイド▲（「CONTRIBUTING」）◇https://swift.org/contributing/◆

　それぞれの内容について簡単にまとめておこう。

　「①▲Swiftとは▲」では「Swiftプロジェクトの目標は、システムプログラミングやモバイル／デスクトップアプリ、クラウドサービスなどあらゆる用途に使えるベストな言語を作ることである」と示されるとともに、「安全」「高速」「表現豊か」という3つの特徴と、ソースコードは全て◆GitHubリポジトリ◇https://github.com/apple◆で管理されていることが述べられている。

　「②▲Swiftブログ▲」ではSwiftの開発状況などが紹介されている。◆Swiftが発表された2014年6月からあった別ブログ◇https://developer.apple.com/swift/blog◆はアップデート情報（新機能紹介など）が中心だが、こちらは、よりコアな内容となっている。

　「③▲Swiftでの開発を始めるに当たって▲」には、特にXcodeを使わずコマンドライン経由でビルド／実行する方法が詳しく記載されている。

　「④▲ドキュメント▲」には◆各ドキュメントがまとめられているページ◇https://developer.apple.com/swift/resources/◆へのリンクが掲載されている。ただし、言語仕様を体系的・網羅的に学びたいときには、◆iBooks Store経由で配布されている電子書籍◇https://itunes.apple.com/us/book-series/swift-programming-series/id888896989?mt=11◆に目を通すことをお勧めする。この他にも◆APIデザインガイドライン◇https://swift.org/documentation/api-design-guidelines/◆へのリンクが掲載されている。

　このAPIデザインガイドラインには◎Swifty△（Appleもセッションや文書で使用おり、△Swiftらしい△という意味）なコードを書くためのポイントが記載されている。前編で述べた通り、現在Swift 3に向けてSwift自身やコアライブラリがあらためてこれに順守するように刷新中であることからも、これは重要なガイドラインである。

　「⑤▲コミュニティガイドライン▲」には◆カテゴリごとに分けられたメーリングリスト◇https://swift.org/community/#mailing-lists◆が紹介されている他（Swiftに関するコミュニケーションは主にメーリングリストを通して行うようになっている）、コミュニティの構造◇https://swift.org/community/#community-structure◆が説明されている。コミュニティの構造についての説明を大まかにまとめると次のようになる。

・ アップルがプロジェクトリードかつ裁定者であり、プロジェクトの指揮を執る

・ Swiftの開発を始めたChris Lattner氏がプロジェクトリードの代表である

・ コアチームメンバーが開発指揮、コードオーナーがコード品質の維持など、責任を持つ人に高い権限がある一方、プルリクエスト／レビューなどはSwift開発に関心を持つ全ての開発者から受け入れている

　コアチームのメンバーは現在6名全員がアップルの社員であるが「コミュニティの貢献によってApple社外の開発者がメンバーに加わることもあり得る」とのことだ。コードオーナーとその担当領域は各レポジトリ直下の◎CODE\_OWNERS.TXT◎に記載されている（例： ◆Swiftレポジトリのコードオーナー◇https://github.com/apple/swift/blob/master/CODE\_OWNERS.TXT◆）。

　Swiftを知るだけでなく開発に貢献したい場合は、「⑥▲Swiftへのコントリビューションガイド▲」に目を通そう。主にコードによる貢献／バグレポート／Swift発展の活動への参加などについて書かれている。

・ ▲◆コードによる貢献◇https://swift.org/contributing/#contributing-code◆：▲ GitHub上でプルリクエストを送る際の、細かい手順／コーディング規約などが記載してあり、対象のプロジェクトがSwiftかどうかに関係なくソフトウェア開発全般に通じる内容も多く参考になる

・ ▲◆バグレポート◇https://swift.org/contributing/#reporting-bugs◆：▲ ◆バグレポートはJiraによって管理◇https://bugs.swift.org/◆されており、アカウントを登録すれば、バグ報告／報告されたバグの閲覧などができる

・ ▲◆Swift発展の活動への参加◇https://swift.org/contributing/#participating-in-the-swift-evolution-process◆：▲ ◆メーリングリスト◇https://lists.swift.org/mailman/listinfo/swift-evolution◆で議論を行い、それが収束すると◆Swift EvolutionリポジトリのProposal（提案）◇https://github.com/apple/swift-evolution/tree/master/proposals◆としてまとめられる（さらにここからレビューがあり、実際にリリースに含めるかなど決定される）

## ●Swiftの開発体制

　次に、◆swift.org◇https://swift.org◆から読み取れるSwiftの開発体制を見ていこう。

　Swift開発の場合、GitHub上で管理されているのは、ソースコード／ドキュメント／各種議論のまとめなど、取りまとめられた情報のみである。途中経過の議論などはGitHub上ではなされていない。オープンソースのライブラリ開発は、GitHubのIssue上で機能追加／仕様変更などについての議論が進められていることが多いが、◆アップルのSwift関連のリポジトリ◇https://github.com/apple◆のほとんどでは、◆Issueが無効化◇https://help.github.com/articles/disabling-issues/◆されている。これは恐らく、Swift開発においてやりとりする情報量が多すぎて、シンプルな作りのGitHub Issueでは収集が付かなくなるという判断がなされたからだろう。

　代わりに、上でも触れた以下の手段が使われている。

・ ▲◆Jiraによって管理されたバグレポート◇https://bugs.swift.org/◆▲

・ ▲◆目的ごとに分かれたメーリングリスト◇https://lists.swift.org◆▲

　次期Swiftに向けた提案やそれについての議論などはメーリングリストでなされるが、バグ報告や明らかに対応した方がベターな既存挙動についての改善提案などはJiraのバグレポートでなされている。

　◆Swift Evolution用のメーリングリスト◇https://lists.swift.org/mailman/listinfo/swift-evolution◆をチェックすることで、次期Swiftの最新動向が把握できる。また、メーリングリストの購読の際、◆Hirundo◇https://stylemac.com/hirundo/◆というmacOS用アプリを使うと、閲覧／検索などがしやすく、お勧めである。

## ●Swift Evolutionとは

　Swiftのロードマップ・現在の開発ステータスなどの管理の場として、◆Swift Evolutionリポジトリ◇https://github.com/apple/swift-evolution◆がある。

　次期Swiftについての込み入った議論はメーリングリスト上で行われると説明したが、それが収束すると、Proposal（提案）としてこのリポジトリに反映される。Proposalはソース管理されており、プルリクエストなどを通じて追加／更新されている（コアチームメンバーは直接◎master◎へコミット可能）。さらに、Proposalのステータスが◆Swiftの変更提案のレビューステータス追跡サイト◇https://apple.github.io/swift-evolution/◆上でまとめられており、どれがSwift 3にすでに実装済みか、どれが今レビュー中かなどが一目で確認できる。

### ○Swift Evolutionに自ら提案していくには？

　そして最後に、自ら提案していくにはどのようにすればよいかについて説明しよう。

　リポジトリ内に、◆運用フロー／ルールなどが記載された文書（英語）◇https://github.com/apple/swift-evolution/blob/master/process.md◆がある。ざっとルールを抜粋すると以下の通りである。

1. これまでの提案で同様のものがないかチェック（現在進行中／リジェクトされたものなども含む）

2. メーリングリストで提案の概要を説明・展開

3. 議論を経て、提案を洗練させていき、◆テンプレート◇https://github.com/apple/swift-evolution/blob/master/0000-template.md◆を使ってまとめる

4. コアチームメンバーにレビュー対象としてもらえるように、Swift EvolutionリポジトリにPull Requestを投げる

5. Pull Requestが通ったら、定められた期間にレビューを受け、採用／不採用の判断がなされる（採用の場合、優先度に応じて順次実装されていく）

　このように提案が採用されるまでには長い道のりがあるが、今後ともSwiftをオープンソースプロジェクトで一貫性のある優れた言語として円滑に進化させていくに当たって、こういった明確なフローが定められているのは重要である。また、例えばメーリングリストを読んでいて気になった点について返信するといったこともフィードバックという大事な貢献なので、そういう小さいところからSwiftプロジェクトに参加していくのもよいだろう。

　過去にリジェクトされた提案は初めにチェックするべきものだが、◆その代表例がまとめられたページもある◇https://github.com/apple/swift-evolution/blob/master/commonly\_proposed.md◆。

　例えば、◆Python◇https://www.python.org◆のように◆◎{}◎をタブで表現する提案◇https://lists.swift.org/pipermail/swift-evolution/Week-of-Mon-20151214/003656.html◆や◆◎&&◎／◎||◎を◎and◎／◎or◎にする提案◇https://lists.swift.org/pipermail/swift-evolution/2015-December/000032.html◆などがある。不採用になった理由を見ることで、Swift開発コミュニティの考えなども読み取れて面白い。基本的に、「今からSwift言語にその仕様変更を加えた場合、そのメリットがデメリットを十分上回るか」という一貫した基準を満たすかどうかで判断されているようだ。あくまでここに載っているのはサマリーであるが、メーリングリストを検索するなどして、過去の実際の議論を全て見ることもできる。

# ■まとめ

　後編では、前編で述べたような情報が◆swift.org◇https://swift.org◆／◆Swift Evolutionリポジトリ◇https://github.com/apple/swift-evolution◆などで事細かにまとめられていることや開発事情などを紹介した。第1回～今回（第3回）まではほとんどSwiftコードも出さずに全体的な動向の説明だったが、次回以降の連載でSwift 3でなされる変更から注目すべき点を取り上げて具体的に紹介していく。