

기업이 iOS 앱 개발에 Swift 5를 사용해야 하는 이유는 무엇입니까?



패트릭 R | 2020년 7월 14일



파트너 포스트 - [Intuz](#) Consulting Beyond Technology



Why Should **BUSINESSES** Use **SWIFT 5** for iOS App Development?



Swift는 개발자가 쉽게 배우고 구현할 수 있는 강력한 프로그래밍 언어입니다. 이 언어는 macOS, watchOS, tvOS 및 iOS 등에서 사용할 수 있습니다. Swift가 가지고 있는 코드는 구문이 간결하고 표현력이 풍부하기 때문에 배우고 구현하고 상호 작용하기가 매우 쉽습니다. 여기에는 개발자가 흥미롭고 재미있는 방식으로 사용하는 것을 좋아하는 최신 기능이 포함되어 있습니다. Swift가 제공하는 코드는 번개처럼 빠르게 실행되는 설계 및 개발된 소프트웨어로 인해 안전합니다. 올바른 iOS 앱 개발 서비스 및 프로세스를 찾는 것은 결국 예상되는 결과를 얻는 데 비즈니스에 도움이 될 것입니다.

Swift 5 및 5.1 업데이트의 새로운 기능

Apple은 새로운 iOS 14를 강력하게 지원하는 [새로운 Xcode 11](#)을 출시했습니다. Apple은 개발 언어인 Swift 5.1의 새 버전을 지원하는 새 버전의 IDE를 출시했습니다.

새 버전의 Swift 5.1은 새로운 언어 기능과 향상된 모듈 안정성을 통합하여 출시되었습니다. 이것은 모듈 안정성을 얻기 위한 지원을 추가한 언어에 도움이 되는 다른 사람들과 I를 공유하기 위한 바이너리 프레

임워크 생성을 가능하게 합니다.

모듈 안정성에 대해 이야기하면 모든 바이너리 프레임워크의 API를 설명하는 텍스트 기반 모듈 인터페이스 파일을 정의하고 다양한 버전의 컴파일러를 사용하여 코드의 본질로 컴파일할 수 있습니다.

다음은 iOS 모바일 애플리케이션 개발에 Swift를 사용하는 브랜드 중 일부입니다.

- 느슨하게
- 우버
- 리프트
- 로빈 후드
- 딜리버리히어로
- ViaVarejo
- 베프로컴퍼니
- 루앙구루 등...

Swift 5.1의 주요 언어 업데이트



바이너리 호환성 및 안정적인 ABI

ABI는 새로운 Swift 5 앱 개발 요구에 대해 안정적이었고 이것이 Swift 라이브러리가 iOS, macOS, watchOS 및 tvOS를 통합한 이유입니다. 개발하기 더 쉽고 크기가 매우 작아 어떤 앱에도 포함되지 않았습니다. 도서관.

표준 라이브러리 업데이트

- 적절한 유형의 컬렉션에서 diff를 처리하고 업데이트하는 지원을 제공합니다.
- 향상된 유연성 및 어레이 초기화
- 연속 문자열의 생성 및 관리를 다루고 String.Index 및 Range에 대한 유니코드 텍스트 및 일반 이니셜라이저 작업을 지원하는 문자열 작업을 간소화하기 위한 추가 API
- 확장 벡터, 축소 및 벡터 스위즐 지원을 포함하는 SIMD 유형과 함께 작동하기 위한 증분식 API 개선

의 변형

- 고유 식별자가 필요한 엔터티를 지원하기 위한 식별 프로토콜

추가 컴파일러

Swift 개발을 사용하면 디버깅을 위해 메모리에 쉽게 액세스하고 빌드를 릴리스해야 합니다. Python, JavaScript 및 Ruby와 같은 동적 언어와의 상호 운용성을 향상시키는 데 도움이 되는 동적으로 호출 가능한 유형에 대한 막대한 지원을 제공합니다. 언어 프로토콜을 구현하고 향후 열거 사례를 처리하는 데 도움이 됩니다. 또한 사용자 정의 동적 "호출 가능" 유형과 강제를 사용한 리터럴 초기화를 도입합니다. 컴파일 조건에서 "미만" 연산자를 완벽하게 지원하고 키 경로를 식별하는 데 도움이 됩니다.

패키지 관리자 업데이트

Swift 패키지 관리자는 Swift 5 애플리케이션 개발 버전에 포함된 일련의 새로운 브랜드 기능으로 구성되며 종속성 미러링, 대상별 빌드 설정, 사용자 정의된 배포 대상을 다루고 코드 평균 데이터 생성을 위한 전체 기능을 보유합니다. 또한 Swift는 실행 파일 빌드와 함께 REPL에서 라이브러리를 가져오는 기능을 포함하는 명령 실행을 선택합니다. Swift 5는 종속성 미러링, 대상별 빌드 설정 및 플랫폼 배포 설정을 포함하는 전체 Swift 진화 프로세스에서 패키지 관리자 제안을 구현합니다.

모듈 안정성

Swift 5.1은 모듈 안정성에 대한 언어의 추가 지원에 도움이 되는 다른 사람과 공유할 수 있는 이진 프레임워크를 만드는 데 도움이 됩니다. 개발자가 iOS 기반 응용 프로그램을 매우 쉽고 빠르게 만들 수 있습니다. Swift 5.1은 바이너리 프레임워크 API를 설명하는 새로운 브랜드 텍스트 기반 모듈 인터페이스 파일을 정의하고 다양한 컴파일러 버전의 완전한 지원과 코드로 컴파일할 수 있도록 합니다.

언어 서버 프로토콜

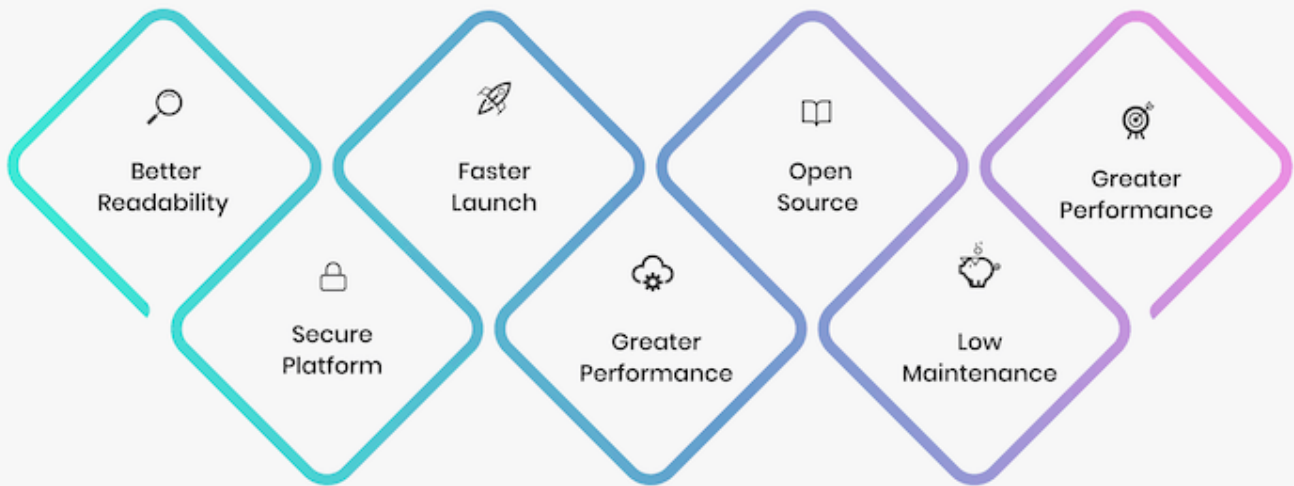
Ubuntu 및 macOS용 Swift 5.1 OSS 도구 체인 패키지는 Swift 및 C 기반 언어용 LSP(Language Server Protocol) 구현인 SourceKit-LSP용 바이너리를 다룹니다.

SwiftSyntax 업데이트

Swift 컴파일러의 파서를 사용하여 성능 향상에 중점을 두고 재설계되었습니다. 전체 내부 데이터 구조를 보다 효율적으로 재설계하여 구문 트리 방문 및 이와 관련된 작업의 성능이 향상되었습니다.

iOS 앱 개발에 Swift 5를 사용할 때의 이점

Benefits of using Swift 5 for iOS App Development



더 작은 앱 크기

Swift 5는 언어를 바이너리와 호환되도록 합니다. 즉, Swift 모바일 개발의 최종 결과는 macOS, iOS 12.2, watchOS 5.2 및 tvOS 12.2와 같은 새로운 운영 체제를 실행하는 사용자에게 더 작은 혜택을 줄 것입니다.

더 빠른 실행

Swift 5의 또 다른 이점은 모든 동적 라이브러리가 메모리에 캐시되고 앱 간에 공유되는 앱에서 활용되므로 시작 시간이 더 빠르다는 것입니다.

더 뛰어난 성능

사용자는 고장난 앱과 결함이 있는 앱 정지 상태에서 작업하기를 원하지 않으며 이것이 앱 제거율의 주요 원인이며 기업이 이를 피하는 것이 중요합니다. Swift 5의 API 안정성은 앱이 동작하고 향상된 성능을 제공하도록 지원합니다.

더 나은 가독성

Swift 5를 선택한 가장 중요한 이유 중 하나는 가독성 때문입니다. 수정, 쓰기 및 읽기가 간단하며 고유하게 만드는 깔끔한 구문이 있습니다. iOS 앱에 비해 Objective-C에 비해 적은 수의 코드 라인이 필요함

니다. 신생 기업 및 기업은 iOS 모바일 애플리케이션 개발의 필요성과 앱 개발 비용 및 전체 개발 시간을 줄이기 위해 Swift를 선택합니다.

낮은 유지 보수

앱이 게시되는 즉시 지속적인 유지 관리가 필요하며 때때로 유지 관리가 필요합니다. 예를 들어, 앱이 사용자에게 도달하면 새 콘텐츠를 추가하고 전체 기능을 작동하기 위해 새 기능을 앱에 통합하기 위한 성능을 추적해야 합니다.

오픈 소스

Swift는 오픈 소스 기술이며 2015년에 발표되었습니다. 다양한 플랫폼에서 Swift를 사용하는 새로운 방법을 엽니다. 오픈 소스는 디자인과 구조에서 점점 더 많은 개선이 이루어질 수 있음을 의미합니다. 향상된 기능과 향상된 성능으로 미래의 모바일 앱을 만들 수 있습니다.

안전한 플랫폼

Swift는 오늘날 가장 빠르게 성장하는 모바일 앱 세계에서 모두에게 필요한 안전한 모바일 앱 개발 플랫폼입니다. Objective-C에서 발생할 수 있는 오류를 방지하는 방언 및 구조화된 개발이 있습니다.

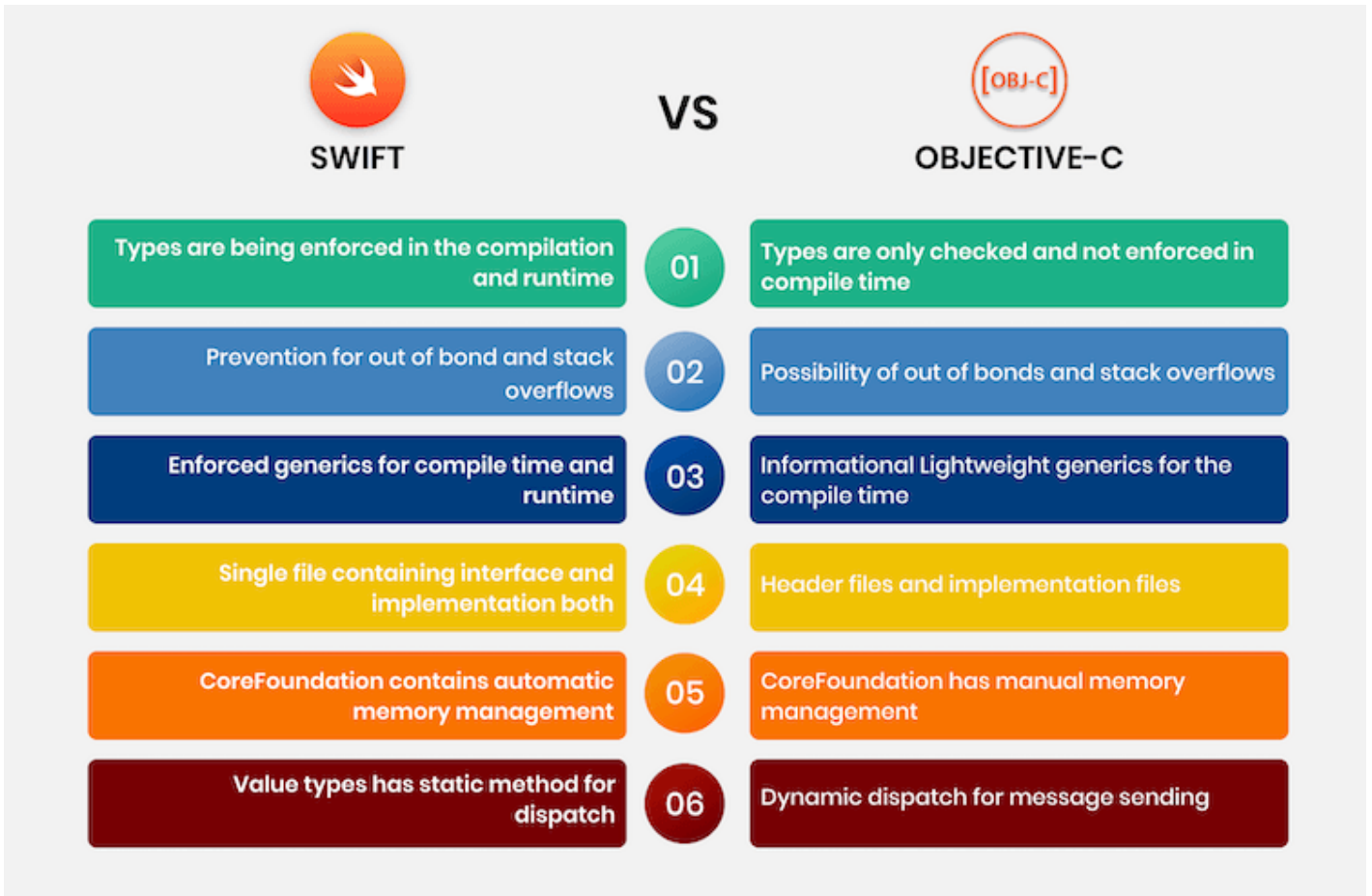
Swift 5.1 업데이트가 iOS 앱 개발 프로세스에 어떤 영향을 미칩니까?

iOS 앱 제작자에 대해 이야기하면 Swift는 가장 강력한 프로그래밍 언어로 간주됩니다. 2018년 Swift는 가장 인기 있는 프로그래밍 언어로 14위에 올랐습니다.

Swift 5.1을 사용하면 공유 바이너리 프레임워크를 구축할 수 있습니다. 그 기능을 통해 프로토콜 확장이 좋고 코드 작성을 쉽게 하는 더 나은 디자인과 API를 만들 수 있습니다. Swift 5.1 출시 이후 개발자가 앱 개발 프로세스를 수행하기가 더 쉬워졌습니다. Swift 및 iOS 앱 제작에 대한 방대한 지식을 보유한 개발자는 전체 앱 구축 프로세스를 쉽고 완벽하게 보안 상태로 처리할 수 있습니다.

Objective-C보다 빠른 Swift - 살펴보겠습니다 -

Swift 및 Objective-C는 이러한 프로그래밍 언어 모두 iOS 앱 개발에 사용됩니다. 이러한 관리는 모두 LLVM 컴파일러의 인프라를 기반으로 하며 두 가지 모두에 대한 단일 iOS SDK만 있습니다. 이것이 이러한 언어가 Cocoa 프레임워크와 작동하는 방식에 큰 차이가 없는 이유입니다.



Swift는 C 언어의 모든 제한 사항을 제거하고 고급 기술을 사용하여 개선되었기 때문에 Objective-C보다 빠르다고 말할 수 있습니다. Swift는 더 빠르게 작동하고 아름답고 기능이 풍부한 iOS 모바일 애플리케이션을 만들기 위해 의도적으로 설계되었습니다.

Swift 5.1의 다음 버전은 무엇입니까?

앞으로 Swift는 프로그래밍 언어에서 더 많은 기술적 변화를 겪게 될 것입니다. Swift가 해당 영역에서 많은 잠재적인 개선 사항을 가지고 있으며 전체 작성 및 실행이 Swift 코드를 더 즐겁게 만들고 Swift 프로그래밍 언어에 대한 새로운 사용 사례를 계속 잠금 해제할 수 있다는 것은 의심의 여지가 없습니다.

새로운 진단 엔진은 Swift 5.2용으로 이미 개발되었으며, 표현식이 컴파일되지 않을 때 Swift 컴파일러에서 생성되는 유형 검사기와 모든 유형의 경고 및 오류 메시지를 개선하는 것을 목표로 합니다. Swift 5.1의 새로운 기능은 더 매력적이고 기능이 풍부한 앱을 만드는 데 도움이 될 것이므로 최고의 iOS 앱 개발 서비스가 필요합니다.

여기에서 우리가 공유한 내용이 매우 통찰력이 있습니까? Swift 모범 사례, 팁 및 요령에 대해 자세히 알아보고 싶으신가요? [iOS 앱 개발에 대한 Intuz의 최종 가이드를 살펴보세요](#).