

About Swift

스위프트에 대해서

문장구조
단어 전체사구 등등
동사

Swift is a fantastic way to write software, whether it's for phones, desktops, servers, or anything else that runs code.

스위프트는 소프트웨어를 작성하기 위한 확장적인 방법입니다. 그것이 코드로 작동하는 것이라면, 핸드폰을 위한 것이든, 데스크탑, 서버, 혹은 어떤 것이라도 상관 없습니다.

It's a safe, fast, and interactive programming language that combines the best in modern language thinking with wisdom from the wider Apple engineering culture and the diverse contributions from its open-source community.

그것은 안전성, 속도, 그리고 상호작용하는 프로그래밍 언어입니다.
현대 언어 사고방식 중 최고의 것과, 폭 넓은 애플 기술 문화로부터 오는 지혜와 오픈 소스 커뮤니티로부터의 다양한 기여들이 합쳐진 language

combine A with B: A와 B를 병합하다

The compiler is optimized for performance and the language is optimized for development, without compromising on either.

<의문문, 조건문> 어느 쪽이
<부정문> 어느 쪽의 시도 (아니다)

어느 쪽과도 타협하지 않고, 컴파일러는 성능에 최적화 되어 있고, 언어는 개발에 최적화 되어 있습니다.

주목보어..?

Swift is friendly to new programmers.

스위프트는 새로운 개발자에게 친근합니다.

It's an industrial-quality programming language that's as expressive and enjoyable as a scripting language.

as A as B : B만큼이나 A하다

스크립트 언어 만큼이나 표현력이 풍부하고 재밌는, 공업용 수준의 프로그래밍 언어입니다.

동명사 → 주어

반동사 (사역동사)

~를 시도하다

Writing Swift code in a playground lets you experiment with code and see the results immediately, without the overhead of building and running an app.

플레이그라운드에서 스위프트 코드를 작성하는 것은, 당신이 앱을 빌드하거나 실행시키지 않고, 코드를 실험해보고 결과를 즉각적으로 볼 수 있게 해줍니다.

참고: <https://forum.wordreference.com/threads/define-away.2386502/>

Swift defines away large classes of common programming errors by adopting modern programming patterns:

스위프트는 현대 프로그래밍 패턴을 채택함으로써, 흔한 프로그래밍 에러의 많은 종류들을 정의합니다. 정의합니다..? 수정합니다..? 규정합니다..? 해결합니다..?

- Variables are always initialized before use.

- 변수는 언제나 사용하기 전에 초기화 됩니다.

- Array indices are checked for out-of-bounds errors.

- 배열 인덱스들은 범위 초과 에러를 위한 체크를 합니다.

- Integers are checked for overflow.

- 정수는 넘치는 것을 막기 위한 체크를 합니다.

- Optionals ensure that nil values are handled explicitly.

- 옵셔널은 nil 값이 명시적으로 다뤄지는 것을 보증합니다.

- Memory is managed automatically.

- 메모리는 자동으로 관리됩니다.

- Error handling allows controlled recovery from unexpected failures.

- 에러 핸들링은 기대하지 않은 실패로부터, 제어되는 복구를 허락합니다.

Swift code is compiled and optimized to get the most out of modern hardware.

스위프트 코드는 현대 하드웨어에서 최대의 결과를 얻기 위해 컴파일 되고, 최적화 됩니다.

The syntax and standard library have been designed based on the guiding principle that the obvious way to write your code should also perform the best.

선택스와 스탠다드 라이브러리는 당신이 코드를 작성하는 명백한 방법 또한 가장 잘 작동해야 한다는 가이드 이론을 기반으로 디자인 되었습니다.

Its combination of safety and speed make Swift an excellent choice for everything from "Hello, world!" to an entire operating system.

이 안정성과 스피드의 결합이, "Hello, world!" 부터 전체 오퍼레이팅 시스템에 이르기까지 모든 것에 대해서, 스위프트가 뛰어난 선택이 되도록 만듭니다.

allow + 목적어 + to do : 목적어에게 ~하는 것을 허락한다

Swift ^Acombines ^Bpowerful type ^{추론}inference and pattern matching ^{동사}with ^{복잡함?}a modern, lightweight syntax ^{목적어}allowing ^{간결한 방법}(complex ideas) to be expressed in a clear and concise manner.

스위프트는 강력한 타입 추론과 패턴 매칭을 현대적이고 가벼운 구문과 결합합니다.
복잡한 아이디어들을 명확하고 간결한 방법으로 표현되도록 하면서? 하기 위해?

As a result, ^{also}code is not just easier to write, but ^{생략됨}easier to read and maintain ^{또한}as well.

그 결과로써, 코드는 쓰기가 더 쉬울 뿐 아니라 읽기와 유지보수 또한 더 쉽습니다.

Swift ^{만드는 데}has been years in the making, and it ^{가능성}continues to evolve with new features and capabilities.

스위프트는 만들어지는 데 몇년이 걸렸고, 새로운 기능과 가능성을 계속 개선 시키고 있습니다.

Our goals ^{아심적인, 어마어마한}for Swift are ambitious.

스위프트를 위한 우리의 목표는 어마어마 합니다.

We can't wait to see ^{가능성}what you create with it.

우리는 당신이 그것으로 만들 것을 보고 싶어서 기다릴 수 없습니다.