

# Live Coding

# Kotlin/Native

# Snake



[github.com/dkandalov/kotlin-native-snake](https://github.com/dkandalov/kotlin-native-snake)



[@dmitrykandalov](https://twitter.com/dmitrykandalov)





# Lightning talk



**what is  
Kotlin/Native?**

◀ Overview

- [Kotlin for Server Side](#)
- [Kotlin for Android](#)
- [Kotlin for JavaScript](#)
- [Kotlin for Native](#)
- [Coroutines](#)
- [Multiplatform](#)
- [What's New in 1.1](#)
- [What's New in 1.2](#)
- [What's New in 1.3](#)

▶ Getting Started

▶ Basics

▶ Classes and Objects

▶ Functions and Lambdas

▶ Collections

▶ Multiplatform Programming

▶ Other

▶ Core Libraries

▶ Reference

▶ Java Interop

▶ JavaScript

▶ Native

# Kotlin/Native for Native

 Edit Page



## Welcome to the native world

Kotlin/Native

Kotlin/Native is a technology for compiling Kotlin code to native binaries, which can run without a virtual machine. It is an [LLVM](#) based backend for the Kotlin compiler and native implementation of the Kotlin standard library.

## Why Kotlin/Native?

Kotlin/Native is primarily designed to allow compilation for platforms where *virtual machines* are not desirable or possible, for example, embedded devices or iOS. It solves the situations when a developer needs to produce a self-contained program that does not require an additional runtime or virtual machine.

## Target Platforms

◀ Overview

- [Kotlin for Server Side](#)
- [Kotlin for Android](#)
- [Kotlin for JavaScript](#)
- [Kotlin for Native](#)
- [Coroutines](#)
- [Multiplatform](#)
- [What's New in 1.1](#)
- [What's New in 1.2](#)
- [What's New in 1.3](#)

▶ Getting Started

▶ Basics

▶ Classes and Objects

▶ Functions and Lambdas

▶ Collections

▶ Multiplatform Programming

▶ Other

▶ Core Libraries

▶ Reference

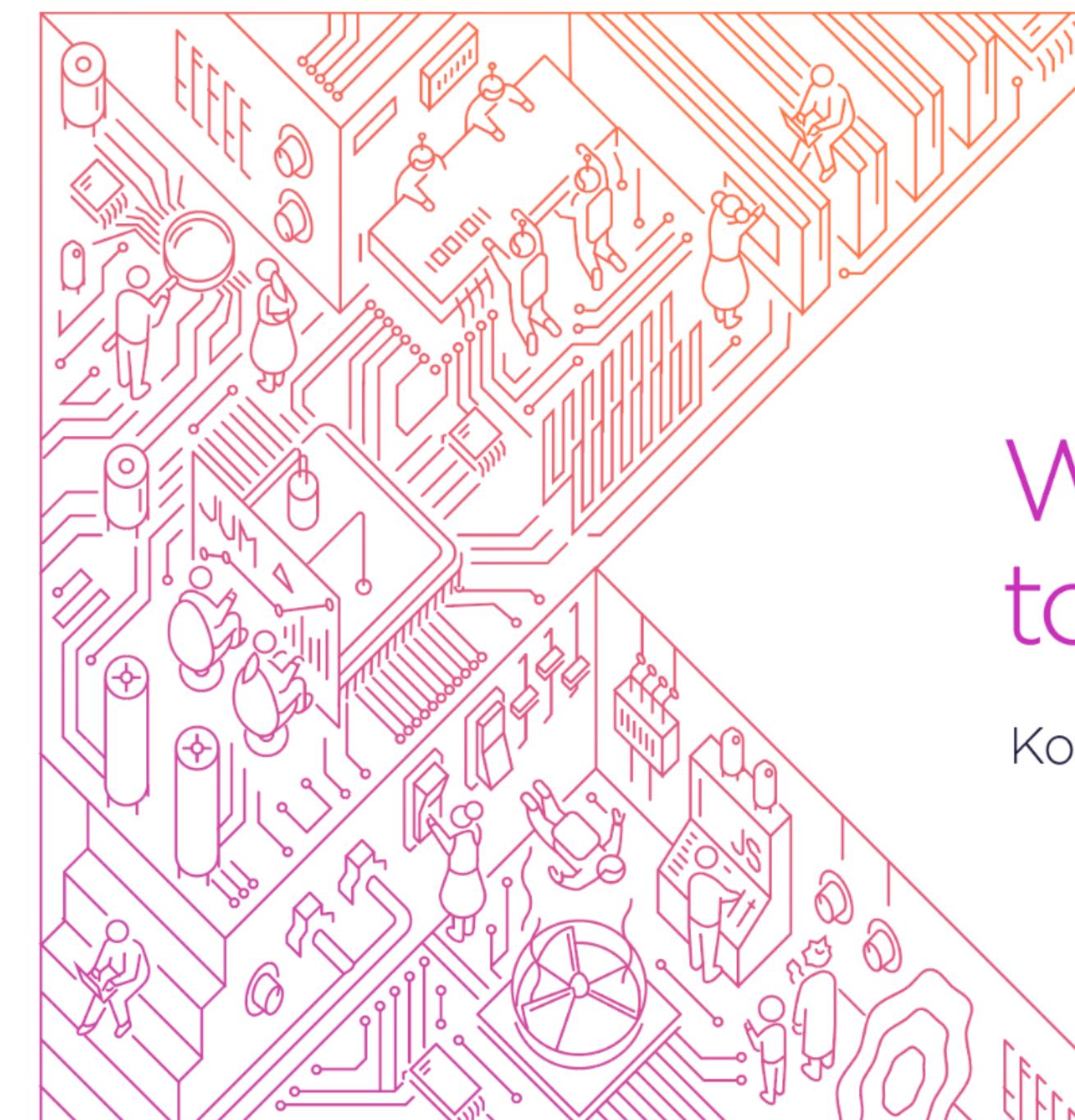
▶ Java Interop

▶ JavaScript

▶ Native

# Kotlin/Native for Native

 Edit Page



Welcome  
to the native world

Kotlin/Native

# BETA

Kotlin/Native is a technology for compiling Kotlin code to native binaries, which can run without a virtual machine. It is an [LLVM](#) based backend for the Kotlin compiler and native implementation of the Kotlin standard library.

## Why Kotlin/Native?

Kotlin/Native is primarily designed to allow compilation for platforms where *virtual machines* are not desirable or possible, for example, embedded devices or iOS. It solves the situations when a developer needs to produce a self-contained program that does not require an additional runtime or virtual machine.

## Target Platforms

JetBrains/kotlin-native: Kotlin/ Native

GitHub, Inc. [US] | github.com/JetBrains/kotlin-native

Search or jump to... Pull requests Issues Marketplace Explore Watch 262 Star 6,131 Fork 492

Code Issues 154 Pull requests 29 Projects 0 Wiki Security Insights

### Kotlin/Native infrastructure

kotlin compiler llvm objective-c c

5,238 commits 195 branches 567 releases 100 contributors Apache-2.0

Branch: master New pull request Create new file Upload files Find File Clone or download

Author	Commit Message	Time Ago
knebekaizer	Follow-up KT-32929: add test dependency	Latest commit 09b21d6 9 hours ago
Interop	Move ABI-specific parts of forward interop to the separate file.	2 days ago
backend.native	Follow-up KT-32929: add test dependency	9 hours ago
build-tools	Added library in benchmarks (#3349)	14 hours ago
cmd	CLI library rework (#3215)	11 days ago
common	Copyright update. (#1944)	last year
dependencies	[REVERTME] filtering out ANDROID_X64, WATCHOS_{X64,ARM64}, TVOS_{X64,...	22 days ago
endorsedLibraries	Rewrote some samples using kotlinx.cli (#3351)	9 hours ago
extracted	CLI library rework (#3215)	11 days ago
gradle	Update Gradle: 5.5	2 months ago
klib	CLI library rework (#3215)	11 days ago
konan	[watchOS][targets] Add support for 32bit Apple Watch.	2 days ago
llvm	Older commit removed due to size (#3324)	5 months ago

Releases · JetBrains/kotlin-native

GitHub, Inc. [US] | github.com/JetBrains/kotlin-native/releases

Search or jump to... Pull requests Issues Marketplace Explore

Watch 262 Star 6,131 Fork 492

Code Issues 154 Pull requests 29 Projects 0 Wiki Security Insights

Releases Tags

build-1.3.60-dev-2260 ...  
a1e26bd zip tar.gz

build-1.3.60-dev-2256 ...  
d153f21 zip tar.gz

build-1.3.60-dev-2250 ...  
d153f21 zip tar.gz

build-1.3.60-dev-2247 ...  
d153f21 zip tar.gz

build-1.3.60-dev-2246 ...  
d153f21 zip tar.gz

build-1.3.60-dev-2242 ...  
d153f21 zip tar.gz

build-1.3.60-dev-2240 ...  
d153f21 zip tar.gz

*Kotlin/Native* is an LLVM  
backend for the Kotlin compiler,  
runtime implementation, and  
native code generation facility  
using the LLVM toolchain.



*Kotlin/Native* is an LLVM  
backend for the Kotlin compiler,  
runtime implementation, and  
native code generation facility  
using the LLVM toolchain.



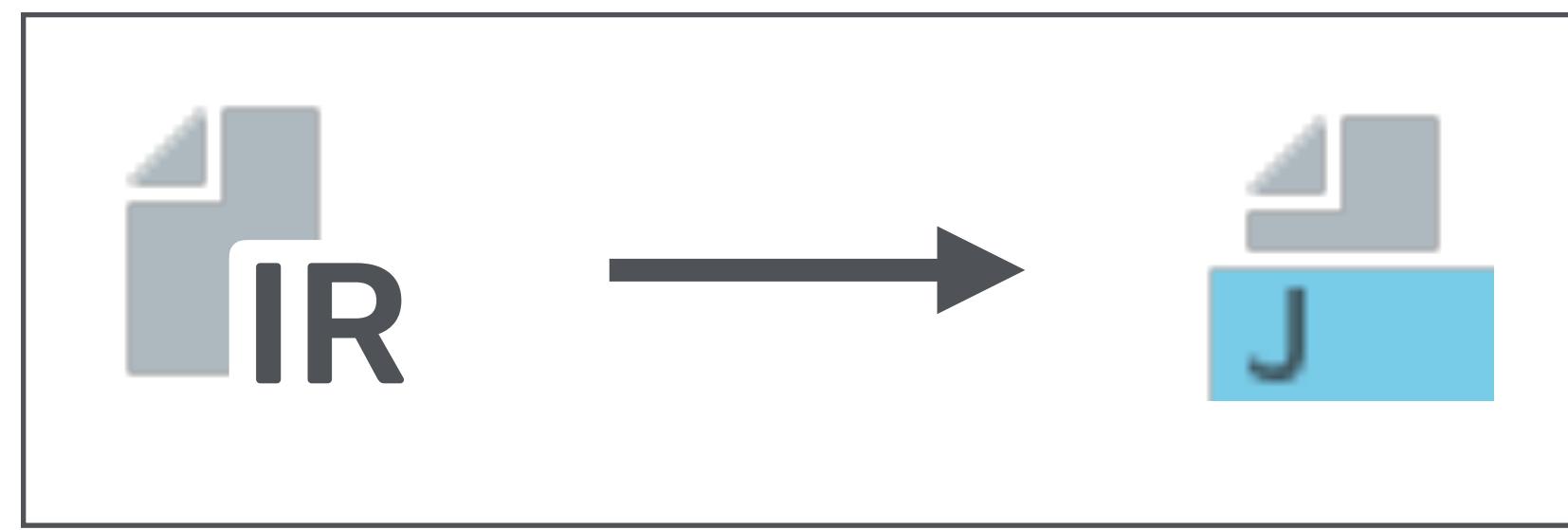




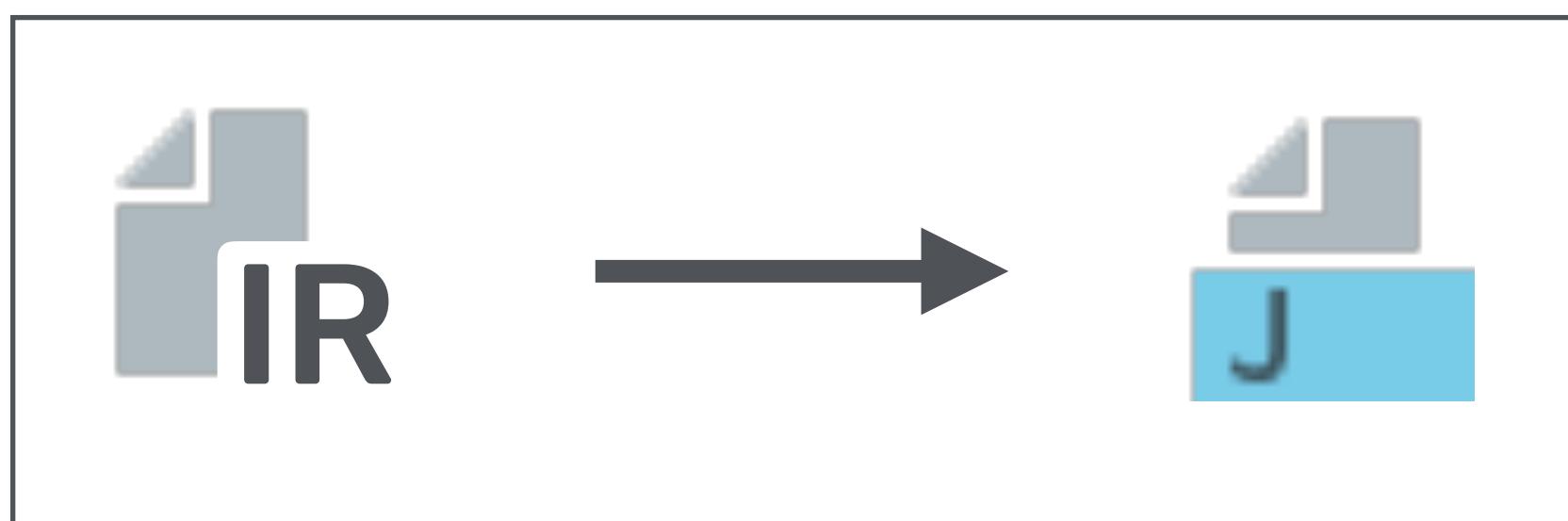
# Compiler frontend



# Compiler backend

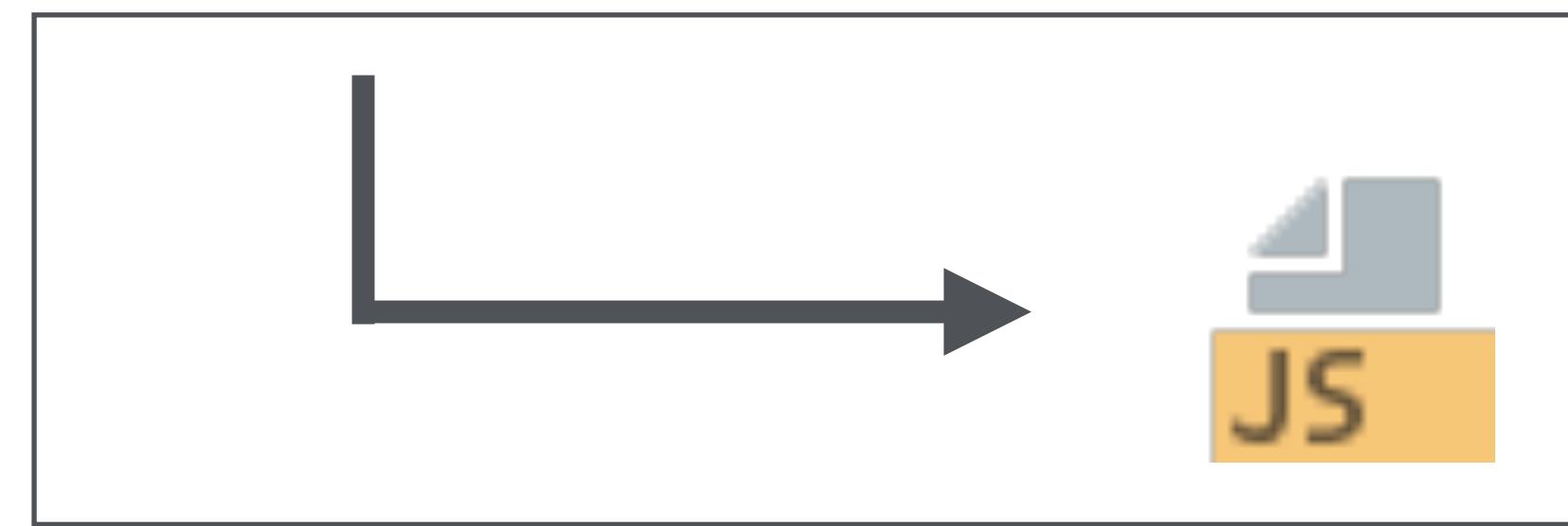


# JVM backend

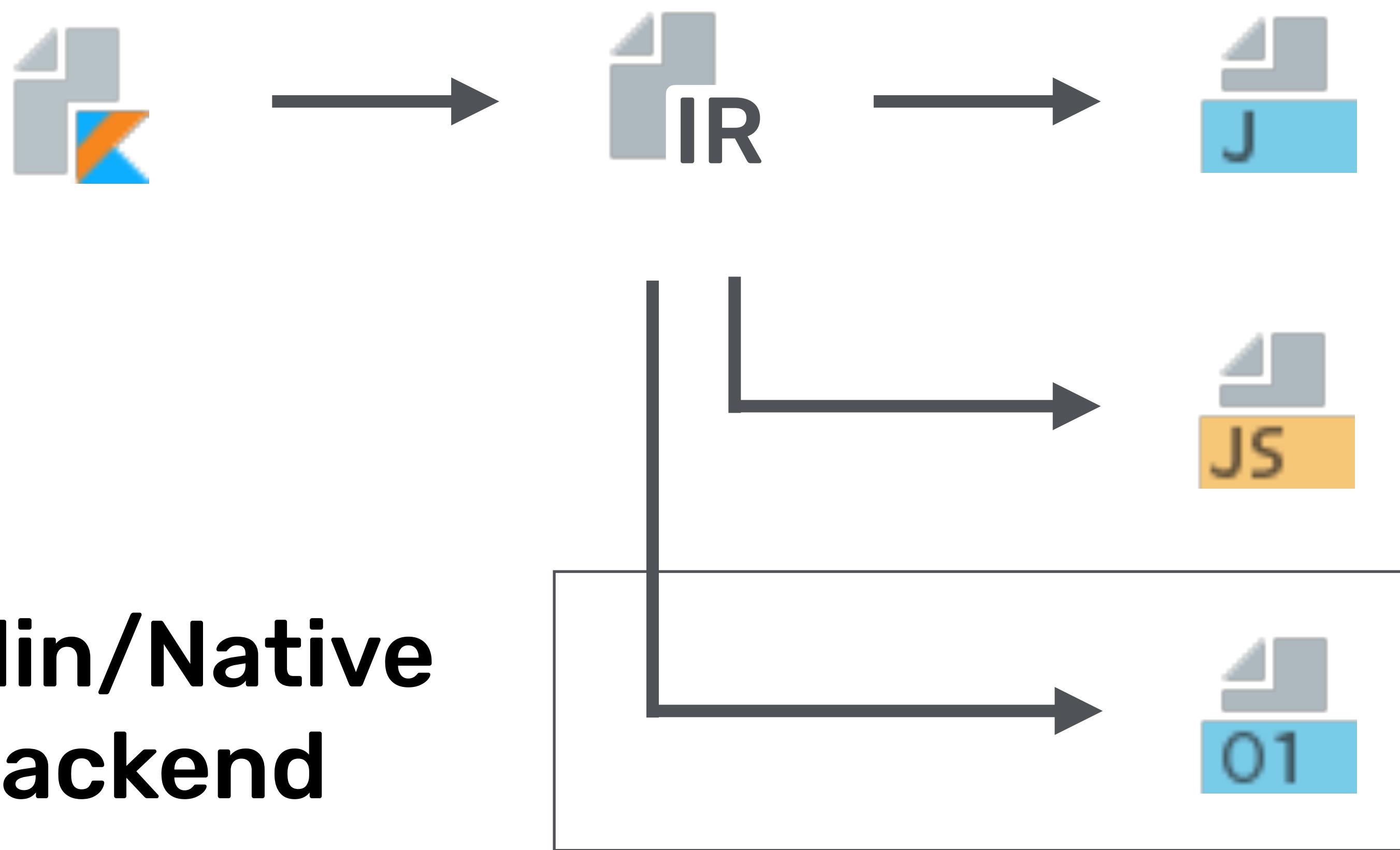




**JavaScript  
backend**



**Kotlin/Native  
backend**



# Kotlin/Native backend



# Kotlin/Native



*Kotlin/Native* is an LLVM  
backend for the Kotlin compiler,  
runtime implementation, and  
native code generation facility  
using the LLVM toolchain.

# LLVM



# The LLVM Compiler Infrastructure

## Site Map:

[Overview](#)  
[Features](#)  
[Documentation](#)  
[Command Guide](#)  
[FAQ](#)  
[Publications](#)  
[LLVM Projects](#)  
[Open Projects](#)  
[LLVM Users](#)  
[Bug Database](#)  
[LLVM Logo](#)  
[Blog](#)  
[Meetings](#)  
[LLVM Foundation](#)

## Download!

Download now:  
[LLVM 8.0.1](#)  
[All Releases](#)  
[APT Packages](#)  
[Win Installer](#)

View the open-source  
[license](#)

## Search this Site

  
Search!

## Useful Links

Mailing Lists:  
[LLVM-announce](#)  
[LLVM-dev](#)  
[LLVM-bugs](#)  
[LLVM-commits](#)  
[LLVM-branch-commits](#)  
[LLVM-testresults](#)

## LLVM Overview

The LLVM Project is a collection of modular and reusable compiler and toolchain technologies. Despite its name, LLVM has little to do with traditional virtual machines. The name "LLVM" itself is not an acronym; it is the full name of the project.

LLVM began as a [research project](#) at the [University of Illinois](#), with the goal of providing a modern, SSA-based compilation strategy capable of supporting both static and dynamic compilation of arbitrary programming languages. Since then, LLVM has grown to be an umbrella project consisting of a number of subprojects, many of which are being used in production by a wide variety of [commercial and open source](#) projects as well as being widely used in [academic research](#). Code in the LLVM project is licensed under the "[Apache 2.0 License with LLVM exceptions](#)"

The primary sub-projects of LLVM are:

1. The **LLVM Core** libraries provide a modern source- and target-independent [optimizer](#), along with [code generation support](#) for many popular CPUs (as well as some less common ones!) These libraries are built around a [well specified](#) code representation known as the LLVM intermediate representation ("LLVM IR"). The LLVM Core libraries are [well documented](#), and it is particularly easy to invent your own language (or port an existing compiler) to use [LLVM as an optimizer and code generator](#).
2. **Clang** is an "LLVM native" C/C++/Objective-C compiler, which aims to deliver amazingly fast compiles (e.g. about [3x faster than GCC](#) when compiling Objective-C code in a debug configuration), extremely useful [error and warning messages](#) and to provide a platform for building great source level tools. The [Clang Static Analyzer](#) is a tool that automatically finds bugs in your code, and is a great example of the sort of tool that can be built using the Clang frontend as a library to parse C/C++ code.
3. The **LLDB** project builds on libraries provided by LLVM and Clang to provide a great native debugger. It uses the Clang ASTs and expression parser, LLVM JIT, LLVM disassembler, etc so that it provides an experience that "just works". It is also blazing fast and much more memory efficient than GDB at loading symbols.
4. The **libc++** and **libc++ ABI** projects provide a standard conformant and high-quality C++11 and C++14 library implementation.

## Latest LLVM Release!

**1 August 2019:** LLVM 8.0.1 is now [available for download](#)! LLVM is publicly available under an open source [License](#). Also, you might want to check out [the new features](#) in SVN that will appear in the next LLVM release. If you want them early, [download LLVM](#) through anonymous SVN.

## ACM Software System Award!

LLVM has been awarded the **2012 ACM Software System Award**! This award is given by ACM to *one* software system worldwide every year. LLVM is [in highly distinguished company](#)! Click on any of the individual recipients' names on that page for the detailed citation describing the award.

## GitHub Migration

The LLVM Project is migrating its source code from SVN to GitHub. SVN will be retired on Oct 21, 2019. Visit the [GitHub Migration Status](#) page for more information.

## Upcoming Releases

### LLVM Release Schedule:

- 9.0.0:
  - 18 July 2019: branch, then rc1
  - 14 August: rc2
  - 28 August: final

**C/C++ (clang)**

**Objective-C**

**Rust**

**Swift**

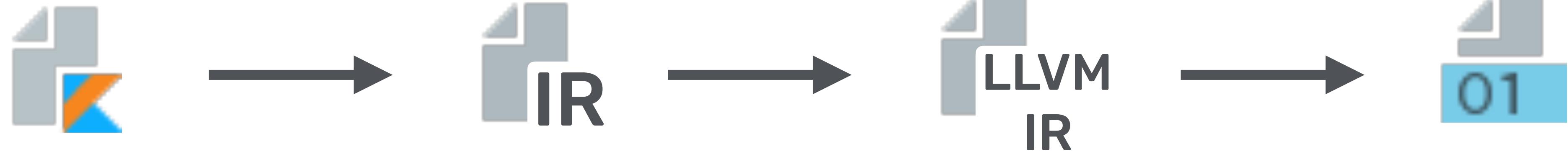
**Julia**

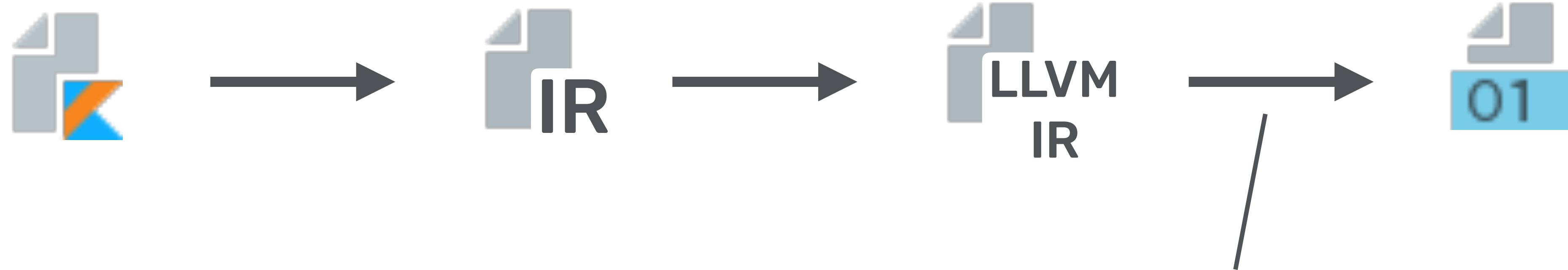
**Graal VM**

# Kotlin/Native backend









**Done by LLVM**

*Kotlin/Native* is an LLVM  
backend for the Kotlin compiler,  
runtime implementation, and  
native code generation facility  
using the LLVM toolchain.

**kotlin.String** → **java.lang.String**

**kotlin.String** → **java.lang.String**

→ **JS string**

**kotlin.String** → **java.lang.String**

→ **JS string**

→ **KString**

**kotlin.\***



**JVM**



**JS**



**Native**

- reference counting
- native-specific code

*Kotlin/Native* is an LLVM  
backend for the Kotlin compiler,  
runtime implementation, and  
native code generation facility  
using the LLVM toolchain.



**why?**

# **1. Performance**

High Performance Managed La X +

← → C https://www.infoq.com/presentations/performance-managed-languages

🔍 ⭐ 🙏 ⋮

# High Performance Managed Languages

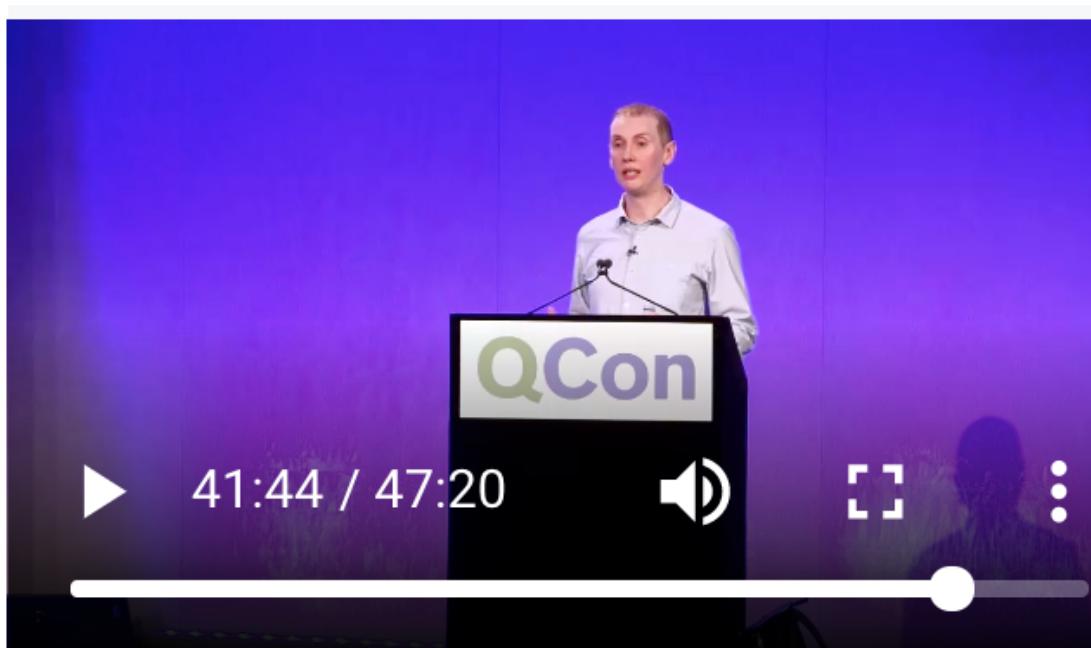
LIKE

DISCUSS

BOOKMARKS

■ ■ ■

1X 1.5X 2X



MP3 SLIDES ANDROID APP  
47:20



## High Performance Managed Languages

Martin Thompson - @mjpt777

### Summary

Martin Thompson explores how their managed runtimes can equal, and even better in some cases, the performance of native languages.

### Bio

Martin Thompson is a Java Champion with over 2 decades of experience building complex and high-performance computing systems. He is most recently known for his work on Aeron and SBE. He blogs at [mechanical-sympathy.blogspot.com](http://mechanical-sympathy.blogspot.com), and can be found giving training courses on performance and concurrency when he is not cutting code to make systems better.

# **2. CAPIs**

- POSIX
- native OS UI
- interop between languages
  - C libraries

(search for “awesome-c”)

# 3. Kotlin

End of the  
Lightning talk





Live coding

# Snake (video game genre)

From Wikipedia, the free encyclopedia

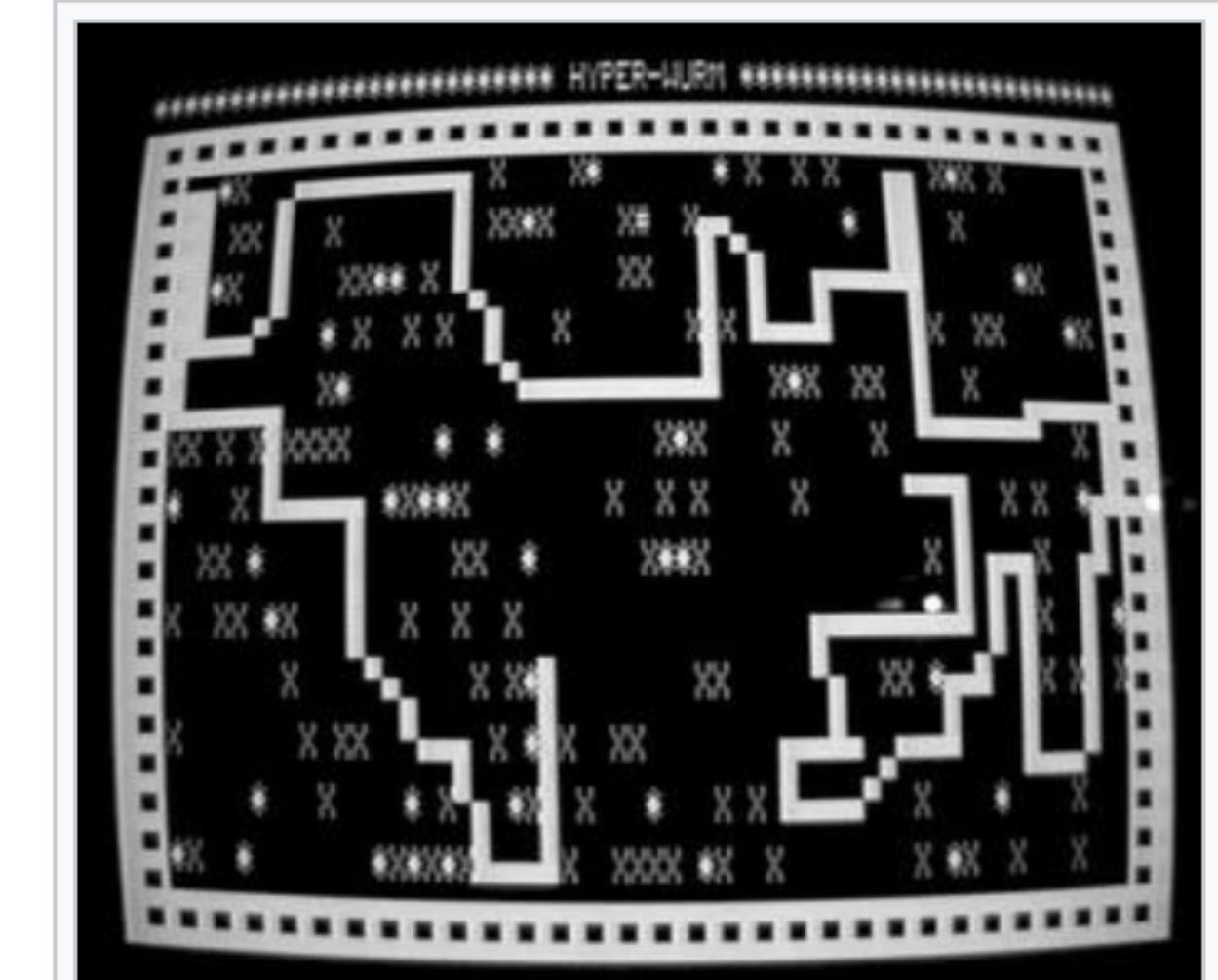
(Redirected from [Snake \(video game\)](#))

**Snake** is the common name for a video game concept where the player maneuvers a line which grows in length, with the line itself being a primary obstacle. The concept originated in the 1976 arcade game *Blockade*, and the ease of implementing *Snake* has led to hundreds of versions (some of which have the word *snake* or *worm* in the title) for many platforms. After a variant was preloaded on [Nokia mobile phones](#) in 1998, there was a resurgence of interest in the snake concept as it found a larger audience. There are over 300 *Snake*-like games for [iOS](#) alone.<sup>[2]</sup>

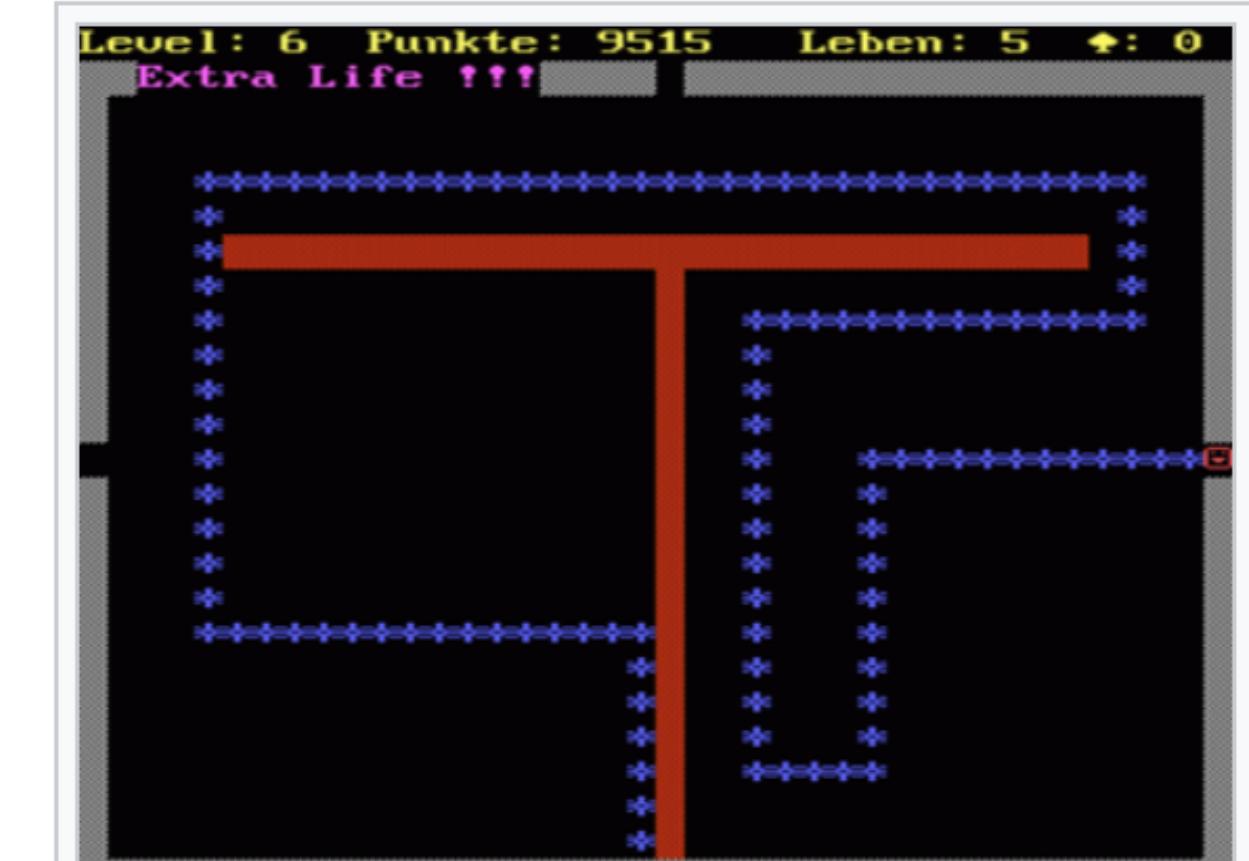
## Contents [hide]

- [1 Gameplay](#)
- [2 History](#)
- [3 Nokia phones](#)
- [4 Reception](#)
- [5 References](#)
- [6 External links](#)

## Gameplay [edit]



Snake on a TRS-80

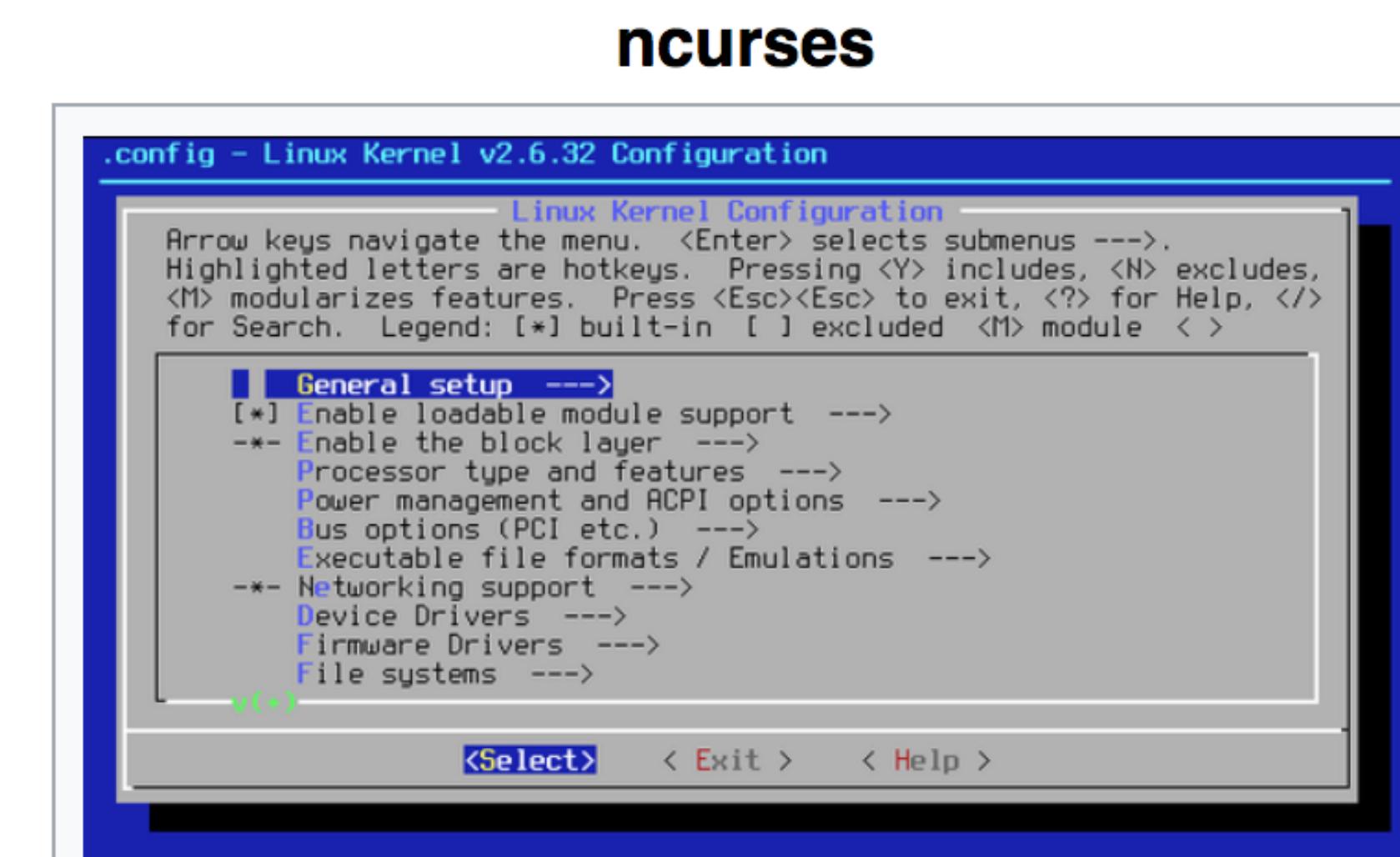


# ncurses

From Wikipedia, the free encyclopedia

**ncurses** (new curses) is a [programming library](#) providing an [application programming interface \(API\)](#) that allows the programmer to write [text-based user interfaces](#) in a terminal-independent manner. It is a toolkit for developing "[GUI-like](#)" [application software](#) that runs under a [terminal emulator](#). It also optimizes screen changes, in order to reduce the [latency](#) experienced when using remote [shells](#).

Contents [hide]	
1	<a href="#">History</a>
1.1	<a href="#">curses</a>
1.2	<a href="#">pcurses</a>
1.3	<a href="#">ncurses</a>
2	<a href="#">Terminal database</a>
3	<a href="#">License</a>
4	<a href="#">Programs using ncurses</a>
5	<a href="#">See also</a>
6	<a href="#">References</a>
7	<a href="#">External links</a>



ncurses-based [menuconfig](#)

<b>Developer(s)</b>	<a href="#">GNU Project</a>
<b>Initial release</b>	1993; 25 years ago
<b>Stable release</b>	6.1 / 27 January 2018; 6 months ago
<b>Repository</b>	<a href="ftp://ftp.gnu.org/gnu/ncurses/">ftp://ftp.gnu.org/gnu/ncurses/</a> ↗
<b>Written in</b>	C
<b>Operating system</b>	POSIX
<b>Type</b>	Widget toolkit
<b>License</b>	<a href="#">X11 License</a> <sup>[1]</sup>

**Disclaimer:**

**Very few unit**

**tests**



# **Disclaimer:**

**This is a happy**

**path**





There are more  
native things...

- C and Objective-C interop
- concurrency with workers
  - object.freeze()
  - object.pin()
- @CName, @ThreadLocal

- memory leaks
- segmentation faults

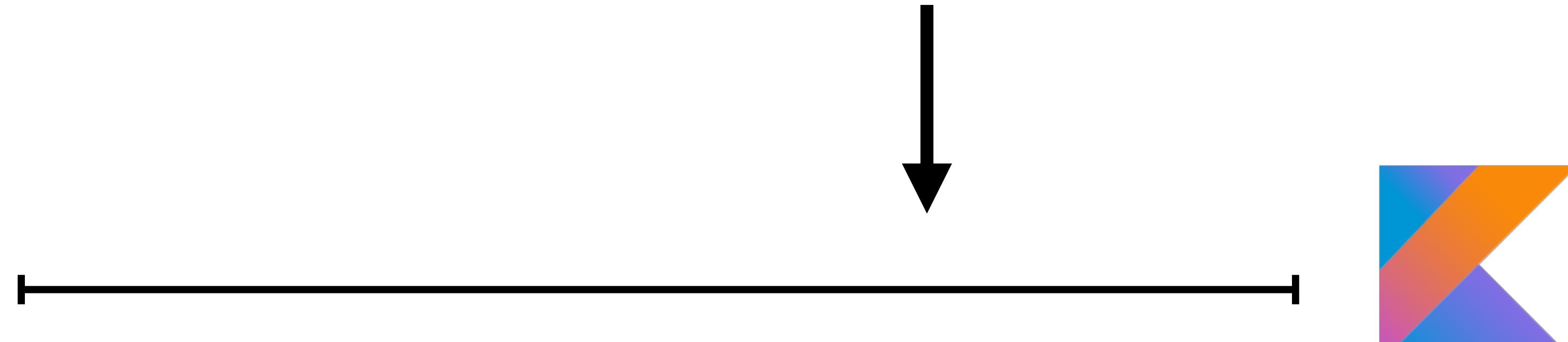
# Takeaways

THE  
**C**  
PROGRAMMING  
LANGUAGE



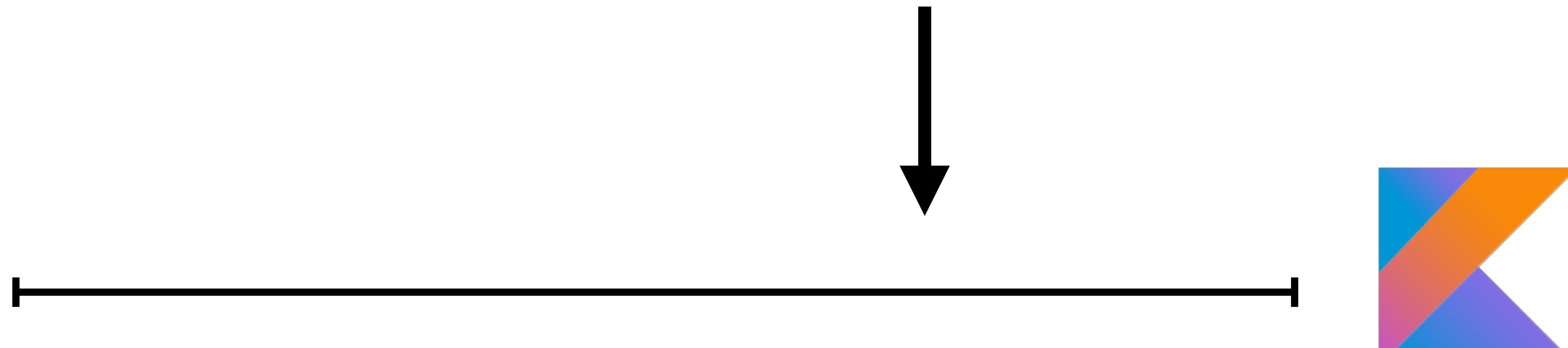
# Kotlin/Native

THE  
**C**  
PROGRAMMING  
LANGUAGE



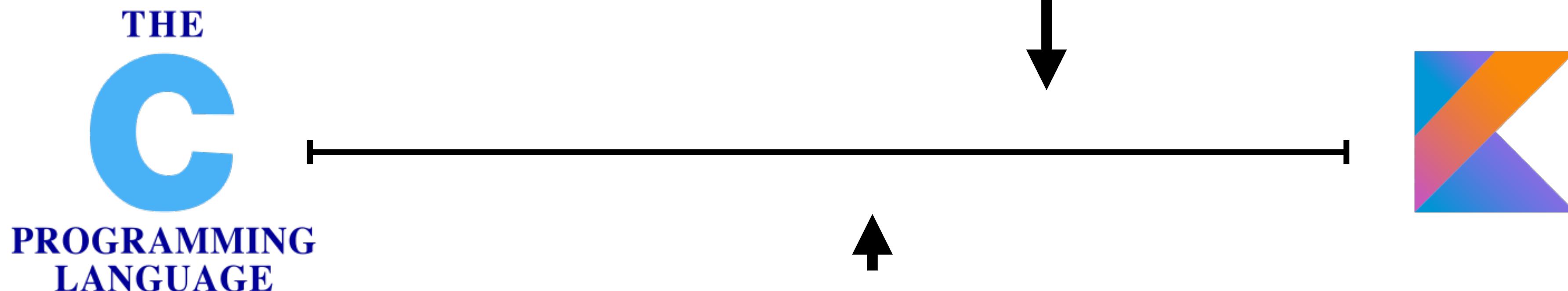
# Kotlin/Native

THE  
**C**  
PROGRAMMING  
LANGUAGE



**Rust**  
**Scala Native**

# Kotlin/Native



**Rust**  
**Scala Native**

# See also:

- [github.com/dkandalov/rust-snake](https://github.com/dkandalov/rust-snake)
- [github.com/dkandalov/scala-native-snake](https://github.com/dkandalov/scala-native-snake)
- [github.com/dkandalov/go-snake](https://github.com/dkandalov/go-snake)
- [github.com/dkandalov/graalvm-snake](https://github.com/dkandalov/graalvm-snake)

**It's early days  
for Kotlin/Native**

- slow compilation 😴
- not ideal IDE support
- tools are not perfect yet

**what can  
you do next?**

kotlin-native/samples at master

GitHub, Inc. [US] | github.com/JetBrains/kotlin-native/tree/master/samples

Search or jump to... Pull requests Issues Marketplace Explore

Watch 262 Star 6,131 Fork 492

Code Issues Pull requests Projects Wiki Security Insights

Branch: master **kotlin-native / samples /**

LepilkinaElena Rewrote some samples using kotlincx.cli (#3351) Latest commit 3b81f12 9 hours ago

..

Commit	Message	Date
androidNativeActivity	Move samples to 1.3.50 version of Kotlin	23 days ago
calculator	Move samples to 1.3.50 version of Kotlin	23 days ago
cocoapods	Move samples to 1.3.50 version of Kotlin	23 days ago
coverage	Changed version in samples to 1.3.40 (#3090)	3 months ago
csvparser	Rewrote some samples using kotlincx.cli (#3351)	9 hours ago
curl	Samples: Fix building for Windows	4 months ago
echoServer	Samples: Enable experimental standard library API	4 months ago
gitchurn	Samples: Fix building for Windows	4 months ago
globalState	Samples: Get rid of KotlinNativeTarget and KonanTarget imports	4 months ago
gradle/wrapper	Update Gradle: 5.5	2 months ago
gtk	Samples: Fix building for Windows	4 months ago
html5Canvas	Provide the proper content-type in wasm sample	2 months ago
libcurl	Samples: Fix building for Windows	4 months ago
nonBlockingEchoServer	Samples: Get rid of KotlinNativeTarget and KonanTarget imports	4 months ago
objc	Samples: Drop MPPTools + migrate to KTS	4 months ago

Create new file Upload files Find file History

Sample commits listed:

- LepilkinaElena: Rewrote some samples using kotlincx.cli (#3351) - 9 hours ago
- ..
- androidNativeActivity: Move samples to 1.3.50 version of Kotlin - 23 days ago
- calculator: Move samples to 1.3.50 version of Kotlin - 23 days ago
- cocoapods: Move samples to 1.3.50 version of Kotlin - 23 days ago
- coverage: Changed version in samples to 1.3.40 (#3090) - 3 months ago
- csvparser: Rewrote some samples using kotlincx.cli (#3351) - 9 hours ago
- curl: Samples: Fix building for Windows - 4 months ago
- echoServer: Samples: Enable experimental standard library API - 4 months ago
- gitchurn: Samples: Fix building for Windows - 4 months ago
- globalState: Samples: Get rid of KotlinNativeTarget and KonanTarget imports - 4 months ago
- gradle/wrapper: Update Gradle: 5.5 - 2 months ago
- gtk: Samples: Fix building for Windows - 4 months ago
- html5Canvas: Provide the proper content-type in wasm sample - 2 months ago
- libcurl: Samples: Fix building for Windows - 4 months ago
- nonBlockingEchoServer: Samples: Get rid of KotlinNativeTarget and KonanTarget imports - 4 months ago
- objc: Samples: Drop MPPTools + migrate to KTS - 4 months ago

JetBrains/kotlin-native: Kotlin/ +

← → C GitHub, Inc. [US] | github.com/JetBrains/kotlin-native ⋮

tools	[watchOS][targets] Add support for 32bit Apple Watch.	2 days ago
utilities	CLI library rework (#3215)	11 days ago
.gitignore	Server and client to show summary performance information (#2765)	6 months ago
.gitmodules	Added JB copyright (#417)	3 years ago
CHANGELOG.md	Update CHANGELOG.md	23 days ago
COCOAPODS.md	Docs: Add a CocoaPods integration readme	6 months ago
CODE_COVERAGE.md	Infrastructure for source-based code coverage	6 months ago
CONCURRENCY.md	docs: fix examples indents (#2222)	11 months ago
DEBUGGING.md	docs: drop debugging unnecessary title (#2091)	last year
DISTRO_README.md	Kotlin/Native samples ported to MPP Gradle DSL (#2261)	11 months ago
FAQ.md	Replace deprecated APIs in FAQ.md with binaries DSL	6 months ago
GRADLE_PLUGIN.md	Rewriting infrastructure to get right measurement of GC work (#3147)	2 months ago
HACKING.md	Included klibs: Rename -Xsource-library -> -Xinclude	22 days ago
IMMUTABILITY.md	Proofread Watson (#2055)	last year
INTEROP.md	Unify compiler and cinterop flags names (#2915)	5 months ago
LIBRARIES.md	Unify compiler and cinterop flags names (#2915)	5 months ago
LICENSE	Add missing LICENSE file	2 years ago
MULTIPLATFORM.md	fix link in MULTIPLATFORM.md (#2164)	last year
OBJC_INTEROP.md	Added a note on read-only property's setters (#3287)	16 days ago
PLATFORM_LIBS.md	Update .md documentation	8 months ago
README.md	Badge with the link to the latest Kotlin/Native release (#3203)	2 months ago
RELEASE_NOTES.md	Support experimental targets. (#2681)	7 months ago
_nav_reference.yml	Docs: Add a CocoaPods integration readme	6 months ago
build.gradle	CLI library rework (#3215)	11 days ago

A screenshot of a web browser window. The title bar shows "Concurrency - Kotlin Programm" and the URL "kotlinlang.org/docs/reference/native/concurrency.html". The page header includes the Kotlin logo, navigation links for "LEARN", "COMMUNITY", "TRY ONLINE", and a search icon. Below the header are four buttons: "Reference" (highlighted with a blue border), "Tutorials", "Books", and "More resources". On the left, a sidebar menu lists various topics under "Reference", with "Native" expanded to show "Concurrency", "Immutability", and "Kotlin Libraries". A red arrow points from the "Native" section in the sidebar to the "Workers" section in the main content area. The main content area features a large heading "Concurrency in Kotlin/Native" and a paragraph about the runtime's approach to concurrency. It also lists several alternative approaches: Workers with message passing, Object subgraph ownership transfer, Object subgraph freezing, Object subgraph detachment, Raw shared memory using C globals, and Coroutines for blocking operations. The "Workers" section is described as a concept where control flow streams are executed concurrently, similar to the Actor Model. It mentions that workers exchange Kotlin objects and can own mutable objects simultaneously. A note states that workers are started with `Worker.start` and can be addressed by their worker id. Other workers or OS threads can send messages to them via `execute`.

## Reference

▶ Overview

▶ Getting Started

▶ Basics

▶ Classes and Objects

▶ Functions and Lambdas

▶ Collections

▶ Multiplatform Programming

▶ Other

▶ Core Libraries

▶ Reference

▶ Java Interop

▶ JavaScript

▶ Native

- Concurrency

- Immutability

- Kotlin Libraries

▶ Platform Libraries

## Tutorials

## Books

## More resources

# Concurrency in Kotlin/Native

 Edit Page

Kotlin/Native runtime doesn't encourage a classical thread-oriented concurrency model with mutually exclusive code blocks and conditional variables, as this model is known to be error-prone and unreliable. Instead, we suggest a collection of alternative approaches, allowing you to use hardware concurrency and implement blocking IO. Those approaches are as follows, and they will be elaborated on in further sections:

- Workers with message passing
- Object subgraph ownership transfer
- Object subgraph freezing
- Object subgraph detachment
- Raw shared memory using C globals
- Coroutines for blocking operations (not covered in this document)

## Workers

Instead of threads Kotlin/Native runtime offers the concept of workers: concurrently executed control flow streams with an associated request queue. Workers are very similar to the actors in the Actor Model. A worker can exchange Kotlin objects with another worker, so that at any moment each mutable object is owned by a single worker, but ownership can be transferred. See section [Object transfer and freezing](#).

Once a worker is started with the `Worker.start` function call, it can be addressed with its own unique integer worker id. Other workers, or non-worker concurrency primitives, such as OS threads, can send a message to the worker with the `execute` call.

Tutorials - Kotlin Programming X +

← → C 🔒 kotlinlang.org/docs/tutorials/ ⋮

Kotlin LEARN COMMUNITY TRY ONLINE ⚡



LEARN

COMMUNITY

TRY ONLINE



Reference Tutorials Books More resources

▶ Getting Started

▶ Android

▶ Java Interop

▶ JavaScript

▶ Multiplatform Projects

Native

- Introduction to Kotlin/Native
- Basic Kotlin/Native Application
- Mapping Primitive Data Types from C
- Mapping Struct and Union Types from C
- Mapping Function Pointers from C
- Mapping Strings from C
- Kotlin/Native as a Dynamic Library
- Kotlin/Native as an Apple

## Tutorials

Edit Page

A series of complete step-by-step tutorials to walk you through how to accomplish different tasks in Kotlin, from Getting Started to writing Android applications and more. If you have a tutorial you'd like featured here, please let us know. We'll be happy to host it and give credit!

- If you like a "hands-on" learning, try [Kotlin Koans online](#) to get familiar with Kotlin;
- If you are using the command line compiler, start with [Working with the Command Line Compiler](#) and then work your way through the [Kotlin Koans](#);
- If you are using IntelliJ IDEA, start with [Getting Started](#) and then work your way through the [Kotlin Koans](#).



**kotlinlang** ▾

Dmitry Kandalov



Jump to...



More Unreads ↗

# getting-started

# gradle

# hiring

# http4k

# intellij

# intellij-plugins

# javascript

# kontributors

# kotlin-native

# kotlinconf

# kotlinlondon

# kotlinx-files

# ktor

# language-proposals

# london

# minutest

# multiplatform

# random

# reflect

More Unreads ↗

**#kotlin-native**☆ | ⚡ 1,828 | ⚡ 2 | Kotlin Native (<https://github.com>...)

Search



Yesterday

**Ola Gawell** 16:15

I need some help with coroutines. I get that kotlin native only support coroutines on the main thread. But it should be possible to execute a background task (for example a http request) without blocking the ui right? I tried to create a simple task that uses the Dispatchers.Main but then the ui is blocked. Here is my code:

**18 replies** Last reply 1 day ago**Scott** 16:29

How close are the different compiler platforms (jvm, js, and native) to sharing the same backend? I feel like this may have been done a while ago, but I haven't heard anything.

**4 replies** Last reply 1 day ago**Sam Schilling** 16:52

Printing a Kotlin/Native object that is exported to an Objective C framework in Swift yields the object name and address, but printing the address of it the Swift way is always that address offset by 0x28. Any idea why this is?

```
let a: MyType = ...
print(a) // MyType@c56528
print(Unmanaged.passUnretained(a).toOpaque()) // 0x0000000100c56500
```

The second print should print the address of `a`, whereas the first would just be the default `toString` for Kotlin objects, I don't know why the addresses are slightly offset though. (edited)

**5 replies** Last reply 1 day ago

Message #kotlin-native

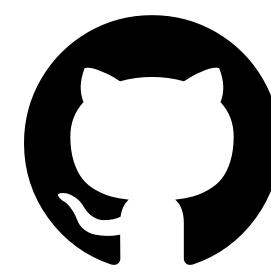




# Kotlin/Native



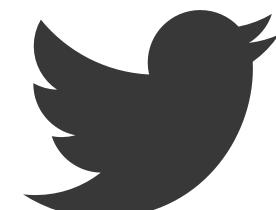
# The End



[github.com/dkandalov/kotlin-native-snake](https://github.com/dkandalov/kotlin-native-snake)



[youtube.com/dkandalov](https://youtube.com/dkandalov)



[@dmitrykandalov](https://twitter.com/dmitrykandalov)

