

Real World
Rewriting Your Front End Every Six Weeks

Trying Stuff Until it Works

Whiteboard Interviews

Blaming the User

O RLY? @ThePracticalDev

O RLY? The Practical Developer @ThePracticalDev

O RLY? The Practical Developer @ThePracticalDev

O RLY? @ThePracticalDev

Your application is a special snowflake

The passionate, functional, micro-serviced approach

Cutting corners to meet arbitrary management deadlines

The internet will make those bad words go away

Kotlin for Frontend Development

Expert
Excuses for Not Writing Unit Tests

Expert
Resurging in Frontend Development
Ruslan Ibragimov

Essential
Copying and Pasting

Essential
Googling the Error Message

O RLY? @ThePracticalDev

O RLY? @ThePracticalDev

O'REILLY* The Practical Developer @ThePracticalDev

O RLY? The Practical Developer @ThePracticalDev

Who are you kidding?

Does it run? Just leave it alone.

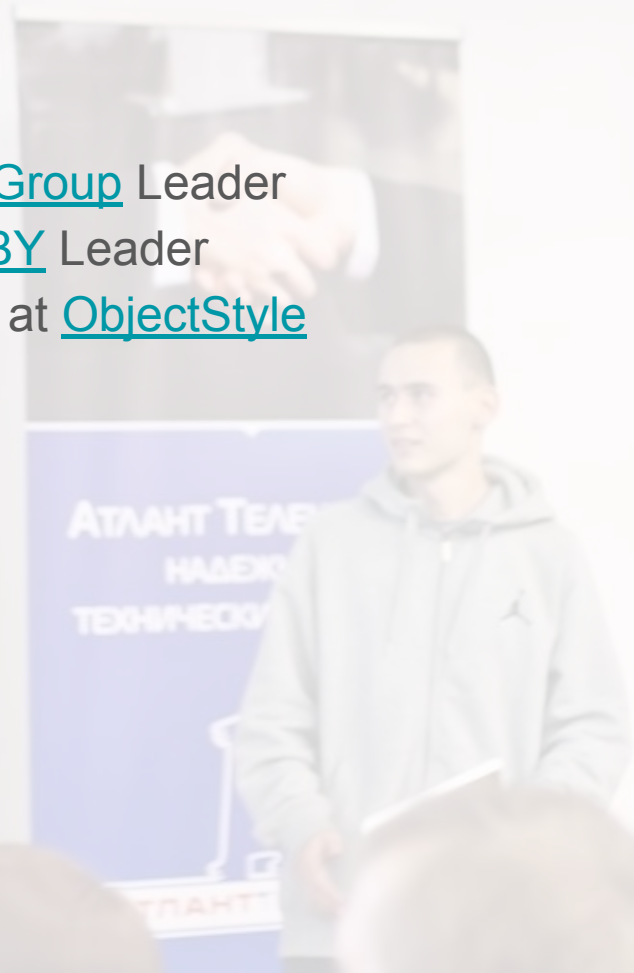
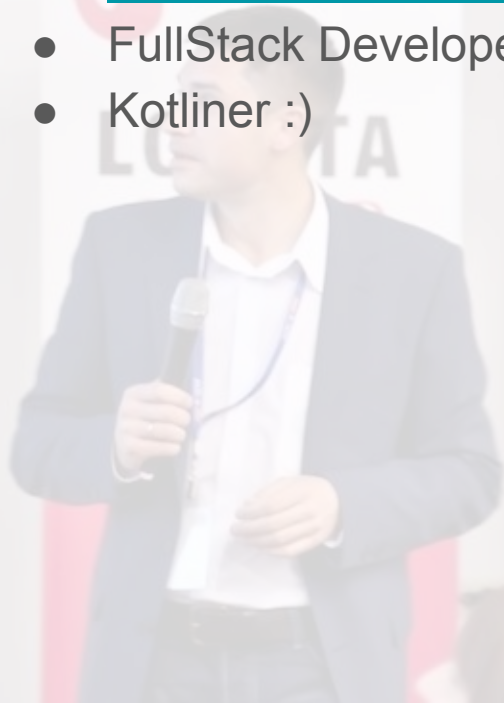
Code written by some stranger on the internet is always perfect

You might as well just kill yourself right now

Web Development With Assembly

About Me

- [Belarus Kotlin User Group](#) Leader
- [Java Professionals BY](#) Leader
- FullStack Developer at [ObjectStyle](#)
- Kotliner :)



Agenda

Kotlin?

Kotlin & TypeScript

Practice

Tooling

The Future

Q&A

Kotlin



Current Kotlin Targets:

- JVM
- Android
- JavaScript







Trending in open source

See what the GitHub community is most excited about today.

Repositories

Developers

Trending: **today**

All languages

JetBrains / kotlin

The Kotlin Programming Language

Java ★ 9,014 🍴 912 Built by

★ Unstar

★ 710 stars today

Unknown languages

CSS

HTML

Java

JavaScript

Kotlin

PHP

Shell

desktop / desktop

Simple collaboration from your desktop

TypeScript ★ 1,144 🍴 43 Built by

★ Star

★ 659 stars today

Other: Languages

kshvmdn / fsqj

Search through your filesystem with SQL-esque queries.

Go ★ 2,146 🍴 55 Built by

★ Star

★ 402 stars today

ProTip! Looking for most starred repositories? [Try this search](#)



In July 2011 JetBrains unveiled Project Kotlin



Filters

LANGUAGES

- C/C++
- C#
- DSL
- Go
- Groovy
- Java
- JavaScript, TypeScript
- Kotlin
- Objective-C
- PHP
- Python
- Ruby
- Scala
- SQL
- Swift

TECHNOLOGIES

PRODUCT TYPE



Toolbox App

1.2

A control panel for your tools and projects

[Learn more](#)



IntelliJ IDEA

2017.1.3

The most intelligent Java IDE

[Learn more](#) | [Buy](#)



PyCharm

2017.1.2

Python IDE
for professional developers

[Learn more](#) | [Buy](#)



WebStorm

2017.1.3

The smartest JavaScript IDE

[Learn more](#) | [Buy](#)



PhpStorm

2017.1.4

Lightning-smart PHP IDE

[Learn more](#) | [Buy](#)



ReSharper

2017.1.2

Visual Studio extension for .NET
developers

[Learn more](#) | [Buy](#)





In July 2011 JetBrains unveiled Project Kotlin



Kotlin Goes Open Source! February 14, 2012

M1-M14

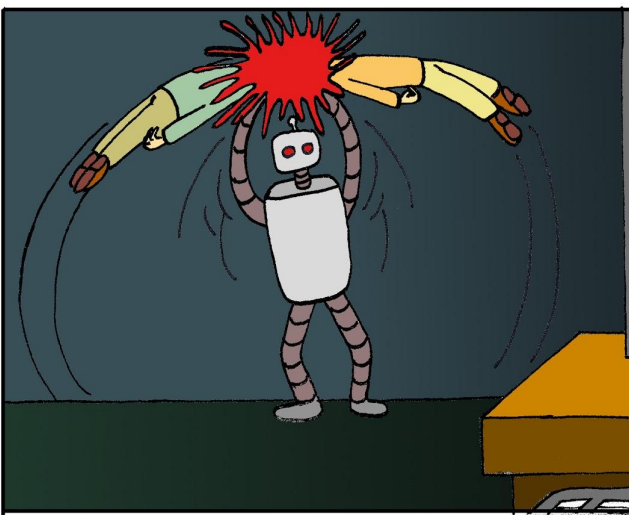
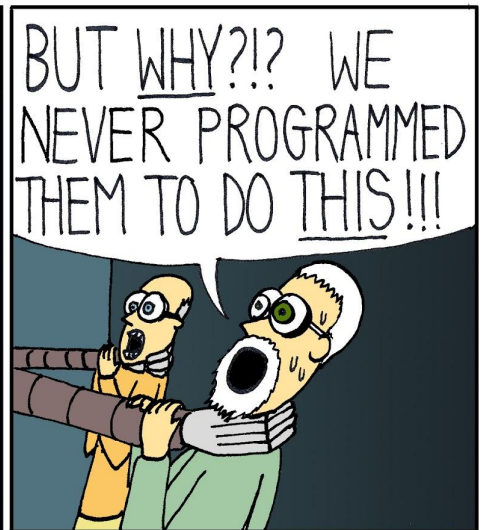
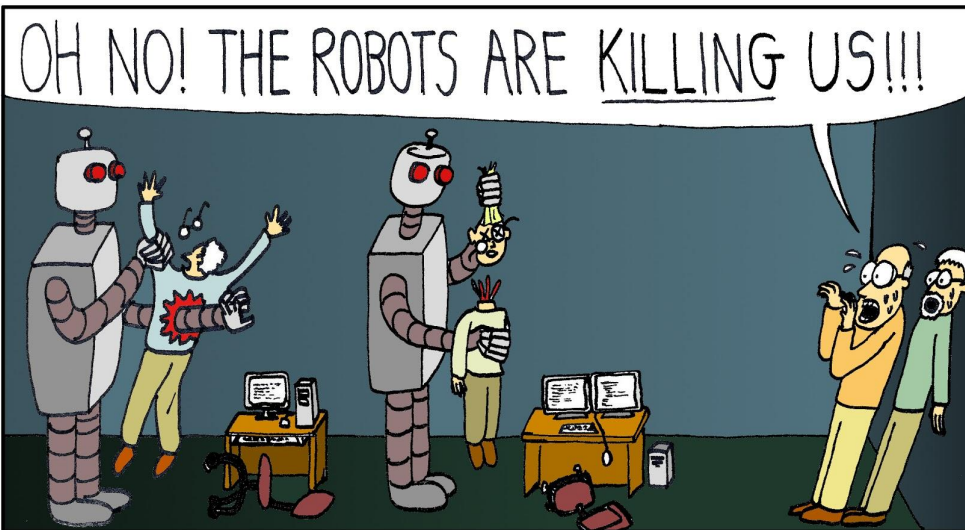


Kotlin v1.0 was released on February 15, 2016

Kotlin v1.1 was released on March 1, 2017

Kotlin - Pragmatic Language

- Interop: JS, JVM(Java)
- Tooling
- Safety



```
static bool isCrazyMurderingRobot = false;  
  
void interact_with_humans (void){  
    if(isCrazyMurderingRobot = true)  
        kill(humans);  
    else  
        be_nice_to(humans);  
}
```

Safety

```
val isCrazyMurderingBot = false  
  
if (isCrazyMurderingBot = true)  
    kill(humans)  
else  
    be_nice_to(humans)
```

Hello, World!

```
// main.ts  
console.log("Hello, TypeScript!");
```

```
$ tsc main.ts  
$ node main.js  
Hello, TypeScript!
```

Hello, World!

```
// main.kt
fun main(args: Array<String>) {
    println("Hello, Kotlin!")
}
```

```
$ kotlinc-js -output main.js -meta-info main.kt
```

```
main/root-package.kjasm
main.js
main.meta.js
```


main.js (plain)

```
if (typeof kotlin === 'undefined') {  
  throw new Error("Error loading module 'main'. Its dependency 'kotlin'  
was not found. Please, check whether 'kotlin' is loaded prior to  
'main'.");  
}  
var main = function (_, Kotlin) {  
  'use strict';  
  var println = Kotlin.kotlin.io.println_s8jyv4$;  
  function main_0(args) {  
    println('Hello, Kotlin!');  
  }  
  _.main_kand9s$ = main_0;  
  Kotlin.defineModule('main', _);  
  main_0([]);  
  return _;  
}(typeof main === 'undefined' ? {} : main, kotlin);
```

?

main

index.js

```
// index.js  
kotlin = require("./node_modules/kotlin/kotlin.js");  
require("./main.js");
```

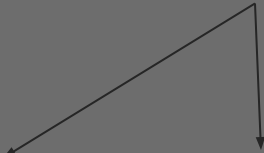
```
$ npm install kotlin  
$ node index.js  
Hello, Kotlin!
```

Functions

```
function foo(bar: string): string {  
  return `Hello, $`  
}
```

```
fun foo(bar: String)  
  return "Hello,  
}
```

```
fun foo(bar: String) = "Hello, $bar!"
```



```
const fn = function () {  
  // ...  
}  
  
(function () {  
  //...  
})()
```

Functions

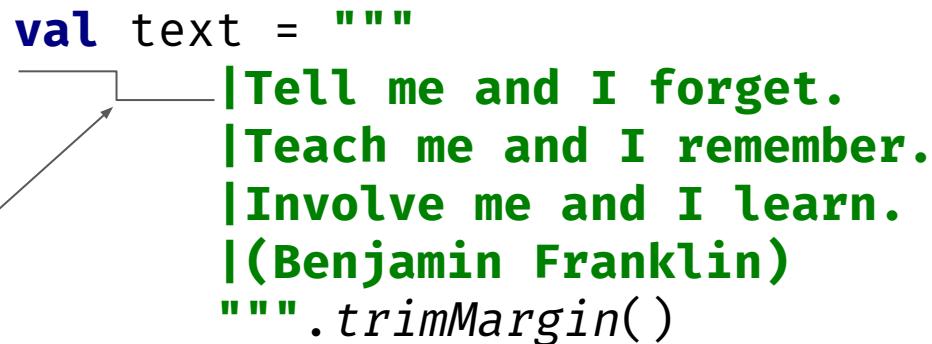
```
fun reformat(str: String,  
            normalizeCase: Boolean = true,  
            upperCaseFirstLetter: Boolean = true,  
            divideByCamelHumps: Boolean = false,  
            wordSeparator: Char = ' ') {  
    // ...  
}
```

```
reformat("abc", true, true, false, '_')
```

```
reformat("abc", wordSeparator = '_')
```

String Template

```
val text = """  
|Tell me and I forget.  
|Teach me and I remember.  
|Involve me and I learn.  
|(Benjamin Franklin)  
""".trimMargin()
```



Collections

```
/*1*/ const sum = [1, 2, 3]
/*2*/ .map(num => num * 2)
/*3*/ .filter(it => it % 2 === 0)
/*4*/ .reduce((acc, i) => acc + i, 0);
```

```
/*1*/ val list = (1..3)
/*2*/ .map({ it * 2 })
/*3*/ .filter { it % 2 == 0 }
/*4*/ .reduce { acc, i -> acc + i }
/*5*/ .sum()
```

```
function sum(acc, i) {
    return acc + i;
}
```

```
const sum = (acc, i) => acc + i;
```

Dynamic Typing

TypeScript:

```
function foo(bar: any): number {...}
```

Kotlin:

```
fun foo(bar: dynamic): Int {  
    bar.prop  
    bar.function()  
}
```

Uncaught TypeError: Cannot read property 'foo' of null

Uncaught ReferenceError: bar is not defined

NullPointerException, NullPointerException


Null

String



Only String

String?



String or null

```
var nullable: String? = null  
nullable.length
```

```
// Kotlin: Compile Error: Only  
safe (?.) or non-null...
```

Operator Overloading

```
class Complex(val i: Int, val j: Int) {  
    operator fun plus(c: Complex) = Complex(this.i + c.i, this.j + c.j)  
}
```

```
val c = Complex(1, 0) + Complex(0, 1) // = Complex(1, 1)
```

If expression

```
let a;  
  
if (smth) {  
    a = "foo"  
} else {  
    a = "bar"  
}
```

```
val a = if (smth) {  
    "foo"  
} else {  
    "bar"  
}
```

Kotlin is Awesome!

- Type Inference
- Named Arguments
- Default Arguments
- Ranges
- Template Strings
- Operator Overloading
- Null Safety
- if-expression
- Properties
- Data Classes
- Sealed Classes (ADT)
- Class Delegation
- Extension Functions
- More Expressions
- Inline Functions
- Generics (reified)
- Object
- Delegated Properties
- Reflection
- Coroutines!
- Type-Safe Builders!
- ...

Future Kotlin Features?

- Annotations for static analyses
- Collection comprehensions
- Format strings
- Subject variable in when
- Overloadable operators | and &
- Static members for Kotlin classes
- **Vararg-like treatment of data classes**
- Support package-private visibility
- Optional (trailing) commas
- **Destructuring assignments**
- **Value classes**
- **Truly immutable data**
- Unsigned arithmetic
- SAM Conversions for Kotlin interfaces
- Use invokedynamic to compile Kotlin
- Collection literals
- Short notation for enum constants
- Slices for lists and arrays
- Private members accessible from tests
- Multi-catch

[Description](#)

Practice

JavaScript

- Target version: ECMAScript 5
- Modules
 - UMD
 - AMD
 - Commonjs
 - Plain (no module system)
- Async/Await, Yield
 - Kotlin Coroutines!
 - rx
 - actors/channels

Calling JavaScript from Kotlin

```
if (typeof kotlin === 'undefined') {  
  throw new Error("Error loading module 'main'. Its dependency 'kotlin'  
was not found. Please, check whether 'kotlin' is loaded prior to  
'main'.");  
}  
var main = function (_, Kotlin) {  
  'use strict';  
  function main$debug() {  
    debugger;  
  }  
  function main_0(args) {  
    var debug = main$debug;  
    debug();  
  }  
  _.$main_kand9s$ = main_0;  
  Kotlin.defineModule('main', _);  
  main_0([]);  
  return _;  
}(typeof main === 'undefined' ? {} : main, kotlin);
```


Calling JavaScript from Kotlin

```
if (typeof kotlin === 'undefined') {  
  throw new Error("Error loading module 'main'. Its dependency 'kotlin' was not found.  
Please, check whether 'kotlin' is loaded prior to 'main'.");  
}  
var main = function (_, Kotlin) {  
  'use strict';  
  function main$jsTypeOf(o) {  
    return typeof o;  
  }  
  function main_0(args) {  
    var jsTypeOf = main$jsTypeOf;  
    jsTypeOf(42);  
  }  
  _.$main_kand9s$ = main_0;  
  Kotlin.defineModule('main', _);  
  main_0([]);  
  return _;  
}(typeof main === 'undefined' ? {} : main, kotlin);
```

Type Definitions

Calling JavaScript from Kotlin

```
// window.kt  
external fun alert(message: Any?): Unit
```

Calling JavaScript from Kotlin

```
// file.js
function MyClass() { }
MyClass.sharedMember = function () { /* ... */ };
MyClass.prototype.ownMember = function () { /* ... */ };
```

```
// file.kt
external class MyClass {
    companion object {
        fun sharedMember()
    }

    fun ownMember()
}
```

JavaScript Modules

```
// file.kt
@jsModule("react-dom")
external object ReactDOM {
    fun render(element: ReactElement?, container: Element?)
    fun<P: RProps, S: RState> findDOMNode(component: ReactComponent<P, S>): Element
    fun unmountComponentAtNode(domContainerNode: Element?)
}
```

Type-Safe Builders vs JSX

JSX in Kotlin

```
render() {  
    return (  
        <div>  
            "Hello, ${this.props.name}!"  
        </div>  
    )  
}
```

```
override fun render(): ReactElement? {  
    return div {  
        + "Hello, ${props.name}!"  
    }  
}
```

CSS in Kotlin (Aza-Kotlin-CSS)

```
val css = Stylesheet {  
    a {  
        width = 10.px  
        color = 0xFFFFFFFF  
        opacity = .8  
  
        hover {  
            color = 0xF2CACF  
        }  
    }  
}
```

```
val css = Stylesheet {  
    const styles = {  
        button: {top:0}  
            fontSize: 12,  
            '&:hover': { 0 }  
            background: 'blue'  
        }  
    }  
    div and span { top = 0 }  
    // div,span{top:0}  
  
    li.nthChild(2) { top = 0 }  
    // li:nth-child(2){top:0}  
  
    input["disabled"] { top = 0 }  
    // input[disabled]{top:0}  
}
```


Tooling

Editor/IDE?

- **IntelliJ Idea Community Edition/Ultimate** ❤️
- Eclipse, Netbeans
- Atom, Sublime

Build

- Whatever + Gradle/Maven
- Plugin for Gradle: <https://github.com/Kotlin/kotlin-frontend-plugin> ☆
 - NPM
 - Webpack
 - Karma
- Webpack Loader: <https://github.com/huston007/kotlin-loader> 🚀
 - `require('./app/app.kt');`

Debug

- source maps
- debugger;
- println/console.log :)

ts2kt

```
$ npm install -g ts2kt
```

```
$ ts2kt [<options>] <d.ts files>
```

Java to Kotlin Conversion

```
User.java x
User getFirstName()
1 package link.kotlin.scripts.model;
2
3 public class User {
4     ... private final String firstName;
5     ... private final String lastName;
6     ... private final String email;
7     ... private final Integer age;
8
9
10    ... public User(String firstName, String lastName, String email, Integer age) {
11        ...     this.firstName = firstName;
12        ...     this.lastName = lastName;
13        ...     this.email = email;
14        ...     this.age = age;
15    ... }
16
17    ... public String getFirstName() {
18        ...     return firstName;
19    ... }
20
21    ... public String getLastName() { return lastName; }
24
25    ... public String getEmail() { return email; }
28
29    ... public Integer getAge() { return age; }
32 }
33
```

Real World Example

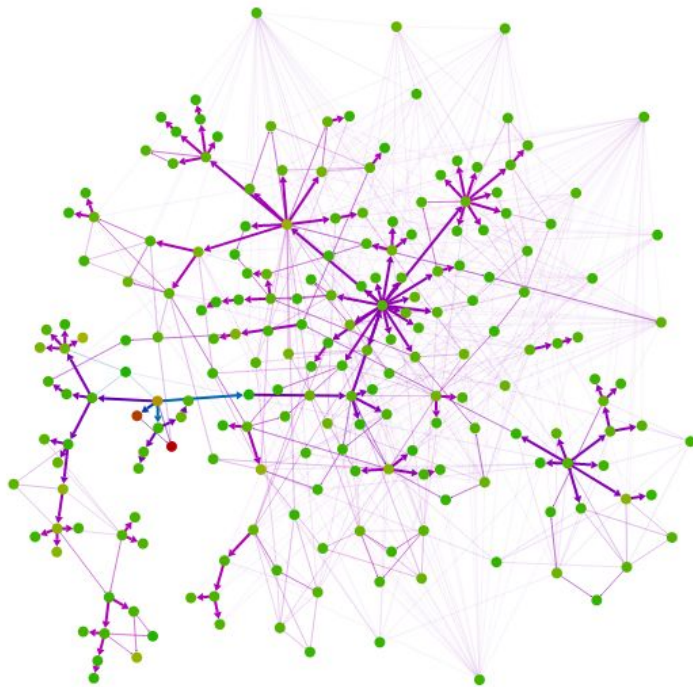
Application:

react
 react-dom
 react-router
 jquery
 react-markdown
 kotlin
 kotlin-html-js

Build: 3mb
 Minimized: 1.3mb
 Zipped: 230kb

Hello World:

Minimized: 600kb
 Prepack: 400kb
[\[src\]](#)



id	name	size	chunks	flags
51	../~/kotlin/kotlin.js	1246 KiB	0	built
83	../~/kotlinx-html-js/kotlinx-html-js.js	583 KiB	0	built
87	./frontend.js	148 KiB	0	built
203	../~/xss-filters/src/xss-filters.js	53 KiB	0	built

The Future

- Bundle Size Optimization [[src](#)]
- Seamless RPC Client-Server [[src](#)]
- WASM? [[src](#)]
- Easy shared code between JVM/JS/Native [[src](#)]
- ...

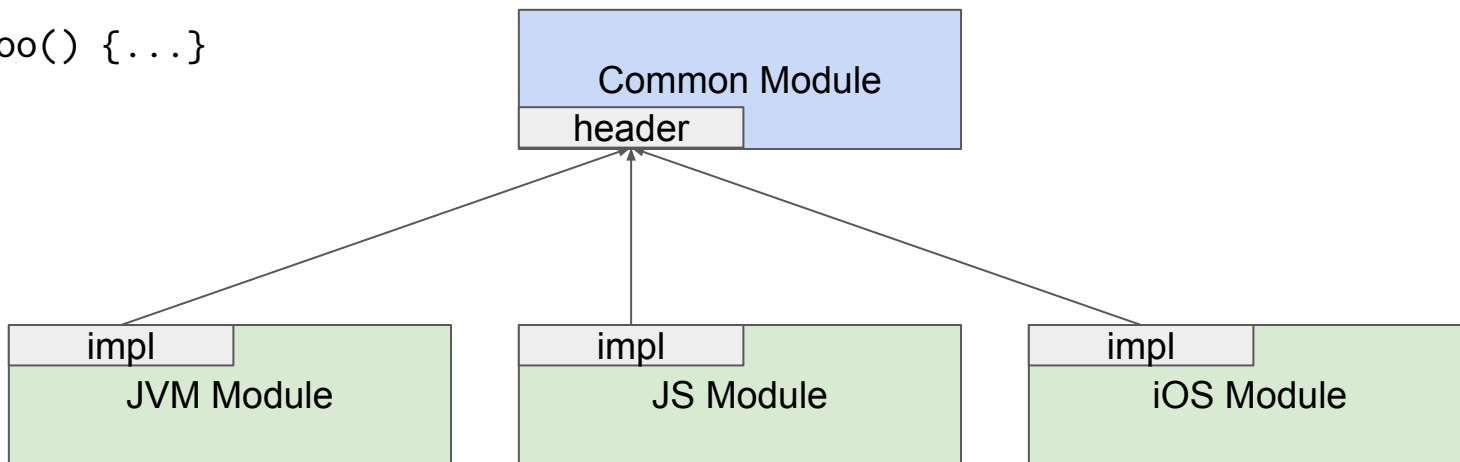
Multiplatform Projects & Libraries [[src](#)]

Common module:

```
header fun foo()
```

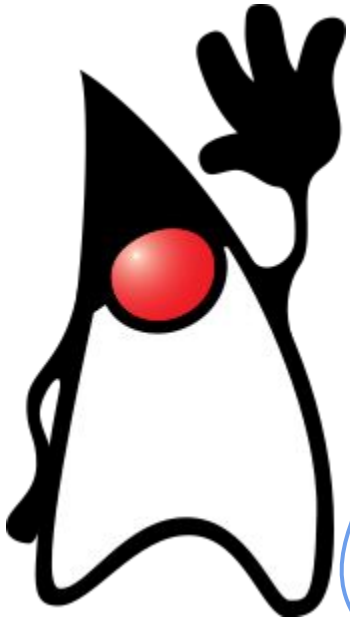
Impl module:

```
impl fun foo() {...}
```

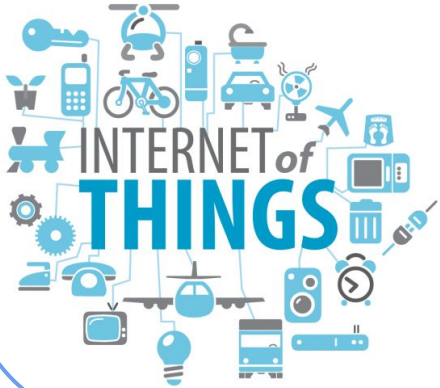


Kotlin/Native

Full-Stack on Kotlin



iOS



How to Learn

Docs: <http://kotlinlang.org/docs/reference/>

In Browser: <https://try.kotlinlang.org/>

Kotlin in Action: <https://www.manning.com/books/kotlin-in-action>

Stepic Kotlin Course: <https://stepik.org/course/Kotlin-2852/>

Awesome Kotlin: <https://kotlin.link/>

Links

- Bindings ready to use - <https://github.com/danfma/kodando>
- Gaming in Browser!? - <https://github.com/perses-games/tetris-kudens>
- React Application Sample - <https://github.com/Kotlin/kotlin-fullstack-sample>
- Application Sample - <https://github.com/JetBrains/kotlin-web-demo>

Conclusion: Pros/Cons

- + Kotlin/JVM, Kotlin/Android, Kotlin/JS, Kotlin/Native, WASM(?)
- + Language
- + IntelliJ IDEA

- IntelliJ IDEA
- Bundle Size
- Gradle/Maven (for frontend devs)

Q&A

<https://kotlin.link/>

