

Kotlinの デフォルト引数の話

Kyoto.kt#3

2025/5/10

京都産業大学 上村太成

自己紹介

- taiseiue

 @taiseiue

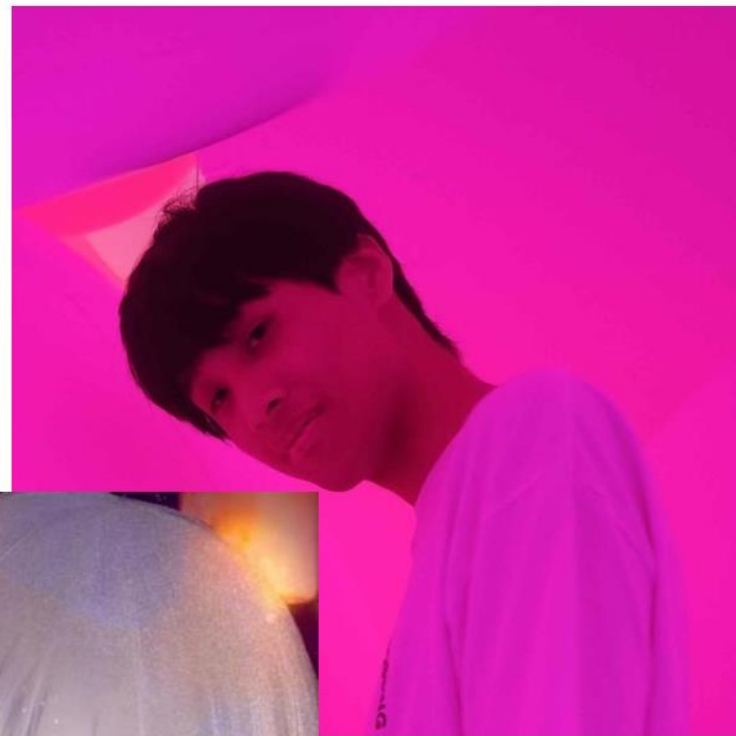
 id:taiseiue

- taiseiue.jp

- C#/TypeScript/Perl

-  WSOFT

- JVM界とは無縁



 alicescrpt Public

Edit Pins Watch 0 Fork 0 Starred 3

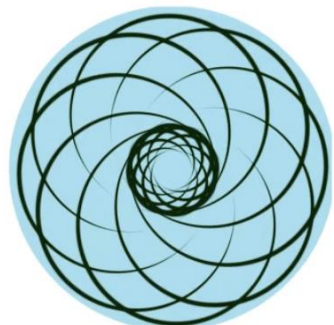
main 1 Branch 0 Tags

Go to file Add file Code

 taiseiue	Update proposal.md	63b4529 · 7 months ago	8 Commits
	.github/ISSUE_TEMPLATE	Update proposal.md	7 months ago
	.gitignore	Initial commit	last year
	README.md	Update README.md	last year

README

AliceScript



Alice Script

About

The repo for the design of the AliceScript programming language

a.wsoft.ws/alice

programming-language

- Readme
- Activity
- Custom properties
- 3 stars
- 0 watching
- 0 forks
- Report repository

mukaikun



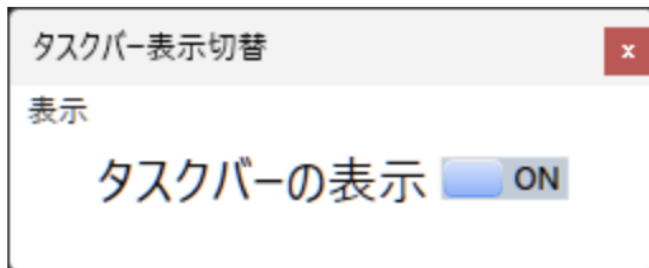
向井くん

Moodleのリソース表示を改善するくん

これは何

- Moodleで掲示されているPDFなどのファイルを、ダウンロードせずにブラウザ内で読み込むChrome拡張機能です。
- Moodle以外のサイトでも、パターンを書くことでブラウザ内で読み込めます。
- 初めましての人は、まず下の設定欄にMoodleのUrlを入力しよう。

Taskbar Hiddener



タスクバー表示切替

タスクバーを隠したり表示したりするシンプルなソフトウェア

What's this?

もしあなたがタスクバーの存在に頭を悩ませているなら、このソフトウェアが役に立つでしょう。このソフトウェアを使うと、Windowsの「タスクバーを自動的に隠す」オプションとは異なり、自分のタイミングでタスクバーを隠したり、表示したりできます。

このソフトウェアは、内部でWin32APIを呼び出し、タスクバーのウィンドウハンドルを取得することで自在に隠したり、表示したりしています。このソフトウェアは.Net Framework 4.5.2以降の環境で動作します。(つまり、Windows 8.1以降の環境では特段の注意なく動作します)

このソフトウェアは2020年秋ごろ作成しました。

Kotlinの話をしてします

デフォルト引数

```
fun greet(name: String = "World") {  
    println("Hello, $name")  
}
```

GET STARTED > KOTLIN

概要

より優れたアプリを作成する

サンプル

ガイド

デベロッパーの事例

≡ フィルタ

Kotlin ファーストの Android 開発

Android と Kotlin をゼロから学ぶ ▼

Android Java デベロッパーのための Kotlin ▲

Kotlin を既存のアプリに追加する

Kotlin-Java 相互運用ガイド

Android での Kotlin の使用に関するよくある質問

▶ Android KTX

参考情報

Kotlin 詳細ガイド ▼

<

translated by Google

た。

このページは Cloud Translation API によって翻訳されました。

Switch to English

Android Developers > Get started > Kotlin > ガイド

この情報は役に立ちましたか？



Kotlin-Java 相互運用ガイド



このドキュメントは、Java と Kotlin で公開 API を作成するための一連のルールです。他の依存関係から消費されても、コードが慣用的に感じられる あります。

最終更新日: 2024 年 7 月 29 日

Java (Kotlin で使用する場合)

ハード キーワードなし

Kotlin の [ハード キーワード](#) をメソッド名として使用しない 定義できます。これらを使用するには、呼び出しの際に

このページの内容

Java (Kotlin で使用する場合)

ハード キーワードなし

Any 拡張機能名を回避する

null 値許容アノテーション

ラムダ パラメータを最後に置く

プロパティの接頭辞

演算子のオーバーロード

Kotlin (Java で使用する場合)

ファイル名

ラムダ引数

Nothing のジェネリクスを回避する

例外を文書化する

防御のためのコピー

同意する

同意しない

Javaってデフォルト引数あったっけ

```
@JvmOverloads
```

```
fun greet(name: String = "World") {  
    println("Hello, $name")  
}
```

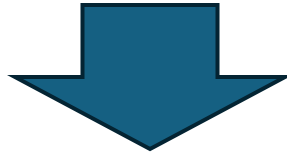
これが内部で
どうなってるのかを見る

内部を見る

```
ノートを入力  
kotlinc hello.kt -include-runtime -d hello.jar
```

内部を見る

```
kotlinc ノートを入力 hello.kt -include-runtime -d hello.jar
```

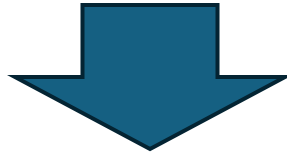


jarを解凍する(tar xfみたいな)

```
jar xf ノートを入力 hello.jar
```

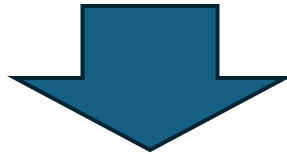
内部を見る

```
kotlinc ノートを入力 hello.kt -include-runtime -d hello.jar
```



jarを解凍する(tar xfみたいな)

```
jar xf ノートを入力 hello.jar
```



バイトコードを逆アセンブル

```
javap ノートを入力 -c HelloKt.class
```

```
@JvmOverloads
```

```
fun greet(name: String = "World") {  
    println("Hello, $name")  
}
```

```
public static final void greet(java.lang.String);
```

Code:

```
0: aload_0
1: ldc          #10          // String name
3: invokestatic #16          // Method kotlin/jvm/internal/Intrinsics.checkNotNullParameter:(Ljava/lang/Object;Ljava/lang/S
6: new          #18          // class java/lang/StringBuilder
9: dup
10: invokespecial #22         // Method java/lang/StringBuilder."<init>":()V
13: ldc          #24          // String Hello,
15: invokevirtual #28         // Method java/lang/StringBuilder.append:(Ljava/lang/String;)Ljava/lang/StringBuilder;
18: aload_0
19: invokevirtual #28         // Method java/lang/StringBuilder.append:(Ljava/lang/String;)Ljava/lang/StringBuilder;
22: invokevirtual #32         // Method java/lang/StringBuilder.toString:()Ljava/lang/String;
25: getstatic    #38          // Field java/lang/System.out:Ljava/io/PrintStream;
28: swap
29: invokevirtual #44         // Method java/io/PrintStream.println:(Ljava/lang/Object;)V
32: return
```

```
public static void greet$default(java.lang.String, int, java.lang.Object);
```

Code:

```
0: iload_1
1: iconst_1
2: iand
3: ifeq         9
6: ldc          #49          // String World
8: astore_0
9: aload_0
10: invokestatic #51         // Method greet:(Ljava/lang/String;)V
13: return
```

```
public static final void greet();
```

Code:

```

10: invokespecial #22          // Method java/lang/StringBuilder.<init>:()V
13: ldc             #24          // String Hello,
15: invokevirtual #28          // Method java/lang/StringBuilder.append:(Ljava/lang/String;)Ljava/lang/StringBuilder;
18: aload_0
19: invokevirtual #28          // Method java/lang/StringBuilder.append:(Ljava/lang/String;)Ljava/lang/StringBuilder;
22: invokevirtual #32          // Method java/lang/StringBuilder.toString:()Ljava/lang/String;
25: getstatic      #38          // Field java/lang/System.out:Ljava/io/PrintStream;
28: swap
29: invokevirtual #44          // Method java/io/PrintStream.println:(Ljava/lang/Object;)V
32: return

```

```
public static void greet$default(java.lang.String, int, java.lang.Object);
```

Code:

```

0: iload_1
1: iconst_1
2: iand
3: ifeq          9
6: ldc           #49          // String World
8: astore_0
9: aload_0
10: invokestatic #51          // Method greet:(Ljava/lang/String;)V
13: return

```

```
public static final void greet();
```

Code:

```

0: aconst_null
1: iconst_1
2: aconst_null
3: invokestatic #53          // Method greet$default:(Ljava/lang/String;ILjava/lang/Object;)V
6: return

```

```
}
```

```
public static final void greet(java.lang.String);
```

Code:

```
0: aload_0
1: ldc          #10          // String name
3: invokestatic #16          // Method kotlin/jvm/internal/Intrinsics.checkNotNullParameter:(Ljava/lang/Object;Ljava/lang/S
6: new          #18          // class java/lang/StringBuilder
9: dup
10: invokespecial #22         // Method java/lang/StringBuilder."<init>":()V
13: ldc          #24          // String Hello,
15: invokevirtual #28         // Method java/lang/StringBuilder.append:(Ljava/lang/String;)Ljava/lang/StringBuilder;
18: aload_0
19: invokevirtual #28         // Method java/lang/StringBuilder.append:(Ljava/lang/String;)Ljava/lang/StringBuilder;
22: invokevirtual #32         // Method java/lang/StringBuilder.toString:()Ljava/lang/String;
25: getstatic    #38          // Field java/lang/System.out:Ljava/io/PrintStream;
28: swap
29: invokevirtual #44         // Method java/io/PrintStream.println:(Ljava/lang/Object;)V
32: return
```

```
public static void greet$default(java.lang.String, int, java.lang.Object);
```

Code:

```
0: iload_1
1: iconst_1
2: iand
3: ifeq         9
6: ldc          #49          // String World
8: astore_0
9: aload_0
10: invokestatic #51          // Method greet:(Ljava/lang/String;)V
13: return
```

```
public static final void greet();
```

Code:

```
public static final void greet(java.lang.String);
```

Code:

```
0: aload_0
1: ldc          #10          // String name
3: invokestatic #16          // Method kotlin/jvm/internal/Intrinsics.checkNotNullParameter:(Ljava/lang/Object;Ljava/lang/S
6: new          #18          // class java/lang/StringBuilder
9: dup
10: invokespecial #22         // Method java/lang/StringBuilder."<init>":()V
13: ldc          #24          // String Hello,
15: invokevirtual #28         // Method java/lang/StringBuilder.append:(Ljava/lang/String;)Ljava/lang/StringBuilder;
18: aload_0
19: invokevirtual #28         // Method java/lang/StringBuilder.append:(Ljava/lang/String;)Ljava/lang/StringBuilder;
22: invokevirtual #32         // Method java/lang/StringBuilder.append:(Ljava/lang/String;)Ljava/lang/StringBuilder;
25: getstatic    #38          // Field java/lang/System.out:Ljava/io/PrintStream;
28: swap
29: invokevirtual #44         // Method java/io/PrintStream.println:()V
32: return
```

```
public static void greet$default(java.lang.String, int, java.lang.Object);
```

Code:

```
0: iload_1
1: iconst_1
2: iand
3: ifeq         9
6: ldc          #49          // String World
8: astore_0
9: aload_0
10: invokestatic #51         // Method greet:(Ljava/lang/String;)V
13: return
```

```
public static final void greet();
```

Code:

```
String str = null;
if ((args.length & 1) != 0) {
    str = "World";
}
greet(str);
```

```
@JvmOverloads
```

```
fun greet(name: String = "World", greeting: String = "Hello") {  
    println("$greeting, $name")  
}
```

```
public static void greet$default(java.lang.String, java.lang.String, int, java.lang.Object);
```

Code:

```
 0: iload_2
 1: iconst_1
 2: iand
 3: ifeq          9
 6: ldc           #51          // String World
 8: astore_0
 9: iload_2
10: iconst_2
11: iand
12: ifeq          18
15: ldc           #53          // String Hello
17: astore_1
18: aload_0
19: aload_1
20: invokestatic  #55          // Method greet:(Ljava/lang/String;Ljava/lang/String;)V
23: return
```

```
// $FF: synthetic method
public static void greet$default(String var0, String var1, int var2, Object var3) { 0件の使用箇所
    if ((var2 & 1) != 0) {
        var0 = "World";
    }

    if ((var2 & 2) != 0) {
        var1 = "Hello";
    }

    greet(var0, var1);
}
```

```
public static void greet$default(String var0, String var1, int var2, Object var3) {  
    if ((var2 & 0b_01) != 0) {  
        var0 = "World";  
    }  
  
    if ((var2 & 0b_10) != 0) {  
        var1 = "Hello";  
    }  
  
    greet(var0, var1);  
}
```

```
public static void greet$default(String var0, String var1, int var2, Object var3) {  
    if ((var2 & 0b_01) != 0) {  
        var0 = "World";  
    }  
  
    if ((var2 & 0b_10) != 0) {  
        var1 = "Hello";  
    }  
  
    greet(var0, var1);  
}
```

ビットマスク

0	0	0	0	0	0	1	1
---	---	---	---	---	---	---	---

int パラメータ

`play$default` の `int` パラメータの値は、デフォルト引数が渡された引数の数と、そのインデックスに基づいて計算されます。Kotlin コンパイラは、どのパラメータを使って `play` 関数を呼び出すかを、このパラメータの値に基づいて判断します。

この例の `play()` の呼び出しでは、インデックス 0 の引数がデフォルト引数を使っています。そのため、`int var1 = 2` を使って `play$default` を呼び出します。

```
play$default((Toy)null, 1, (Object)null);
```

これで、`play$default` の実装は、`var0` の値をデフォルト値で置き換えなければならないことを認識できます。

引数が32個以上の時は？

@JvmOverloads

```
fun greet(  
    p1: String = "1", p2: String = "2", p3: String = "3", p4: String = "4",  
    p5: String = "5", p6: String = "6", p7: String = "7", p8: String = "8",  
    p9: String = "9", p10: String = "10", p11: String = "11", p12: String = "12",  
    p13: String = "13", p14: String = "14", p15: String = "15", p16: String = "16",  
    p17: String = "17", p18: String = "18", p19: String = "19", p20: String = "20",  
    p21: String = "21", p22: String = "22", p23: String = "23", p24: String = "24",  
    p25: String = "25", p26: String = "26", p27: String = "27", p28: String = "28",  
    p29: String = "29", p30: String = "30", p31: String = "31", p32: String = "32",  
    p33: String = "33", p34: String = "34", p35: String = "35", p36: String = "36",  
    p37: String = "37", p38: String = "38", p39: String = "39", p40: String = "40",  
    p41: String = "41", p42: String = "42", p43: String = "43", p44: String = "44",  
    p45: String = "45", p46: String = "46", p47: String = "47", p48: String = "48",  
    p49: String = "49", p50: String = "50", p51: String = "51", p52: String = "52",  
    p53: String = "53", p54: String = "54", p55: String = "55", p56: String = "56",  
    p57: String = "57", p58: String = "58", p59: String = "59", p60: String = "60",  
    p61: String = "61", p62: String = "62", p63: String = "63", p64: String = "64"  
) {  
    println(  
        listOf(  
            p1, p2, p3, p4, p5, p6, p7, p8,  
            p9, p10, p11, p12, p13, p14, p15, p16,  
            p17, p18, p19, p20, p21, p22, p23, p24,  
            p25, p26, p27, p28, p29, p30, p31, p32,  
            p33, p34, p35, p36, p37, p38, p39, p40,  
            p41, p42, p43, p44, p45, p46, p47, p48,  
            p49, p50, p51, p52, p53, p54, p55, p56,  
            p57, p58, p59, p60, p61, p62, p63, p64  
        ).joinToString(", ")  
    )  
}
```

実行してみる

▼ **untitled** ~/source/untitled

> .gradle

> .idea

> .kotlin

> build

> gradle

▼ src

▼ main

▼ kotlin

Main.kt

resources

> test

.gitignore

build.gradle.kts

gradle.properties

gradlew

gradlew.bat

settings.gradle.kts

> 外部ライブラリ

> スクラッチとコンソール

4 **fun** greet(
22 println(
23 listOf(
24 p1, p2, p3, p4, p5, p6, p7, p8,
25 p9, p10, p11, p12, p13, p14, p15, p16,
26 p17, p18, p19, p20, p21, p22, p23, p24,
27 p25, p26, p27, p28, p29, p30, p31, p32,
28 p33, p34, p35, p36, p37, p38, p39, p40,
29 p41, p42, p43, p44, p45, p46, p47, p48,
30 p49, p50, p51, p52, p53, p54, p55, p56,
31 p57, p58, p59, p60, p61, p62, p63, p64
32).joinToString(separator: ", ")
33)
34 }
35
36 **fun** main(){
37 greet();
38 }

⚠ 1 ^ ▼

デコンパイル ☒ インライン化 ☒ 最適化

1695 ICONST_M1
1696 ICONST_M1
1697 ACONST_NULL
1698 INVOKESTATIC org/example/MainKt.greet\$defa
1699 L1
1700 LINENUMBER 38 L1
1701 RETURN
1702 MAXSTACK = 67
1703 MAXLOCALS = 0
1704
1705 // access flags 0x19
1706 **public final static** greet(Ljava/lang/String;
1707 @Lkotlin/jvm/JvmOverloads;() // invisible
1708 // annotable parameter count: 63 (invisibl
1709 @Lorg/jetbrains/annotations/NotNull;() //
1710 @Lorg/jetbrains/annotations/NotNull;() //
1711 @Lorg/jetbrains/annotations/NotNull;() //
1712 @Lorg/jetbrains/annotations/NotNull;() //
1713 @Lorg/jetbrains/annotations/NotNull;() //
1714 @Lorg/jetbrains/annotations/NotNull;() //
1715 @Lorg/jetbrains/annotations/NotNull;() //

実行 MainKt ×

↑
↓
⇌
⇕
🖨
🗑

/opt/homebrew/Cellar/openjdk/23.0.2/libexec/openjdk.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ I
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31,

プロセスは終了コード 0 で終了しました

パフォーマンス >

結果の表示 00:00

プロセスは終了しました

▼ **untitled** ~/source/untitled

> .gradle

> .idea

> .kotlin

> build

> gradle

▼ src

▼ **main**

▼ kotlin

Main.kt

resources

> test

.gitignore

build.gradle.kts

gradle.properties

gradlew

gradlew.bat

settings.gradle.kts

> 外部ライブラリ

> スクラッチとコンソール

4 **fun** greet(
22 println(
23 listOf(
24 p1, p2, p3, p4, p5, p6, p7, p8,
25 p9, p10, p11, p12, p13, p14, p15, p16,
26 p17, p18, p19, p20, p21, p22, p23, p24,
27 p25, p26, p27, p28, p29, p30, p31, p32,
28 p33, p34, p35, p36, p37, p38, p39, p40,
29 p41, p42, p43, p44, p45, p46, p47, p48,
30 p49, p50, p51, p52, p53, p54, p55, p56,
31 p57, p58, p59, p60, p61, p62, p63, p64
32).joinToString(separator: ", ")
33)
34 }
35
36 **fun** main(){
37 greet();
38 }

デコンパイル ☒ インライン化 ☒ 最適化

1695 ICONST_M1
1696 ICONST_M1
1697 ACONST_NULL
1698 INVOKESTATIC org/example/MainKt.greet\$defa
1699 L1
1700 LINENUMBER 38 L1
1701 RETURN
1702 MAXSTACK = 67
1703 MAXLOCALS = 0
1704
1705 // access flags 0x19
1706 **public final static** greet(Ljava/lang/String;
1707 @Lkotlin/jvm/JvmOverloads;() // invisible
1708 // annotable parameter count: 63 (invisibl
1709 @Lorg/jetbrains/annotations/NotNull;() //
1710 @Lorg/jetbrains/annotations/NotNull;() //
1711 @Lorg/jetbrains/annotations/NotNull;() //
1712 @Lorg/jetbrains/annotations/NotNull;() //
1713 @Lorg/jetbrains/annotations/NotNull;() //
1714 @Lorg/jetbrains/annotations/NotNull;() //
1715 @Lorg/jetbrains/annotations/NotNull;() //

実行 MainKt ×

↑
↓
⇌
⇕
🖨
🗑

/opt/homebrew/Cellar/openjdk/23.0.2/libexec/openjdk.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ I
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31,
プロセスは終了コード 0 で終了しました

パフォーマンス
結果の表示 00:00
プロセスは終了しました

```
Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stder
```

```
53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64
```

動いてるやん

```
public static void greet$default(String var0, String var1, String var2, ... String var32, ... int var63, int var64, int var65, Object var66) {  
    if ((var64 & 1) != 0) {  
        var0 = "1";  
    }  
  
    if ((var64 & 2) != 0) {  
        var1 = "2";  
    }  
  
    if ((var64 & 1073741824) != 0) {  
        var30 = "31";  
    }  
  
    if ((var64 & Integer.MIN_VALUE) != 0) {  
        var31 = "32";  
    }  
  
    if ((var65 & 1) != 0) {  
        var32 = "33";  
    }  
    greet(var0, var1, var2, ... var32, ...var63);  
}
```

```
public static void greet$default(String var0, String var1, String var2, ... String var32, ... int var63, int var64, int var65, Object var66) {  
    if ((var64 & 1) != 0) {  
        var0 = "1";  
    }  
  
    if ((var64 & 2) != 0) {  
        var1 = "2";  
    }  
  
    if ((var64 & 1073741824) != 0) {  
        var30 = "31";  
    }  
  
    if ((var64 & Integer.MIN_VALUE) != 0) {  
        var31 = "32";  
    }  
  
    if ((var65 & 1) != 0) {  
        var32 = "33";  
    }  
    greet(var0, var1, var2, ... var32, ...var63);  
}
```

まとめ

- @JvmOverloads + デフォルト引数を持つ関数
 - オーバーロードの関数とfunc\$defaultみたいなメソッドに展開される
- Func\$defaultは引数に、どの引数が指定されてるかを表すフラグをとる
 - これはInt32型
 - 32ビットに収まらない場合はもうひとつ引数ができる

```
if ((var64 & 8192) != 0) {  
    var13 = "14";  
}
```

```
if ((var64 & 16384) != 0) {  
    var14 = "15";  
}
```

```
if ((var64 & '耀') != 0) {  
    var15 = "16";  
}
```

```
if ((var64 & 8192) != 0) {  
    var13 = "14";  
}
```

```
>>> ord("耀")  
32768
```

```
if ((var64 & 16384) != 0) {  
    var14 = "15";  
}
```

```
if ((var64 & '耀') != 0) {  
    var15 = "16";  
}
```

完

Kotlinの デフォルト引数の話

Kyoto.kt#3

2025/5/10

京都産業大学 上村太成