March 2020

CodersCraft

Kotlin for Android

Ndongo Tonux SAMB

About me

- Ndongo SAMB -> Tonux
- Software Engineer
- SONATEL GROUP, Senegal
- @tonux_samb
- sambndongo@gmail.com
- Member of Jiggen Tech

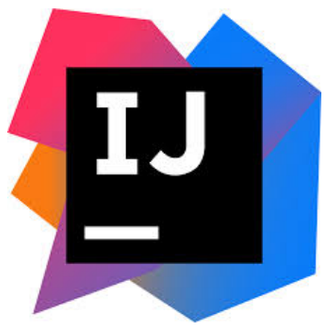
Definition

- Statically typed language
- First version in 2011
- Kotlin is an Open-Source Language.
- Object and Functional Paradigms
- Compilation en bytecode
- Execution in JVM

Authors

- JetBrains company founded in 2000
- IntelliJ IDEA, the Java IDE
- Android Studio is based on IntelliJ

- Under the Apache 2.0 license



...

- Officially supported by Google (I/O 2017)
- Excellent support in Android Studio

Why kotlin?

Concise

Drastically reduce the amount of boilerplate code.

Safe

Avoid entire classes of errors such as null pointer exceptions.

Interoperable

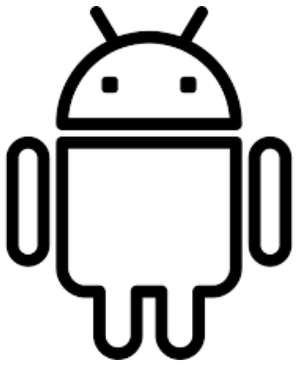
Leverage existing libraries for JVM, Android and the browser.

Tool-friendly

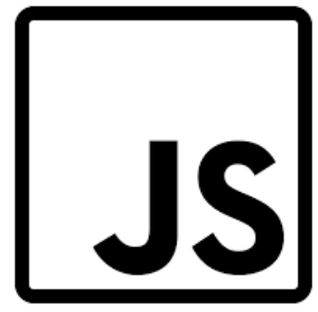
Choose any Java IDE or build from the command line.

Possibilities

Server

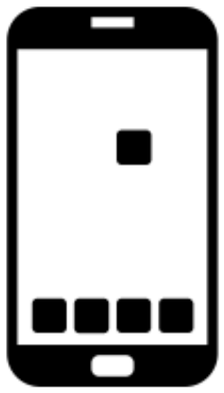


Android

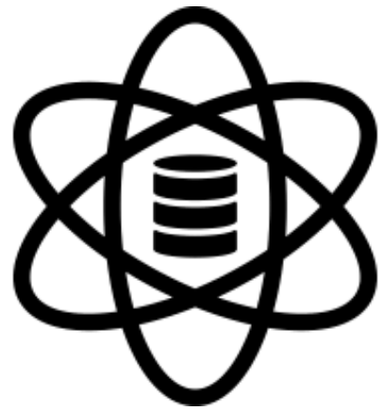


Javascript

Cross-platform



Native



Data-science

Kotlin features

- Null safety
- No checked exceptions
- Extension functions
- Higher-order functions
- Function types & lambdas
- Default & named arguments
- Properties
- Type inference
- Operator overloading
- Smart casts
- Data classes
- Immutable collections
- Enhanced switch-case
- String interpolation
- Ranges
- Inline functions
- Infix notation
- Tail recursion
- Coroutines (async/await)
- Great Standard Library
- Sealed classes
- Delegated Lazy properties
- Class delegation
- Singleton
- Nested functions
- Object decomposition
- Top-level functions
- Reified generics
- Raw Strings
- And more…


SHUT UP AND SHOW ME THE CODE!
imgflip.com

Functions

```java
// Java
public int sum(int a, int b) {
    return a + b;
}
```

```kotlin
// Kotlin
fun sum(a: Int, b: Int) = a + b
```

Immutability/mutability

```
// Assign-once (read-only) variable
val age = 10
age = 25 // Compile-time error
```

```
// Mutable variable
var number = 1
number = 3        // OK
```

```
// Always start with immutable "val"
// Change to "var" only when necessary
```

Null safety

```
var str: String= "localhost"
str = null // Compile-time erro
```

```
var str: String?= " Corona virus "
 str= "Stay at home!"  //OK
```

when expression

```
fun calcul(a:Int, b: Int, operator: String){
    val result = when (operator) {
        "+" -> a + b
        "-" -> a - b
        "*" -> a * b
        "/" -> a / b
        else -> "invalid operator."
    }
    println("result = $result")
}
```

```
when(view.visibility){
    View.VISIBLE -> toast("visible")
    View.INVISIBLE -> toast("invisible")
    else -> toast("gone")
}
```

```
when (view) {
    is TextView -> toast("TextView")
    is RecyclerView -> toast("RecyclerView")
    is SearchView -> toast("SearchView")
    else -> toast("View type not supported")
}
```

Lambdas

```
listOf(10, 3, null, 0, 5, null)
  .filterNotNull()
  .filter { it > 3 }
  .forEach {print(it)}
// 10 5
```

Group the same actions on an object

```
// Java
TextView textView = new TextView(this);
textView.setVisibility(View.VISIBLE);
textView.setText("CodersCraft!");
textView.setTextSize(15f)
textView.setMaxLines(4);
```

```
// kotlin
val textView = TextView(this)
textView.apply{
    visibility = View.VISIBLE
    text = "CodersCraft!"
    size = 15f
    maxLines = 4
}
```

RxJava Vs RxKotlin

```java
// Java
Observable.just("20", "13", "30", "45")
      .map(new Func1<String, Integer>() {
          @Override
          public Integer call(String s) {
              return Integer.parseInt(s);
          }
      })
      .filter(new Func1<Integer, Boolean>() {
          @Override
          public Boolean call(Integer integer) {
              return integer > 18;
          }
      })
      .subscribe(new Action1<Integer>() {
          @Override
          public void call(Integer integer) {
                System.out.println(integer);
          }
});
```

```kotlin
// Kotlin
Observable.just("20", "13", "30", "45")
 .map { it.toInt() }
  .filter { it > 18 }
  .subscribe {print(it)}
```

Coroutines

```
class ArticlesViewModel : ViewModel(), CoroutineScope {
      private val job = Job()
      private lateinit var articlesLiveData: LiveData<List<Articles>> = MutableLiveData()
      override val coroutineContext: CoroutineContext
                get() = job + Dispatchers.Main

     fun getAllArticles() {
            launch(Dispatchers.Main) {
                  val articles: List<Article> = async(Dispatchers.IO) {
                      articlesRepository.queryArticles()
                  }.await()
                  articlesLiveData.value = articles
            }
      }
}
```

Data class

//java

```java
public class User {
 String name;
 int age;
 public User(String name, int age) {
  this.name = name;
  this.age = age;
 }
 public String getName() {
  return name;

 }
 public void setName(String name) {
  this.name = name;
 }
 public int getAge() {
  return age;
 }
 public void setAge(int age) {
  this.age = age;
 }
 @Override
 public boolean equals(Object o) {
  if (this == o) return true;
  if (o == null || getClass() != o.getClass()) return false;
  User user = (User) o;
  if (age != user.age) return false;
  if (name != null ? !name.equals(user.name) : user.name != null)
   return false;
 }
 @Override
 public int hashCode() {
  int result = 31 * name != null ? name.hasCode() : 0;
  result = 31 * result + age;
  return result;
 }
 @Override
 public String toString() {
  return "User{" +
   "name='" + name + '\'' +
   ", age=" + age +
   '}';
 }
}
```

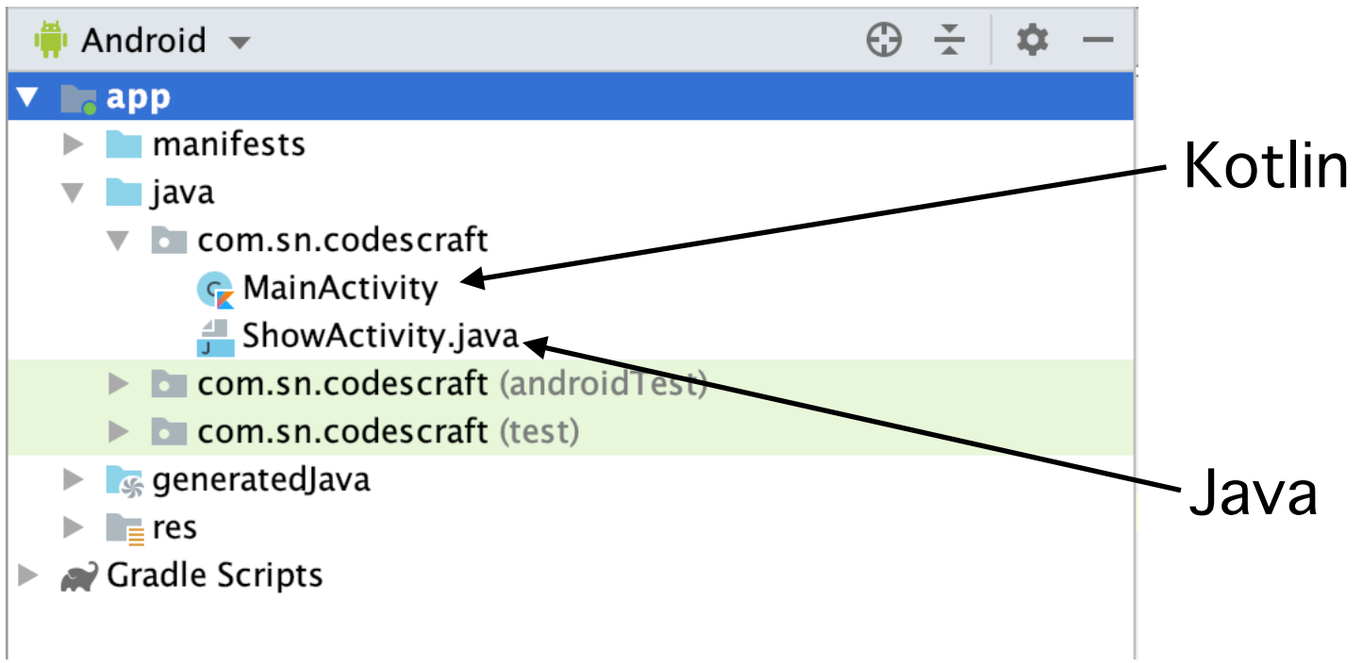
//kotlin

```kotlin
data class Person(name: String?, var age: Int)
```

data = equals() + hashCode() + toString() + copy()

Strong compatibility

- All versions of Android are compatible
- Most Java libraries are compatible
- Interoperable with Java in the same project



More easy to read

- Declaration of Activity

Java

```java
public class MyActivity extends AppCompatActivity {
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity);
  }
}
```

Kotlin

```kotlin
class MyActivity : AppCompatActivity() {
  override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity)
  }
}
```

More easy to read

- Create an on-click listener

Java

```java
FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
fab.setOnClickListener(new View.OnClickListener() {
  @Override
  public void onClick(View view) {
    ...
  }
});
```

Kotlin

```kotlin
val fab = findViewById(R.id.fab) as FloatingActionButton
fab.setOnClickListener {
  ...
}
```

More easy to read

- Create a click listener item

Java

```java
private BottomNavigationView.OnNavigationItemSelectedListener onNavigationItemSelectedListener
    = new BottomNavigationView.OnNavigationItemSelectedListener() {
  @Override
  public boolean onNavigationItemSelected(@NonNull MenuItem item) {
    switch (item.getItemId()) {
      case R.id.navigation_home:
        textMessage.setText(R.string.title_home);
        return true;
      case R.id.navigation_dashboard:
        textMessage.setText(R.string.title_dashboard);
        return true;
    }
    return false;
  }
};
```

Kotlin

```kotlin
private val onNavigationItemSelectedListener
    = BottomNavigationView.OnNavigationItemSelectedListener { item ->
  when (item.itemId) {
    R.id.navigation_home -> {
      textMessage.setText(R.string.title_home)
      return@OnNavigationItemSelectedListener true
    }
    R.id.navigation_dashboard -> {
      textMessage.setText(R.string.title_dashboard)
      return@OnNavigationItemSelectedListener true
    }
  }
  false
}
```

fastest growing

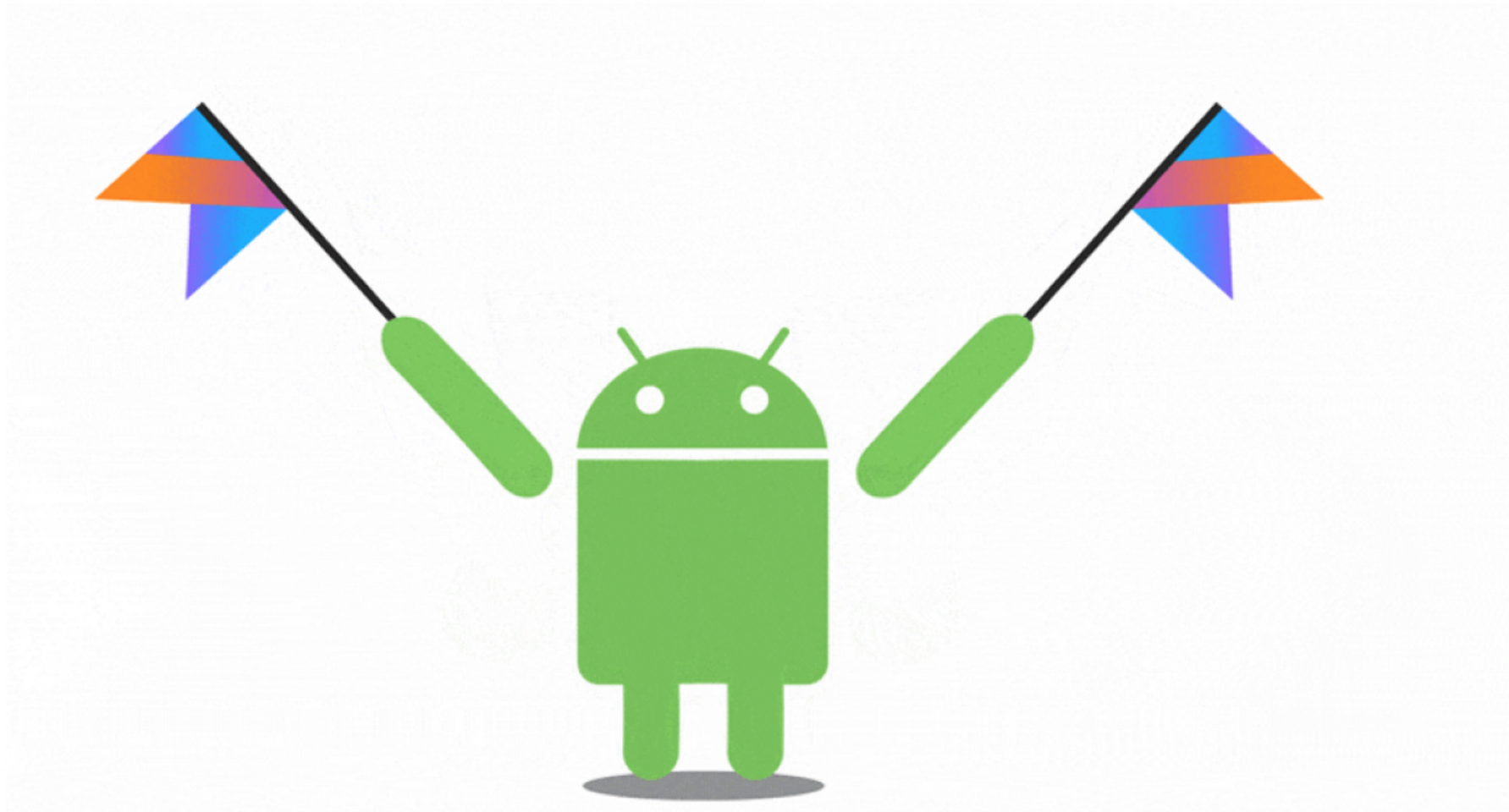
Language	Change in Use (%)
Dart	532
Rust	235
HCL	213
Kotlin	182
TypeScript	161
PowerShell	154
Apex	154
Python	151
Assembly	149
Go	147

Source: GitHub

TO BE

CONTINUED

Dieureudieuf



Links

- **kotlinlang.org** Great documentation

Online IDE - Hello World, Examples, Advent of Code…

- **developer.android.com** Develop Android apps with Kotlin

- **Try.kotlin**

- **learn Kotlin**

- **Coroutines**