



Mikael Desertot
mikael.desertot@uphf.fr



Applications Android

Langage de programmation

- Java

- Historique



- Kotlin

- Recommandé par Google





Kotlin

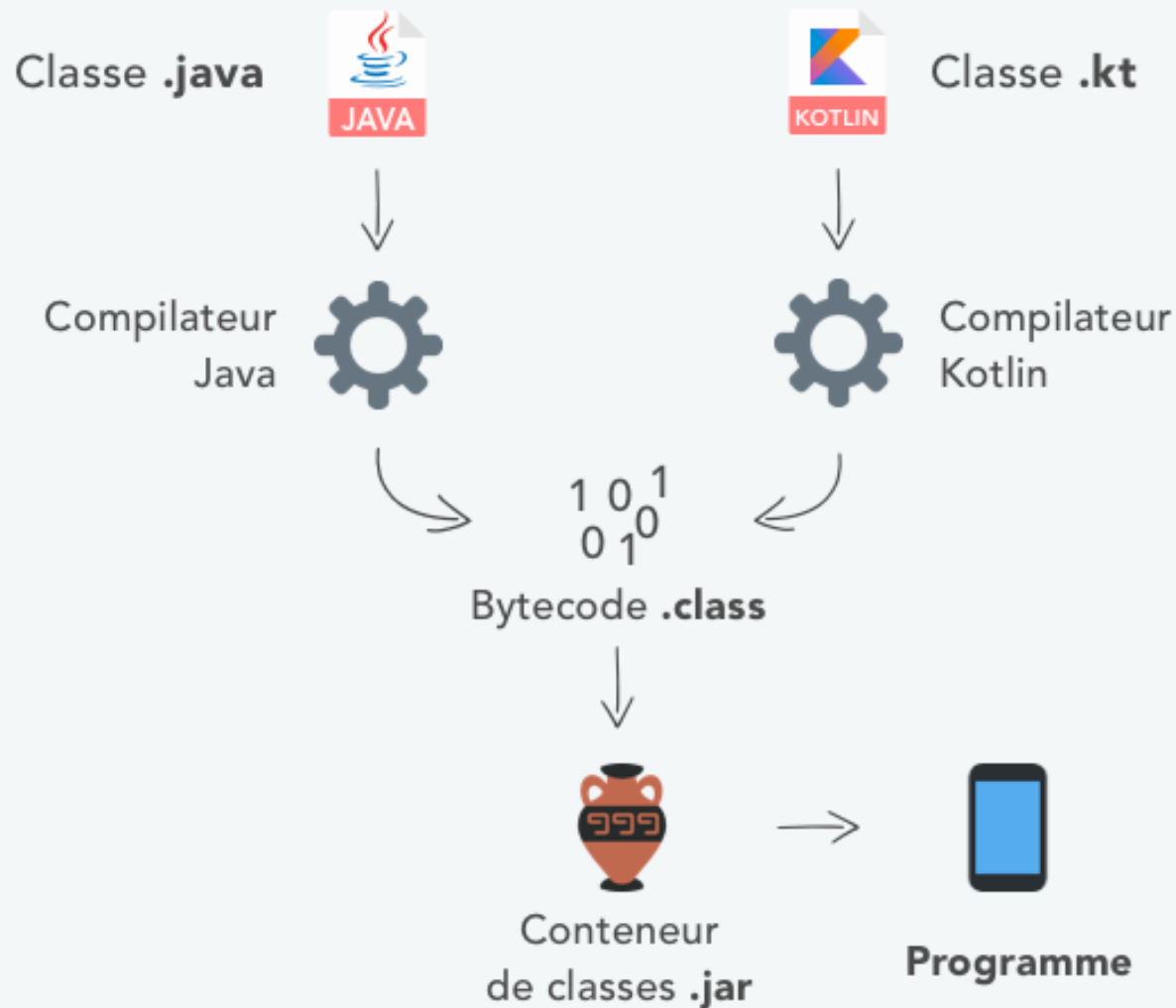
Historique

- **Kotlin** est un langage de programmation orienté objet et fonctionnel, avec un typage statique qui permet de compiler pour la machine virtuelle Java, JavaScript, et vers plusieurs plateformes en natif (grâce à LLVM).
- 2011 Première version
- 2017 Second langage pour Android (après Java)
- 2019 Langage officiel et recommandé par Google pour Android
- Support sur de nombreux autres frameworks (serveur, etc.)

Objectif

- Créé par des développeurs Java
- Doit être
 - Plus concis
 - Plus sûr
 - Plus pragmatique
 - Et.... 100% compatible avec Java

Fonctionnement



Example (Java)

```
public class Aquarium {  
  
    private int mTemperature;  
  
    public Aquarium() { }  
  
    public int getTemperature() {  
        return mTemperature;  
    }  
  
    public void setTemperature(int mTemperature) {  
        this.mTemperature = mTemperature;  
    }  
  
    @Override  
    public String toString() {  
        return "Aquarium{" +  
            "mTemperature=" + mTemperature +  
            '}';  
    }  
}
```


Exemple (Kotlin)

```
data class Aquarium (var temperature: Int = 0)
```

```
fun main() {  
    val aq = Aquarium()  
    aq.temperature = 12  
    println(aq)  
}
```

Affichage → Aquarium(temperature=12)

Exemples de code

```
fun birthdayGreeting(name: String, age: Int): String {  
    val nameGreeting = "Happy Birthday, $name!"  
    val ageGreeting = "You are now 5 years old!"  
    return "$nameGreeting\n$ageGreeting"  
}
```

Arguments nommés

```
println(birthdayGreeting(name = "Rex", age = 2))
```

```
println(birthdayGreeting(age = 2, name = "Rex"))
```

Paramètres par défaut

```
fun birthdayGreeting(name: String = "Rover", age: Int): String {  
    return "Happy Birthday, $name! You are now $age years old!"  
}
```

Exemples de code

```
fun main() {  
    val trafficLightColor = "Black"  
  
    val message =  
        if (trafficLightColor == "Red") "Stop"  
        else if (trafficLightColor == "Yellow") "Slow"  
        else if (trafficLightColor == "Green") "Go"  
        else "Invalid traffic-light color"  
  
    println(message)  
}
```

Gestion des nullable

```
fun main() {  
    var favoriteActor: String? = "Sandra Oh"  
    println(favoriteActor?.length)  
}
```

Exemples de code

Intégrer des fonction dans des variables

```
fun trickOrTreat(isTrick: Boolean, extraTreat: (Int) -> String): () -> Unit {  
    if (isTrick) {  
        return trick  
    } else {  
        println(extraTreat(5))  
        return treat  
    }  
}
```

Fonctions d'ordre supérieur

```
val softBakedMenu = cookies.filter {  
    it.softBaked  
}
```

Pour approfondir

- <https://developer.android.com/courses/pathways/android-basics-compose-unit-1-pathway-1?hl=fr>
- <https://developer.android.com/courses/pathways/android-basics-compose-unit-2-pathway-1?hl=fr>
- <https://developer.android.com/courses/pathways/android-basics-compose-unit-3-pathway-1?hl=fr>

Points clés pour une application de base

- Créer une application
- Créer une mise en page
- Ajouter des actions

Points importants

- Deux types de mises en pages
 - XML
 - Compose
- Disposition des éléments → un layout
- Différences dp et sp
 - dp : Density-independant Pixels, adapte à la résolution
 - sp : Scale-independant Pixels, préférence utilisateur (taille de police)
 - <https://developer.android.com/training/multiscreen/screendensities>

Exemple avec layout XML

Gérer les vues

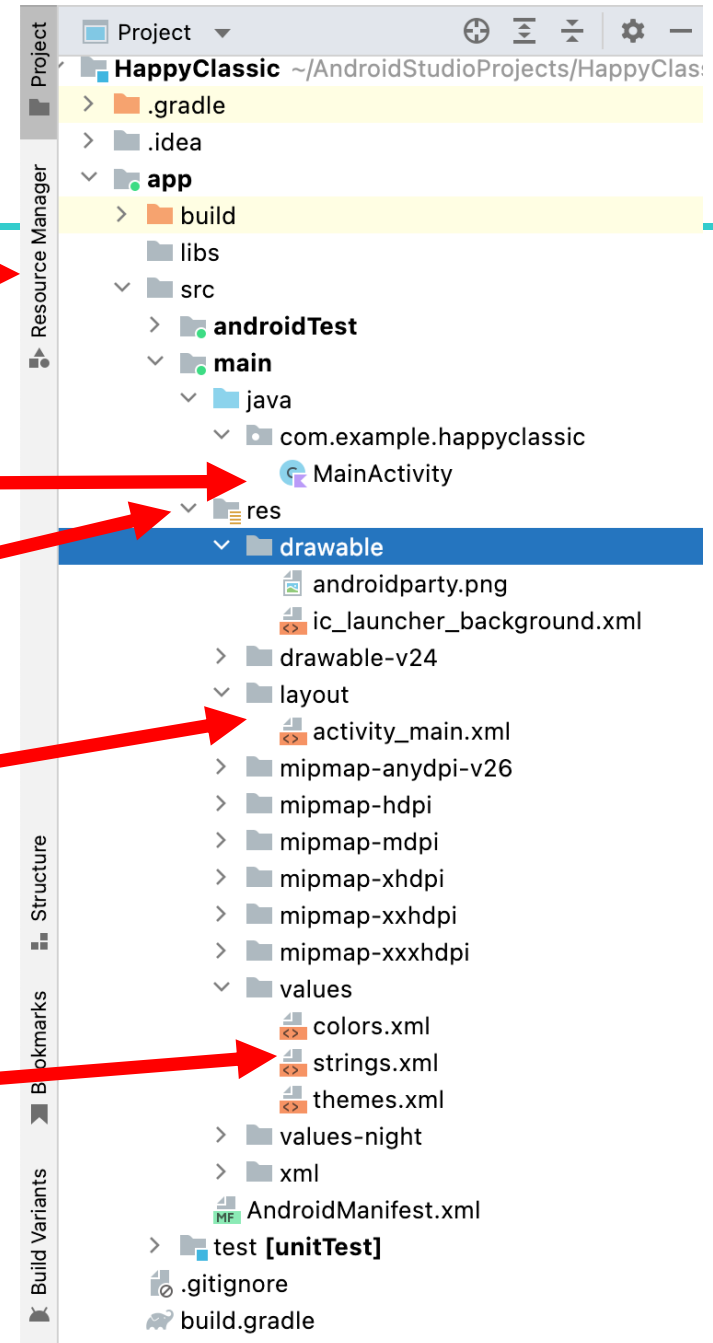
Gestion des ressources
(view → tool windows)

Point d'entrée

Ressources

Mise en forme

Valeurs-Constantes



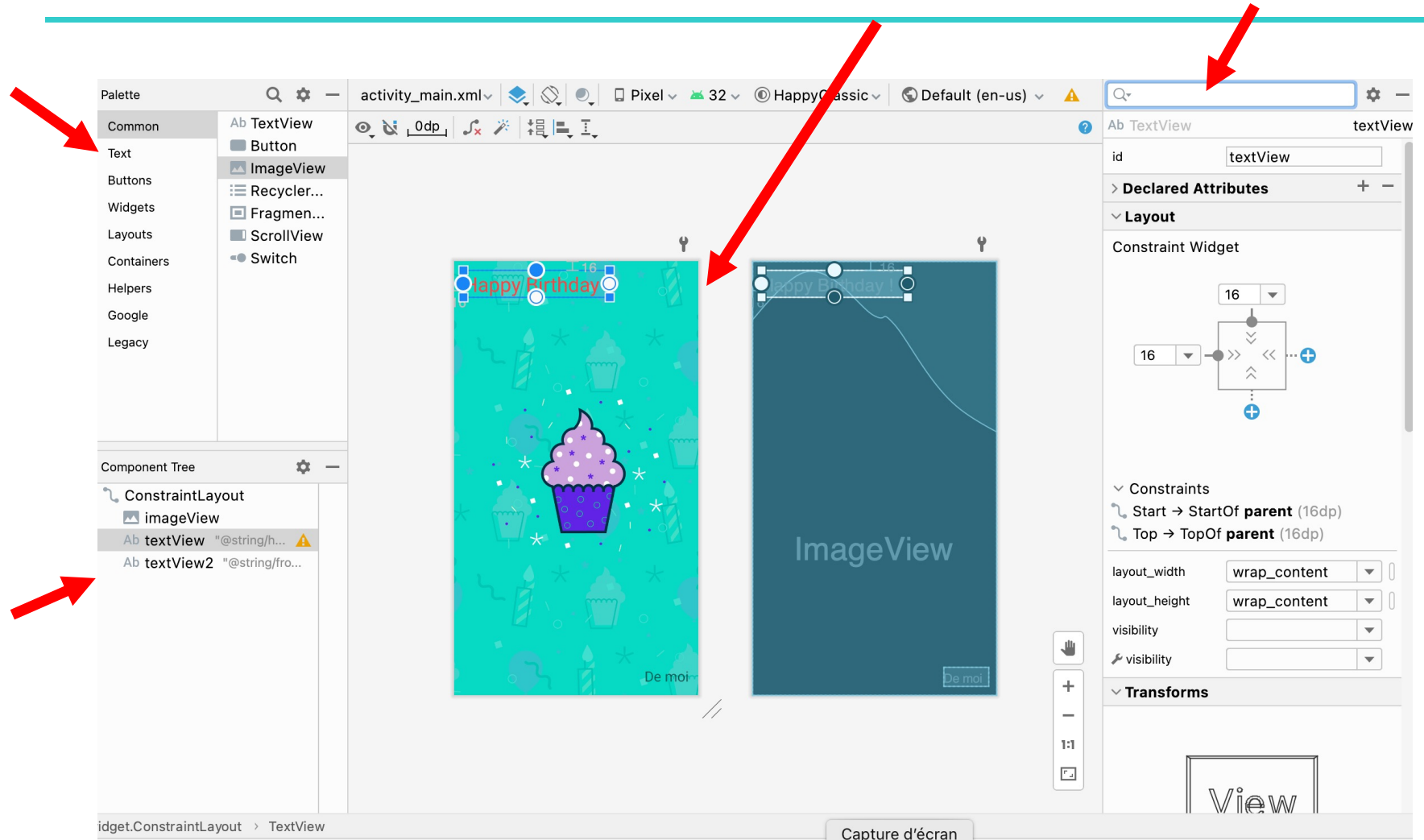
Gérer les vues

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">
```

```
<ImageView
    android:id="@+id/imageView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:importantForAccessibility="no"
    android:scaleType="centerCrop"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/androidparty" />
```

```
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="16dp"
    android:text="@string/happy_birthday_text"
    android:textColor="@android:color/holo_red_light"
    android:textSize="34sp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

Gérer les vues



Gérer les vues



Pour approfondir XML

- <https://developer.android.com/courses/pathways/android-basics-kotlin-two>
- <https://developer.android.com/courses/pathways/android-basics-kotlin-three>

Actions utilisateur



Bouton (XML)

<Button

```
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:text="@string/roll"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/imageView" />
```

```
private fun rollDice() {
    val dice = Dice(6)
    val diceRoll = dice.roll()
    val diceImage: ImageView = findViewById(R.id.imageView)
```

```
    val drawableResource = when (diceRoll) {
        1 -> R.drawable.dice_1
        2 -> R.drawable.dice_2
        3 -> R.drawable.dice_3
        4 -> R.drawable.dice_4
        5 -> R.drawable.dice_5
        else -> R.drawable.dice_6
    }
    diceImage.setImageResource(drawableResource)
```

```
val rollButton: Button = findViewById(R.id.button)
rollButton.setOnClickListener { rollDice() }
```

Pour approfondir XML

- <https://developer.android.com/courses/pathways/android-basics-kotlin-four?hl=fr>
- Regarder test et debuggage