Growing with Kotlin: Learning Kotlin after JavaScript

July 23, 2020

Hi there 👋

Software Engineer @ Procter and Gamble

Working on ecommerce, direct to consumer websites

Coding Bootcamp Grad

Plant Parent

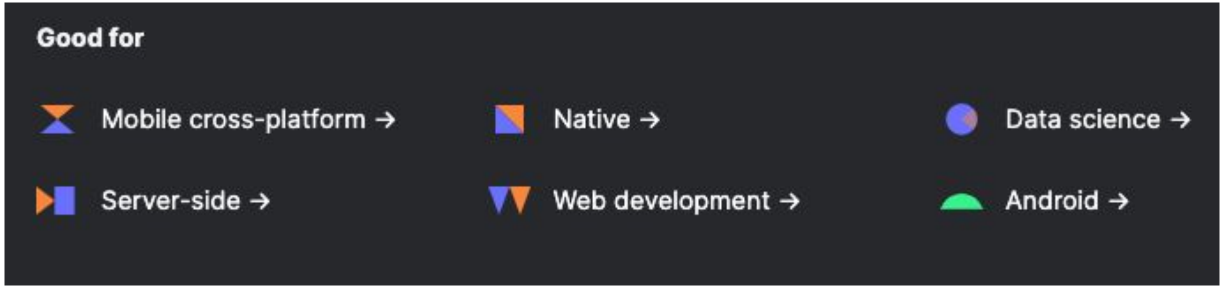
Linkedin: /alexxmitchell
Twitter: @alexxm025

Agenda

- What is Kotlin?
- Kotlin vs JS
- Declaring Variables and Types
- Nullability/Null Safety
- Coroutines
- Wrap Up

What is Kotlin?

- a cross-platform, statically typed, general-purpose programming language with type inference
- the official language of Android



Good for

Mobile cross-platform →
Native →
Data science →
Server-side →
Web development →
Android →

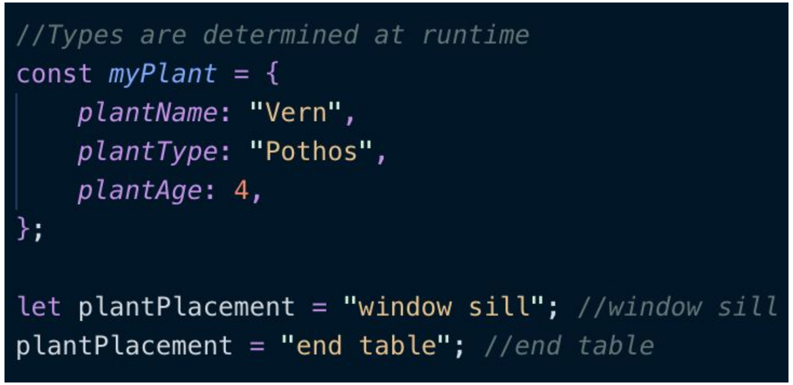
Kotlin vs JS

- Syntax
- How errors present themselves
- Type coercion
- Inheritance

But there are also similarities

Declaring Variables and Types

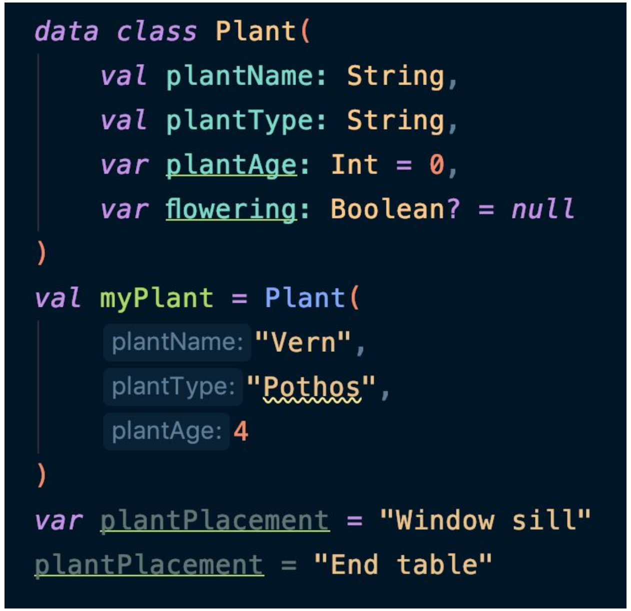
JavaScript



```javascript
//Types are determined at runtime
const myPlant = {
    plantName: "Vern",
    plantType: "Pothos",
    plantAge: 4,
};

let plantPlacement = "window sill"; //window sill
plantPlacement = "end table"; //end table
```

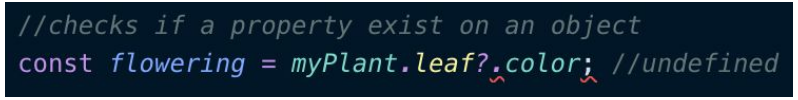
Kotlin



```kotlin
data class Plant(
    val plantName: String,
    val plantType: String,
    var plantAge: Int = 0,
    var flowering: Boolean? = null
)
val myPlant = Plant(
    plantName: "Vern",
    plantType: "Pothos",
    plantAge: 4
)
var plantPlacement = "Window sill"
plantPlacement = "End table"
```

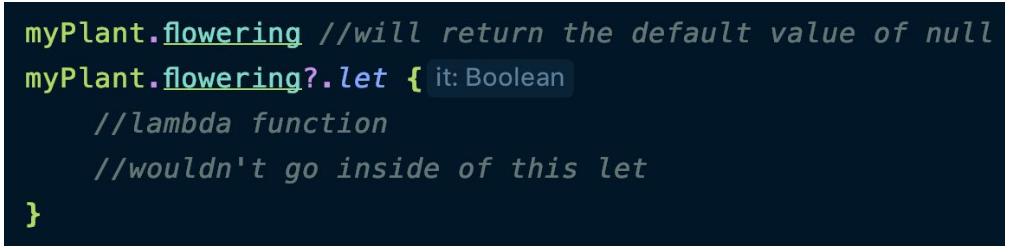
Null Safety

JavaScript

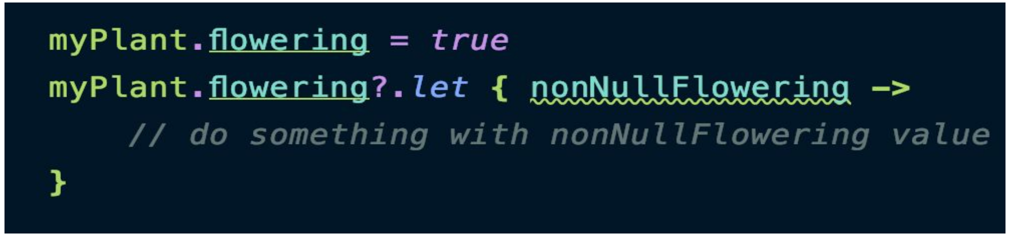


```javascript
//checks if a property exist on an object
const flowering = myPlant.leaf?.color; //undefined
```

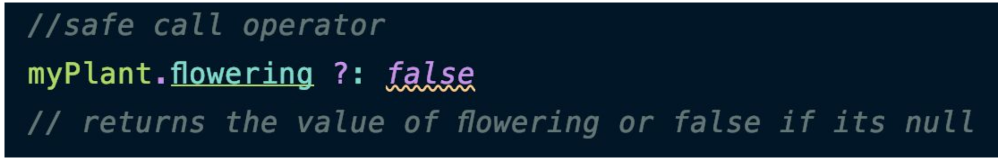
Kotlin



```kotlin
myPlant.flowering //will return the default value of null
myPlant.flowering?.let { it: Boolean
    //lambda function
    //wouldn't go inside of this let
}
```



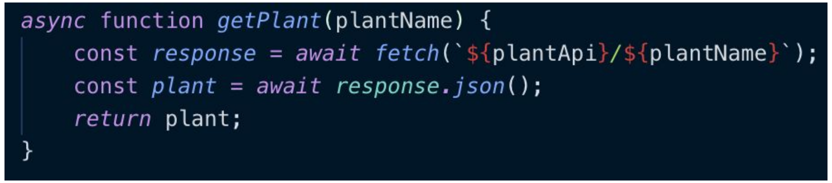
```kotlin
myPlant.flowering = true
myPlant.flowering?.let { nonNullFlowering ->
    // do something with nonNullFlowering value
}
```



```kotlin
//safe call operator
myPlant.flowering ?: false
// returns the value of flowering or false if its null
```

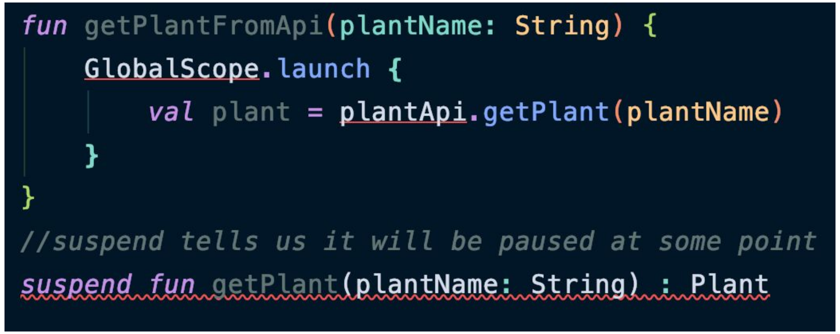
Async/Await and Coroutines

JavaScript



```javascript
async function getPlant(plantName) {
    const response = await fetch(`${plantApi}/${plantName}`);
    const plant = await response.json();
    return plant;
}
```

Kotlin



```kotlin
fun getPlantFromApi(plantName: String) {
    GlobalScope.launch {
        val plant = plantApi.getPlant(plantName)
    }
}
//suspend tells us it will be paused at some point
suspend fun getPlant(plantName: String) : Plant
```

Reasons I enjoy

- Static types
- A function for everything
- Associated IDEs make developing easier
- Ability to create for more than mobile

?