

# Lesson 1:

## Your first Android app



Instructor: Ahmet Geymen

# About this lesson

- Lesson 1: Your first Android app
  - Introduction to Kotlin
  - Meet Android Studio
  - Build a basic layout

# Get started

## Introduction to Kotlin

# Kotlin Playground

<https://developer.android.com/training/kotlinplayground>



# Your first program

```
fun main() {  
    println("Hello, world!")  
}
```

# Parts of a function

```
fun name ( inputs ) {  
    body  
}
```

```
      name inputs  
      ↓   ↓  
fun main() {  
    println("Hello, world!") ← body  
}
```

# Operators

# Operators

- Mathematical operators

+ - \* / %

- Increment and decrement operators

++ --

- Comparison operators

< <= > >=

- Assignment operator

=

- Equality operators

== !=



# Math operators with integers

$1 + 1 \Rightarrow 2$

$53 - 3 \Rightarrow 50$

$50 / 10 \Rightarrow 5$

$9 \% 3 \Rightarrow 0$

# Math operators with doubles

$1.0 / 2.0 \Rightarrow 0.5$

$2.0 * 3.5 \Rightarrow 7.0$

# Math operators

1 + 1           =>     kotlin.Int = 2

53 - 3           =>     kotlin.Int = 50

50 / 10           =>     kotlin.Int = 5

1.0 / 2.0       =>     kotlin.Double = 0.5

2.0 \* 3.5       =>     kotlin.Double = 7.0

# Numeric operator methods

`2.times(3)`      `=>`      `kotlin.Int = 6`

`3.5.plus(4)`      `=>`      `kotlin.Double = 7.5`

`2.4.div(2)`      `=>`      `kotlin.Double = 1.2`

# Variables

# Variables

- Powerful type inference
  - Let the compiler infer the type
  - You can explicitly declare the type if needed
- Mutable and immutable variables
  - Immutability not enforced, but recommended

Kotlin is a statically-typed language. The type is resolved at compile time and never changes.

# Variable declaration

`val` name : data type = initial value

name            data type            initial value

↓                   ↓                   ↙

`val count: Int = 2`

**Important:** Once a type has been assigned by you or the compiler, you can't change the type or you get an error

# Mutable and immutable variables

- Mutable (Changeable)

```
var score = 10
```

- Immutable (Unchangeable)

```
val name = "Jennifer"
```

Although not strictly enforced, using immutable variables is recommended in most cases.



# Mutable and immutable variables

```
var count = 1
```

```
count = 2
```

```
val size = 1
```

```
size = 2
```

=> Error: val cannot be reassigned

# Coding conventions

- Variable names should be in camel case and start with a lowercase letter.
- In a variable declaration, there should be a space after a colon when you specify the data type.

space



```
val discount: Double = .20
```

- There should be a space before and after an operator.

space



```
var pet = "bird"
```

space



```
val sum = 1 + 2
```

# Commenting

# Commenting conventions

```
// This is an end-of-line comment
```

```
/* This is a block comment  
   on multiple lines. */
```

```
/*  
 * This is a very long comment that can  
 * take up multiple lines.  
 */
```

# Data types

# Numbers - Integer types

Type	Bits	Notes
Long	64	From $-2^{63}$ to $2^{63}-1$
Int	32	From $-2^{31}$ to $2^{31}-1$
Short	16	From -32768 to 32767
Byte	8	From -128 to 127

# Numbers - Floating-point types

Type	Bits	Notes
Double	64	16 - 17 decimal digits
Float	32	6 - 7 decimal digits

# Strings

- Strings are any sequence of characters enclosed by double quotes.

```
val s1 = "Hello world!"
```

- String literals can contain escape characters

```
val s2 = "Hello world!\n"
```

- Multiline strings are delimited by a triple quote ("\"")

```
val text = """  
    for (c in "foo")  
        print(c)  
    """
```



# String concatenation

```
val numberOfDogs = 3
```

```
val numberOfCats = 2
```

```
val sentence = "I have $numberOfDogs dogs" + " and  
$numberOfCats cats"
```

```
println(sentence) => I have 3 dogs and 2 cats
```

# String templates

- A template expression starts with a dollar sign ( `$` ) and can be a simple value:

```
val i = 10
println("i = $i")
=> i = 10
```

- Or an expression inside curly braces:

```
val s = "abc"
println("$s.length is ${s.length}")
=> abc.length is 3
```

# String template expressions

```
val numberOfShirts = 10
```

```
val numberOfPants = 5
```

```
val sentence = "I have ${numberOfShirts +  
numberOfPants} items of clothing"
```

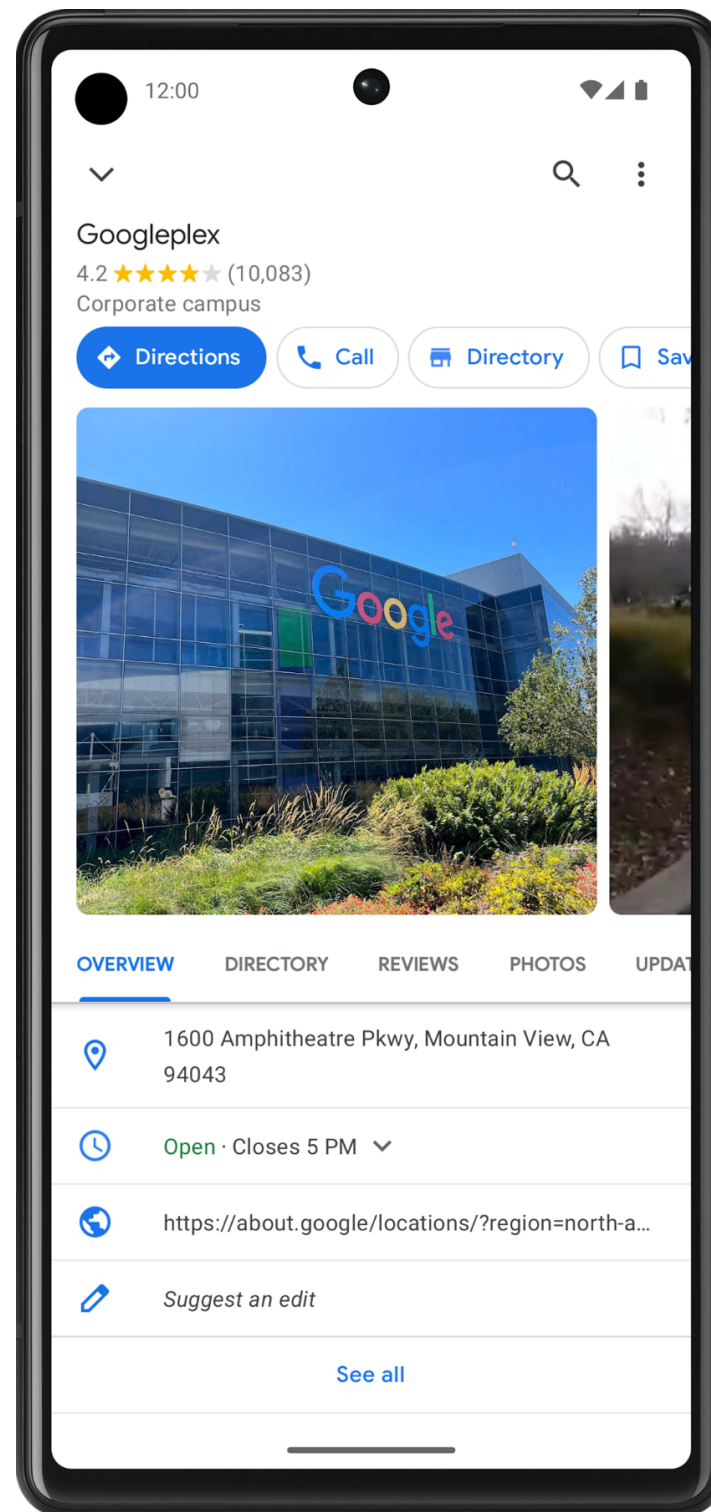
```
println(sentence) => I have 15 items of clothing
```

# Most common types with samples

Kotlin data type	What kind of data it can contain	Example values
String	Text	"Search" "Sign in" "Send"
Int	Integer number	32 172630
Double	Decimal number	2.0 501.0292 -2138.9999
Float	Decimal number (less precise than a Double)	5.0f -1630.209f 1.34238F
Boolean	true or false	true false

# Simple practice

Can you guess, which data type would be appropriate for each UI element on this screen?

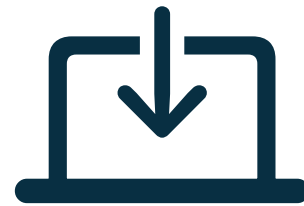


# Meet Android Studio

Overview of the official IDE

# Download Android Studio

<https://developer.android.com/studio>



Android Studio Giraffe

# Workshop

**Build a basic layout**