اختصارات لغة Kotlin

الاسم	الأمر	مثال	الشرح
Variable المتغير	var	var age = 5 age = 10	يتم تعريف متغير باستخدام var, حيث يمكننا تغير قيمته في أي وقت.
الثابت	val	"val name = "Salem	يتم تعريف ثابت باستخدام val, حيث لا يمكننا تغير قيمته بعد تعريفه.
Data نوع البيانات type	Type :	var age: Type	يمكننا تعيين نوع البيانات للمتغيرات و الثوابت عن طريق إضافة : بعد اسم المتغير ثم إضافة نوع البيانات بعد ذلك.
Int العدد الصحيح	Int	var age: Int = 1	يمكننا تحديد نوع البيانات لتمثيل عدد صحيح باستخدام Int.
العدد العشري Float/Double	Float / Double	var grade: Float = 1.5	يمكننا تحديد نوع البيانات لتمثيل عدد عشري باستخدام Float أو Double.
قيمة منطقية Boolean	Boolean	var isFood: Boolean = true	يمكننا تحديد نوع البيانات لتمثيل قيمة منطقية باستخدام Boolean, حيث تكون قيمتها إما true أو false.
Char حرف أو رمز	Char	var sign: Char = '@'	يمكننا تحديد نوع البيانات لتمثيل حرف أو رمز باستخدام Char.
String نص	String	var name: String = "Salem"	يمكننا تحديد نوع البيانات لتمثيل نص باستخدام String.

Method / الدالة Function	fun myFun() { }	fun printName() {
		print("Salem")
		}
			لتعريف دالة نقوم باستخدام الأمر fun ثم اسم الدالة يليه الأقواس.

تعيين نوع للدالة	: Type	fun getName(): String {
		return "Salem"
		}
			يمكننا تحديد نوع لناتج الدالة باستخدام نفس طريقة تحديد أنواع للمتغيرات, حيث نقوم بإضافة : بعد الأقواس () ثم نحدد النوع.

معاملات الدوال	fun myFun(a: Type, b: Type) { }	fun doubleTheNumber(num: Int): Int {
		return num*2
		}
			لإرسال معاملات للدوال يمكننا إضافتها بداخل الأقواس () مع تحديد نوع لكل معامل.

تعيين قيم افتراضية لمعاملات الدوال		fun theYear(year: Int = 2019): String
		{
		return "%{year} السنة"
		}
			يمكننا تعيين قيم افتراضية لمعاملات الدوال عن طريق إضافة علامة = ثم القيمة الافتراضية. وعند استدعاء الدالة يمكننا عدم إرسال أي معامل وسيتم استخدام 2019 كقيمة افتراضية للمعامل year.

Array المصفوفة	arrayOf()	arrayOf(1, 2, 3)	يمكننا تعريف مصفوفة بكل بساطة عن طريق استخدام الأمر arrayOf وتعيين قيم المصفوفة بداخل الأقواس ().
حجم الصفوفة	.size	var myArray = arrayOf(1, 2, 3) myArray.size	يمكننا معرفة حجم المصفوفة عن طريق استخدام الأمر size على المصفوفة.
الوصول إلى عنصر في المصفوفة	[ ]	myArray[0]	نستخدم الأقواس المربعة مع مؤشر العنصر في المصفوفة للوصول إلى العنصر بداخل المصفوفة.
تغيير قيمة في المصفوفة	[ ]	myArray[0] = 5	نستخدم الأقواس المربعة مع مؤشر العنصر في المصفوفة وتعيين القيمة الجديدة للعنصر بداخل المصفوفة.

if else جملة	if else	var age = 1
		if (age != 0) {
		print(age)
		} else {
		print("العمر غير مطابق")
		}
			لاستخدام جملة if else نقوم بكتابة الكلمة if ثم الأقواس () وبداخلها الشرط ثم نكتب كلمة else.

when جملة	when	when(age) {
		1 -> print(age)
		20 -> print(age - 10)
		else -> print("العمر غير مطابق")
		}
			في Kotlin تم استبدال جملة switch بجملة when. نكتب جملة when ثم يليها القيم المتوقعة و الكود الخاص بكل قيمة وفي النهاية يمكننا إضافة جملة else لتمثل عدم تطابق القيمة.

for جملة	for	for (item in myArray) {
		print(item)
		}
			لاستخدام جملة for نقوم بكتابة الكلمة for ثم الأقواس () وبداخلها item ويمثل أحد العناصر بداخل المصفوفة myArray في كل مرة يتم تشغيل الجملة.

while جملة	while	var age = 1
		while (age != 0) {
		print(age)
		}
			لاستخدام جملة while نقوم بكتابة الكلمة while ثم الأقواس () وبداخلها الشرط.

Class الكلاس	class	class Human { }	لتعريف class في Kotlin نستخدم كلمة class ويليها الاسم. كل class في Kotlin هو public ولكن لا يمكنك الوراثة منه.

جعل الكلاس يورث	open class	open class Human { }	لنتمكن من الوراثة من class في Kotlin يجب علينا تعريفه كـ open, بإضافة الكلمة open إليه.
الدالة المنشئة Constructor	constructor	class Human( ) { }	لإضافة Constructor إلى class نقوم بإضافة الأقواس () بعد الاسم. وهذا النوع من الـ Constructors يعرف بـ Primary Constructor.

الدالة المنشئة Constructor	constructor	class Human( ) {
		constructor(name: String): this() {
		
		}
		}
			يمكن إ ضافة أكثر من Constructor عن طريق تعريفها داخل الـ class بإضافه الكلمة constructor ثم الأقواس () والمعاملات إن وجدت. مع التأكد من استدعاء الـ Primary Construtor كما في المثال : this() إن وجد.

كلاس البيانات Data Class	data class	data class Human(val name: String, val age: Int)	في بعض الأحيان نحتاج إلى كلاس ليمثل البيانات فقط, يمكننا استخدام data class لذلك. عند تعريف data class يمكننا إضافة الحقول التي نريد وسيتم إضافة كل التفاصيل الأخرى التي نحتاجها من قبل Kotlin تلقائيا.
الكائن Object	object	object Sun { }	في بعض الأحيان تريد تعريف بعض الدوال والحقول في مكان واحد لتتمكن من استخدامها في كل مكان. يمكننا استخدام object وتعريف singleton في Kotlin.

الكائن الرفيق Companion Object	companion object	class Human( ) {
		companion object {
		
		}
		}
			في بعض الأحيان تحتاج إلى تعريف بعض الدوال والحقول في الكلاس ولكن تريدها أن تكون ثابته لكل الكوائن من هذا الكلاس, يمكننا استخدام companion object وتعريفه بداخل الكلاس.

Null Safety	?	var age: String? = null	في Kotlin حتى نتمكن من تعيين null لمتغير يجب علينا إصافة ? إلى النوع.
? الاستدعاء من متغير	?.	age?.lenght	عند استدعاء أي دالة أو حقل من متغير يمكن أن تكون قيمته null يجب علينا إضافة ? قبل النقطة. وفي حال كون قيمة age تساوي null سيتم إرجاع null كقيمة للأمر.
!! الاستدعاء من متغير	!!	age!!.lenght	عند استدعاء أي دالة أو حقل من متغير يمكن أن تكون قيمته null يمكننا إضافة !! قبل النقطة. وفي حال كون قيمة age تساوي null سنحصل على استثناء Null Pointer Exception