

لغة kotlin

لغة kotlin هي احدي اللغات التي لاقت روجا كبيرا في السنين الماضيه وزادت شعبيتها بعد دعمها بشكل رسمي في اندرويد استديو لبرمجة تطبيقات الاندرويد. لغه ال kotlin قادرة علي برمجة تطبيقات الويب والosi بشكل طبيعي باستخدام ادوات للتطوير مدعومة بشكل كبير من الشركة المطورة للغة kotlin.

❖ مزايا لغة الkotlin :-

- 1-لغة مفتوحة المصدر.
 - 2-توفير الوقت والجهد.
 - 3-متوافقة مع Android Studio
 - 4-أخطأ أقل وتتميز بقضائها علي مشكلة NullPointerException.
 - 5- تتميز بالانسيابية.
 - 6- أكثر أمان.
 - 7- متعددة الاستخدامات ويمكن من خلالها تطوير مواقع الانترنت السيرفرات سطح المكتب.....الخ.
- المتغيرات :-
- عبارة عن قيمة قابلة للتغير أثناء كتابة البرنامج ولتعريف المتغيرات في لغه الكوتلن نستخدم كلمة var
- اسناد قيم للمتغيرات:-

يمكنك التعريف بالمتغير عن طريق تعريفه واسناد القيمة له مباشرة .

```
var myName="Ahmed"  
var myAge=27
```

في المثال أعلاه لم نذكر النوع ولكن اسندنا القيمة كيف يستطيع البرنامج معرفة نوع المتغير؟ بواسطة compiler الذي يقوم بمعرفة النوع من خلال القيمة المسندة له.

إذا قمت باسناد قيمة للمتغير من نوع Int مثلاً فانك لن تستطيع تغييرها الى اي نوع آخر.

```
var age = 12  
age = "12 years old" // Error: type mismatch
```

❖ تعريف بالمتغير والتعريف بنوعه واسناد القيمة مباشرة

يمكنك التعريف بالمتغير والتعريف بنوعه مع مراعاة كتابة أول حرف من نوعه بحرف كبير.

```
var myName : String ="Ahmed"  
var myAge:Int=27
```

❖ تعريف بالمتغير والتعريف بنوعه واسناد القيمة لاحقاً

يمكنك التعريف بالمتغير والتعريف بنوعه مع مراعاة كتابة أول حرف من نوعه بحرف كبير في سطر ويمكننا اسناد القيمة خلال اسطر البرنامج أو في السطر الذي يليه.

```
var myName : String  
myName="Ahmed"
```

يراعى عند تعريف المتغيرات في لغة kotlin أنها لا تبدأ بأي حرف من الرموز المخصصة باستثناء (.) ما تعرف بالشرطة التحتية أو underscore.

ثانياً أنواع المتغيرات:-

العدد الصحيح	Int
عدد صحيح قصير	Short
عدد صحيح طويل	Long
قيمة عشرية بسيطة	Float
قيمة عشرية أكثر دقة وتحديد	Double
حرف	Char
سلسلة نصية	String
متغير منطقي	Boolean

للتفريق بين ال Double و ال Float هو عندما نريد تعريف متغير عشري من النوع float نضع نهايته حرف f.

مع ملاحظة ان المتغير double يتكون من bit64 بعكس float الذي يتكون من bit32 فقط

```
var cost:Double=12.9001111  
var cost = 14.1655f
```

❖ الثوابت

تنطبق عليه نفس مواصفات المتغيرات باستثناء أنه قيمة غير قابلة للتعديل أثناء البرنامج ولتعريف الثابت نستخدم كلمة `val` ، يمكننا الاستفادة منه في تعريف القوانين الرياضية أو لو كان هنالك رقم وظيفي فهو ثابت بالطبع لا يمكن تغييره أو رقم الهوية الوطني.

```
val id:Int=601121  
id = 1234 // Error
```

ثالثاً: تحويل المتغيرات

ربما كتبت متغير و اردت تحويل الى اي نوع آخر فانك لن تستطيع ذلك بسبب انك قد حددت نوع المتغير او ان Compiler قد قام بتحديد نوع المتغير اذا كان اسناد القيمة بشكل مباشر. تحويل المتغيرات ايسر مما تتخيل لن تحتاج الى اكثر من سطر او ذاكرة كبيره لحفظ الطريقة. تابع الشفرة القادمة.

```
val number1: Int = 55  
val number2: Long = number1.toLong()
```

لاحظ قمنا بتحويل Int الى Long باستخدام دالة `toLong()` وبشكل سلس وبسيط.

الرموز الرياضية:-

في البداية نحب أن نقوم بتعريف العمليات الأساسية وهي : الجمع، الطرح، الضرب، القسمة وباقي القسمة.

الرمز	العملية
+	الجمع
-	الطرح
*	الضرب
/	القسمة
%	باقي القسمة

في العمليات الرياضية يراعى عند تنفيذها إذا كانت تتضمن الأقواس، وعلية الضرب / القسمة، والجمع / الطرح التالي:

نبدأ بتنفيذ ما بين الأقواس ومن ثم عملية الضرب / القسمة ومن ثم الجمع / الطرح.

هذا مثال يحسب لنا مساحة المستطيل ومحيطه، نجد أن المساحة هي مجرد عملية ضرب عادية ، وبالنسبة للمحيط فهو يعطي أولوية التنفيذ إلى الأقواس ومن ثم يضرب الناتج في ٢.

```
var length:Int=5
var width:Int=7
var distance=length * width
println(distance)
var circumference= 2*(length+width)
println(circumference)
```

هنا قمنا باختصار فلدي متغير يحمل قيمة في البرنامج و اردنا اجراء عملية الضرب عليه فقمنا بكتابة `num *=3` بدلا من كتابة `num=num*3`.

كذلك الأمر لو أردنا اجراء طرح او قسمة نقوم بكتابة اسم المتغير ومن ثم رمز العملية علامة المساواة ومن ثم العدد الذي نرغب بإجراء العملية عليه.

```
var num=9
println(num)
num *=3
println(num)
```

كذلك الأمر لو أردنا اجراء طرح او قسمة نقوم بكتابة اسم المتغير ومن ثم رمز العملية علامة المساواة ومن ثم العدد الذي نرغب بإجراء العملية عليه.

```
var num=9
println(num)
num *=3
println(num)
```

عملية الجمع `fun main(args:array<string>) {`

`var m:int=7`

`var y:int=2`

`var sum=m+y`

`print(sum)`

`}`

عملية الطرح:-

`fun main(args:array<string>) {`

`var m:int=7`

`var y:int=2`

```
var z=m-y
```

```
print(z)
```

```
}
```

عملية القسمة:-

```
fun main(args:array<string>) {
```

```
var m:int=7
```

```
var y:int=2
```

```
var z= m/y
```

```
print(z)
```

```
}
```

عملية باقي القسمة:-

```
fun main(args:array<string>) {
```

```
var m:int=7
```

```
var y:int=2
```

```
var z= m%y
```

```
print(z)
```

```
}
```

عملية الضرب:-

```
fun main(args:array<string>) {
```

```
var m:int=7
```

```
var y:int=2
```

```
var z= m*y
```

```
print(z)
```

```
}
```