**متغير ثابت: Val keyword  
(Immutable variable: val keyword)**

يتم التصريح عن المتغير الثابت باستخدام الكلمة الأساسية ***var***في ***Kotlin***. في هذا المثال، أعلنا عن متغير غير قابل للتغيير ***myName***باستخدام كلمة رئيسية ***val***ثم عرضنا قيمته لاحقًا.

**fun** main**(**args : Array**<**String**>){**

/\*\*

\* This is an immutable variable

\* also called unchangeable variable

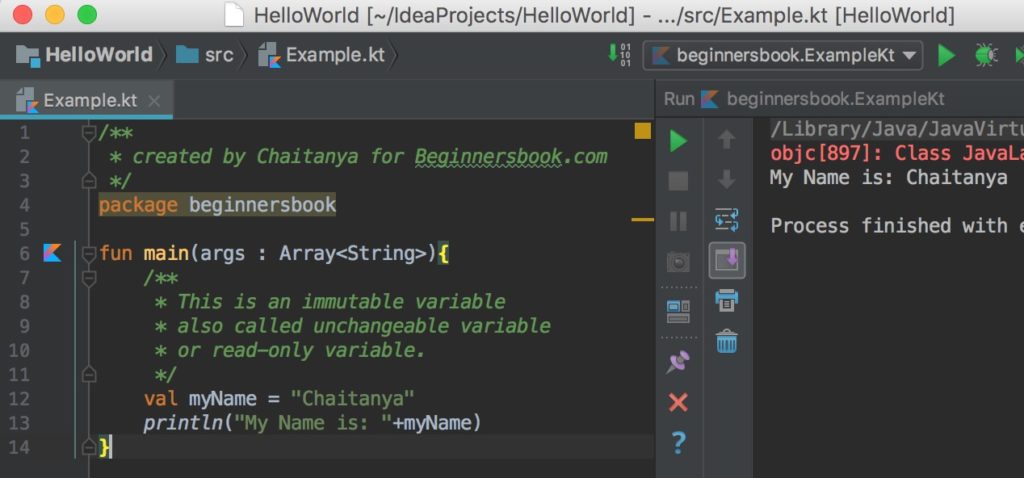
\* or read-only variable.

\*/

**val** myName = "Chaitanya"

println**(**"My Name is: "+myName**)**

**}**

الناتج

**ماذا يحدث عندما نحاول تغيير قيمة متغير غير قابل للتغيير؟**  
إذا حاولنا تغيير قيمة متغير غير قابل للتغيير ، فسنحصل على خطأ في التجميع، تمامًا كما حصلنا في المثال التالي.

**package** beginnersbook

**fun** main**(**args : Array**<**String**>){**

/\*\*

\* This is an immutable variable

\* also called unchangeable variable

\* or read-only variable.

\*/

**val** myName = "Chaitanya"

myName="Chaitanya Singh"

println**(**"My Name is: "+myName**)**

**}**

الناتج:

Error:**(**13, 5**)** Kotlin: Val cannot be reassigned

**متغير كوتلن المتغير: var keyword  
(Mutable variable: var keyword)**

على عكس المتغيرات الثابتة ، يمكننا تغيير قيمة المتغير القابل للتغيير. في kotlin ، نستخدم الكلمة الأساسية ***var***للإعلان عن متغير قابل للتغيير. لنأخذ مثالاً لفهم هذا.

في هذا المثال ، أعلنا عن متغير قابل للتغيير باستخدام الكلمة الأساسية ***var***، لإثبات أنه يمكننا تغيير قيمته ، قمنا بإعادة تعيين قيمة مختلفة لمتغير myName.

**fun** main**(**args : Array**<**String**>){**

/\*\*

\* This is an mutable variable

\* we can change the value of this

\* variable

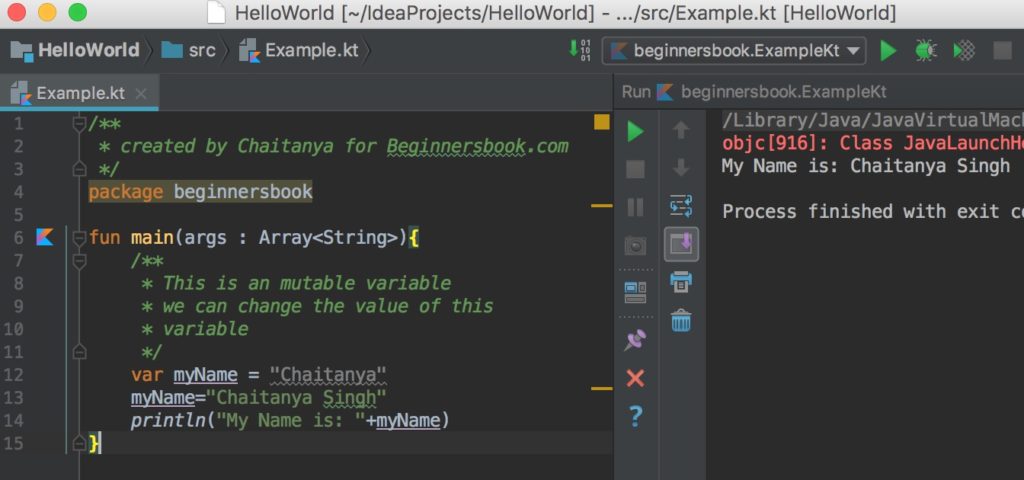
\*/

**var** myName = "Chaitanya"

myName="Chaitanya Singh"

println**(**"My Name is: "+myName**)**

**}**

انتاج

**ما هو المتغيرفي Kotlin؟**

المتغير هو اسم يُعطى لموقع في الذاكرة يمكنه الاحتفاظ بالبيانات. على سبيل المثال ، عندما أعلن عن المتغير مثل هذا:

**var** website = "coursee.org/"

يتم تخزين البيانات “***coursee.org***” في الذاكرة في موقع معين ، يسمى ***website***.

هنا var هي كلمة أساسية تستخدم للإعلان عن متغير ، website هو معرف (اسم المتغير) ، “***coursee.org***” هو البيانات (قيمة المتغير) ونوع المتغير هو String (سنناقشه لاحقًا) .

**Type inference** :

كما ذكرنا سابقًا ، يمكننا التصريح عن المتغير وتهيئته في عبارة واحدة مثل هذا:

**var** website = "coursee.org/"

في البيان أعلاه ، لم نحدد نوع المتغير ، تعلم kotlin أن website المتغير عبارة عن سلسلة. يمكن للمجمع أن يفهم نوع المتغير من خلال النظر إلى القيم.

ومع ذلك ، إذا كنت تريد أن تذكر نوع المتغير صراحة ، فيمكنك القيام بذلك على النحو التالي:

**var** website: String = "coursee.org"

هنا ذكرنا صراحة نوع المتغير “website” على أنه String.

**الإعلان أولاً والتهيئة لاحقًا:**  
يمكننا الإعلان عن المتغير أولاً ثم يمكننا تهيئته لاحقًا في البرنامج. النقطة المهمة التي يجب ملاحظتها هنا هي أننا نقوم بذلك على هذا النحو ، يجب عليك تحديد نوع المتغير أثناء الإعلان.

//It is mandatory to specify the type in this case

**var** website: String

website = "coursee.org"

**أنواع بيانات Kotlin**

**1. Numbers – Byte, Short, Int, Long, Float, Double  
2. Boolean – True, false  
3. Characters  
4. Arrays  
5. Strings**

**Numbers**

لدينا العديد من أنواع البيانات لتمثيل الأرقام في Kotlin.

**1. Byte**

نطاق نوع بيانات بايت هو -128 إلى 127. يستخدم هذا لتمثيل قيم عدد صحيح أصغر.

**fun** main**(**args : Array**)** **{**

**val** num: Byte = 99

println**(**"$num"**)**

**}**

النتيجة:

99

يجب أن تكون على دراية بنطاق نوع البيانات لتجنب الأخطاء. على سبيل المثال ، قد ينتج عن التعليمة البرمجية التالية خطأ لأن القيمة المعينة لمتغير نوع البايت ليست في النطاق.

**fun** main**(**args : Array**)** **{**

//Range is -128 to 127

**val** num: Byte = 300

println**(**"$num"**)**

**}**

النتيجة:

error

يمكننا أيضًا معرفة الحد الأدنى والحد الأقصى لقيمة نوع بيانات مثل هذا:

**fun** main**(**args : Array**<**String**>){**

**var** bigByte: Byte = Byte.MAX\_VALUE

**var** smallByte: Byte = Byte.MIN\_VALUE

println**(**"Biggest Byte value is: " + bigByte**)**

/\*\*

\* I can use String interpolation and put

\* the variable inside quotes to display its

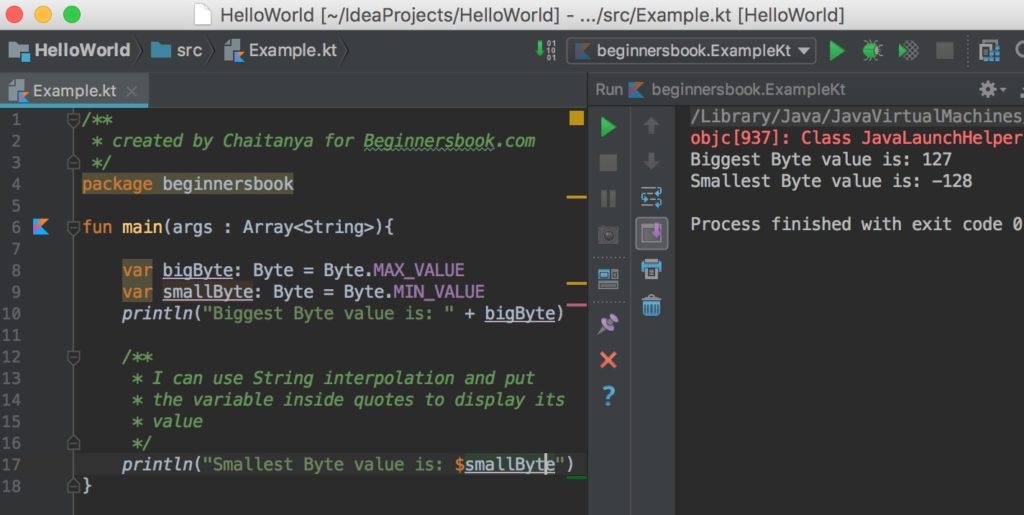
\* value

\*/

println**(**"Smallest Byte value is: $smallByte"**)**

**}**

الناتج :



**2. Short**

نطاق نوع البيانات القصيرة هو -32768 إلى 32767.

**fun** main**(**args : Array**)** **{**

**val** num: Short = 29999

println**(**"$num"**)**

**}**

الناتج :

29999

قد تتساءل عندما يكون لدينا نطاق أكبر في نوع البيانات القصيرة ، فلماذا نستخدم نوع بيانات بايت؟ هذا لحفظ الذاكرة. يحتوي نوع البيانات القصيرة على ذاكرة أكبر مقارنة بنوع بيانات بايت ، لذا إذا كنت متأكدًا من أن القيمة ستكون في حدود -128 إلى 127 ، فإن نوع بيانات بايت سيكون خيارًا أفضل من منظور الذاكرة

**3. Int**

نطاق نوع البيانات Int هو -231  الی 231-1

**fun** main**(**args : Array**)** **{**

**val** num: Int = 123456

println**(**"$num"**)**

**}**

الناتج :

123456

**ملاحظة:** إذا لم تحدد نوع المتغير صراحةً ، فسيعامل المترجم هذا المتغير على أنه Int ، إذا كانت القيمة تتراوح بين-231 الی 231-1

**4. Long**

نطاق نوع البيانات الطويل هو263 الی 263-1

**fun** main**(**args : Array**)** **{**

**val** num: Long = 12345678

println**(**"$num"**)**

**}**

الناتج :

12345678

**ملاحظة:** إذا لم تحدد نوع المتغير صراحةً ، فسيعامل المترجم هذا المتغير على أنه طويل ، إذا كانت القيمة خارج النطاق من-231 الی 231-1 ولكنها تقع في النطاق من -263 الی 263-1

**5. Double**

**fun** main**(**args : Array**)** **{**

// all floating numbers are double by default

// unless you suffix the value with F letter

**val** num = 101.99

println**(**"$num"**)**

**}**

الناتج :

101.99

**6. Float**

**fun** main**(**args : Array**)** **{**

// This is a float data type as we have suffixed

// the value with letter 'F'

**val** num = 101.99F

println**(**"$num"**)**

**}**

النتيجة :

101.99

**7.Boolean**

قيمة المتغير المنطقي إما صحيحة أو خاطئة.

**fun** main**(**args : Array**)** **{**

**val** boolValue = **false**

println**(**"$boolValue"**)**

**}**

النتيجة :

**false**

**8.Char**

يتم تضمين جميع الحروف الهجائية الإنجليزية (صغيرة أو كبيرة) في نوع البيانات Char. لا يمكنك تعيين أرقام لمتغير نوع بيانات Char.

**fun** main**(**args : Array**)** **{**

**val** ch = 'A'

println**(**"$ch"**)**

**val** ch2: Char

ch2 = 'Z'

println**(**"$ch2"**)**

**}**

النتيجة :

A

Z