# Exercises about data types

* Solve them in Visual Studio or use <https://dotnetfiddle.net/>.

## Exercise 04.01

* Create a variable (amount) and give it the value 123,4567.
  + Do the exercise with data type “double”.
* Add 234,5678 to variable amount.
* Show the result to the console.

|  |  |
| --- | --- |
|  |  |
|  | Do the exercise again with the data type “decimal”. |
|  |  |

## Exercise 04.02

* Define a variable that contains the date and time of the current moment. (DateTime.Now)
* Show on the console in different lines:
  + Year: the year of now.
  + Month: the month of now.
  + Day: the day of now.
  + Hour: the hour of now.
  + Minutes: the minutes of now.
  + Seconds: the seconds of now.

## Exercise 04.03

* Define a variable of the data type double and add the value 0,1234567890123456789 to it.
* Show on the console that value.

|  |  |
| --- | --- |
|  |  |
|  | What do you see?  You have done already this kind of exercise, so this is a kind of repeating yourself.  Do you know how to correct this? |
|  |  |

## Exercise 04.04

The code below generates random a number between 1 and 20.

* Lower number is included.
* Bigger number is excluded.

static void Main()

{

Random bytRandom = new Random();

Console.WriteLine(bytRandom.Next(1, 20));

}

* Create using this method “Next” to generate a password with minimum 8 and maximum 14 random characters.
* Get a random number.
* Find in the alphabet “abcdefghijklmnopqrstuvwxyz” the corresponding character.
  + Pay attention that there is sometimes a “z” in the password.

### An example of a correct result

* adqrazzgup

## Exercise 04.05

* Build on the result of 04.04 or restart.
* Every letter can be used only once. This means that a letter can’t occur twice in the password.

### An example of a correct result

* adqrzgupb

## Exercise 04.06

* Build on the result of 04.05 or restart.
* Capitals, digits and &§!+/.?$ can also be used.
* When your solution method is correct, the routine / method that generated the password should not change at all.

## Exercise 04.07

|  |  |
| --- | --- |
|  |  |
|  | You will have unexpected results in this routine. Mark them with a comment in your exercise.  This exercise is easy, but hard because of the many steps. Add code, test, add something, test again, and so on … |
|  |  |

* Declare a variable with name “myByte” of data type “byte”.
* Give that variable the value 123.
* Show the value of “myByte” to the console.
* Declare a variable with name “myShort” of data type “short”.
* Give that variable the value of “myByte”.
* Show the value of “myShort” to the console.
* Declare a variable with name “myInt” of data type “int”.
* Give that variable the value of “myShort”.
* Show the value of “myInt” to the console.
* Declare a variable with name “myLong” of data type “long”.
* Give that variable the value of “myInt”.
* Show the value of “myLong” to the console.
* Declare a variable with name “myFloat” of data type “float”.
* Give that variable the value of “myLong”.
* Show the value of “myFloat” to the console.
* Declare a variable with name “myDouble” of data type “double”.
* Give that variable the value of “myFloat”.
* Show the value of “myDouble” to the console.
* Declare a variable with name “myDecimal” of data type “decimal”.
* Give that variable the value of “myDouble”.
* Show the value of “myDecimal” to the console.
* Give the value 123456 to “myLong”.
* Give the value of “myLong” towards “myInt”. (Use casting)
* Show the value of “myLong” to the console.
* Give the value 123456789123456789 to “myLong”.
* Give the value of “myLong” towards “myInt”. (Use casting)
* Show the value of “myInt” to the console.
* Give the value 1234 to “myInt”.
* Give the value of “myInt” towards “myShort”. (Use casting)
* Show the value of “myShort” to the console.
* Give the value 1234567 to “myInt”.
* Give the value of “myInt” towards “myShort”. (Use casting)
* Show the value of “myShort” to the console.
* Give the value 123 to “myInt”.
* Give the value of “myInt” towards “myByte”. (Use casting)
* Show the value of “myByte” to the console.
* Give the value 1234 to “myInt”.
* Give the value of “myInt” towards “myByte”. (Use casting)
* Show the value of “myByte” to the console.