

# MQL4 COURSE

By Coders' guru  
[www.forex-tsd.com](http://www.forex-tsd.com)

-3-

## DATA TYPES

-----

Welcome to my third lesson in my MQL4 course.

I hope you enjoyed the "SYNTAX" lesson, which tried to give you the answers for:

- What format you can use to write MQL4 code?
- How to make the world better by commenting your code?
- What the Identifiers are, and what are the rules of choosing them?
- What are the MQL4 Reserved words?

If you didn't read the "SYNTAX" lesson please download it from here:

<http://forex-tsd.com/attachment.php?attachmentid=399>

And you can download the "Welcome" lesson from here:

<http://forex-tsd.com/attachment.php?attachmentid=372>

*Don't forget to login first.*

Now, let's **enjoy** the DATA TYPES.

### What's the Data type mean?

Any programming language has a set of names of the memory representation of the data. For example if the memory holds numbers between -2147483648 to 2147483647, the most of the programming languages will name this data as "**Integer**" data type.

### Variables?

Variables are the names that refer to sections of memory into which data can be stored.

To help you think of this as a picture, imagine that memory is a series of different size boxes. The box size is memory storage area required in bytes.

- In order to use a box to store data, the box must be given a name; this process is known as **declaration**.
- In the declaration process you use a word tell the computer what's the kind and size of the box you want to use, this word known as **keyword**.

- It helps if you give a box a meaningful name that relates to the type of information which make it easier to find the data, this name is the **variable constant**.
- Data is placed into a box by **assigning** the data to the box.
- When we set the value of the box you have created in the same line you declared the variable; this process is known as **initialization**.

When we create a variable we are telling the computer that we want him to assign a specified memory length (in bytes) to our variable, since storing a simple number, a letter or a large number is not going to occupy the same space in memory, so the computer will ask us what's the kind of data and how much the length of the data? That is the Data type for.

For example if we said this line of code to the computer:

```
int MyVariable=0;
```

That's mean we are asking the computer to set a block of 4 bytes length to our variable named "MyVariable".

In the previous example we have used:

**int** ← Keyword

**int** ← Integer data type.

**int** ← Declaration

**MyVariable** ← Variable's constant.

**=0** ← Initialization

We will know more about variables in a coming lesson.

In MQL4, these are the kinds of Data types:

- [Integer](#) (int)
- [Boolean](#) (bool)
- [Character](#) (char)
- [String](#) (string)
- [Floating-point number](#) (double)
- [Color](#) (color)
- [Datetime](#) (datetime)

## 1- Integer

An integer, is a number that can start with a + or a - sign and is made of digits. And its range value is between -2147483648 to 2147483647.

MQL4 presents the integer in [decimal or hexadecimal format](#).

For example the next numbers are Integers:

```
12, 3, 2134, 0, -230  
0x0A, 0x12, 0X12, 0x2f, 0xA3, 0Xa3, 0X7C7
```

We use the keyword **int** to create an integer variable.

For example:

```
int intInteger = 0;  
int intAnotherIntger = -100;  
int intHexIntger=0x12;
```

*Decimal and Hexadecimal:*

*Decimal notation is the writing of numbers in the base of 10, and uses digits (0, 1, 2, 3, 4, 5, 6, 7, 8 and 9) to represent numbers. These digits are frequently used with a decimal point which indicates the start of a fractional part, and with one of the sign symbols + (plus) or - (minus) to indicate sign.*

*Hexadecimal is a numeral system with a base of 16 usually written using the symbols 0–9 and A–F or a–f.*

*For example, the decimal numeral 79 can be written as 4F in hexadecimal.*

## 2- Boolean

Boolean variable is a data type which can hold only two values, true and false (or their numeric representation, 0 and 1). And it occupies 1 bit of the memory. In MQL4, false, FALSE, False, true, TRUE and True are equals.

*Boolean named like this in the honor of the great mathematician Boole George.*

We use the keyword **bool** to create a boolean variable.

For example:

```
bool I = true;  
bool bFlag = 1;  
bool bBool=FALSE;
```

### 3- Character

*MQL4 names this Data type “Literal”.*

A character is one of 256 defined alphabetic, numeric, and special key elements defined in the [ASCII](#) (American Standard Code for Information Interchange) set. Characters have integer values corresponding to location in the ASCII set. You write the character constant by using single quotes (') surrounding the character.

For example:

```
'a' , '$' , 'Z'
```

We use the keyword **int** to create a character variable.

For example:

```
int chrA = 'A';  
int chrB = '$';
```

Some characters called Special Characters can't present directly inside the single quotes because they have a reserved meanings in MQL4 language. Here we use something called **Escape Sequence** to present those special characters, And that by prefixing the character with the backslash character (\).

For example:

```
int chrA = '\\'; //slash character  
int chrB = '\n'; //new line
```

This is the list of Escape Sequence characters used in MQL4.

carriage return	\r
new line	\n
horizontal tab	\t
reverse slash	\\
single quote	'
double quote	"
hexadecimal ASCII-code	\xhh

## ASCII table

Char	Dec	Oct	Hex	Char	Dec	Oct	Hex	Char	Dec	Oct	Hex	Char	Dec	Oct	Hex
(nul)	0	0000	0x00	(sp)	32	0040	0x20	@	64	0100	0x40	`	96	0140	0x60
(soh)	1	0001	0x01	!	33	0041	0x21	A	65	0101	0x41	a	97	0141	0x61
(stx)	2	0002	0x02	"	34	0042	0x22	B	66	0102	0x42	b	98	0142	0x62
(etx)	3	0003	0x03	#	35	0043	0x23	C	67	0103	0x43	c	99	0143	0x63
(eot)	4	0004	0x04	\$	36	0044	0x24	D	68	0104	0x44	d	100	0144	0x64
(enq)	5	0005	0x05	%	37	0045	0x25	E	69	0105	0x45	e	101	0145	0x65
(ack)	6	0006	0x06	&	38	0046	0x26	F	70	0106	0x46	f	102	0146	0x66
(bel)	7	0007	0x07	'	39	0047	0x27	G	71	0107	0x47	g	103	0147	0x67
(bs)	8	0010	0x08	(	40	0050	0x28	H	72	0110	0x48	h	104	0150	0x68
(ht)	9	0011	0x09	)	41	0051	0x29	I	73	0111	0x49	i	105	0151	0x69
(nl)	10	0012	0x0a	*	42	0052	0x2a	J	74	0112	0x4a	j	106	0152	0x6a
(vt)	11	0013	0x0b	+	43	0053	0x2b	K	75	0113	0x4b	k	107	0153	0x6b
(np)	12	0014	0x0c	,	44	0054	0x2c	L	76	0114	0x4c	l	108	0154	0x6c
(cr)	13	0015	0x0d	-	45	0055	0x2d	M	77	0115	0x4d	m	109	0155	0x6d
(so)	14	0016	0x0e	.	46	0056	0x2e	N	78	0116	0x4e	n	110	0156	0x6e
(si)	15	0017	0x0f	/	47	0057	0x2f	O	79	0117	0x4f	o	111	0157	0x6f
(dle)	16	0020	0x10	0	48	0060	0x30	P	80	0120	0x50	p	112	0160	0x70
(dc1)	17	0021	0x11	1	49	0061	0x31	Q	81	0121	0x51	q	113	0161	0x71
(dc2)	18	0022	0x12	2	50	0062	0x32	R	82	0122	0x52	r	114	0162	0x72
(dc3)	19	0023	0x13	3	51	0063	0x33	S	83	0123	0x53	s	115	0163	0x73
(dc4)	20	0024	0x14	4	52	0064	0x34	T	84	0124	0x54	t	116	0164	0x74
(nak)	21	0025	0x15	5	53	0065	0x35	U	85	0125	0x55	u	117	0165	0x75
(syn)	22	0026	0x16	6	54	0066	0x36	V	86	0126	0x56	v	118	0166	0x76
(etb)	23	0027	0x17	7	55	0067	0x37	W	87	0127	0x57	w	119	0167	0x77
(can)	24	0030	0x18	8	56	0070	0x38	X	88	0130	0x58	x	120	0170	0x78
(em)	25	0031	0x19	9	57	0071	0x39	Y	89	0131	0x59	y	121	0171	0x79
(sub)	26	0032	0x1a	:	58	0072	0x3a	Z	90	0132	0x5a	z	122	0172	0x7a
(esc)	27	0033	0x1b	;	59	0073	0x3b	[	91	0133	0x5b	{	123	0173	0x7b
(fs)	28	0034	0x1c	<	60	0074	0x3c	\	92	0134	0x5c		124	0174	0x7c
(gs)	29	0035	0x1d	=	61	0075	0x3d	]	93	0135	0x5d	}	125	0175	0x7d
(rs)	30	0036	0x1e	>	62	0076	0x3e	^	94	0136	0x5e	~	126	0176	0x7e
(us)	31	0037	0x1f	?	63	0077	0x3f	_	95	0137	0x5f	(del)	127	0177	0x7f

### 4- String

The string data type is an array of characters enclosed in double quote (").

The array of characters is an array which holds one character after another, starting at index 0. After the last character of data, a NULL character is placed in the next array location. It does not matter if there are unused array locations after that.

A NULL character is a special character (represented by the ASCII code 0) used to mark the end of this type of string.

See figure 1 for a simple representation of the string constant "hello" in the characters array.

h	e	l	l	o	NULL				
0	1	2	3	4	5	6	7	8	9

**Figure 1 – Characters array**

MQL4 limits the size of the string variable to 255 characters and any character above 255 characters will generate this error: *(too long string (255 characters maximum))*.

You can use any special character -mentioned above- in your string constant by prefixing it with the backslash (\).

We use the keyword **string** to create a string variable.

For example:

```
string str1 = "Hello world1, with you coders guru";
string str2 = "Copyright © 2005, \"Forex-tds forum\"."; //Notice the use of (") character.
string str3 = "1234567890";
```

## 5- Floating-point number (double)

Floating point number is the Real Number (that is, a number that can contain a fractional part beside the integer part separated with (.) dot).Ex: 3.0,-115.5, 15 and 0.0001.

And its range value is between 2.2e-308 to 1.8e308.

We use the keyword **double** to create a floating-point variable.

For example:

```
double dblNumber1 = 10000000000000000;
double dblNumber3 = 1/4;
double dblNumber3 = 5.75;
```

## 6- Color

Color data type is a special MQL4 data type, which holds a color appears on the MetaTrader chart when you create your own Expert Advisor or Custom Indicator and the user can change it from the property tab of your Expert Advisor or Custom Indicator.

You can set the Color variable constant in three ways:

1- **By the color name:** For the well know colors (called Web Colors Set) you can assign the name of the color to the color variable, see the list of the Web Colors Set.

2- **By Character representation** (MQL4 named it this name): In this method you use the keyword (C) followed by two signal quotations (!). Between the two signal quotations you set the value of the red, green and blue (know as RGB value of the color). These values have to be between: 0 to 255. And you can write these values in decimal or hexadecimal format.

3- **By the integer value:** Every color in the Web Colors Set has its integer value which you can write it in decimal or hexadecimal format. And you can assign the Integer value of the color to the color variable. The hexadecimal color format looks like this: 0xBBGGRR where BB is the blue value, GG is green value and RR is the red value.

For example:

```
// symbol constants
C'128,128,128' // gray
C'0x00,0x00,0xFF' // blue
// named color
Red
Yellow
Black
// integer-valued representation
0xFFFFFFFF // white
16777215 // white
0x008000 // green
32768 // green
```

We use the keyword **color** to create a color variable.

For example:

```
color clr1= Red;
color clr1= C'128,128,128' ;
```

```
color clr1=32768;
```

## Web Colors Set

Black	DarkGreen	DarkSlateGray	Olive	Green	Teal	Navy	Purple
Maroon	Indigo	MidnightBlue	DarkBlue	DarkOliveGreen	SaddleBrown	ForestGreen	OliveDrab
SeaGreen	DarkGoldenrod	DarkSlateBlue	Sienna	MediumBlue	Brown	DarkTurquoise	DimGray
LightSeaGreen	DarkViolet	FireBrick	MediumVioletRed	MediumSeaGreen	Chocolate	Crimson	SteelBlue
Goldenrod	MediumSpringGreen	LawnGreen	CadetBlue	DarkOrchid	YellowGreen	LimeGreen	OrangeRed
DarkOrange	Orange	Gold	Yellow	Chartreuse	Lime	SpringGreen	Aqua
DeepSkyBlue	Blue	Magenta	Red	Gray	SlateGray	Peru	BlueViolet
LightSlateGray	DeepPink	MediumTurquoise	DodgerBlue	Turquoise	RoyalBlue	SlateBlue	DarkKhaki
IndianRed	MediumOrchid	GreenYellow	MediumAquamarine	DarkSeaGreen	Tomato	RosyBrown	Orchid
MediumPurple	PaleVioletRed	Coral	CornflowerBlue	DarkGray	SandyBrown	MediumSlateBlue	Tan
DarkSalmon	BurlyWood	HotPink	Salmon	Violet	LightCoral	SkyBlue	LightSalmon
Plum	Khaki	LightGreen	Aquamarine	Silver	LightSkyBlue	LightSteelBlue	LightBlue
PaleGreen	Thistle	PowderBlue	PaleGoldenrod	PaleTurquoise	LightGrey	Wheat	NavajoWhite
Moccasin	LightPink	Gainsboro	PeachPuff	Pink	Bisque	LightGoldenRod	BlanchedAlmond
LemonChiffon	Beige	AntiqueWhite	PapayaWhip	Cornsilk	LightYellow	LightCyan	Linen
Lavender	MistyRose	OldLace	WhiteSmoke	Seashell	Ivory	Honeydew	AliceBlue
LavenderBlush	MintCream	Snow	White				

## 7- Datetime

Datetime data type is a special MQL4 data type, which holds a date and time data. You set the Datetime variable by using the keyword (D) followed by two signal quotations (!). Between the two signal quotations you write a character line consisting of 6 parts for value of year, month, date, hour, minutes, and seconds. Datetime constant can vary from Jan 1, 1970 to Dec 31, 2037.

For example:

```
D'2004.01.01 00:00' // New Year
D'1980.07.19 12:30:27'
D'19.07.1980 12:30:27'
D'19.07.1980 12' //equal to D'1980.07.19 12:00:00'
D'01.01.2004' //equal to D'01.01.2004 00:00:00'
```

We use the keyword **datetime** to create a datetime variable.



For example:

```
datetime dtMyBirthDay= D'1972.10.19 12:00:00';  
datetime dt1= D'2005.10.22 04:30:00';
```

I hope you enjoyed the lesson.

The next lesson will be about the “Operations & Expressions”.

I welcome very much the questions and the suggestions.

See you  
**Coders' Guru**  
22-10-2005