

# MQL4 COURSE

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

-3  
DATA TYPES

**Vítám** vás ve třetí lekci svého MQL4 kurzu.

Doufám, že předchozí lekce “ SYNTAX ” se vám líbila. V té jsme se pokoušeli zodpovědět:

- Jaký formát můžete použít k zápisu MQL4 kódu?
- Jak zlepšit pohled na svět pomocí komentářů vašeho kódu?
- Co jsou identifikátor a jaká jsou pravidla pro jejich volbu?
- Co jsou rezervovaná slova u MQL4? Pokud jste si lekci “ SYNTAX ” nepřečetli, stáhněte si ji, prosím, odtud:

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

Uvítací lekci můžete stáhnout odsud:

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

Nezapomeňte se nejprve zalogovat.

Nyní se podívejme na datové typy – DATA TYPES. Co znamená pojem datový typ?

Všechny programovací jazyky mají sadu pojmenování pro paměťovou reprezentaci dat. Např., pokud paměť obsahuje čísla v -2 147 483 648 až 2 147 483 647, většina programovacích jazyků pojmenuje tento datový typ jako “Integer”.

Proměnné?

Proměnné jsou jména, která odkazují na sekce v paměti, do kterých mohou být data uložena.

Abych vám pomohl, zkuste si představit, že paměť je série různě velkých přihrádek. Rozměry přihrádek odpovídají množství bajtů požadovaných pro uložení.

- Abychom byli schopni vidět příhrádku pro uložení dat , musí jí být přiřazeno jméno. Tento proces je znám jako deklarace.
- V procesu deklarace použijete slovo, které sdělí počítači, jaké je jméno a velikost příhrádky, kterou si přejete použít. Toto slovo se nazývá klíčové slovo - keyword .
- Pomůže, pokud dáte příhrádce smysluplné jméno, má souvislost s typem informace, což pomůže ke snadnějšímu vyhledání dat; jméno je proměnnou konstanty.
- Data jsou umístěna do příhrádky přiřazením.
- Pokud nastavíme hodnotu příhrádky, vytvořili jsme stejný řádek, jakým deklarujeme proměnnou; tento proces se nazývá inicializace.

Když vytvoříme proměnnou, sdělujeme tím počítači, že po něm žádáme vytvoření místa v paměti o specifikované délce (v bajtech) pro naši proměnnou. Jelikož uložení jednoduchého čísla písmena nebo velké množství hodnot nezabírá stejné množství paměti, počítač se nás táže na typ a délku těchto dat. K tomu slouží funkce Datatype.

Např. pokud přeneseme tento řádek do počítače:

```
int MyVariable=0;
```

Znamená to, že žádáme počítač o sestavení bloku o délce 4 bajty pro naši proměnnou pojmenovanou "My Variable".

V předchozím příkladu jsme použili:

int ← Klíčové slovo

int ← Integer - datový typ.

int ← Deklarace

My Variable ← Konstanta proměnné.

=0 ← Inicializace

O proměnných se dozvíme více v další lekci.

Toto jsou datové typy v MQL 4:

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

## 1- Integer (celé číslo)

Integer je číslo, které začíná znakem + nebo – a je sestaveno z jednotlivých číslic. Jeho rozsah se pohybuje v rozmezí -2147483648 to 2147483647.

MQL 4 prezentuje integer v [decimálním nebo hexadecimálním formátu](#).

Zde je uveden příklad celých čísel:

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

K vytvoření proměnné celého čísla použijeme klíčové slovo. Např.:

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

Decimální a hexadecimální:

Decimální zápis čísel na základě desítkové soustavy, použitím číslic (0,1,2,3,4,5,6,7,8 a 9) Tyto číslice jsou v často využívány v decimálním systému, který indikuje začátek frakční části, spolu se symboly + (plus) nebo – (minus) pro indikaci znaku.

Hexadecimální je číselný systém na základě 16 běžně zapisovaný použitím symbolů 0–9 a A–F nebo a–f.

např., decimální číslo 79 může být v hexadecimální soustavě zapsáno jako 4F.

## 2-Boolean

Boolean proměnné jsou datové typy, které mohou obsahovat pouze dvě hodnoty, true nebo false (nebo jejich numerické zástupce v podobě 0 a 1). V paměti zabírají jeden bit. V MQL4 false, FALSE, False, true, TRUE a True znamená totéž.

Pojmenování Boolean vzniklo na počest velkého matematika Boole George.

Pro vytvoření proměnné typu boolean použijeme klíčové slovo bool.

Příklad:

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

### 3-Character - znak

MQL 4 pojmenovává tento datový typ jako "Literal".

Znak je jedním z 256 definovaných alfabetských, numerických a speciálních klíčových elementů definovaných v [ASCII](#)(Americký standardní kód pro výměnu informací) soustavě.

Znaky mají hodnotu celého čísla, odpovídající lokalizaci v systému ASCII.

Konstantu znaku zapíšete za pomoci apostrofů (') ohraničující znak.

Příklad:

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

Pro vytvoření proměnné znaku použijeme klíčové slovo int. Např.:

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

Některé znaky, zvané „zvláštní“, nemohou být prezentovány přímo uvnitř samostatné kvóty, protože mají v jazyku MQL4 vyhrazený význam. K prezentování znaků používáme sekvenci zvanou Escape Sequence označením znaku pomocí obráceného lomítka (\).

Příklad:

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

Toto je seznam znaků sekvence Escape Sequence používané v MQL 4 .

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

## ASCII tabulka

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

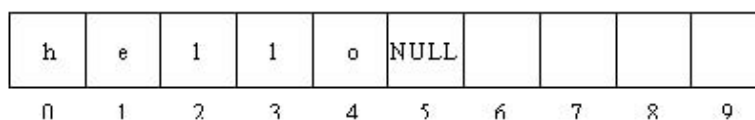
## 4-String - řetězec

Datový typ string je v poli znaků ohraničen uvozovkami (").

Pole znaků se nazývá pole, které obsahuje jeden znak za druhým s počátečním indexem 0. Poté, co je umístěn poslední datový znak, znak NULL je vložen do dalšího prvku. Nezáleží na tom, zda jsou se po něm vyskytují nevyužité prvky.

Znak NULL je zvláštní znak (reprezentovaný v kódu ASCII code 0), používaný k označení konce tohoto typu řetězce.

Viz. obr. 1 – jednoduchá reprezentace řetězce konstanty "hello" ve znacích pole .



Obr. 1 – znaky pole

MQL4 limituje rozsah řetězce proměnné na 255 znaků, a pokud dojde k přesažení tohoto limitu, bude generována chyba: (too long string (255 characters maximum) ). Jakýkoliv zvláštní znak, který bude vyobrazen ve vašem řetězci, bude označen obráceným lomítkem (\).

K vytvoření proměnné řetězce se používá klíčové slovo string.

Příklad:

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

## 5- Číslo s plovoucím desetinným bodem (double)

Desetinné číslo je reálné číslo (tj., číslo obsahující frakční část mimo celého čísla, oddělenou tečkou (.) ).Např.: 3.0 , -115.5, 15 a 0.0001.

A hodnota jeho rozsahu se pohybuje v rozmezí 2.2e-308 až 1.8 e308 .

K vytvoření proměnné plovoucího bodu používáme klíčové slovo double.

Příklad:

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

## 6-Color - barva

Color patří v MQL ke zvláštním datovým typům, které jsou nosiči barevného vzhledu grafů MetaTraderu při vytváření Expert advisoru a v uživatelském indikátoru je uživatel může měnit z tabulky vlastností.

Proměnné barev můžete nastavit třemi způsoby:

1- Podle jména barvy: u známých barev (pojmenovaných Web Colors Set) můžete přiřadit jméno proměnné barvy ze seznamu Web Colors Set.

2- Pomocí znakové reprezentace (Podle pojmenování MQL4): U této metody použijete klíčové slovo (C ) následované dvěma signálními apostrofy ('). Mezi oba signální apostrofy zadáte hodnotu červené, zelené a modré barvy (RGB hodnota barvy). Tyto hodnoty se musí pohybovat v rozmezí: 0 až 255. Hodnoty můžete zapsat v decimálním nebo hexadecimálním formátu.

3- Pomocí hodnoty celého čísla: Každá barva v sadě Web Colors Set má v sobě obsaženou hodnotu celého čísla zapsanou v decimálním nebo hexadecimálním formátu. Hodnotu celého čísla barvy můžete tak přiřadit do proměnné barvy. Hexadecimální formát vypadá takto: 0xBBGGRR, přičemž BB je hodnota pro modrou, GG pro zelenou a RR pro červenou barvu.

Příklad:

```
// 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
```

K vytvoření proměnné barvy použijeme klíčové slovo color.

Příklad:

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

### 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
Gold	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	ComflowerBlue	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 (datum, čas)

Datové typy data a času určenými pro MQL4 uchovávají údaje o datu a čase. Proměnné funkce datetime nastavíte prostřednictvím klíčového slova (D), následovaného dvěma signálními apostrofy ('). Mezi tyto dva signální apostrofy zapíšete řádek obsahující 6 částí pro hodnoty roku, měsíce, dne hodiny, minuty a sekundy. Konstanty funkce datetime se mohou pohybovat v rozmezí 1. ledna 1970 až 31. prosince 2037.

Příklad:

```
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'
```

Příklad:



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

Doufám, že se vám lekce líbila.

Další lekce se bude týkat “ Operací & Výrazů” .

Uvítám jakékoliv dotazy a připomínky.

S pozdravem

Coders'G uru

22-10 -20 0 5