

ایجاد سیستم‌های معاملاتی خودکار

در زبان MQL 5

برنامه نویسی اکسپرت برای متاتریدر ۵

تالیف و ترجمه:

فرهاد سلطانی

آراد کتاب

سرشناسه	: سلطانی، فرهاد، ۱۳۵۹.
عنوان و پدیدآور	: ایجاد سیستم‌های معاملاتی خودکار در زبان MQL 5: برنامه نویسی اکسپرت برای متاتریدر ۵ / تالیف فرهاد سلطانی.
مشخصات نشر	: تهران، آراد کتاب،
مشخصات ظاهری	: ۳۲۰ ص، مصور،
شابک	: ۹۷۸-۶۰۰-۱۸۶-۳۳۰-۱
عنوان دیگر	: برنامه نویسی اکسپرت برای متاتریدر ۵
موضوع	: ارز - داده‌پردازی
موضوع	: زبان‌های برنامه نویسی کامپیوتر
رده بندی کنگره	: ۱۳۹۶ الف۸س / HG ۳۸۵۱
رده بندی دیویی	: ۳۳۲ / ۴۵۰۲۸۵
شماره کتابخانه ملی	: ۴۸۵۵۶۹۴

ایجاد سیستم‌های معاملاتی خودکار در زبان MQL 5

برنامه نویسی اکسپرت برای متاتریدر ۵

<input checked="" type="checkbox"/> ترجمه و تالیف : فرهاد سلطانی	<input checked="" type="checkbox"/> ناشر: آراد کتاب
<input checked="" type="checkbox"/> نوبت چاپ: اول ۱۳۹۶	<input checked="" type="checkbox"/> تیراژ: ۱۰۰۰ جلد
<input checked="" type="checkbox"/> چاپ و صحافی : عطا - امید	<input checked="" type="checkbox"/> قیمت: ۳۲۵۰۰۰ ریال
شابک: ۹۷۸-۶۰۰-۱۸۶-۳۳۰-۱	

حق چاپ برای ناشر محفوظ است. کلیه حقوق و حق چاپ متن، طرح روی جلد و عنوان کتاب با نگرش به قانون حمایت حقوق مؤلفان، مصنفان و هنرمندان مصوب ۱۳۴۸ برای انتشارات آراد کتاب محفوظ است و متخلفین تحت پیگرد قانونی قرار می‌گیرند.

مرکز پخش و فروش:

انتشارات آراد کتاب تلفن: ۶۶۹۷۵۲۸۵ - ۶۶۴۸۲۲۲۶ - ۰۹۱۲۳۰۶۲۴۵۸

خرید آنلاین از سایت: www.aradbook.com

فهرست مطالب

صفحه	عنوان
9	فصل 1: اصول MQL5
19	فصل 2: انواع متغیرها و داده‌ها
43	فصل 3: عملیات ریاضی
49	فصل 4: عملگرهای شرطی و حلقه‌ای
59	فصل 5: توابع
67	فصل 6: برنامه‌نویسی با رویکرد شیء‌گرا
77	فصل 7: ساختار یک برنامه MQL5
85	فصل 8: اصول اکسپرت‌ها
95	فصل 9: سفارش‌گذاری
111	فصل 10: ایجاد یک کلاس سفارش‌گذاری
129	فصل 11: حد ضرر و حد سود
143	فصل 12: مدیریت، اصلاح و بستن پوزیشن‌ها
151	فصل 13: سفارش‌های انتظاردار
175	فصل 14: حد ضرر دنبال‌کننده
187	فصل 15: مدیریت سرمایه و اندازه معامله
193	فصل 16: داده‌های مربوط به کندل و قیمت
205	فصل 17: استفاده از اندیکاتورها در اکسپرت
227	فصل 18: کار کردن با زمان و تاریخ
249	فصل 19: کنار هم گذاشتن تمامی مفاهیم قبلی
267	فصل 20: نکات و ترفندها
287	فصل 21: اندیکاتورها، اسکریپت‌ها و کتابخانه‌ها
303	فصل 22: اشکال‌زدایی و تست کردن

درباره این کتاب

هدف از نگارش این کتاب دادن دانش و ابزار لازم به خوانندگان آن برای خلق اکسپرت‌هایی در زبان برنامه نویسی MQL5 می‌باشد. ما همچنین مروری بر نحوه ساخت اندیکاتورها، اسکریپت‌ها و کتابخانه‌ها می‌کنیم. در طول مطالب این کتاب ساختاری برای توسعه سریع اکسپرت‌ها ایجاد خواهیم کرد.

با استفاده از قابلیت‌های شی گزایی زبان MQL5، کلاس‌ها و توابعی را برای عملیات متداول معاملاتی از قبیل باز کردن، بستن و مدیریت سفارش‌ها، درست می‌کنیم. ما همچنین کلاس‌ها و توابعی را برای استفاده در ویژگی‌هایی مانند مدیریت سرمایه، حدضرر دنبال کننده¹، تایمر معاملاتی، اندیکاتورها و ... ایجاد می‌کنیم. MQL5 کتابخانه مخصوص خودش را دارد که دسترسی سریع به توابع معاملاتی متداول را می‌دهد. کتابخانه استاندارد زمانی که اکسپرت‌ها با کمک جادوگر² MQL5 تولید می‌شوند، استفاده می‌شود. ما از کتابخانه استاندارد MQL5 در این کتاب استفاده نخواهیم کرد، اگرچه خواننده کتاب می‌تواند از کلاس‌های کتابخانه استاندارد MQL5 در پروژه‌های خود استفاده کند.

زبان برنامه نویسی MQL5 بسیار قدرتمند و وسیع می‌باشد خیلی بیشتر از آنچه که MQL4 بوده است. کارهای زیادی در MQL5 وجود دارد که می‌توان انجام داد اما از حوصله این کتاب خارج است و این کتاب قصد ندارد که به عنوان مرجع زبان MQL5 تلقی شود. زمانی که مفاهیم پایه‌ای برنامه نویسی MQL5 را فراگیرید، سایت رسمی MQL5 به نام www.mql5.com می‌تواند به عنوان مرجع برنامه نویسی MQL5 برای راهنمایی شما در تمام زمینه‌ها به کار می‌رود.

این کتاب فرض می‌کند که خواننده با نرم افزار متاتریدر 5 و معاملات ارز به طور معمول آشنایی دارد. خواننده کتاب باید با نحوه استفاده از نرم افزار MQL5 آشنا باشد و یک فهم پایه‌ای از تحلیل تکنیکال و معاملات سیستمی داشته باشد. برای این کتاب هیچ تجربه قبلی برنامه نویسی ضروری نیست اگرچه خوانندگان از مزایای داشتن مهارت قبلی در سایر زبان‌های برنامه نویسی می‌توانند بهره ببرند. تلاش‌های زیادی در نوشتن این کتاب شده که اطلاعات آن درست باشد اما ممکن است که چند خطا در این بین پیدا شود. MQL5 هنوز در فاز پیشرفت و تغییر قرار دارد بنابراین ممکن است بعضی ویژگی‌ها بعد از انتشار این کتاب تغییر یابند یا ویژگی‌های جدیدی اضافه شود.

اگر شما در حین خواندن کتاب به اشتباهی پی بردید، لطفاً به ایمیل من contact@expertadvisorbook.com اطلاع دهید. غلط نامه این کتاب در وبسایت موجود است و هر گونه اشتباهی در چاپ‌های بعدی تصحیح خواهد شد.

1- Trailing
2- Wizard

کد منبع

کد منبع این کتاب رایگان در وبسایت رسمی آن <http://www.expertadvisorbook.com> قابل دانلود می‌باشد. این کد منبع شامل فایل‌های کتابخانه‌ای¹ اکسپرت و یک لیست قابل چاپ می‌باشد و چندین برنامه که در طول این کتاب به وجود می‌آید. ما اغلب به این فایل‌ها ارجاع می‌دهیم، بنابراین پیشنهاد می‌شود که آنها را دانلود کنید و این فایل‌ها را در فولدر برنامه متا تریدر 5 خود قرار دهید. شما می‌توانید این کد منبع را در پروژه‌های شخصی خود استفاده کنید. حتی می‌توانید آنها را ویرایش کرده و برای خود استفاده کنید. اما از آن‌ها نمی‌توانید در پروژه‌های تجاری استفاده کنید و اگر آنها را به اشتراک بگذارید، باید نویسنده آن ذکر شود.

مقدمه

در بدو ورود به بازارهای مالی، شایسته‌تر است، بازار هدف خود را انتخاب کنیم که می‌تواند بازار سهام، آتی سکه، بازارهای جهانی باشد. سپس سازوکار بازار هدف خود را مانند نحوه افتتاح حساب در کارگزاری، شیوه معامله کردن و ساعات معامله کردن و ... فارغ از تحلیل، بیاموزیم. پس از گذر این مرحله، با روش‌های تحلیل: فاندامنرال، تکنیکال، تکنو فاندامنرال¹ در حد آشنایی بیاموزیم و مشخص نماییم کدام‌یک از روش‌های تحلیل با ساختارهای ذهنی و رفتاری ما سازگاری دارد. مثلاً بازار هدف خود را، به خاطر کم ریسک بودن، سهام و روش تحلیل خود را، تحلیل تکنیکال، انتخاب می‌نماییم.

در مرحله بعد تمامی روش‌های موردبحث در روش تحلیلی انتخابی خود را (مثلاً روش تحلیل تکنیکال)، مطالعه می‌کنیم تا روش متناسب با ساختار ذهنی خود را انتخاب نماییم. مثلاً در روش تحلیل تکنیکال روش‌های تحلیلی زیر مطرح است: اندیکاتوری، روانشناسی کندلها، سایکل‌های زمانی، امواج الیوت، روندی، فیبوناچی، پرایس اکشن، الگوها، ایچی موکو، فراکتالی، بیل ویلیامز، چنگال اندروز، حجمی، الگوهای شمعی ژاپنی، الگوهای هارمونیک، روش‌های آماری، روش‌های هوش مصنوعی، گن و ... تمام تمرکز خود را در یکی از روش‌های فوق گذاشته تا به صورت یک متخصص در این حوزه گردیم.

در مرحله بعد به مطالعه و بررسی استراتژی‌های موجود در کتاب‌ها، سایت‌ها، مقالات و ... جهت گرفتن ایده اولیه برای طراحی استراتژی شخصی خود پرداخته و نقاط ضعف و قوت هر استراتژی استخراج نموده و همه موارد استخراجی را در استراتژی خودمان لحاظ می‌کنیم. همه قواعد استراتژی خود را به صورت مکتوب ثبت می‌نماییم.

پس از تدوین قواعد، به مرحله اجرا می‌رسیم. یکی از پلتفرم‌های معاملاتی معروف که قابلیت برنامه‌نویسی دارد و در همه بازارهای هدف اشاره شده، قابل اجراست، نرم‌افزار Meta Trader است. این پلتفرم معاملاتی در دو نسخه 4 و 5 معروف است. نسخه 4 آن بانام تجاری پاریس رسا که تا این لحظه فقط قابلیت نمایش داده‌های سهام، بورس، دلار و ... را دارد و نسخه 5 آن، مفید تریدر نام دارد که علاوه بر قابلیت‌های قبل، قابلیت معامله کردن² را دارد. به زبان برنامه‌نویسی متاتریدر MQL گویند. در این مرحله استراتژی مدون شده خود را، تبدیل به نرم‌افزار اتوماتیک معاملاتی³ نموده تا به جای ما، عملیات خرید و فروش را انجام دهد. نتیجه اینکه اینکه خطای انسانی در زمان معامله کردن به شدت کاهش پیدا می‌کند. موضوع کتاب ما در مورد آموزش برنامه‌نویسی MQL5 با فرض، آشنایی اولیه با MQL است.

1- ترکیبی از روش تحلیل فاندامنرال و تکنیکال است گویند

مرحله بعد چرخه بهبود مستمر سیستم معاملاتی است؛ یعنی اینکه چطور بتوانیم میزان ریسک سیستم خود را کم و میزان سود را افزایش دهیم. در این مرحله، مطالعه سایر روش‌های تحلیلی و ارتباط آنها با یکدیگر که بتواند سیستم طراحی شده ما را بهبود بخشد، بررسی روش‌های آماری، روش‌های خطایابی و برخورد با خطا و... اهمیت پیدا می‌کند. در مرحله بعد دقیق کردن استراتژی معاملاتی به روش‌های هوش مصنوعی مانند الگوریتم‌های ژنتیک، سایر روش‌های شبکه‌های عصبی و... اهمیت پیدا می‌کند. و در کنار همه مراحل فوق مدیریت سرمایه، مدیریت ریسک و روش‌های محاسبه ریسک و ضریب سودآوری و... جزو الزامات هر سیستمی می‌باشد و مطالعه و تحقیق در این حوزه می‌تواند به شما آرامشی همیشگی در این بازار پرتلاطم مالی دهد. جهت بهره‌وری از اثر، سوره کدهای این اثر، در سایت جهان بورس¹ منوی MQL زیر منوی کتاب‌ها موجود است.

در این میان، از استادان، معامله‌گران و دانشجویان علاقه‌مند تقاضا می‌شود تا با همکاری، راهنمایی و پیشنهادهای سازنده خود، به مجموعه آموزشی آسا سرمایه² و یا سایت جهان بورس³، ما را در انتشار آثاری دیگر، یاری دهند.

در عبور از مسیرهای فوق همواره عزیزانی بودند که در تکامل بنده در مسیر عبور از بازارهای مالی نقش برجسته‌ای داشته‌اند؛ جا دارد از همه این بزرگواران، در اینجا تشکر کنم. از مهندس جواد زرین اقبال به خاطر دلگرمی، انرژی دادن و انتقال تجربیات، از مهندس یاسر پاکزاد به خاطر دلگرمی، صبوری و دقیق بودن ایشان در مباحث علمی، جناب شهاب موسوی مدیریت مجموعه آموزشی آسا سرمایه به خاطر پیشنهاد ترجمه و انعطاف‌پذیر بودن ایشان، مهندس سلیمان زاده به خاطر پشتکار ایشان در ترجمه کتاب‌های دیگر و آشنا نمودن با مباحث جدید، مهندس مجتبی شیعه زاده به خاطر تحقیقات آماری سریع و دقیق و پرانرژی بودن ایشان، مهندس خوش‌پی به خاطر تحقیقات ایشان در مباحث مدیریت سرمایه و از همسرم به خاطر صبوری و کمک به ویراستاری کتاب و محیا نمودن محیطی آرام جهت تألیف کتاب و از پدر و مادر عزیزم که دعای خیر ایشان همیشه پشت‌وپناه من بوده و از جناب رضایی مدیریت انتشارات آراد به خاطر همکاری صمیمانه ایشان، سپاسگزارم.

فرهاد سلطانی

شهریور 96

1 <http://jahanbourse.ir/>

2 <http://asasarmaye.com/>

3 <http://jahanbourse.ir/>

پیشگفتار

"معامله‌گری در دنیای امروز علاوه بر توان فنی، تحلیلی و مدیریتی، به رصد سریع و همه‌جانبه، بهینه‌سازی ابزارهای تحلیلی و واکنش سریع به تغییر و تحولات بازار نیازمند است." این جمله‌ای است که احتمالاً هر فرد فعال در بازارهای مالی داخلی و بین‌المللی پس از مدتی آزمون و خطا به اصول آن پی برده است.

امروزه مفهوم "سرعت" در همه ابعاد آنو بیش از هر زمان دیگری در دنیای معاملات مالی نقش تعیین‌کننده و سرنوشت‌ساز در گزارش‌های معاملاتی افراد دارد. به لطف سیستم‌های کامپیوتری و ابزارهای وسیعی که در این زمینه وجود دارد، می‌توان گفت هر معامله‌گر باهوش و البته مستعد قادر است با یادگیری و افزایش مهارت عملیاتی در حوزه یادشده، ابزارهای وسیع رصد، بهینه‌سازی، هشداردهنده و... را برای بهبود مستمر نتایج خود به خدمت گیرد.

با توجه به اهمیت موضوع یادشده در بهبود عملکرد فعالان بازار از سال 94 و با همکاری جناب آقای مهندس فرهاد سلطانی که سابقه طولانی و عملیاتی در زمینه برنامه‌نویسی در بازارهای مالی دارند، برگزاری دوره‌های آموزشی با محوریت زبان برنامه‌نویسی MQL در مجموعه آسارمیه آغاز شد که به لطف پروردگار با استقبال چشمگیر معامله‌گران بازارهای مالی داخلی و بین‌المللی همراه گردید. پس از گذشت دو سال و به پیشنهاد ایشان، کار ترجمه و تألیف کتاب حاضر، به‌عنوان مرجعی معتمد و جامع در زمینه برنامه‌نویسی MQL به زبان فارسی آغاز گردید و اکنون خوشحالم که جای خالی کتاب مرجع در زمینه MQL برای علاقه‌مندان و مشتاقان این حوزه، با این اثر جبران گردیده است.

در انتها جا دارد از زحمات جناب آقای سلطانی، عضو محترم هیئت‌علمی مجموعه آسارمیه بابت زحماتی که در تألیف و ترجمه این اثر متقبل شدند، جناب آقای رضایی، مدیریت محترم نشر آراد که با پیگیری و صبوری دلسوزانه خود نقش بسزایی در تولید این کتاب داشتند و در نهایت سرکار خانم آرامی، قائم‌مقام و مدیر محترم اجرایی مجموعه آسارمیه که زحمت هماهنگی و پیگیری مستمر این پروژه را به دوش کشیدند صمیمانه تقدیر و تشکر به‌عمل آورم. امیدوارم کتاب حاضر، آغازی باشد برای تولید و نشر اطلاعات به‌روز و کاربردی در زمینه معاملات و آنالیز کامپیوتری در میهن عزیزمان و همین‌طور از شما خواننده محترم تقاضا می‌کنم انتقادات و پیشنهادهای خود را در مورد این اثر با نشر آراد و همچنین مجموعه آسارمیه در میان بگذارید.

شهاب موسوی

مؤسس و مدیرعامل مجموعه آسارمیه

تابستان 96

فصل 1

اصول MQL5



برنامه‌های MQL5

شما می‌توانید سه نوع برنامه در زبان MQL5 تولید کنید: اکسپرت¹، اندیکاتور² و اسکریپت³. یک اکسپرت برنامه‌ای اتوماتیک است که می‌تواند سفارش‌های یک معامله را باز، ویرایش کند و ببندد. شما تنها می‌توانید در یک زمان یک اکسپرت را روی نمودار⁴ داشته باشید. بیشتر مطالب این کتاب چگونگی ایجاد یک اکسپرت را در زبان MQL5 پوشش می‌دهد. یک اندیکاتور داده‌های تحلیل تکنیکال را روی یک نمودار یا یک پنجره جداگانه با استفاده از خطوط، هیستوگرام‌ها، فلش‌ها، میله‌ها یا اشیاء نمودار نمایش می‌دهد. شما می‌توانید چندین اندیکاتور را به نمودار اضافه کنید. فصل بیست و یکم نحوه‌ی ایجاد اندیکاتورها را در MQL5 نمایش می‌دهد. یک اسکریپت برنامه‌ای تخصصی است که یک وظیفه خاص را انجام می‌دهد. زمانیکه اسکریپت به داخل نمودار کشیده شود، تنها یک بار اجرا می‌گردد. شما در یک زمان تنها یک اسکریپت را می‌توانید به نمودار اضافه کنید. ما در فصل بیست و یکم چگونگی ایجاد آنرا نشان می‌دهیم.

فرمت فایل‌ها در MQL5

یک فایل (mq5) حاوی کد منبع⁵ یک برنامه MQL5 است، از قبیل اکسپرت، اندیکاتور، اسکریپت. این فایل، زمانیکه ما در ویرایشگر متا برنامه MQL5 می‌نویسیم، ایجاد می‌شود. این فایل در ویرایشگر متا یا هر ویرایشگر متن می‌تواند باز ویرایش گردد.

1- Expert
2- Indicator
3- Script
4- Chart
5- Source Code

یک فایل EX5(.ex5)، برنامه کامپایل شده قابل اجرا می‌باشد، زمانیکه یک فایل MQL5 در ویرایشگر متا کامپایل می‌شود، یک فایل EX5 با همان اسم ساخته می‌شود. این فایلی است که زمانیکه شما برنامه MQL5 را به نمودار می‌چسبانید، اجرا می‌شود. فایل‌های EX5، باینری می‌باشند و ویرایش نمی‌شوند.

یک فایل MQH(.mqh) یک فایل اینکلود¹ است که حاوی کد منبع برای استفاده هر برنامه MQL5 می‌باشد. مانند فایل‌های MQL5، فایل MQH نیز می‌تواند در ویرایشگر متا باز و ویرایش شود.

انواع دیگر فایل‌ها

یک فایل اینکلود (.mqh) یک فایل کدمنبع است که حاوی کلاسها، توابع و متغیرهایی برای استفاده در برنامه MQL می‌باشد. فایل‌ها اینکلود حاوی کدهایی سودمند برای استفاده مجدد است و به دفعات زیادی می‌تواند مورد استفاده قرار گیرد. زمانیکه یک برنامه کامپایل می‌شود، محتوای هر فایل اینکلودی که در برنامه استفاده شده، مشمول برنامه کامپایل شده قرار می‌گیرد. ما در طول مطالب این کتاب تعداد زیادی فایل اینکلودی درست می‌کنیم.

یک کتابخانه، فایلی اجرایی است که حاوی توابعی برای استفاده مجدد می‌باشد، مشابه با یک فایل اینکلود. کتابخانه‌ها به فرمت EX5 با فایل DLL ویندوز می‌باشد. کتابخانه در حافظه کامپیوتر بصورت جداگانه در کنار برنامه MQL اجرا می‌شود.

کتابخانه‌ها در جاهایی سودمند هستند که بخواهیم توابع خود را برای دیگران در دسترس قرار دهیم، بدون آنکه منبع را در دسترس بگذاریم. ما در فصل بیست و یکم به تفصیل در مورد کتابخانه‌ها بحث می‌کنیم.

یک فایل تنظیمات اکسپرت یا فایل از پیش تعیین شده با فرمت (.set). حاوی پارامترهای معامله برای اکسپرت می‌باشد. فایل‌های تنظیمات در باکس ورودی اطلاعات اکسپرت در زیر تب ورودیها² ذخیره یا بارگذاری می‌شود. دکمه بارگذاری³، پارامترها را از یک فایل (.set) بارگذاری کرده و دکمه ذخیره⁴، پارامترهای اخیر را در یک فایل .set ذخیره می‌کند. شما همچنین می‌توانید پارامترها را در استراتژی تستر⁵ زیر تب ورودیها ذخیره یا بارگذاری کنید که با استفاده از منوی باز شده کلیک راست ماوس می‌باشد.

1- Include
2- Inputs
3- Load
4- Save
5- strategy tester

محل فایل های MQL5

پوشه متاتریدر 5 بصورت پیش فرض در آدرس C:\program files\ (اگر شما نرم افزار متاتریدر 5 را در پوشه یا درایو دیگری نصب کرده اید، محل نصب شما متفاوت خواهد بود). تمام برنامه های MQL5 در پوشه MQL5\ در داخل پوشه محل نصب متاتریدر 5 شما می باشد. با فرض محل نصب پیش فرض¹ برای متاتریدر 5، محل پوشه MQL5 در آدرس C:\program files\metaTrader5\Trade5\MQL5 می باشد. (بروکر شما می تواند نام متفاوتی را برای پوشه metaTrader5\ در نظر بگیرد.)

از آنجا که تمام برنامه های MQL5 از همان فرمت استفاده می کنند، انواع برنامه ها در زیر پوشه هایی در پوشه MQL5\ از متاتریدر 5 قرار می گیرند. محتوای زیر پوشه های پوشه MQL5 عبارتند از:

Experts\ این پوشه حاوی فایل های MQ5 و EX5 اکسپرتها می باشد.

Indicators\ این پوشه حاوی فایل های MQ5 و EX5 برای اندیکاتورها می باشد.

Scripts\ این پوشه حاوی فایل های MQ5 و EX5 اسکریپت ها می باشد.

Include\ این پوشه حاوی فایل های اینکلود MQH می باشد.

Libraries\ این پوشه حاوی فایل های MQ5, EX5 و DLL برای کتابخانه می باشد.

Images\ اگر برنامه شما از تصاویر استفاده می کند، آن ها باید در پوشه Images\ با فرمت bmp ذخیره شوند.

Files\ هر فایل خارجی که در برنامه تان استفاده می کنید، غیر از فایل های اینکلود، کتابخانه ها، تصاویر یا سایر برنامه های MQL، باید در این پوشه ذخیره شوند.

Presets\ این پوشه به صورت پیش فرض برای فایل های set. که از دیالوگ باکس اکسپرت بارگذاری یا ذخیره می شوند، یا از تب ورودیها در استراتژی تستر، اختصاص دارد.

Logs\ پوشه پیش فرض برای فایل های ورودی² اکسپرتها می باشد. شما می توانید این لاگها را در تب اکسپرتها در جعبه ابزار رابط اصلی متاتریدر پیدا کنید.

هر ارجاعی به پوشه های بالا در این کتاب فرض می کند که آنها در پوشه MQL5\ و در پوشه محل نصب نرم افزار متاتریدر می باشد. بنابراین ارجاع به پوشه Experts\ ما را به آدرس MQL5\Experts\ از پوشه نصب متاتریدر 5 می برد.

ویرایشگر متا

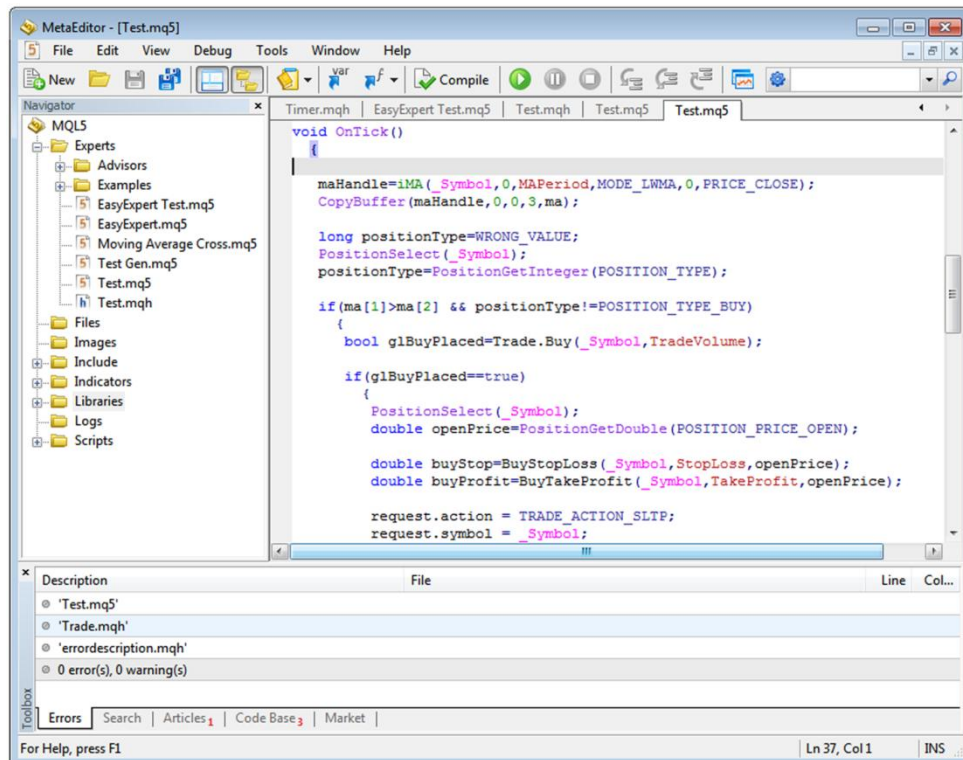
ویرایشگر متا یک محیط توسعه یکپارچه³ برای MQL5 است که همراه با نرم افزار متاتریدر 5 می باشد. شما می توانید این ویرایشگر را با کلیک بر دکمه ویرایشگر متا در جعبه ابزار استاندارد رابط

1- Default

2- Log

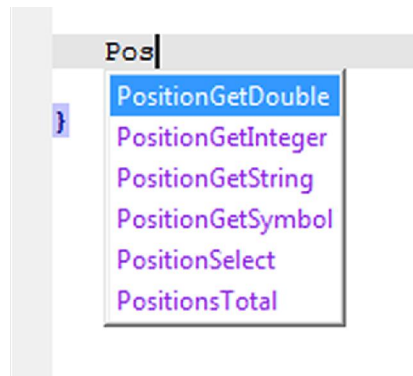
3- IDE

کاربری متاتریدر باز کنید یا کلید F4 را فشار دهید. همچنین می‌توانید آنرا از منوی استارت ویندوز باز کنید. ویرایشگر متا ویژگی‌های سودمند زیادی برای خلق برنامه‌های MQL5 دارد از جمله تکمیل خودکار لیست اسامی، راهنمای ابزار اطلاعات پارامترها، ابزارهای جستجو، ابزارهای اشکال زدایی و غیره..



شکل 1-1. رابط کاربری ویرایشگر متا، از بالا سمت چپ پنجره هدایتگر، ویرایشگر کد و جعبه ابزار

اولین حرف از حروف یک مؤلفه زبان MQL5، یک متغیر یا تابع را تایپ کنید و یک لیست کشویی با کلمات محتمل ظاهر می‌شود. با کلیدهای بالا و پایین در لیست حرکت کنید کلمه مورد نظر را انتخاب کرده و کلید ENTER را فشار دهید. همچنین با کلیک سمت چپ ماوس نیز می‌توان آن کلمه را انتخاب کرد. شما می‌توانید لیست اسامی کشویی را هر زمان با فشار دادن ترکیبی از کلیدهای Ctrl +Space فراخوانی کنید یا با انتخاب لیست اسامی از منوی edit.



شکل 2-1. لیست اسامی کشویی

زمانیکه پارامترهای یک تابع را پر می‌کنید، ابزار راهنمای اطلاعات نمایان می‌شود تا پارامترهای آن تابع را به شما خاطر نشان کند. متن برجسته در داخل ابزار اطلاعات پارامترها، بیانگر پارامتر اخیر می‌باشد. بعضی توابع چندین نوع دارند، مثلاً تابع `SymbolInfoDouble()` در شکل ۳-۱ دو نوع دارد همچنانکه در متن اطلاعات پارامترها با [1 of 2] نشان داده شده است. با استفاده از کلیدهای بالا و پایین¹ می‌توانید از هر دو تابع استفاده کنید. شما در هر زمان می‌توانید ابزار اطلاعات پارامتر را با فشار دادن ترکیبی از کلیدهای `Ctrl + shift + space` فراخوانی کنید یا اینکه از منوی `Edit` گزینه اطلاعات پارامتر را انتخاب کنید.

```
[1 of 2] double SymbolInfoDouble(const string symbol_name,ENUM_SYMBOL_INFO_DOUBLE property_id)
SymbolInfoDouble ( _Symbol, )
```

شکل 3-1. راهنمای اطلاعات پارامترها را نشان می‌دهد.

دو پنجره دیگر نیز در داخل رابط کاربری ویرایشگر متا وجود دارد. پنجره هدایتگر² محتوای پوشه MQL5 را به صورت ساختار درختی نمایش می‌دهد که اجازه دسترسی سریع به برنامه‌های MQL را فراهم می‌کند. پنجره جعبه ابزار شامل چندین تب³ می‌باشد که شامل تب خطاها که نمایشگر خطاهای کامپایل می‌باشد، تب جستجو که نمایشگر نتایج جستجوهایست و مقالات، پایگاه کد و تب مارکت که اطلاعات وبسایت MQL5 را لیست می‌کنند.

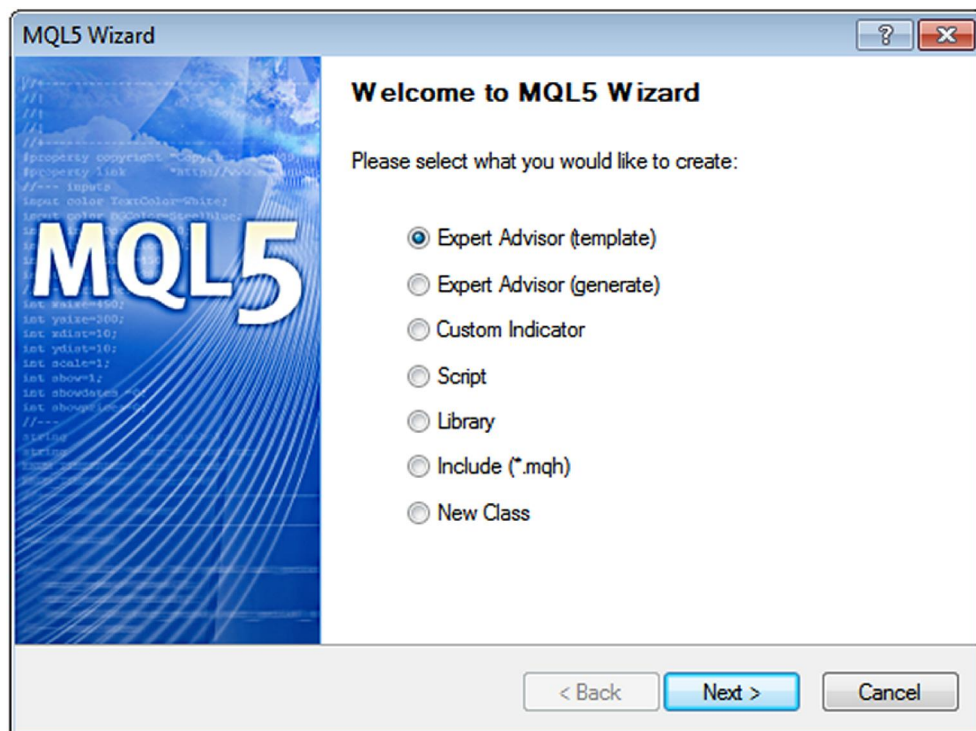
ویرایشگر متا یک راهنمای مرجع MQL5 است که برای پیدا کردن توابع و سایر مؤلفه‌های زبان برنامه نویسی سودمند است. به سادگی و با قرار دادن ماوس بر روی کلمه کلیدی و با فشردن دکمه `F1`

1- Arrow Key
2- Navigator
3- Tab

از صفحه کلیدتان نیز می‌توان آنرا باز کرد. مرجع MQL5 در صفحه دیگری باز می‌شود. شما همچنین می‌توانید این مرجع را از منوی Help باز کنید.

ویزارد (جادوگر) MQL5

از جادوگر MQL5 برای خلق یک برنامه جدید MQL5 استفاده می‌شود برای باز کردن آن، روی دکمه New در جعبه ابزار کلیک کنید یا New را از منوی File انتخاب کنید. یک پنجره با گزینه‌های زیر پدیدار می‌شود.



شکل 4-1. جادوگر MQL5

- **Expert Advisor (template)** - این گزینه یک فایل اکسپرت جدید را از قالب معینی ایجاد می‌کند. فایل ایجاد شده در فولدر `MQL5\Experts` یا یک زیر پوشه مشخص شده، ذخیره می‌شود.
- **Expert Advisor(generate)** - این گزینه به کاربر این امکان را می‌دهد که بدون کد نویسی یک اکسپرت ایجاد می‌کند، اکسپرت تولید شده از کتابخانه استاندارد MQL5 استفاده می‌کند.
- **Custom Indicator** - این گزینه یک اندیکاتور سفارشی جدید از یک قالب مشخص ایجاد می‌کند. فایل ایجاد شده در پوشه `MQL5\Indicator` یا یک زیر پوشه مشخص، ذخیره می‌شود.

- **Script** - این گزینه یک فایل اسکریپت خالی از قالبی مشخص ایجاد می‌کند. فایل ایجاد شده در پوشه `\MQL5\Script` یا یک زیر پوشه مشخص، ذخیره می‌شود.
 - **Library** - این گزینه یک کتابخانه خالی از قالبی مشخص ایجاد می‌کند. فایل ایجاد شده در پوشه `\MQL5\Libraries` یا یک زیر پوشه مشخص، ذخیره می‌شود.
 - **Include(*.mqh)** - این گزینه یک فایل اینکلود را در قالب مشخص با فرمت `.mqh` ایجاد می‌کند. فایل ایجاد شده در پوشه `\MQL5\Include` یا زیر پوشه مشخص ذخیره می‌شود.
 - **New Class** - این گزینه یک فایل اینکلود با تعریف کلاس از قالبی مشخص ایجاد می‌کند. فایل ایجاد شده در پوشه `\MQL5\Experts` یا یک زیر پوشه مشخص ذخیره می‌شود.
- ما در طول مطالب این کتاب بطور عمیق‌تری وارد بحث ایجاد برنامه‌ها به کمک جادوگر MQL5 می‌شویم.

کامپایل کردن

برای کامپایل کردن برنامه MQL5، دکمه Compile را در جعبه ابزار ویرایشگر متا فشار دهید. خطاهای فایل کنونی MQL5 و تمام فایل‌های اینکلود آن، بررسی شده و یک فایل `Ex5` تولید می‌شود. هر خطا و یا اختطاری در تب Errors از پنجره جعبه ابزار پدیدار می‌شود. خطاها می‌بایست قبل از تولید فایل `EX5` برطرف گردند. فصل بیست و دوم در مورد اشکال زدایی و برطرف کردن خطاهای برنامه بحث می‌کند. به اختطارها نیز باید رسیدگی شود اما می‌توانند در مورد عدم ایجاد مشکل، نادیده گرفته شوند. برنامه با وجود اختطارها، همچنان قابل کامپایل با موفقیت می‌باشد.

نگارش برنامه

MQL5 شبیه به سایر برنامه‌های مدرن از قبیل `C#`، `C++` یا `Java` می‌باشد. اگر در زبانهایی با شکل و فرم زبان‌های خانواده C برنامه نویسی کرده‌اید، نگارش و ساختار MQL برای شما آشنا خواهد بود. یک عبارت یا اپراتور در زبان MQL5 باید به سمی کالون (;) ختم شود. یک عبارت می‌تواند چندین طول داشته باشد اما باید در انتهای آن علامت سمی کالون (;) باشد. اضافه نکردن (;) در انتهای عبارت یک اشتباه متداول است که بیشتر برنامه نویسان تازه کار انجام می‌دهند.

// یک عبارت ساده

```
x = y + z;
```

// یک عبارت که چند خط طول می‌کشد.

```
x = (y + z)
    / (q - r);
```

یک استثنا در مورد قرار دادن سمی کالون (;) اپراتور ترکیبی است. یک اپراتور ترکیبی شامل یک اپراتور است که دنباله آن یک جفت آکولاد {} می‌آید. در مثال زیر، اپراتور عبارت `x==0` است. هیچ سمی کالونی بعد از آکولاد آخر نمی‌آید. هر عبارتی در داخل آکولاد ها باید به (;) ختم شود.

```
// یک اپراتور ترکیبی ساده
If (x==0)
{
    Print (" x is equal to zero ");
    return;
}
```

نام یا شناسه¹

زمانی که متغیرها²، توابع³ و کلاس‌ها⁴ را نامگذاری می‌کنید، نیاز به استفاده از یک نام منحصر بفرد و توصیفی دارید. نام نباید با نام دیگری در برنامه همسان باشد و نباید با یکی از اجزاء زبان MQL5 یکی باشد.

شما می‌توانید از حروف، اعداد و کاراکتر زیر خط دار (_) استفاده کنید اگر چه اولین کاراکتر یک نام نباید عدد باشد. حداکثر طول یک نام 64 کاراکتر است. این باعث می‌شود که شما امکان زیادی برای خلاقیت داشته باشید، بنابراین از نامی استفاده کنید که واضح و توصیف کننده باشد.

نام‌ها به بزرگی و کوچکی حروف حساسند، به این معنی که مثلاً `MyIdentifier` و `myIdentifier` یکی نمی‌باشند! برنامه نویسان از این ویژگی برای تمایز بین انواع مختلف متغیرها، توابع و کلاسها استفاده می‌کنند. استفاده از حروف بزرگ در این کتاب، ابتدای کلمات زیر می‌باشد:

- در متغیرهای سراسری⁵، اشیاء، کلاس‌ها و توابع اولین حرف هر کلمه بزرگ می‌باشد. به عنوان مثال: `MyFunction` یا `MyVariable`
- متغیرهای محلی⁶ و اشیاء که داخل یک تابع اعلان می‌شوند، از روش دیگری استفاده می‌کنند که در آن اولین کلمه از حروف کوچک استفاده می‌شود و اولین حرف از تمام کلمات بعدی، بزرگ می‌باشد. به عنوان مثال: `myVariable` یا `localObject`
- ثابت‌ها، تمام حروفشان بزرگ است. برای جدا کردن کلمات از هم از علامت (_) استفاده می‌شود. به عنوان مثال: `MY_CONSTANT`

1- Identifiers
2- Variables
3- functions
4- classes
5- Global variables
6- Local variables

توضیحات یا کامنت¹

کامنتها برای شرح دادن یک قسمت از کد برنامه بکار می‌رود. شما از کامنت‌ها می‌توانید برای منظم کردن برنامه خود بهره ببرید. همچنین با استفاده از کامنت‌ها بطور موقت خطوطی از کد را می‌توانید از برنامه حذف کنید. هر خطی از برنامه به شکل کامنت در آید، توسط کامپایلر نادیده گرفته می‌شود. برای اضافه کردن یک کامنت، اولین کاراکترها در هر جمله باید دو علامت اسلش (//) باشد. این علامت یک خط را به صورت کامنت در می‌آورد.

// این یک کامنت است //

// خط زیرین بصورت یک کامنت در آمده //

// X = Y + Z;

برای تبدیل چند خط از کد به کامنت، از علامت اسلش - ستاره (*) در ابتدای متن استفاده کرده و در انتهای آن ستاره - اسلش (*/) قرار می‌دهیم.

/* این یک کامنت چند خطی است */

این خطوط بوسیله کامپایلر نادیده گرفته می‌شوند

X = Y + Z; /*

ویرایشگر متا دارای یک سری از دستورات کامنت کردن می‌باشد. با انتخاب خطوطی که می‌خواهید آنها را به صورت کامنت در آورید، به کمک ماوس شروع کنید. در منوی ویرایش در زیر منوی Comments منوی Comment lines خطوط انتخاب شده را به صورت کامنت در می‌آورد در حالی که uncommented کامنتها را از خطوط انتخاب شده، حذف خواهد کرد. منوی Function Header یک سربرگ کامنت شده، مشابه با آنچه به صورت خودکار در MQL5 ایجاد می‌شود، در کد می‌گنجانند.

```
// Function header generated by Edit menu -> Comments -> Function Header
//+-----+
//|                                     |
//+-----+
```


فصل 2

انواع متغیرها¹ و داده‌ها²



متغیرها

یک متغیر اولین واحد ذخیره‌سازی در هر زبان برنامه‌نویسی می‌باشد. متغیرها اطلاعاتی را که برای برنامه ما ضروری است، مانند قیمت‌ها، مقادیر اندیکاتور یا پارامترهای معامله، در خود نگاه می‌دارد. قبل از آنکه یک متغیر بتواند استفاده شود، می‌بایست تعریف گردد. شما یک متغیر را با مشخص کردن نوع داده و یک نام منحصر به فرد به آن اعلان می‌کنید. به صورت اختیاری شما می‌توانید یک مقدار اولیه به متغیر اختصاص دهید. شما معمولاً یک متغیر را در آغاز یک برنامه یا تابع یا زمانی که برای اولین بار استفاده می‌شود اعلان می‌کنید. اگر شما یک متغیر را بیش از یک بار اعلان کنید یا اصلاً اعلان نکنید، یک خطای کامپایل دریافت می‌کنید. این مثالی از اعلان کردن یک متغیر می‌باشد:

```
int mynumber=1;
```

در این مثال، نوع داده (integer) می‌باشد و نام آن mynumber است و ما مقدار اولیه یک را به آن اختصاص داده‌ایم، زمانی که برای اولین بار متغیر اعلان شد، می‌توانید مقدار آن را با اختصاص یک مقدار جدید، تغییر دهید.

```
mynumber =3;
```

متغیر mynumber حالا مقدار عدد 3 را دارد. شما همچنین می‌توانید مقدار یک متغیر را به متغیر دیگری اختصاص دهید.

```
int mynumber;
int yournumber=2;
mynumber= yournumber;
```

متغیر mynumber اکنون مقدار عددی 2 دارد. اگر شما مقدار اولیه‌ای به متغیر ندهید، مقدار پیش فرض خالی برای آن در نظر گرفته می‌شود. برای انواع عددی، مقدار اولیه آن صفر است و برای انواع رشته‌ای مقدار آن null یا یک رشته خالی است (" ").

انواع داده‌ها

زمانی که یک متغیر را اعلان می‌کنید، نوع داده تعیین می‌کند که متغیر چه نوع داده‌ای را می‌تواند نگاه دارد. انواع داده‌ها در MQL می‌تواند به سه نوع اصلی تقسیم شود.

- نوع صحیح (Integer) که اعداد صحیح هستند، به‌عنوان مثال: 0, 1, 10563
- نوع حقیقی که اعداد اعشاری با نقطه ممیز می‌باشند، مانند: 1.35635
- رشته‌ها که متنی هستند و تشکیل شده از کاراکترها می‌باشند، به‌عنوان مثال: جمله: "روبه قهوه‌ای از روی سگ تنبل پرید"

انواع صحیح¹

MQL4 تنها یک نوع داده صحیح دارد؛ int، MQL5 چند نوع داده دیگر به آن اضافه کرده که عددها را در بازه‌های مختلفی نگهداری می‌کند. بگذارید با نوع صحیح علامت‌دار آغاز کنیم، نوع علامت‌دار می‌تواند عددهای مثبت و منفی را نگاه دارد:

- **Char**: نوع Char یک بایت از حافظه را اشغال می‌کند. محدوده مقادیر بین 128- تا 127 می‌باشد.
- **Short**: این نوع از دو بایت از حافظه استفاده می‌کند. محدوده مقادیر آن بین 32768- تا 32767 می‌باشد.
- **Int**: نوع int از چهار بایت حافظه استفاده کرده و محدوده آن بین 2,147,483,648 تا 2,147,483,647 است.
- **Long**: این نوع داده هشت بایت از حافظه را اشغال کرده و محدوده آن بین 9,223,372,036 تا 9,223,372,036 می‌باشد.

1- Integer

پس چه نوع صحیحی باید استفاده شود؟ شما به‌طور متناوب در توابع MQL5 استفاده از انواع int و long را مشاهده می‌کنید، بنابراین این‌ها انواعی هستند که بیشتر استفاده می‌شوند، شما می‌توانید از انواع char و short نیز اگر بخواهید استفاده کنید.

انواع داده صحیح بدون علامت نیز موجود می‌باشد که مقادیر منفی در آنها استفاده نمی‌شود. انواع بدون علامت همان مقدار حافظه‌ای را مصرف می‌کنند که انواع علامت‌دار استفاده می‌کنند، اما حداکثر مقدار آنها دو برابر حداکثر مقدار علامت‌داری‌ها می‌باشد.

- **Uchar**: نوع Uchar یک بایت حافظه را اشغال کرده و محدوده آن بین صفر تا 255 می‌باشد.
- **Ushort**: این نوع دو بایت از حافظه را اشغال کرده و محدوده آن بین صفر تا 535، 65 می‌باشد.
- **Uint**: این نوع داده، چهار بایت حافظه را اشغال کرده و محدوده آن بین صفر تا 295، 967، 294، 4 است.
- **Ulong**: این نوع داده، هشت بایت از حافظه را اشغال کرده و محدوده آن بین صفر و 615، 551، 709، 744، 446، 18 می‌باشد.

در عمل، به‌ندرت از متغیرهای صحیح بدون علامت استفاده می‌شود. اما برای استفاده در دسترس می‌باشند.

انواع حقیقی¹ یا اعشاری

اعداد نوع حقیقی برای ذخیره مقادیر عددی کسری از قبیل قیمت‌ها بکار می‌رود. دو نوع عدد حقیقی در MQL5 وجود دارد. اختلاف بین این دو نوع درجه دقت آنهاست زمانیکه نماینده اعداد کسری می‌باشند.

- **Float**: نوع Float از چهار بایت حافظه بهره می‌برد. حدود 7 رقم اعشار دقت دارد.
 - **Double**: نوع Double از هشت بایت حافظه سود می‌برد و تا 15 رقم اعشار دقت دارد.
- شما معمولاً به‌طور متناوب نوع double را در برنامه MQL5 استفاده می‌کنید. نوع float می‌تواند برای صرفه‌جویی در حافظه استفاده شود و کاربرد آن زمانی است که با آرایه‌ها² بزرگ از اعداد حقیقی سروکار داریم. اما در توابع MQL5 استفاده نمی‌شود.

نوع رشته‌ای³

نوع رشته‌ای برای ذخیره متن بکار می‌رود. رشته می‌بایست داخل دو علامت (" ") کوتیشن قرار گیرد. در زیر مثالی از اعلان متغیر رشته‌ای آورده شده:

```
string mystring= "This is a string";
```

1- Real Type
2- Array
3- String

اگر شما نیاز به استفاده از یک یا دو (" ") در داخل رشته دارید، از یک بک اسلش (\) قبل از علامت (" ") استفاده کنید. این کاراکتر فرار نامیده می‌شود.

```
Strings myQuote= "We are\"escaping\"double quotes";
Print (myQuote);
//output: we are "escaping" double quotes.
```

اگر شما نیاز به استفاده از یک بک اسلش (\) در رشته خود دارید، از دو بک اسلش پشت سر هم استفاده کنید. مثلاً:

```
string mySlash= "This is a backslash:\\";
Print (mySlash);
// output: This is a backslash:\
```

شما همچنین می‌توانید یک خط جدید را به یک رشته با استفاده از کاراکتر فرار \n اضافه کنید. مثال:

```
String myNewline= "This string has \n a New line character";
Print (myNewline);
// output: This string has
//          a New line character
```

شما می‌توانید رشته‌ها را با کمک اپراتور الحاق (+) باهم ترکیب کنید. این عمل باعث ترکیب چند رشته و ادغام آن در یک رشته می‌شود:

```
String insert= "Concatenated";
String mycontact= "This is an example of a " + insert + "string";
Print (mycontact);
//output: This is an example of a Concatenated string
```

تابع StringConcatenate() نیز می‌تواند برای ادغام رشته‌ها استفاده شود. این عمل از نظر حافظه بهتر از استفاده از عملگر (+) ادغام می‌باشد. اولین پارامتر از تابع StringConcatenate() متغیر رشته‌ای برای کپی رشته ادغامی و بقیه پارامترها رشته‌هایی هستند که در آن ادغام می‌شوند:

```
string newString;
string insert = "concatenated";
StringConcatenate(newString, "This is another example of a ", insert,
" string.");
Print(newString);
// Output: This is another example of a concatenated string.
```

متغیر newstring حاوی رشته ادغامی می‌باشد، توجه کنید که رشته‌هایی که در هم ادغام می‌شوند، به وسیله کاما (,) در داخل تابع StringConcatenate() از هم جدا می‌شوند.

سرانجام اگر شما یک رشته طولانی دارید، می‌توانید آن را در چند خط بشکنید. شما نیازی به استفاده از عملگر الحاق (+) ندارید. هر خط باید با علامت (" ") محصور شود و باید یک سمیکالون (;) در انتهای عبارت قرار داده شود.

```
string myMultiline = "This is a multi-line string. "
    "These lines will be joined together.";
Print(myMultiline);
Output: This is a multi-line string. These lines will be joined
together.
```

داده‌های منطقی¹

متغیر منطقی یا bool برای ذخیره مقادیر درست یا غلط (True/False) استفاده می‌شود. از نظر تکنیکی نوع بولین یک نوع صحیح است، از آنجاکه مقادیر صفر (غلط) و یک (درست) می‌گیرد. در زیر مثالی از یک متغیر نوع بولین آمده است:

```
bool myBool = true;
Print(myBool);
//Output: True
```

اگر به یک متغیر بولین مقدار اولیه True نداده باشیم، مقدار پیش‌فرض آن صفر یا غلط خواهد بود و مقدار غیر صفر در یک متغیر بولین به درست (True) نسبت داده می‌شود.

```
bool myBool;
Print(myBool);
// Output: false
```

```
myBool = 5;
if(myBool == true) Print("myBool is true");
//Output: myBool is true
```

در مثال بالا، ما ابتدا متغیر بولین را (myBool) بدون مقدار تعریف کرده‌ایم. بنابراین مقدار آن برابر صفر یا غلط (false) می‌باشد. زمانی که ما مقدار عدد 5 را به متغیرمان نسبت داده‌ایم، myBool در یک اپراتور به‌عنوان درست (True) عمل می‌کند. ما در فصل سوم بیشتر درباره عملگرهای بولین صحبت خواهیم کرد.

نوع رنگ²

متغیر نوع رنگ برای ذخیره اطلاعاتی درباره رنگ‌ها بکار می‌رود. رنگ‌ها می‌توانند با ثابت‌های پیش‌تعریف‌شده انتخاب شوند: مقادیر RGB یا مقادیر هگزادسیمال

1- Boolean Type
2- Color Type

شما بیشتر از ثابت‌های رنگ استفاده خواهید کرد. این رنگ‌ها همان‌هایی هستند که شما در خطوط اندیکاتور یا یک شیء نمودار از آن استفاده می‌کنید. شما می‌توانید تمام سری ثابت‌های رنگ را در مرجع MQL5 در قسمت

Standard constants ... > objects constants > web colors

مشاهده کنید. در زیر مثالی از متغیر رنگ که با استفاده از یک ثابت رنگ اعلان شده، مشاهده می‌شود:

```
color lineColor = clrRed;
```

متغیر رنگ lineColor به وسیله ثابت رنگ قرمز مقداردهی اولیه شده است، clrRed. این مثالی از کاربرد مقادیر RGB برای قرمز است:

```
color lineColor = C'255,0,0';
```

مقدار RGB رنگ قرمز 255, 0, 0 است. یک ثابت RGB با حروف بزرگ C آغاز می‌شود و مقدار RGB در داخل علامت تک کوت (' ') قرار داده می‌شود. سرانجام، مثالی از کاربرد مقدار هگزادسیمال برای رنگ قرمز:

```
color lineColor = 0xFF0000;
```

مقدار هگزادسیمال با '0x' شروع می‌شود و شش کاراکتر مقدار هگزادسیمال در دنباله آن می‌آید. در این مثال این مقدار FF0000 می‌باشد.

RGB و مقدار هگزادسیمال برای رنگ‌های سفارشی استفاده می‌شوند که در ثابت‌های رنگ تعریف نشده‌اند. اگر شما با کار کردن یا رنگ‌های RGB و هگزادسیمال راحت هستید، می‌توانید رنگ‌های دلخواه خودتان را تعریف کنید. در غیر این صورت، شما ثابت‌های رنگ را بسیار راحت و سودمند خواهید دانست.

متغیر نوع تاریخ و زمان¹

این نوع متغیر برای ذخیره زمان و تاریخ بکار می‌رود. زمان و تاریخ در متغیر تاریخ زمان نسبت به زمان unix ذخیره می‌شوند که تعداد ثانیه‌هایی است که از تاریخ 1 ژانویه سال 1970 سپری شده است. تاریخ 1 ژانویه 2012 نیمه‌شب در زمان unix برابر 1,325,397,600 می‌باشد.

اگر شما نیاز دارید که به یک متغیر تاریخ زمان یک مقدار خاصی زمان و تاریخ بدهید، از ثابت تاریخ زمان بهره ببرید. یک ثابت تاریخ زمان با یک حرف بزرگ D شروع می‌شود که در آن تاریخ و

1- Datetime

زمان در (' ') می‌باشد. تاریخ و زمان به فرمت hh:mm:ss yyyy.mm.dd ذخیره می‌شوند. مانند مثال زیر:

```
datetime myDate = D'2012.01.01 00:00:00';
```

مثال بالا متغیر myDate را با تاریخ 1 ژانویه 2012 در نیمه‌شب مقداردهی می‌کند. شما می‌توانید قسمت‌هایی از ثابت تاریخ زمان را حذف کنید اگر استفاده نمی‌شوند. مثال‌های زیر بیانگر این قضیه می‌باشند:

```
datetime myDate = D'2012.01.01 00:00';           // Hour and minute
datetime myDate = D'2012.01.01 00';           // Hour only
datetime myDate = D'2012.01.01';             // Date only
```

تمام مثال‌های ذکرشده، متغیر datetime را با یک مقدار، مقداردهی اولیه می‌کنند. یعنی زمان اول ژانویه 2012 در نیمه‌شب از آنجاکه قسمت hh:mm:ss تاریخ زمان استفاده نشده است، می‌توان آن را حذف کرد. پس چه اتفاقی می‌افتد اگر قسمت تاریخ را حذف کنیم؟ کامپایلر تاریخ امروز را جایگزین می‌کند. تاریخ کامپایل برنامه. فرض می‌کنیم امروز یک ژانویه 2012 است. حال چه اتفاقی می‌افتد زمانی که ما از یک ثابت تاریخ زمان بدون تاریخ استفاده کنیم:

```
datetime myDate = D'';                          // 2012.01.01 00:00
datetime myDate = D'02:00';                     // 2012.01.01 02:00
```

در اولین مثال، متغیر myDate به زمان نیمه‌شب امروز تنظیم شده است. در دومین مثال، ما زمانی را برای ثابت تاریخ زمان تعیین کرده‌ایم که تاریخ را برای امروز بازمان تعیین شده قرار می‌دهد. MQL5 چندین ثابت از پیش تعریف شده برای تاریخ و زمان دارد. ثابت (__DATE__) تاریخ کامپایل برنامه را برمی‌گرداند. این درست مانند استفاده از یک ثابت زمانی خالی می‌باشد. ثابت (__DATETIME__) زمان و تاریخ کنونی کامپایل را نشان می‌دهد. توجه کنید که دو علامت خط زیرین (__) قبل و بعد از هر ثابت وجود دارد. در زیر مثالی از استفاده ثابت (__DATE__) را نشان می‌دهد. فرض می‌کنیم که تاریخ امروز اول ژانویه 2012 است.

```
datetime myDate = __DATE__;                     // 2012.01.01 00:00
datetime myDate = D'';                         // 2012.01.01 00:00
```

و حال مثالی از ثابت (__DATETIME__) فرض می‌کنیم زمان و تاریخ کامپایل برنامه اول ژانویه در ساعت 3:15:05 می‌باشد:

```
datetime myDate = __DATETIME__;                // 2012.01.01 03:15:05
```