

ویژوال استودیو به توسعه دهندگان این امکان را می‌دهد تا کدهایی را که تکراری بوده و به دفعات در متن برنامه مورد استفاده هستند به شکل یک قطعه کد آماده (در صورت نیاز با مقادیر پیش فرض) ذخیره کنند، سپس در مواقع نیاز بدون اینکه مجبور باشند آن را دوباره و دوباره بنویسند، تنها با تایپ کردن نام قطعه کد ذخیره شده و دو بار فشردن کلید Tab، کد تعریف شده توسط ویژوال استودیو در محل تعیین شده اضافه می‌گردد. به این قطعه کدهای آماده [code snippet](#) گفته می‌شود. خود ویژوال استودیو تعدادی [code snippet آماده](#) دارد که آشنایی با آنها می‌تواند سرعت کدنویسی را افزایش دهد. برای دیدن لیست کامل و مدیریت آنها به مسیر

Tools -> Code Snippets Manager (Ctrl+K, Ctrl+B)

بروید.

در ویژوال استودیو 2010 دو نوع snippet وجود دارد:

1- Expansion snippets: که در محل کرسر (Cursor) اضافه می‌شوند. مثل cw و enum که به ترتیب دستور writeLine و ساختار یک enum را ایجاد می‌کنند.

2- SurroundsWith snippets: که می‌توانند یک تکه کد انتخاب شده را در بر بگیرند مثل for و do که کد انتخاب شده را در یک حلقه for و do-while محصور می‌کنند.

نکته ای که باید توجه داشت این است که یک snippet می‌تواند از هر دو نوع باشد. برای مثال for و do و یا if، در صورتی که کدی انتخاب شده باشد آن را محصور می‌کنند و گرنه ساختار خالی مرتبط را در محل cursor اضافه می‌کنند.

همانطور که در ابتدا هم ذکر شد، علاوه بر snippetهای آماده‌ی موجود، توسعه دهنده می‌تواند قطعه کدهایی را خود ایجاد کرده و مورد استفاده قرار دهد.

در اینجا یک expansion snippet خواهیم ساخت تا کار اضافه کردن بلاک try-catch-finally را برای ما انجام دهد.

- ابتدا یک فایل xml به پروژه اضافه می‌کنیم و آنرا TryCatchFinally.snippet می‌نامیم. توجه کنید که نام فایل باید به snippet. ختم شود.

- فایل را باز و درون آن راست کلیک کرده و گزینه Insert snippet > Snippet را انتخاب می‌کنیم. با اینکار یک [قالب پایه snippet](#) (که یک ساختار xml) است به فایل اضافه می‌شود. هر فایل snippet از دو بخش اصلی [header](#) و [snippet](#) تشکیل شده که بخش header اطلاعاتی کلی درباره قطعه کد را نگهداری می‌کند و بخش snippet مربوط به تعریف محتوای قطعه کد است.

```
<codesnippet format="1.0.0" xmlns="http://schemas.microsoft.com/VisualStudio/2005/CodeSnippet">
  <header>
    <title>title</title>
    <author>author</author>
    <shortcut>shortcut</shortcut>
    <description>description</description>
    <snippettypes>
      <snippettype>SurroundsWith</snippettype>
      <snippettype>Expansion</snippettype>
    </snippets>
  </header>
  <content>
    <code>
try
{
}
catch
{
}
finally
{
}
    </code>
  </content>
</codesnippet>
```

```
</snippettypes>
</header>
<snippet>
  <declarations>
    <literal>
      <id>name</id>
      <default>value</default>
    </literal>
  </declarations>
  <code language="XML">
    <!--[CDATA[<test-->
      <name>$name$</name>
      $selected$ $end$]]>
  </code>
</snippet>
</codesnippet>
```

- قالب پیش فرض شامل هر دو نوع snippet است .

```
<codesnippet format="1.0.0" xmlns="http://schemas.microsoft.com/VisualStudio/2005/CodeSnippet">
  <header>
    ...
    <snippettypes>
      <snippettype>SurroundsWith</snippettype>
      <snippettype>Expansion</snippettype>
    </snippettypes>
  </header>
  ...
</codesnippet>
```

از آنجا که قصد داریم یک Expansion snippet بسازیم پس تگ SurroundsWith را حذف می‌کنیم .

```
<snippettypes>
  <snippettype>Expansion</snippettype>
</snippettypes>
```

- در بخش header مقدار تگ [Title](#) را به "Try Catch Finally" و مقدار تگ [Shortcut](#) را به "trycf" و [Description](#) را به "Adds a try-catch-finally block" تغییر می‌دهیم . Title عنوان snippet است و وجود آن ضروری است . اضافه کردن shortcut اختیاری است و به عنوان یک متن میانبر برای اضافه کردن snippet استفاده می‌شود .

```
<Header>
  <Title>Try Catch Finally</Title>
  <Author>mohsen.d</Author>
  <Shortcut>trycf</Shortcut>
  <Description>Adds a try-catch-finally block</Description>
```

- تگ [Literal](#) برای تعریف جایگزین برای بخشی از کد درون snippet که احتمال دارد پس از اضافه شدن ، توسط برنامه نویس و یا در صورت استفاده از [function](#) توسط خود ویژوال استودیو تغییر کند استفاده می‌شود . در قطعه کد try-catch-finally ، ما می‌خواهیم به کاربر اجازه بدهیم که نوع استثنائی را که catch می‌شود تغییر دهد .  
تگ [id](#) نامی برای بخش قابل ویرایش تعریف می‌کند ( که از آن در ادامه در تعریف خود قطعه کد استفاده می‌کنیم ) . آنرا به "ExceptionName" تغییر می‌دهیم . تگ [default](#) هم مقدار پیش فرضی را برای آن بخش مشخص می‌کند . ما می‌خواهیم تمام استثناها را Catch کنیم پس مقدار پیش فرض را برابر "Exception" قرار می‌دهیم .

```
.....
<declarations>
  <literal>
```

```
<id>ExceptionName</id>
<default>Exception</default>
</literal>
</declarations>
...
```

- و در تگ [Code](#) ، خود قطعه کدی که ویژوال استودیو باید آن را اضافه کند ، تعریف می‌شود . مقدار مشخصه Language آن را به CSharp تغییر می‌دهیم و محتویات داخل آنرا به شکل زیر اضافه می‌کنیم .

```
<code language="CSharp">
<!--[CDATA[
try
{
    $end$
}
catch($ExceptionName$)
{

}
finally
{

}
}]-->
</code>
```

به نحوه استفاده از ExceptionName که در قسمت Literal تعریف کردیم توجه کنید . عبارت \$end\$ هم یک کلمه رزرو شده است که محل قرار گرفتن cursor را بعد از اضافه شدن قطعه کد مشخص می‌کند .

- در نهایت snippet ما به شکل زیر خواهد بود :

```
<codesnippet format="1.0.0" xmlns="http://schemas.microsoft.com/VisualStudio/2005/CodeSnippet">
  <header>
    <title>Try Catch Finally</title>
    <author>mohsen.d</author>
    <shortcut>trycf</shortcut>
    <description>Adds a try-catch-finally block</description>
    <snippettypes>
      <snippettype>Expansion</snippettype>
    </snippettypes>
  </header>
  <snippet>
    <declarations>
      <literal>
        <id>ExceptionName</id>
        <default>Exception</default>
      </literal>
    </declarations>
    <code language="CSharp">
      <!--[CDATA[
try
{
    $end$
}
catch($ExceptionName$)
{

}
finally
{

}
}]-->
      </code>
    </snippet>
  </codesnippet>
```

## اضافه کردن snippet ساخته شده به visual studio

دو راه برای اضافه کردن snippet تعریف شده به ویژوال استودیو وجود دارد :  
روش اول قرار دادن فایل snippet در پوشه code snippets ویژوال استودیو است که مسیر پیش فرض آن

```
C:\Users\<UserName>\Documents\Visual Studio 2010\Code Snippets\
```

است . این پوشه به ازای هر زبان دارای یک زیر پوشه است . این snippet را باید در پوشه #C قرار دهیم . همین که فایل را در پوشه مناسب قرار دهیم ویژوال استودیو بدون نیاز به restart شدن آن را خواهد شناخت .

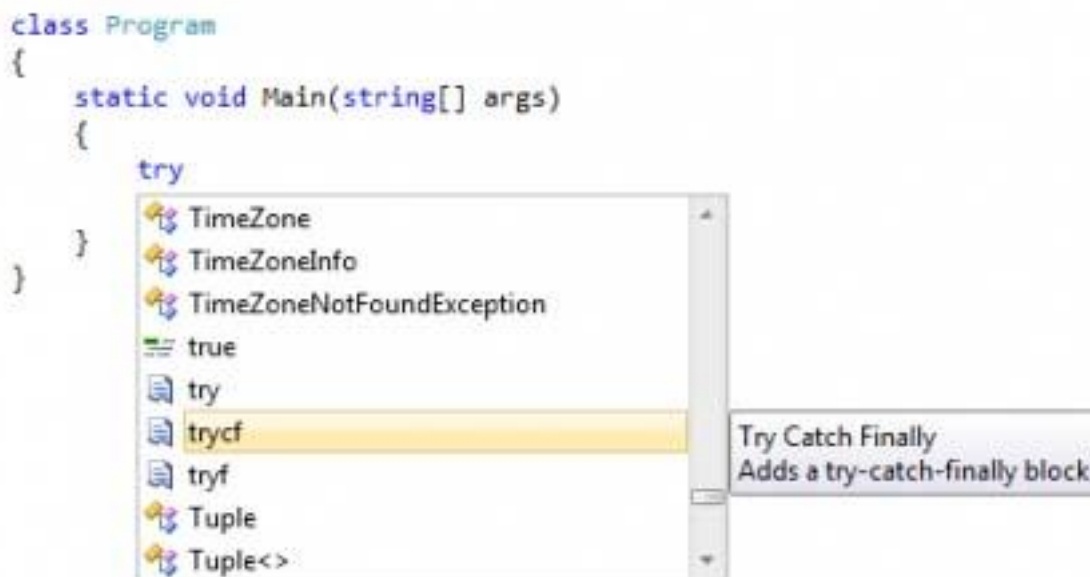
گزینه دوم import کردن فایل snippet به داخل ویژوال استودیو است . در ویژوال استودیو به مسیر

```
Tools -> Code Snippets Manager (Ctrl+K, Ctrl+B)
```

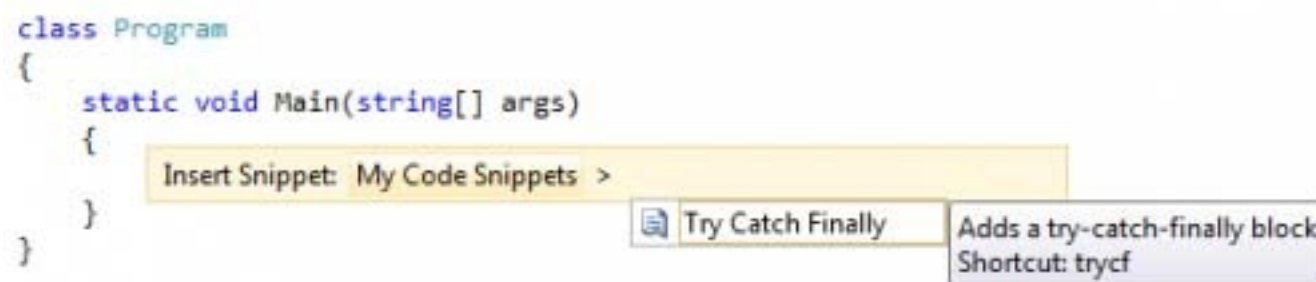
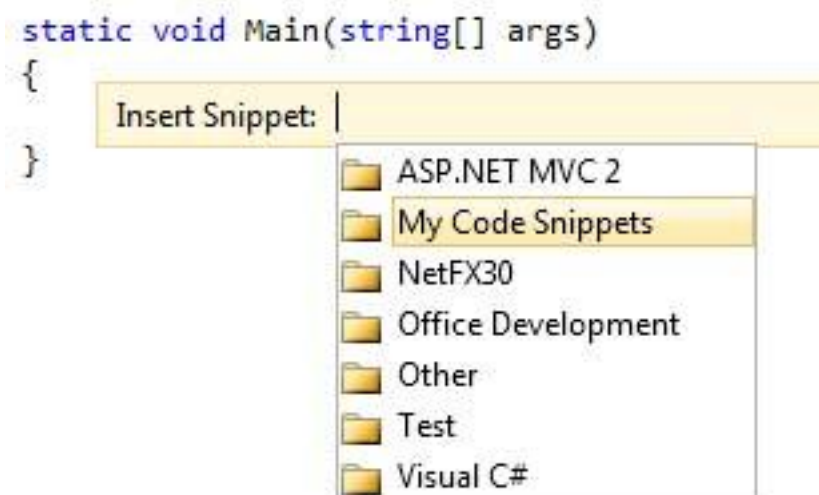
می‌رویم . در پنجره Code Snippets Manager ، بر روی کلید import کلیک و فایل موردنظر را یافته و انتخاب کرده و پوشه‌ی محل ذخیره شدن آن را تعیین می‌کنیم .

## استفاده از snippet ساخته شده

برای استفاده از snippet می‌توانیم متن میانبر تعریف شده را تایپ کنیم و با دو بار فشردن کلید tab قطعه کد تعریف شده به محل کرسر اضافه می‌شود



همینطور با فشردن کلیدهای Ctrl+X و Ctrl+K به صورت پشت سر هم منوی "Insert Snippet" ظاهر می‌شود که از طریق آن می‌توانیم Snippet موردنظر را یافته ( بدنبال Title تعریف شده برای snippet در پوشه ای که آنرا ذخیره کرده اید بگردید ) و با انتخاب آن کد تعریف شده اضافه خواهد شد .



برای آشنایی با روش‌های مختلف دسترسی به snippet ها [اینجا را](#) بررسی کنید .

## ابزارها

دستکاری خود فایل xml چندان جالب و خالی از خطا نیست . روش‌های بهتری برای ساخت و ویرایش snippet ها وجود دارد . [Snippet Editor](#) ابزاری برای ویرایش و ساخت snippet هاست و [Snippet Designer](#) هم یک پلاگین برای ویژوال استودیو است که کار مشابهی را انجام می‌دهد . یکی از کارهای جالبی که با این ابزار می‌توانید انجام دهید انتخاب یک قطعه از کد ( مثل یک تابع ) و سپس ساختن یک snippet از روی آن است .

[در این پروژه](#) هم مجموعه snippet های موجود در ویژوال استودیو 2010 برای زبان سی شارپ ، جهت سازگاری با [stylecop](#) ویرایش و refactor شده اند ( در کنار تعریف snippet های دیگر ) .