

# بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِیْمِ

## آموزش سی شارپ - بخش ۲۵ ← اجرای منوی Edit

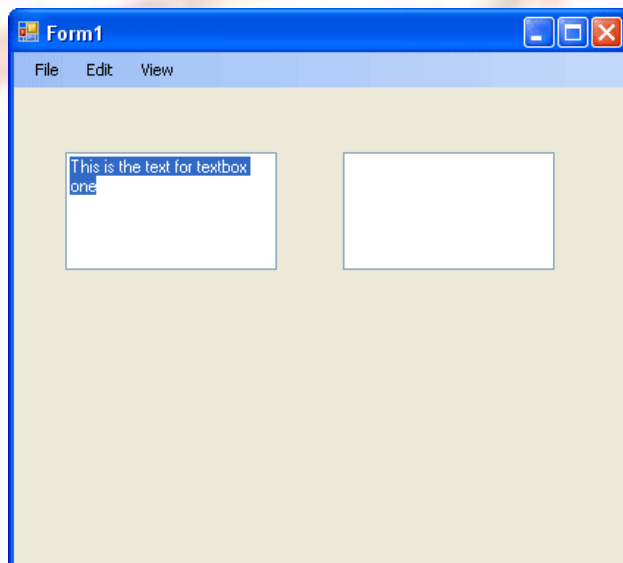
### • مقدمه

در بخش قبلی به صورت کوتاه تنها در رابطه با نحوه راه اندازی گزینه Quit از منوی File را به شما آموزش دادیم. در این بخش شما فراخواهید گرفت که چگونه منوی Edit را نیز اجرا کرده و باعث شود که هر گزینه از این منو، عملیات به خصوصی را انجام دهد.

### • اجرای منوی Edit در سی شارپ

قاعده‌تاً می‌بایست که ما قبل از کدنویسی منوی Edit، سایر گزینه‌های منوی File را راه اندازی کنیم، اما بدلیل پیچیدگی آن نسبت به سایر منوها، آن را فعلاً رها می‌کنیم و ابتدا با منوی Edit شروع می‌کنیم؛ در نهایت مجدداً به سراغ منوی File باز خواهیم گشت.

برای اینکه عملکرد گزینه‌های Cut، Copy، Paste و Undo را در عمل ببینید، ابتدا دو Text Box در کنار هم در منوی خود قرار دهید و برای جفت Text Box ها، در پنجره Properties آن‌ها، مقدار Property مربوط به MultiLine را به True تغییر دهید. با True کردن آن‌ها، خواهید توانست که به جای یک خط، چندین خط در یک Text Box بنویسید. برای اولین Text Box، در Property مربوط به Text، یک مقدار دلخواه بنویسید. فرم شما به شکل زیر خواهد شد:



کاری که می‌کنیم این هست که گزینه‌های Cut، Copy و Past را فعال می‌کنیم. Undo را نیز فراموش نخواهیم کرد! ابتدا ما متن داخل Text Box اول را Cut خواهیم کرد و سپس این عمل را Undo می‌کنیم.

بنابراین، به فرم خود بازگردید و بر روی گزینه Cut از منوی Edit، دابل-کلیک کنید؛ سپس وارد بخش کدنویسی خواهید شد:

```
private void mnuCut_Click(object sender, EventArgs e)
{
    |
}
```

کدی که برای انجام عملیات Cut در یک Text Box انجام می‌شود، خیلی راحت و ساده می‌باشد. تنها کافیسیت که کد زیر را بنویسید:

```
textBox1.Cut();
```

Cut() یک متدی در سی‌شارپ می‌باشد که همراه با خود زبان سی‌شارپ عرضه می‌شود و دقیقاً همان کاری را می‌کند که نام آن انتخاب شده‌است.

قبل از اینکه برنامه‌ی خود را اجرا کنید، یک‌بار دیگر به فرم خود بازگردید و بر روی undo دابل-کلیک کنید و کد زیر را بنویسید:

```
textBox1.Undo();
```

اکنون، کدهای خود را امتحان کنید؛ برنامه خود را اجرا کنید و متن خود را در Text Box انتخاب کنید. توسط گزینه Cut، متن را Cut کنید و سپس با استفاده از گزینه Undo، متن خود را بازگردانید.

کار دیگری که می‌توانیم انجام دهیم این است که بررسی کنیم که آیا متنی جهت Cut کردن انتخاب شده است یا نه! پس کافیسیت که کد مربوط به Cut را به شکل زیر تغییر دهید:

```
if (textBox1.SelectedText != "")
{
    textBox1.Cut();
}
```

فکر نمی‌کنم نیازی به شرح کدهای بالا وجود داشته باشد! کدها کاملاً گویای عملکرد خود هستند.

برای گزینه Undo نیز می‌توانیم چنین کاری را انجام دهیم؛ یعنی بررسی کنیم که آیا عملی برای Undo کردن وجود دارد یا نه! پس کدهای Undo را به شکل زیر تغییر دهید:

```
if (textBox1.CanUndo == true)
{
    textBox1.Undo();
}
```

اگر برنامه‌ی خود را اجرا کنید و ابتدا بر روی Cut کلیک کنید، متن شما Cut خواهد شد. اگر یک‌بار بر روی Undo کلیک کنید، متن شما باز خواهد گشت. اگر یک بار دیگر Undo کنید، مجدداً متن شما Cut می‌شود. این بدلیل این هست که شما بعد از دستور Undo، مراحل قبلی را پاک نکردید، برای این کار، کد زیر را جایگزین کنید:

```
if (textBox1.CanUndo == true)
{
    textBox1.Undo();
    textBox1.ClearUndo();
}
```

کد بالا را اجرا کنید؛ خواهید دید که با دوبار Undo کردن، عملیات Cut مجدداً انجام نخواهد شد.

حال وقت آن رسید که دو گزینه Copy و Past را نیز راه‌اندازی کنیم. پس ابتدا بر روی گزینه Copy دابل-کلیک کنید تا وارد بخش کدنویسی آن شویم. کدی که برای کپی استفاده خواهیم کرد، به شکل زیر است:

```
textBox1.Copy();
```

اما برای بررسی اینکه آیا متنی برای کپی کردن انتخاب شده است، باید به شکل زیر عمل کنیم:

```
if (textBox1.SelectionLength > 0)
{
    textBox1.Copy();
}
```

ما در واقع در حال بررسی مقدار `SelectionLength` هستیم که بینیم چه تعداد کاراکتر انتخاب شده است. اگر که برابر با صفر باشد، یعنی هیچ کاراکتری برای کپی شد انتخاب نشده است، پس وارد بدنه `IF` نمی‌شود؛ در غیراینصورت، کدهای داخل بدنه `IF` اجرا خواهند شد.

برای `Past` کردن، از `Text Box` دومی استفاده خواهیم کرد. پس کد زیر را برای گزینه `Past` اضافه کنید:

```
textBox2.Past();
```

**نویسنده: سید جعفر موسوی (Jeff Mosawy)**

**منبع مقاله: مجله گویا آی تی**

**لینک ثابت مقاله (دسترسی آنلاین):** <http://www.gooyait.com/1391/12/16/c-sharp-tutorial-part-25.html>

**با آرزوی موفقیت – GOOYAIT.COM**