

فرم

❖ همانطور که گفته شد با استفاده از تگ `<form>`، یک فرم در صفحه `html` ایجاد می گردد و شما می توانید کنترل‌های خود را برای گرفتن اطلاعات از کاربر درون آن تعریف کنید.

```
<form method="get/post" action="test.asp">
```

```
</form>
```

تفاوت متد Get و متد Post

❖ دو راه برای انتقال اطلاعات از یک فرم در سمت **client** به سمت سرور وجود دارد:

❖ اولین روش استفاده از **querystring** می باشد. **Querystring** اطلاعات اضافی است که به انتهای **url** می چسبد و به سرور ارسال می شود. برای انتقال به این روش مقدار خصوصیت **Method** را باید برابر با **get** قرار داد.

❖ شروع یک **querystring** با یک علامت سوال می باشد که بدنبال آن تعدادی زوج نام و مقدار آورده می شود. وقتی تعداد زوج نام و مقادیرها بیش از یکی باشد آنگاه با **&** از یکدیگر جدا میشوند.

❖ زمانی که تعداد فیلدها کم و نمایش داده شدن مقادیر وارد شده بر روی **address bar** برای انتقال اهمیت نداشته باشد از این روش استفاده می کنیم.

تفاوت متد Get و متد Post (ادامه)

❖ متد دیگر `post` می باشد که اطلاعات کاربر را با استفاده نکردن از `querystring` پنهان می کند.

❖ زمانی که تعداد فیلدها زیاد و نمایش ندادن آنها در `address bar` اهمیت دارد از این متد استفاده می شود.

خصوصیت action

❖ این خصوصیت آدرس محل ذخیره اسکریپت **server side** است که قرار است اطلاعات فرستاده شده از سمت کاربر را گرفته و پردازشهایی بر روی آنها انجام دهد. (مثلاً ذخیره کردن در DB)

❖ همچنین از این خصوصیت می توان استفاده کرد تا اطلاعات درون فرم را به یک آدرس E-mail ارسال کرد.

خصوصیت action در فرمها و دکمه ارسال

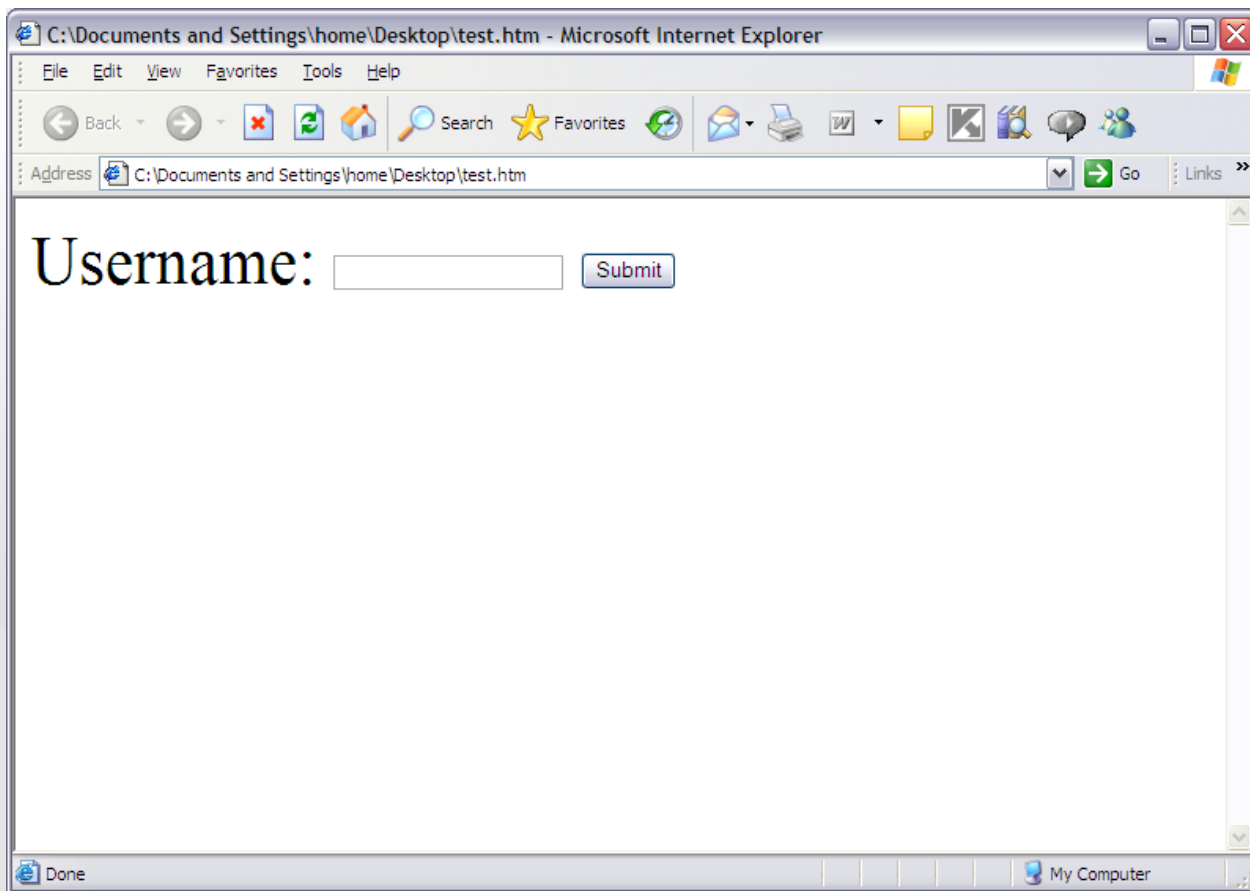
❖ وقتی که کاربر روی دکمه **Submit** کلیک می کند، محتویات فرم به فایل یا آدرس دیگری فرستاده خواهد شد. مشخصه **action**، نام آن فایل یا آدرس را مشخص می کند، آن فایل معمولاً اطلاعات وارد شده را دریافت کرده و پس از پردازش یک خروجی را بر می گرداند.

خصوصیت action در فرمها و دکمه ارسال (ادامه)

مثال ❖

```
<html>
<body>
<form name="input" action="action.asp" method="get">
Username:
<input type="text" name="user">
<input type="submit" value="Submit">
</form>
</body>
</html>
```

خصوصیت action در فرمها و دکمه ارسال (ادامه)



خروجی ❖

فرم حاشیه دار

❖ توسط برچسب **<fieldset>** شما می توانید یک حاشیه دور عناصر فرم خود ایجاد کنید و توسط برچسب **<legend>** می توانید کلماتی را روی قسمت بالایی حاشیه بنویسید.

فرم حاشیه دار (ادامه)

مثال ❖

```
<html>
<body>
<fieldset>
<legend>Health information:</legend>
<form>
Height <input type="text" size="3">
Weight <input type="text" size="3">
</form>
</fieldset>
</body>
</html>
```

فرم حاشیه دار (ادامه)

C:\Documents and Settings\home\Desktop\test.htm - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites Home Page Print Stop

Address <C:\Documents and Settings\home\Desktop\test.htm> Go Links >>

Health information:

Height Weight

Done My Computer

خروجی

قابها (frame)

❖ با استفاده از قاب ها می توان چندید فایل html را بطور همزمان بر روی صفحه مرورگر نمایش داد. برای اینکار از دو برچسب زیر استفاده می شود:

❖ برچسب `<frameset>` : این برچسب مشخص می کند که پنجره مرورگر چگونه بین قاب ها تقسیم شود.

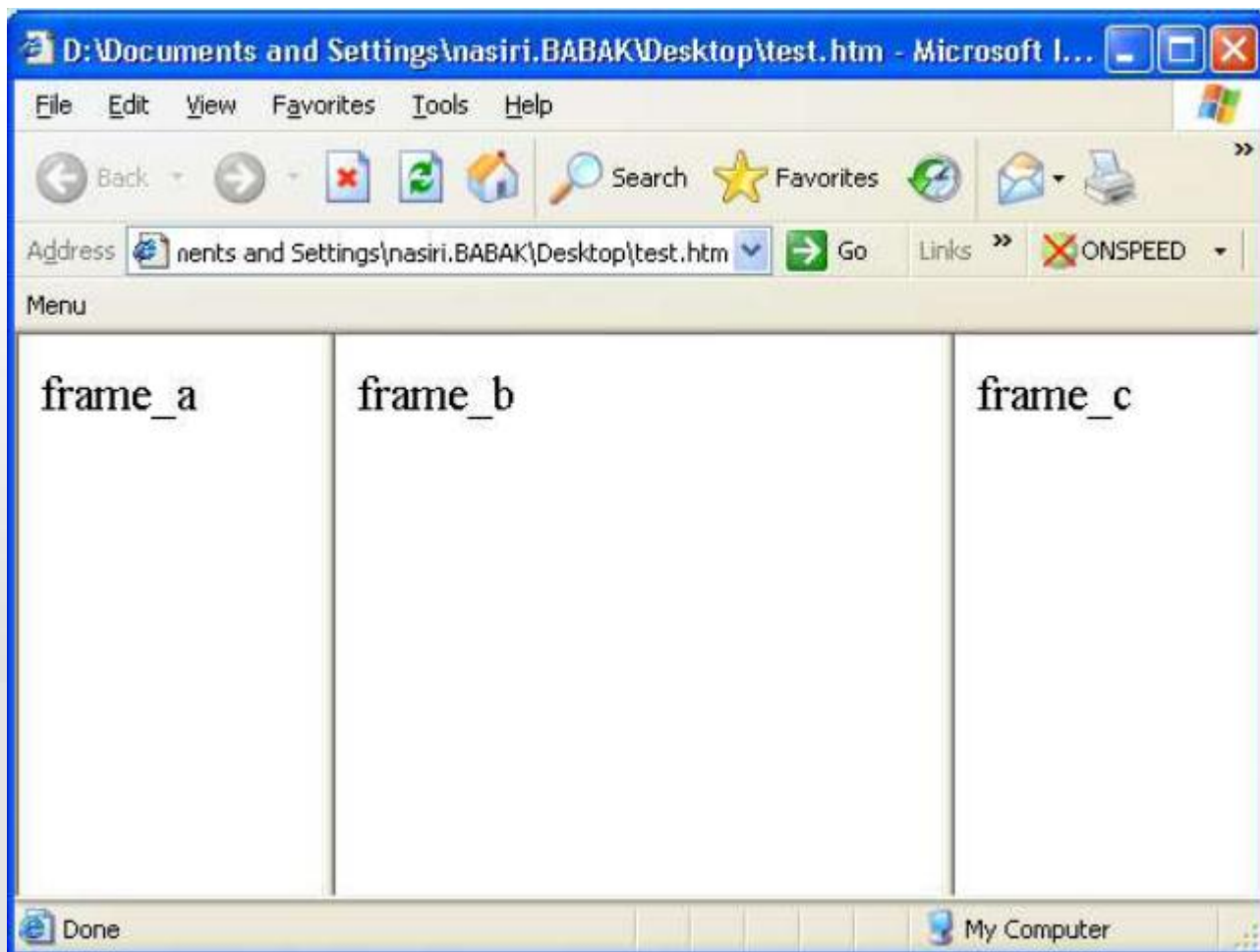
❖ برچسب `<frame>` : این برچسب مشخص کننده قاب ها یا همان صفحات html می باشد.

قابها (frame) (ادامه)

❖ مثال : در این مثال صفحه به سه ستون تقسیم می شود. ستون اول و سوم ۲۵ درصد و ستون دوم ۵۰ درصد پهنای مرورگر را اشغال می کنند. فایل `farme_a.htm` در ستون اول، فایل `farme_b.htm` در ستون دوم، فایل `farme_c.htm` در ستون سوم قرار می گیرند.

```
<html>
<frameset cols="25%,50%,25%">
<frame src="frame_a.htm">
<frame src="frame_b.htm">
<frame src="frame_c.htm">
</frameset>
</html>
```

قابها (frame) (ادامه)



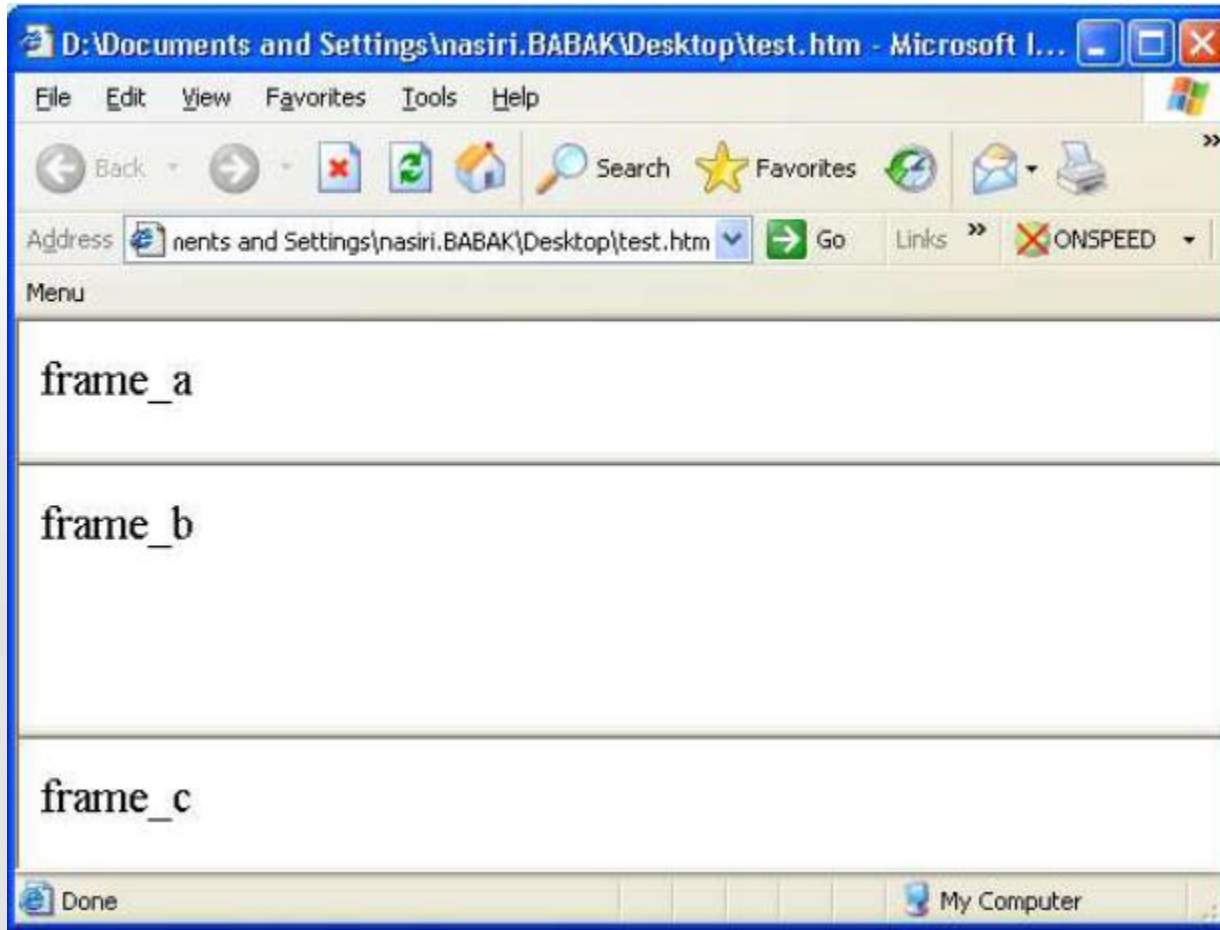
خروجی ❖

قابها (frame) (ادامه)

❖ مثال :

```
<html>  
<frameset rows="25%,50%,25%">  
<frame src="frame_a.htm">  
<frame src="frame_b.htm">  
<frame src="frame_c.htm">  
</frameset>  
</html>
```

قابها (frame) (ادامه)



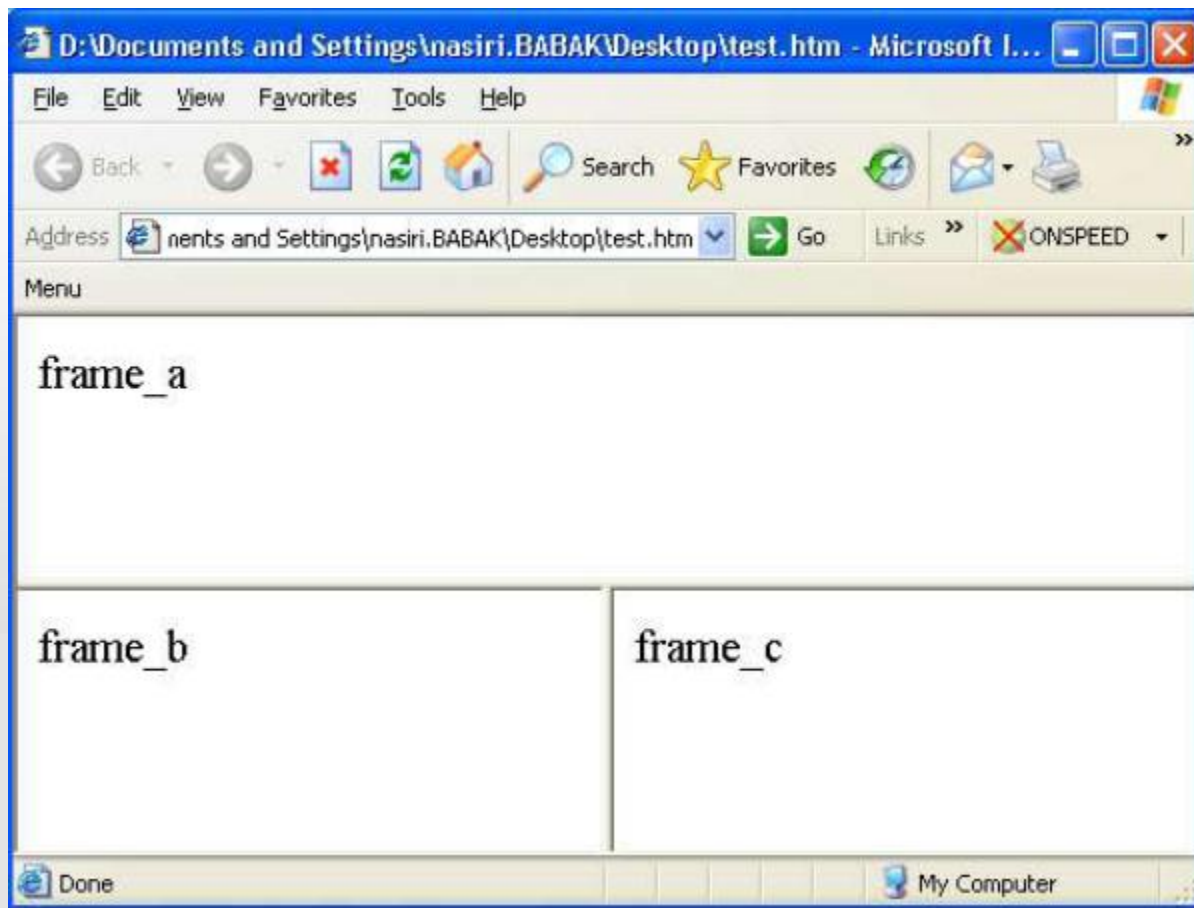
خروجی ❖

قابها (frame) (ادامه)

❖ مثال :

```
<html>
<frameset rows="50%,*">
<frame src="frame_a.htm">
<frameset cols="50%,*">
<frame src="frame_b.htm">
<frame src="frame_c.htm">
</frameset>
</frameset>
</html>
```


قابها (frame) (ادامه)



خروجی ❖

تگ span و div

- ❖ این دو تگ جزو تگهای بسیار پر کاربرد در هنگام کار با CSSها می باشد.
- ❖ معمولاً اکثر طراحان از این دو تگ به یک صورت استفاده می کنند.
- ❖ اما دارای تفاوت‌های کوچکی هستند.

تگ span

❖ تگ `` سبب تعریف محدوده ای مشخص می شود. تمامی عناصرهایی که دورن تگهای `` و `` قرار خواهند گرفت خواص آن را به ارث خواهند برد. خواصی از قبیل استیل (style).

❖ برخلاف تگ های `<p>` و `<div>` مرورگرها هیچ سطر جدیدی به اول یا آخر عنصر `` اضافه نمی کنند.

تگ span (ادامه)

❖ در روش قدیمی :

```
<font color="#FF00FF">Hello!</font>
```

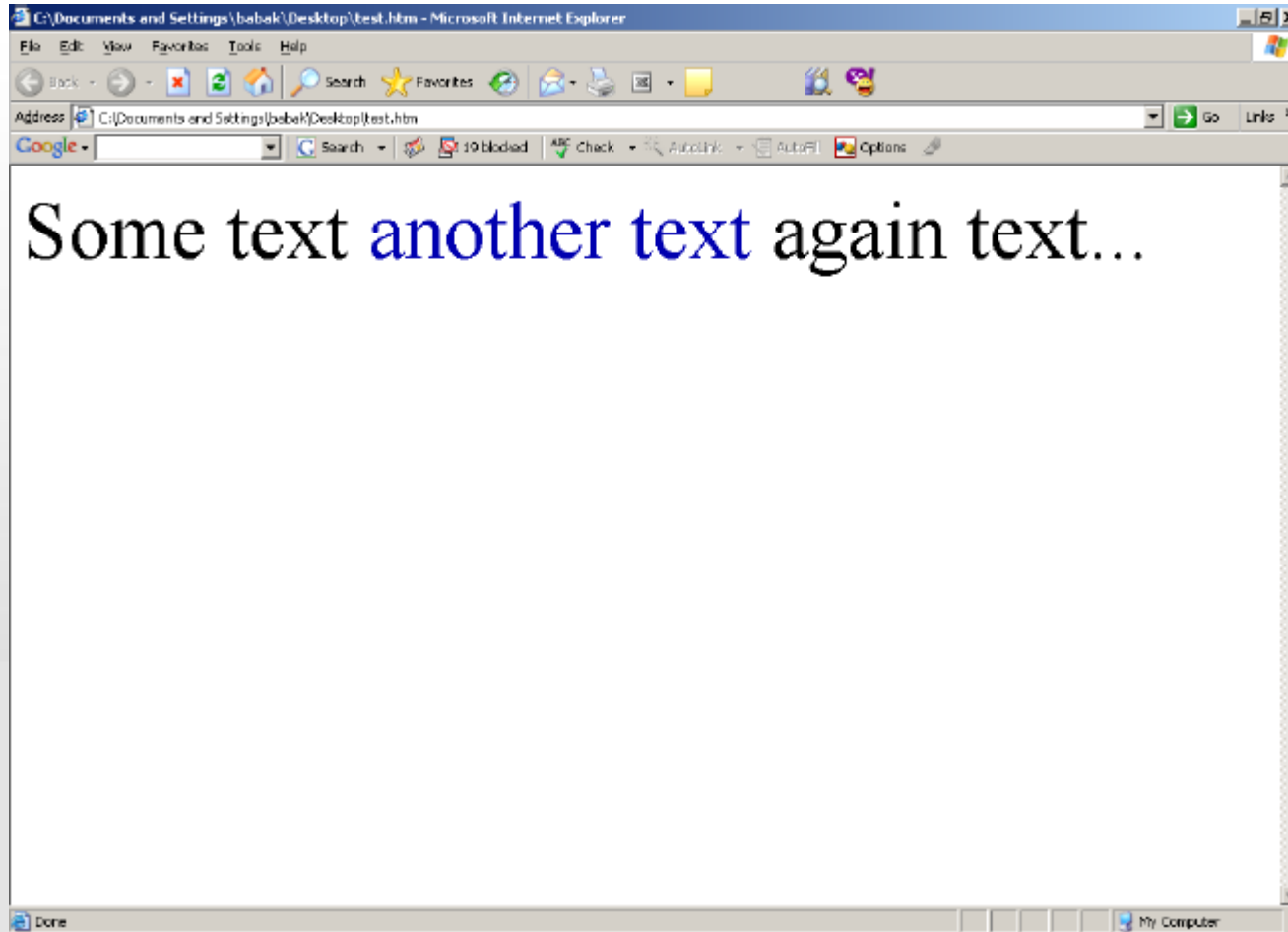
❖ در روش جدید :

```
<span style="color:#FF00FF">Hello!</span>
```

❖ مثالی دیگر:

```
<p>Some text  
<span style="color:#0000AF;">another text</span>  
again text...</p>
```

تگ span (ادامہ)



خروجی ❖

تگ div

- ❖ این تگ نیز سبب تعریف محدوده ای مشخص می شود و کاربرد آن شبیه به تگ `` است.
- ❖ برخلاف تگ `` مرورگرها یک سطر جدید به اول و آخر عنصر `div` اضافه می کنند. مانند عنصر `<p>`.

تگ div (ادامه)

مثال ❖

```
<p>Some text  
<div style="color:#0000AF;">  
another text  
</div>  
again text...  
</p>
```

تگ div (ادامہ)

خروجی ❖

