```
        window.alert("عدد وارد شده معتبر نیست");
    }
    </script>
```

بررسی کد:

- عدد n و رشته s از ورودی خوانده می‌شود،
- اگر n بزرگ‌تر از صفر باشد، یک‌بار روی صفحه نوشته می‌شود و سپس از مقدار n یک‌واحد کسر می‌شود و به همین ترتیب اجرای حلقه ادامه می‌یابد.
- چنان‌چه عدد صفر یا کمتر باشد، پیغام خطا برای کاربر نمایش داده می‌شود.

کدی بنویسید که اسامی افراد را به صورت تک‌تک خوانده و نهایتاً همه آن‌ها را روی صفحه نمایش دهد. نشانه اتمام اسامی، ورود رشته "end" است.

```
<script type="text/javascript">
var name="", s="";
name= window.prompt("یک نام را وارد نمایید");
do
{
    s = s + name + "<br/>";
    name= window.prompt("یک نام را وارد نمایید");
}
while (name!="end")
document.write(s);
</script>
```

بررسی کد:

- ابتدا متغیرهای name و s تعریف و با یک رشته خالی ("") مقداردهی می‌شوند چون اگر به آن‌ها مقدار اولیه داده نشود، مقدار پیش‌فرض undefined را می‌پذیرند.
- سپس یک نام از ورودی خوانده شده و حلقه شروع می‌شود.
- از s برای نگه‌داری کلیه اسامی وارد شده استفاده می‌شود و هر بار، نام وارد شده به مقدار فعلی s افزوده می‌شود و مجدداً در s ذخیره می‌گردد.

از آن‌جا که شـرط اتمام حلقه، وارد شـدن رشـته "end" اسـت بنابراین عمل خواندن نام را در انتهای حلقه انجام می‌دهیم تا شـرط بررسـی شـود و در صورت مخالف بودن با مقدار "end" کنترل برنامه به ابتدای حلقه منتقل شود.

۳-۲-۲۰ دستور for

در مثال ۳ دستور do…while مشاهده کردید که می‌توان از این دستور برای عملیات‌های تکراری که تعداد دفعات آن‌ها ممکن است بسته به شرایط تغییر کند استفاده کرد؛ چرا که معلوم نبود کاربر قصد دارد چند نام وارد کند.

در این میان از دستور for برای تکرار یک عملیات به تعداد دفعات مشخص استفاده می‌گردد و نگارشی به صورت زیر دارد:

```
for(a;b;c)
{
    execute code
}
```

a: در این بخش از دستور، مقداردهی اولیه شمارنده انجام می‌شود.

b: در این قسـمت، شـرطی قرار داده می‌شود که اگر نتیجه آن درست باشد، حلقه ادامه پیدا می‌کند و در غیراین‌صورت، به اتمام می‌رسد.

c: هر بار که اجرای دستورات حلقه به پایان می‌رسد، دستور موجود در این بخش نیز به اجرا در می‌آید. عمدتاً دستور تغییر شمارنده در این بخش صورت می‌گیرد.

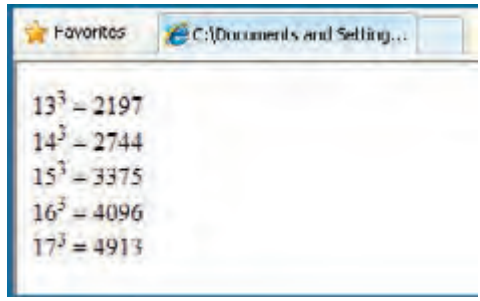
کدی بنویسید که اعداد ۱ تا ۱۰ را روی صفحه بنویسد.

```
<script type="text/javascript">
for (i=1;i<=10;++i)
{
    document.write(i + "<br/>");
}
</script>
```

مثال Example

کدی بنویسید که توان سوم اعداد بین ۱۳ تا ۱۷ را نشان دهد.

```
<script type="text/javascript">
for (i=13;i<=17;++i)
{
      document.write(i +"<sup>3</sup> = " + i*i*i + "<br/>");
}
</script>
```



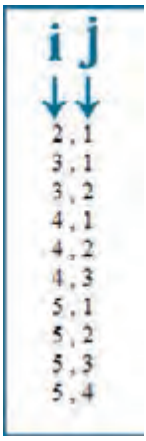
بررسی کد:

در این مثال با اســتفاده از برچسب که برای نمایش توان اعداد استفاده می‌شود، توان سوم شمارنده i (یعنی مقدار i*i*i) نمایش داده شده است.

مثال Example

خروجی کد زیر چیست؟

```
for (i=2;i<=5;++i)
{
      j=1;
      while (j<i)
      {
            document.write(i +" , " + j + "<br/>");
            ++j;
      }
}
```



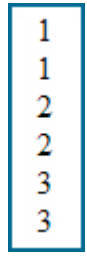
بررسی کد:

- در این کد، ابتدا متغیر i با عدد ۲ مقداردهی می‌شود.
- سپس کدهای درون حلقه for به اجرا درمی‌آید. لذا j با عدد ۱ مقداردهی شده و به دلیل درست بودن شرط j<i عبارت 2,1 روی صفحه نوشته می‌شود.
- در ادامه مقدار j به ۲ افزایش می‌یابد و از آن‌جا که دیگر شـرط i<j صدق نمی‌کند، حلقه while پایان یافته و کنترل به for منتقل می‌گردد.
- ایـن بـار i یک‌واحد افزایش یافتـه و مراحل قبل تکرار می‌گردد. این روش اسـتفاده از حلقه‌ها، کاربرد حلقه‌های تودرتو نام دارد.

نتیجه اجرای عبارت کد زیر چیسـت؟ اگر علامت‌های { } برداشـته شـوند چه تغییری در نتیجه حاصل می‌شود؟

```
for (i=1;i<=3;++i)
{
document.write(i+ "<br/>");
document.write(i+ "<br/>");
}
```

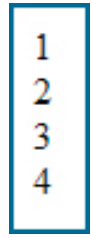
بـا جـرای کد فوق نتیجه زیر در خروجی ظاهر می‌شـود چون متغیر i از ۱ تـا ۳ تغییر می‌کند و در هر بار اجرای حلقه دوبار نوشته می‌شود.



اما وقتی آکولادها برداشته می‌شود و کد به صورت زیر درمی‌آید:

```
for (i=1;i<=3;++i)
document.write(i+ "<br/>");
document.write(i+ "<br/>");
```

صرفاً اولین دسـتور write جزو دستورات حلقه محسوب می‌شود (مانند آن‌چه در مورد if مشاهده کردید) و لذا خروجی به صورت زیر خواهد بود.



اعداد ۱ تا ۳ توسـط دسـتور write اول و عدد ۴ توسط دسـتور write دوم نوشته شده است. این قاعده در مورد همه دستوراتی که کدهای آن‌ها درون آکولاد قرار می‌گیرند مانند while هم صادق است.

با اجرای کد زیر چند بار کلمه JavaScript روی صفحه نوشته می‌شود؟

```
<script type="text/javascript">
for (i=1;i<=4;++i)
for (j=1;j<=3;++j)
document.write("JavaScript <br/>");
</script>
```

۱۲ بار! چون حلقه اول چهار بار و حلقه دوم در هر بار اجرای حلقه اول، سه بار به اجرا در می‌آید. در این مثال هم for دوم تنها دستورِ for محسوب می‌شود و نیازی به گذاشتن آکولاد نیست.

۴-۲-۲۰ دستور break

دستور break برای خروج از حلقه تکرار کاربرد دارد؛ یعنی درون حلقه‌ای مانند while چنان‌چه دستور break اجرا شود، صرف‌نظر از مقادیر و درستی یا نادرستی شروط، کنترل برنامه به خارج از حلقه منتقل می‌شود. کاربرد دیگر این دستور، همان‌گونه که در بخش‌های قبل مشاهده کردید، پایان دادن به بررسی شرط‌ها در دستور switch بود.

خروجی کد زیر چیست؟

```
<script type="text/javascript">
var i=0;
while (true)
{
    document.write(i++ + "<br/>");
    if (i==5)
    break;
}
</script>
```

اعداد صفر تا چهار روی صفحه نوشته می‌شود و وقتی مقدار i برابر با ۵ شد، دستور break اجرا و حلقه به پایان می‌رسد.

کدی بنویسید که اسامی افراد را به صورت تک‌تک خوانده و نهایتاً همه آن‌ها را روی صفحه نمایش دهد. نشانه اتمام اسامی، ورود رشته "end" است.

```
<script type="text/javascript">
var s="" , name="";
while (true)
{
```

```
        name= window.prompt("نام موردنظر را وارد کنید");
        if (name=="end")
        break;
        s= s + name + "<br/>";
    }
    document.write(s);
    </script>
```

کدی بنویسید که تعدادی عدد بزرگ‌تر از صفر را از ورودی خوانده و تعداد اعداد زوج را نمایش دهد. ورود عدد صفر نشان‌دهنده اتمام ورود اعداد است.

```
<script type="text/javascript">
var cnt=0 , n;
while (true)
{
    n= window.prompt("عدد موردنظر را وارد کنید");
    if (n==0)
    break;
    if (n%2==0)
    cnt++;
}
document.write("تعداد اعداد زوج وارد شده: " + cnt);
</script>
```

بررسی کد:

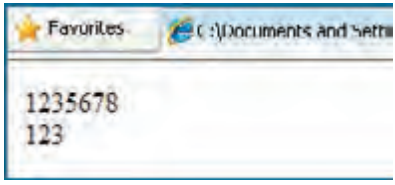
- ابتدا شمارنده cnt با صفر مقداردهی می‌شود.
- شـــرط حلقه while همواره true اســـت بنابراین تکرار حلقه آن‌قدر ادامه می‌یابد تا دستور break; اجرا شود.
- در هر بار اجرای حلقه، عدد n خوانده می‌شود؛ اگر این عدد برابر با صفر بود، حلقه خاتمه می‌یابد.
- اگر n زوج باشـــد، یک‌واحد به مقدار شمارنده افزوده می‌شود. نهایتاً، متد write مقدار شمارنده را روی صفحه درج خواهد کرد.

۵-۲-۲۰ دستور continue

وجود این دستور درون حلقه، باعث می‌شود دستورات پس از آن نادیده گرفته شده و کنترل برنامه به ابتدای حلقه منتقل شود. در این حالت، مانند روند طبیعی اجرای حلقه، شرط مورد بررسی قرار می‌گیرد و در صورت صحیح بودن، اجرای حلقه ادامه پیدا می‌کند؛ در غیراین‌صورت، اجرای حلقه پایان می‌پذیرد.

خروجی کد زیر چیست؟

```
<script type="text/javascript">
for (i=1;i<=8;i++)
{
      if (i==4)
      continue;
      document.write(i);
}
document.write("<br/>");
for (i=1;i<=8;i++)
{
      if (i==4)
      break;
      document.write(i);
}
</script>
```



بررسی کد:

در حلقه for اول، اعداد ۱ تا ۸ شمارش شده و روی صفحه نوشته می‌شوند، فقط وقتی شمارنده برابر با ۴ می‌شود، دستور continue اجرا شده و در نتیجه کنترل برنامه به ابتدای حلقه منتقل می‌شود، لذا عدد ۴ نوشته نخواهد شد.

در حلقه for دوم، اعداد ۱ تا ۴ شمارش می‌شوند، اما وقتی مقدار شمارنده برابر با ۴ می‌شود، اجرای حلقه متوقف می‌گردد لذا فقط اعداد ۱تا ۳ روی صفحه درج خواهند شد.

فصل

20

کنترل روند اجرای برنامه

چکیده‌ی فصل

- در زبان‌های برنامه‌نویسی و اسکریپت‌نویسی برای کنترل روند برنامه از «عبارت‌های شرطی» و نیز «حلقه‌های تکرار» استفاده می‌شود.
- برای مدیریت حالت‌های شرطی از دستورات if ، if … else و switch استفاده می‌گردد.
- برای انجام عملیات‌های تکراری در جاوا اسکریپت از حلقه‌های while، do…while و for استفاده می‌شود.
- می توانیم اجرای توابع را مشروط به وقوع رویدادی از عناصر صفحه کنیم؛ برای نمونه رویداد onClick عنصر دکمه.
- دستور break باعث اتمام حلقه می‌شود.
- دستورات بعد از contiune نادیده گرفته می‌شوند و کنترل برنامه به ابتدای حلقه منتقل می‌شود.

پرسش‌ها و تمرین‌ها

۱. صفحه‌ای حاوی یک دکمه ایجاد کنید تا کاربر با کلیک روی آن، اعداد فرد دو رقمی را مشاهده کند.

۲. کدی بنویسید که تعدادی عددِ بزرگ‌تر از صفر را از کاربر دریافت نموده و میانگین آن‌ها را روی صفحه بنویسد. ورود عدد صفر نشانه اتمام اعداد است.

۳. خروجی کد زیر چیست؟

```
<script type="text/javascript">
var i=5;
while (--i>1)
document.write(i + "<br/>");
</script>
```

۴. کد زیر چه نتایجی را برمی‌گرداند؟

```
<script type="text/javascript">
for (i=0;i<=25;i=i+2)
document.write(i + “<br>”);
</script>
```

۵. کدی بنویسید که با استفاده از حلقه‌های تودرتو شکل زیر را روی صفحه نمایش دهد.

```
*
**
***
****
*****
```

فصل بیست و یکم

کار با
اشیاء جاوا اسکریپت

پس از مطالعه این فصل از فراگیر انتظار می‌رود:

۱. با ماهیت اشیاء پیش‌ساخته در جاوا اسکریپت آشنا شود.

۲. روش دستیابی به خصوصیت‌ها و استفاده از متدها را فرا بگیرد.

۳. توانایی کار با اشیاء متداول را کسب نماید.

« مطالعه آزاد »

کلیات

همان‌گونه که در فصل‌های پیشـــین این کتاب توضیح داده شـــد، جاوا اسکریپت یک زبان اسکریپت‌نویسی شـــیءگرا محسوب می‌شـــود و با بهره‌گیری از این خاصیت می‌توانید اشـــیاء موردنظر را ایجاد و در طول برنامه از آن‌ها اســـتفاده کنید. شـــیئی که در جاوا اســـکریپت می‌سازید در واقع نوعی از داده اســـت که مطابق با نیاز خود ایجاد کرده‌اید و در ســـاماندهی اطلاعات و پردازش آن‌ها نقش تعیین‌کننده‌ای دارد. علاوه بر این در جاوا اسکریپت تعدادی شیء پیش‌ساخته[1] وجود دارد که در این فصل با روش استفاده از آن‌ها آشنا خواهید شد.

۱-۲۱ تعریف یک شیء (مطالعه آزاد)

پیش از آن‌که با اشـــیاء پیش‌ساخته جاوا اسکریپت آشنا شوید بهتر است در عمل، روش ایجاد و استفاده از یک شیء را فرابگیرید تا چنان‌چه ابهامی پیرامون این مفهوم در ذهن شما وجود دارد برطرف شود.

فرض کنید در حال ایجاد برنامه‌ای برای یک فروشگاه هستیم که با اطلاعات کالاها و مشتریان کار می‌کند، بنابراین می‌توانیم در این برنامه، کلاس (مفهوم) مشتری را ایجاد و از روی آن اشیاء موردنظر را بسازیم. طبیعتاً هـــر شـــیء تعدادی خصوصیت (مانند نام، نـــام خانوادگی و ...) دارد که همان Propertyها هســـتند و نیز تعداد عملکرد یا متد (Method) خواهد داشت.

- برای ایجاد کلاس، از کلمه کلیدی function استفاده می‌شود.
- نام موردنظر برای کلاس را وارد کنید.
- مقادیر لازم برای مقداردهی به این شیء را درون پرانتز وارد نمایید.

1 . Built-in

- کلمه کلیدی this را تایپ و پس از آن یک نقطه درج کنید. حالا خصوصیت موردنظر برای شیء را وارد نمایید. سپس مقدار متناظر با این خصوصیت را که در تعریف کلاس قید شده به آن منتسب کنید. این کار را برای همه خصوصیت‌های شیء تکرار نمایید.
- اکنون نوبت به متدهای شیء می‌رسد. کلمه کلیدی this را وارد و پس از درج نقطه، نام متد را وارد نمایید. حال نام متد را به این مقدار منتسب نمایید.
- حال متدهای تعریف شده برای کلاس را در خارج از آن پیاده‌سازی نمایید.

```
function customer(n,l,a)
{
    //properties
    this.name=n;
    this.lastname=l;
    this.age=a;

    //methods
    this.changeLastname=changeLastname;
    this.increaseAge=increaseAge;
}

function changeLastname(new_l)
{
    this.lastname=new_l;
}

function increaseAge()
{
    this.age= this.age +1;
}
```

در کد فوق، ابتدا کلاس مشتری تعریف شده که حاوی سه خصوصیت نام، نام خانوادگی و سن است. سپس دو متد برای آن پیاده‌سازی گردیده که changeLastname، نام خانوادگی جدیدی را به مشتری نسبت می‌دهد و increaseAge، سن وی را یک‌سال افزایش می‌دهد. اکنون باید از این کلاس تعریف شده، درون برنامه استفاده نماییم.

- برای ایجاد شیء از روی کلاس، کلمه کلید var و نام شیء نوشته می‌شود.
- پس از علامت انتساب، نام کلاس با مقادیر لازم برای مقداردهی به شیء ساخته شده درون پرانتز و به همان ترتیبی که در تعریف کلاس وجود داشت قید می‌گردد.

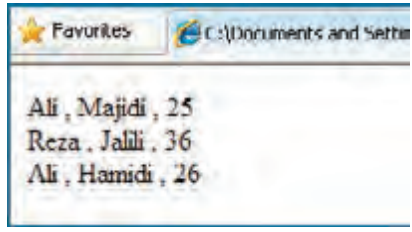
اکنون شیء، ساخته شده و می‌توان از خصوصیات یا متدهای آن استفاده کرد.

```
<body>
<script type=”text/javascript”>
var c1=new customer(“Ali”,”Majidi”,25);
var c2=new customer(“Reza”,”Jalili”,36);
document.write(c1.name + “ , “ + c1.lastname + “ , “ + c1.age );
document.write(“<br/>”);
document.write(c2.name + “ , “ + c2.lastname + “ , “ + c2.age );
c1.changeLastname(“Hamidi”);
c1.increaseAge();
document.write(“<br/>”);
document.write(c1.name + “ , “ + c1.lastname + “ , “ + c1.age );
</script>
</body>
```

در کد فوق، شیء c1 ساخته شده و خصوصیت‌های نام، نام خانوادگی و سن وی با مقادیر Ali، Majidi و 25 مقداردهی شده است. شیء c2 هم از روی کلاس مشتری ایجاد و با مقادیر Reza، Jalili و 36 مقداردهی گردیده است. اکنون می‌توانیم این مقادیر را خوانده و روی صفحه نمایش دهیم.

حال با فراخوانی متدهای تعریف شده برای کلاس مشتری قصد داریم مقادیر ذخیره شده در شیء c1 را تغییر دهیم. متد changeLastname("new name") نام جدید را جایگزین نام قبلی می‌کند و متد increase-Age باعث افزایش یک‌واحدی سن می‌شود. حال وقتی مجدداً خصوصیات شیء c1 را روی صفحه می‌نویسیم، متوجه تغییر مقادیر می‌شویم.

نتیجه اجرا مجموعه کدهای درج شده در این بخش به صورت زیر است:



۲۱-۲ اشیاء پیش‌ساخته جاوا اسکریپت

در جاوا اسکریپت تعدادی شیء پیش‌ساخته وجود دارد که استفاده از آن‌ها باعث تسهیل در انجام عملیات موردنظر می‌شود. این اشیاء دارای تعدادی خصوصیت هستند که اطلاعاتی را در مورد داده‌ها در اختیار شما قرار

می‌دهند و ضمناً با اســتفاده از متدهــای آن‌ها می‌توانید عملیات موردنظر را به ســادگی انجام دهید. برای مثال همان‌گونه در فصول گذشــته مشــاهده کردیده استفاده از خصوصیت length شــیء String، طول رشته را برای شما محاسبه می‌کند. در صورت عدم استفاده از این خصوصیت مجبور هستید متد محاسبه طول رشته را شخصاً بنویسید که طبیعتاً کار زمان‌بری است.

۱-۲-۲۱ شیء String

این شیء برای ذخیره‌سازی عبارت‌های متنی، دست‌کاری آن‌ها و نیز استخراج اطلاعاتی در مورد آن‌ها استفاده می‌شود.

برای تعریف یک شیء String می‌توانید از نگارش زیر استفاده نمایید:

```
var txt = new String("My Text!");
```

یا به شکل ساده‌تری از عبارت زیر را به کار ببرید:

```
var txt = "My Text!";
```

مهم‌ترین خصوصیت این شیء، length است که طول رشته را برمی‌گرداند، بنابراین، نتیجه اجرا دستور زیر، درج عدد ۸ روی صفحه است.

```
document.write(txt.length);
```

پرکاربردترین متدهای این شیء را در جدول زیر مشاهده می‌کنید.

متد	کاربرد
charAt(index)	نویسه موجود در نمایه داده شده را برمی‌گرداند.
contact(str1,str2,…)	چند رشته را دریافت نموده و آن‌ها را به رشته جاری متصل می‌کند. عملکرد این متد همانند عملگر + است.
replace(substring,newstring)	در کل رشته، عبارت newstring را جایگزین substring می‌کند.
search(str)	عبارت str را درون رشته جستجو می‌کند و محل شروع آن را برمی‌گرداند. عدد ۱- نشانه عدم وجود عبارت در رشته است.
split()	رشته به مجموعه‌ای از زیررشته‌ها تفکیک می‌کند.
substr(start,length)	بخشی از رشته را برمی‌گرداند که از نمایه start شروع شده و به طول length ادامه می‌یابد.
toLowerCase()	رشته را به حروف کوچک تبدیل می‌کند.
toUpperCase()	رشته را به حروف بزرگ تبدیل می‌کند.

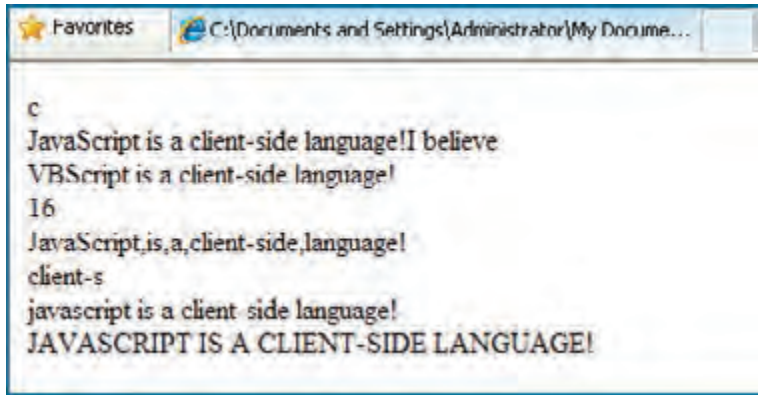
مثال Example

کد زیر چه مقادیری را برمی‌گرداند؟

```
<script type="text/javascript">
var str1=new String("JavaScript is a client-side language!");
var str2="I believe";
document.write(str1.charAt(5) + "<br/>");
document.write(str1.concat(str2,"<br/>"));
document.write(str1.replace("JavaScript","VBScript") + "<br/>");
document.write(str1.search("client") + "<br/>");
document.write(str1.split(" ") + "<br/>");
document.write(str1.substr(16,8) + "<br/>");
document.write(str1.toLowerCase() + "<br/>");
document.write(str1.toUpperCase() + "<br/>");
</script>
```

نکته

نمایه (index) نویسه‌ها از صفر شروع می‌شود، بنابراین در یک رشته، نویسه سوم دارای نمایه ۲ است.



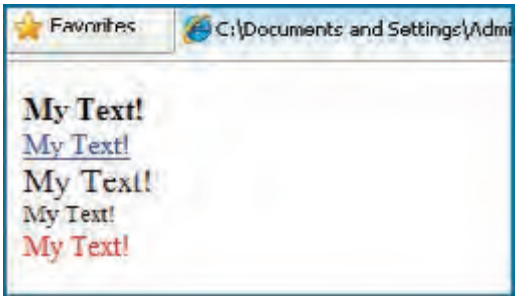
برای شـــیء String متدهای دیگری نیز تعریف شـــده که برای دســـت‌کاری ظاهر رشته‌ها در صفحات وب کاربرد دارند و همانند عناصر HTML عمل می‌کنند. تعدادی از این متدها را درون جدول بعد مشاهده می‌کنید.

متد	کاربرد
big()	اندازه فونت را بزرگ می‌کند.
blink()	متن را در حالت چشمک‌زن نمایش می‌دهد (در مرورگرهای IE، Chrome و Safari) عمل نمی‌کند.
bold()	متن را به صورت توپر نمایش می‌دهد.
fontcolor()	متن را با رنگ مشخص شده نشان می‌دهد.
Italics()	متن را به صورت مایل نمایش می‌دهد.
link()	متن را به صورت پیوند در می‌آورد.
small()	اندازه فونت را کوچک می‌کند.

مثال Example

نتیجه اجرای کد زیر چیست؟

```
<script type="text/javascript">
var str1=new String("My Text!");
document.write(str1.bold() + "<br/>");
document.write(str1.link("http://www.google.com") +"<br/>");
document.write(str1.big() + "<br/>");
document.write(str1.small() + "<br/>");
document.write(str1.fontcolor("red") + "<br/>");
</script>
```



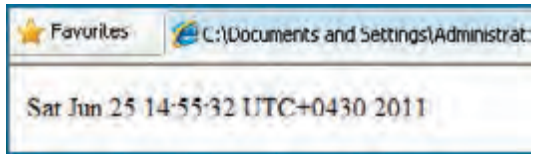
۲-۲-۲۱ شیء Date

با استفاده از این شیء می‌توانید زمان و تاریخ جاری را به دست آورده و عملیات موردنظر روی آن را پیاده‌سازی کنید. برای استخراج زمان و تاریخ جاری و قرار دادن آن درون یک شیء متغیر از دستور زیر استفاده می‌شود:

```
var now=new Date();
```

کدی بنویسد که زمان و تاریخ جاری را روی صفحه نمایش دهد.

```
var now=new Date();
document.write(now);
```



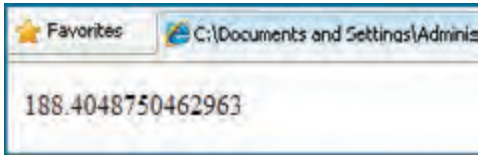
نحوه نمایش زمان و تاریخ به ما یادآوری می‌کند که برای استخراج زمان در قالب موردنظر باید از متدهای پیش‌بینی شده برای این شیء استفاده نماییم. متدهای پرکاربرد را در این جدول مشاهده می‌کنید.

متد	کاربرد
getDate()	شماره روز را در ماه برمی‌گرداند که عددی بین ۱ تا ۳۱ است.
getDay()	شماره روز را در هفته برمی‌گرداند که عددی بین ۰ تا ۶ است.
getMonth()	شماره ماه را به صورت عددی بین ۰ تا ۱۱ برمی‌گرداند.
getFullYear()	سال را به صورت چهاررقمی از تاریخ استخراج می‌کند.
getSeconds()	ثانیه را از شیء Date جدا می‌کند.
getMinutes()	دقیقه را از شیء Date استخراج می‌کند.
getHours()	عددی بین ۰ تا ۲۳ را به عنوان ساعت برمی‌گرداند.
getTime()	زمان سپری شده از روز ۱۹۷۰/۱/۱ را برحسب میلی‌ثانیه نمایش می‌دهد.

مثال Example

کدی بنویسید که نشان دهد چند روز تا پایان سال ۲۰۱۱ میلادی مانده است.

```
<script type="text/javascript">
var now=new Date();
var endOfYear=new Date(2011,11,31);
var msRemaining= endOfYear - now;
var daysRemaining = msRemaining/86400000;
document.write(daysRemaining);
</script>
```



بررسی کد:

- زمان جاری با ایجاد یک نمونه از شیء Date در متغیر now ذخیره می‌شود.
- انتهای سال میلادی به عنوان مقادیر اولیه برای شیء Date فرستاده می‌شود تا شیئی با زمان موردنظر ایجاد و در متغیر endOfYear ذخیره شود.
- زمان جاری از متغیر حاوی انتهای سال کسر می‌گردد و نتیجه درون متغیر msRemaining ذخیره می‌گردد. این مقدار برحسب میلی‌ثانیه است و باید آن را تبدیل کرد.
- با تقسیم عدد فوق بر ۸۶۴۰۰۰۰۰ که حاصل‌ضرب ۱۰۰۰×۶۰×۶۰×۲۴ است می‌توان تعداد روزهای باقی‌مانده را استخراج کرد و نمایش داد.

مثال Example

کدی بنویسید که نشان دهد امروز چه روزی از هفته است.

```
<script type="text/javascript">
var now=new Date();
```

```
var dayNum=now.getDay();
var day;
switch (dayNum)
{
    case 0:
    day = "یکشنبه";
    break;
    case 1:
    day = "دوشنبه";
    break;
    case 2:
    day = "سه شنبه";
    break;
    case 3:
    day = "چهارشنبه";
    break;
    case 4:
    day = "پنجشنبه";
    break;
    case 5:
    day = "جمعه";
    break;
    case 6:
    day = "شنبه";
    break;
}
document.write("امروز: " + day);
</script>
```

بررسی کد:

- متد ()getDay شماره روز را از تاریخ استخراج می‌کند.
- در تقویم میلادی، یکشنبه با عدد صفر نشان داده می‌شود.
- با استفاده از دستور switch، شماره روز بررسی شده و مقدار متناظر به متغیر day نسبت داده می‌شود.
- این مقدار توسط دستور write روی صفحه نوشته می‌شود.

مثال Example

کدی بنویسید که زمان جاری را نمایش دهد.

```
<script type="text/javascript">
var now=new Date();
document.write(now.getHours() + ":" + now.getMinutes() +":" + now.getSeconds());
</script>
```

در این مثال، زمانی که روی صفحه نمایش داده می‌شود، زمان بارگذاری صفحه است و برای مشاهده زمان فعلی باید صفحه را به صورت دستی، تازه‌سازی (Refresh) کرد. آیا راهی وجود دارد تا زمان به صورت خودکار روی صفحه تغییر کند؟ بله! کد زیر را امتحان کنید.

```
<html>
<head>
<script type="text/javascript">
function startTime()
{
    var today=new Date();
    var h=today.getHours();
    var m=today.getMinutes();
    var s=today.getSeconds();
    // اگر عدد کوچکتر از ۱۰ است یک صفر به ابتدای آن اضافه کن //
    m=checkTime(m);
    s=checkTime(s);
    document.getElementById('txt').innerHTML=h+":"+m+":"+s;
    t=setTimeout('startTime()',500);
}
function checkTime(i)
{
```

```
if (i<10)
  {
      i="0" + i;
  }
return i;
}
</script>
</head>

<body onload="startTime()">
<div id="txt"></div>
</body>
</html>
```

بررسی کد:

- در این کد تابعی به نام startTime تعریف شـده که همانند مثال قبل، زمان جاری را در قالب مناسـب استخراج می‌کند.
- از تابع checkTime برای تبدیل ثانیه‌ها و دقیقه‌های تک‌رقمی به دو رقمی استفاده می‌شود. این کار با افزودن یک صفر به سمت چپ آن‌ها انجام می‌شود.
- رویداد setTimeOut هر نیم ثانیه (۵۰۰ میلی‌ثانیه) یک‌بار تابع تولید زمان را فراخوانی می‌کند.
- درون صفحه، لایه‌ای با شناسه txt قرار داده شده و با استفاده از دستور .('document.getElementById('txt' innerHTML زمان درون آن درج می‌شود.
- تابع startTime در رویداد onload برچسب body (هنگام بارگذاری این برچسب) فراخوانی می‌شود.

در فصل‌های آینده مطالب بیش‌تری را در مورد رویدادها فراخواهید گرفت.

۳-۲-۲۱ شیء Math

با اسـتفاده از این شیء می‌توانید محاسـبات ریاضی موردنیاز را در زبان جاوا اسکریپت پیاده‌سازی نمایید. این شیء نیز همانند سایر اشیاء دارای تعدادی خصوصیت و تعدادی متد است که از میان خصوصیت‌ها می‌توان به PI اشاره نمود که عدد پی را برمی‌گرداند. متدهای پرکاربرد این شیء هم در جدول زیر فهرست شده‌اند.

متد	کاربرد
pow(x,y)	عدد x را به توان y می‌رساند و نتیجه را برمی‌گرداند.
random()	عددی تصادفی بین ۰ و ۱ را تولید می‌کند.
round(x)	x را به نزدیک‌ترین عدد صحیح گرد می‌کند.
floor(x)	x را به کوچک‌ترین عدد صحیح گرد می‌نماید.
ceil(x)	x را به بزرگ‌ترین عدد صحیح گرد می‌کند.
توابع مثلثاتی	این توابع عبارتند از sin، cos، tan، asin، acos، atan و ...

تابعی بنویسید که شعاع دایره را دریافت نموده و مساحت آن را برگرداند. سپس عددی را به عنوان شعاع دایره برای تابع ارسال نموده و نتیجه را روی صفحه بنویسید.

```
<html><head>
<script type="text/javascript">
function CalcArea(r)
{
      return r*r*Math.PI;
}
</script>
</head>
<body>
<script type="text/javascript">
document.write(CalcArea(2));
</script>
</body></html>
```

کدی بنویسید که یک عدد صحیح تصادفی بین ۰تا ۶ تولید نماید.

```
var n= Math.random()*7;
document.write(Math.floor(n);
```

بررسی کد:

- متد Math.random() یک عدد بین ۰ تا ۱ تولید می‌کند. توجه داشته باشید که هیچ‌گاه، اعداد صفر و یک توسط این تابع تولید نمی‌شوند.
- با ضرب عدد تولید شده در ۷، متغیر n حاوی عددی بین ۰ تا ۷ خواهد شد.
- متد floor، عدد تولید شده را به کوچک‌ترین عدد صحیح گرد می‌کند. بنابراین خروجی، یکی از اعداد ۰ تا ۶ خواهد بود.

کدی بنویسید که پنج عدد صحیح تصادفی بین ۰ و ۱۰۰۰ تولید نموده و روی صفحه نمایش دهد.

```
<script type="text/javascript">
var n;
for (i=1;i<=5;i++)
{
    n= Math.random()*1000;
    document.write(Math.round(n) + "<br/>");
}
</script>
```

بررسی کد:

- در هر بار اجرای حلقه، متد Math.random() یک عدد بین ۰ و ۱ تولید می‌کند.
- این عدد را در ۱۰۰۰ ضرب می‌کنیم تا در بازه ۰ تا ۱۰۰۰ قرار گیرد.
- با استفاده از متد Math.round() عدد تولید شده را گرد کرده و روی صفحه نمایش می‌دهیم.

کدی بنویسید که ۱۰ عدد صحیح تصادفی بین ۵۰۰ تا ۱۰۰۰ برگرداند.

```
var i=1,n;
while (i<=5)
{
    n=Math.round(Math.random()*1000);
    if (n<500)
```

```
        continue;
        document.write(n + "<br/>");
        i++;
    }
```

بررسی کد:

- شمارنده i با عدد یک مقداردهی می‌شود.
- همانند مثال قبل، عددی بین ۰ تا ۱۰۰۰ تولید می‌شود.
- اگر عدد تولید شده، کمتر از ۵۰۰ باشد، دستور continue اجرا می‌گردد یعنی از اجرای ادامه دستورات صرف‌نظر می‌شود. لذا عدد روی صفحه نوشته نمی‌شود و شمارنده هم اضافه نخواهد شد.
- اگر عدد تولید شـــده بیش‌تر از ۵۰۰ باشـــد، دستور continue اجرا نخواهد شد، بنابراین عدد تولید شده روی صفحه نوشته می‌شود و شمارنده i هم یک‌واحد افزایش خواهد یافت.

۴-۲-۲۱ شیء Array

شیء Array یا آرایه، متغیر ویژه‌ای است که در یک زمان می‌تواند چندین مقدار را در خود نگه‌داری کند. برای نمونه چنان‌چه لیستی از مقادیر داشته باشید می‌توانید آن‌ها را درون یک شیء Array ذخیره نموده و از طریق نمایه، به تک‌تک عناصر دسترسی داشته باشید. برای تعریف آرایه از نگارش زیر استفاده می‌شود:

```
var ar=new Array();
```

حتی هنگام تعریف آرایه می‌توانید اندازه آن را نیز مشخص کنید.

```
var ar=new Array(4)
```

حال آرایه ar در حافظه رایانه، شکلی شبیه به تصویر زیر دارد. اعدادی که در این تصویر مشاهده می‌کنید، نمایه عناصر موجود در آرایه هستند، برای نمونه، نمایه شماره ۲ به خانه سوم آرایه اشاره دارد.

برای قرار دادن مقادیر موردنظر در آرایه از نگارش زیر استفاده می‌شود که در آن، عدد i بیان‌گر نمایه خانه موردنظر است.

```
ar[i] = value;
```

بنابراین وقتی مقادیر زیر را به آرایه نسبت می‌دهید،

```
ar[0] = "بهار";
ar[1] = "تابستان";
```

```
ar[2] = "پاییز";
ar[3] = "زمستان";
```

آرایه در حافظه به صورت زیر در می‌آید.

بهار	تابستان	پاییز	زمستان
0	1	2	3

اکنون اگر دستور جاوا اسکریپت زیر اجرا شود:

```
document.write(ar[2]);
```

عبارت «پاییز» روی صفحه نوشته می‌شود. چنان‌چه نام آرایه را هم برای متد write ارسال کنید، تمامی عناصر آرایه روی صفحه نوشته می‌شوند.

```
document.write(ar);
```

بهار,تابستان,پاییز,زمستان

برنامه‌ای بنویسید که ۵ عدد را از ورودی دریافت نموده و آن‌ها را از آخر به اول روی صفحه بنویسد. برای نمونه اگر کاربر به ترتیب اعداد ۳۰،۱۱،۲۵،۲۰ و ۴۰ را وارد کرده است، عبارت 30-11-25-20-40 روی صفحه نوشته شود.

```
<script type="text/javascript">
var ar = new Array(5);
for (i=0;i<=4;i++)
ar[i] = window.prompt("عدد موردنظر را وارد کنید","۰" );
for (i=4;i>=0;i--)
document.write(ar[i] + "-");
</script>
```

بررسی کد:

- ابتدا یک آرایه با ۵ خانه ایجاد می‌شود.
- حلقه اول، اعداد را از ورودی خوانده و درون آرایه قرار می‌دهد. دقت داشته باشید که نمایه آرایه از صفر شروع می‌شود.

- حلقه دوم، اعداد وارد شده را از انتهای آرایه به ابتدای آن خوانده و نمایش می‌دهد.

کدی بنویسید که نشان دهد امروز، چه روزی از هفته است.

```
<script type="text/javascript">
var ar=new Array("یکشنبه","دوشنبه","سه‌شنبه","چهارشنبه","پنج‌شنبه","جمعه","شنبه");
var now=new Date();
var dayNum=now.getDay();
document.write("امروز: " + ar[dayNum]);
</script>
```

بررسی کد:

- آرایه‌ای حاوی روزهای هفته ایجاد می‌شود و نام هر روز درون خانه با نمایه متناظر قرار داده می‌شود. در این مثال با روش دیگری برای تعریف آرایه و مقداردهی اولیه به آن آشنا شدید.
- روز هفته توسط تابع getDay() استخراج می‌شود و نام روز متناظر روی صفحه نوشته می‌شود. قبلاً این مثال را با استفاده از دستور switch نوشته بودیم که طبیعتاً کد فوق ساده‌تر و کوتاه‌تر است.

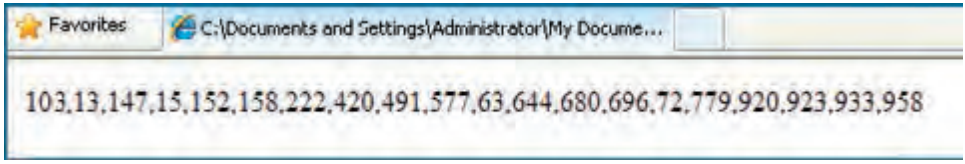
شیء Array دارای تعدادی خصوصیت و متد است که در این میان، خصوصیت length و متد sort() کاربرد بیش‌تر دارند. خصوصیت length، تعداد عناصر آرایه را برمی‌گرداند و متد sort() عناصر درون آرایه را مرتب می‌سازد.

کدی بنویسید که ۲۰ عدد تصادفی بین ۰ تا ۱۰۰۰ تولید نموده و درون آرایه قرار دهد. سپس آرایه مرتب شده را روی صفحه نمایش دهد.

```
<script type="text/javascript">
var arr=new Array(20)
for (i=0;i<=19;i++)
arr[i]=Math.round(Math.random()*1000);
arr.sort();
```

```
document.write(arr);
</script>
```

اگر خروجی کد فوق را مشاهده کنید، متوجه یک مشکل خواهید شد

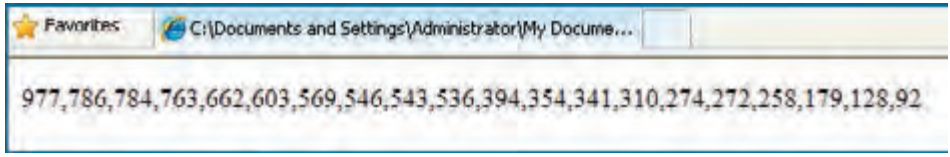


عـددی مثـل ۱۰۳ قبل از ۱۳ قرار گرفته و این نکته نشـان می‌دهد که متد sort برای مرتب‌سـازی اعداد همانند مرتب‌سازی رشته‌ها عمل کرده و همان‌طور که bac قبل از bc قرار می‌گیرد عدد ۱۰۳ را هم قبل از ۱۳ قرار داده است. برای حل این مشکل، تابع زیر را درون پرانتز متد sort قرار دهید:

```
function(a,b){return a - b}
```

این متد وظیفه مقایسه اعداد را برعهده دارد. چنان‌چه به جای مقدار a-b مقدار b-a را قرار دهید مرتب‌سازی به صورت نزولی انجام خواهد شد.

```
var arr=new Array(20)
for (i=0;i<=19;i++)
arr[i]=Math.round(Math.random()*1000);
arr.sort(function(a,b){return b - a});
document.write(arr);
</script>
```



چکیده‌ی فصل

- جاوا اسکریپت یک زبان شیءگرا محسوب می‌شود و در آن می‌توان مفاهیم موردنظر را به صورت کلاس تعریف نمود.
- به هر نمونه‌ای که از روی یک کلاس ساخته می‌شود، شیء گفته می‌شود. اشیاء می‌توانند تعدادی خصوصیت و متد داشته باشند.
- در جاوا اسکریپت تعدادی شیء پیش‌ساخته وجود دارد که می‌توان برای تسهیل انجام عملیات موردنظر از متدها و خصوصیت‌های آن‌ها استفاده نمود.
- از شیء String برای ذخیره‌سازی عبارت‌های متنی و دست‌کاری رشته‌ها استفاده می‌شود.
- شیء Date برای استخراج زمان و تاریخ کاربرد دارد.
- از شیء Math برای انجام محاسبات ریاضی استفاده می‌شود.
- شیء Array متغیر ویژه‌ای است که می‌تواند چندین مقدار را در خود نگه‌داری نماید.

پرسش‌ها و تمرین‌ها

۱. (اختیاری) کلاسی حاوی مشخصات یک دایره (مختصات مرکز و اندازه شعاع) ایجاد کنید. همچنین متدهایی برای محاسبه محیط و مساحت آن اضافه نمایید.

۲. کدی بنویسید که دو رشته از ورودی دریافت نماید و تشخیص دهد که آیا رشته دوم در رشته اول وجود دارد یا خیر.

۳. عملکرد متد ()blink را در مرورگر FireFox بررسی نمایید.

۴. کدی بنویسید که نام ماه میلادی جاری را بنویسید.

۵. کدی بنویسید که ۵۰ عدد صحیح تصادفی بین ۴۰۰ و ۶۰۰ تولید نماید.

۶. کدی بنویسید که تعدادی عدد صحیح غیرصفر را از ورودی بخواند. نشانه خاتمه اعداد، ورود صفر است. سپس تمامی اعداد و میانگین آن‌ها را نمایش دهد.

فصل بیست و دوم

فرم‌های تعاملی در جاوا اسکریپت

پس از مطالعه این فصل از فراگیر انتظار می‌رود:

۱. با فرم‌های تعاملی و نحوه کارکرد آن‌ها آشنا شود.

۲. روش استفاده از خصوصیات، متدها و رویدادهای عناصر فرم را فرابگیرد.

۳. با روش‌های موجود برای اعتبارسنجی فرم‌ها آشنا شود.

۴. از کدهای جاوا اسکریپت برای بهبود تعامل فرم با کاربر استفاده نماید.

« مطالعه آزاد »

کلیات

در فصل شـــانزدهم ایـــن کتاب با روش ایجاد فرم‌های HTML و جمع‌آوری داده‌ها از آن‌ها آشـــنا شـــدید. همچنین دانستید که در نرم‌افزار Dreamweaver با استفاده از کنترل‌های Spry می‌توانید تعامل فرم را با کاربر بهبود ببخشـــید تا کاربر بتواند به ازای مقادیر وارد شـــده در فرم، پیغام‌های مناسب را دریافت نماید. کنترل‌های Spry برای پیاده‌سازی قابلیت اعتبارسنجی در فرم‌های HTML، یک فایل خارجی جاوا اسکریپت به وب‌سایت اضافـــه می‌کننـــد. در این فصل قصد داریم ضمن آشـــنایی بـــا رویدادهای هر یک از عناصر فـــرم، کدهای جاوا اسکریپت موردنیاز برای اعتبارسنجی و بهبود عمل‌کرد فرم را شخصاً بنویسیم.

۲۲-۱ اعتبارسنجی فرم

هر فرم HTML از مجموعه‌ای از عناصر مانند کادرهای متنی، لیست‌های انتخاب، دکمه‌ها و ... تشکیل شده است که هر یک خصوصیات، متدها و رویدادهای خاص خود را دارند. برای تسلط بر روش اعتبارسنجی یک فرم و بهبود کارایی آن باید با متدها و رویدادهای هر یک از عناصر، روش دسترســـی به مقادیر آن‌ها و نیز متدها و رویدادهای فرم آشنا شوید. در این بخش، روش ایجاد یک فرم به صورت مرحله به مرحله توضیح داده می‌شود و در هر مرحله، چگونگی اعتبارسنجی کنترل افزوده شده به صفحه مورد بررسی قرار خواهد گرفت.

۲۲-۱-۱ ایجاد فرم

در این بخش قصد داریم فرمی ایجاد نماییم تا بازدیدکنندگان، برای عضویت در وب‌سایت آن را تکمیل و ارسال نمایند. در این فرم کاربر باید نام کاربری و رمزعبور، میزان تحصیلات و ، شهر محل سکونت را وارد نماید. به روشی که در فصل شانزدهم آموختید، یک فرم روی صفحه وب ایجاد نموده و مشخصه name آن را با frm1 مقداردهی کنید.

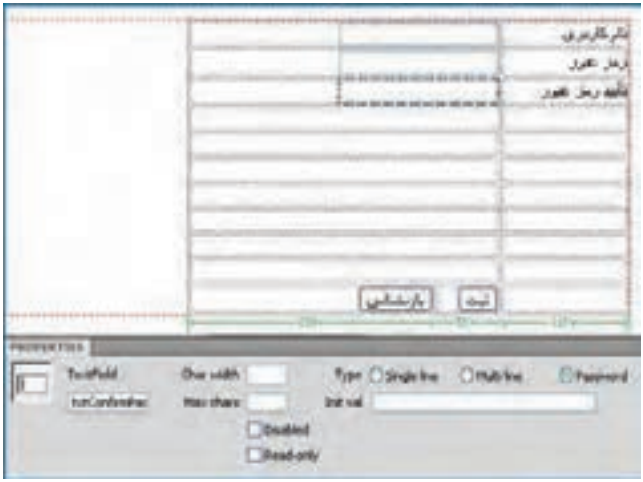
```
<form action="process.html" method="get" name="frm1">
</form>
```

مقادیر وارد شـده برای action و mathod فعلاً اهمیتـی ندارد چون روش دریافت و پردازش اطلاعات یک فرم در این فصل بررسی نخواهد شد. درون فرم، جدولی حاوی دو ستون و تعدادی سطر ایجاد نمایید تا عناصر فرم درون آن قرار گیرند.

همچنیـن در انتهـای فـرم، دکمه‌هـای Submit و Reset را به فـرم اضافه نموده و بـر روی آن‌ها به ترتیب عبارت‌های «ثبت» و «بازنشـانی» را بنویسـید. یادآوری می‌شود که وقتی دکمه‌ای را به صفحه اضافه می‌کنید به صورت پیش‌فرض از نوع submit است و باید با کلیک روی آن و رجوع به پنجره Properties، نوع آن را تغییر دهید.

۲۲-۱-۲ کادرهای متنی

همان‌گونه که پیش از این آموختید از کادرهای متنی برای ورود داده‌های متنی یا عددی استفاده می‌شود. سـه کادر متنـی برای ورود نام کاربری، رمز عبور و تأیید رمز عبـور روی صفحه قرار دهید و آن‌ها را به ترتیب txtUsername، txtPassword و txtConfirmPassword نام‌گذاری کنیم. یادآوری این نکته ضروری اسـت که باید با انتخاب کادرهای متنی وارد شده برای رمز عبور، از پنجره Properties، نوع آن‌ها را به Password تغییر دهید تا نویسه‌ها وارد شده درون آن‌ها قابل تشخیص نباشد.



برای کادرهای متنی وارد شده باید اعتبارسنجی‌های زیر صورت گیرد:

- طول نام کاربری باید بین ۶ تا ۱۲نویسه باشد.
- طول رمز عبور نباید کمتر از ۶ نویسه باشد.
- رمز عبور و تأیید آن باید یکسان باشند.

تابعی به نام checkForm در بخش سرصفحه (برچسب <head>) ایجاد نمایید. برای دسترسی به مقادیر کادرهای متنی موجود درون فرم از نگارش زیر استفاده می‌شود:

```
document. FormName.TextFieldName.value
```

بنابراین برای بررسی طول رشته وارد شده در کادر متنی نام کاربری، باید عبارت زیر را به تابع checkForm اضافه نماییم:

```
<head>
<script type="text/javascript">
function checkForm()
{
    var username=document.frm1.txtUsername.value;
    if (username.length <6 || username.length >12)
    {
        window.alert("نام کاربری باید ۶ تا ۱۲ نویسه داشته باشد");
        return (false);
    }
    return(true)
}
</script>
</head>
```

در این قطعه کد، مقدار وارد شده در عنصر txtUsername درون متغیر username قرار داده شده و سپس طول این متغیر ارزیابی شده است. در صورتی که مقدار آن کمتر از ۶ و بیش‌تر از ۱۲ باشد، یک پیغام خطا روی صفحه ظاهر و مقدار false به عنوان نتیجه تابع برگردانده می‌شود. اگر هم نتیجه اعتبارسنجی صحیح باشد، مقدار true برگردانده خواهد شد.

اکنون باید تغییراتی را هم در برچسب form ایجاد کنیم تا این تابع هنگام کلیک روی دکمه «ثبت» فراخوانی شود. در برچسب سازنده فرم، عبارت "onsubmit="return function_name() را وارد کنید؛ به این معنی که وقتی رویداد onsubmit فرم به وقوع پیوست، تابع function_name فراخوانی شود.

```
<form action="process.html" method="get" name="frm1" onsubmit="return checkForm()">
```

رویداد onsubmit فرم چه زمانی فراخوانی می‌شـود؟ زمانی که روی دکمه Submit فرم کلیک شود. وقتی تابـعِ بررسـی فرم فراخوانی می‌شـود یک نتیجه منطقی (true یـا false) برمی‌گرداند؛ اگر نتیجه true باشـد، اطلاعات فرم برای صفحه‌ای که در مشـخصه action قید شده فرسـتاده می‌شود، در غیراین‌صورت، پیغام‌های متناسب نمایش داده شده و از ارسال فرم جلوگیری خواهد شد.

اکنون باید در تابع تعریف شده، شرایط لازم برای معتبر بودن رمزعبورهای وارد شده را نیز بررسی نماییم. برای انجام این کار، تغییرات زیر رادر تابع checkForm اعمال نمایید.

```
function checkForm()
{
var username=document.frm1.txtUsername.value;
var password=document.frm1.txtPassword.value;
var cpassword=document.frm1.txtConfirmPassword.value;
if (username.length<6 || username.length>12)
{
    window.alert("نام کاربری باید ۶ تا ۱۲ نویسه داشته باشد");
    return (false);
}
else if (password.length < 6)
{
    window.alert("طول رمزعبور نباید کمتر از ۶ نویسه باشد");
    return (false);
}
else if (password!= cpassword)
{
    window.alert("رمزعبور با تأیید آن یکسان نیست");
    return (false);
}
return(true)
}
```

۳-۱-۲۲ آشنایی با رویدادها

در روشـی که برای اعتبارسـنجی مورد اسـتفاده قرار گرفت، کاربر همه اطلاعات را وارد می‌نماید و پس از کلیک روی دکمه «ثبت» عملیات اعتبارسنجی صورت گرفته و پیغام‌های لازم نمایش داده می‌شود. فرض کنید قصـد داریم طراحی فرم را به‌گونه‌ای تغییر دهیم که اعتبارسـنجی هر عنصر بعد از وارد کردن اطلاعات در آن

صورت پذیرد. در این حالت باید با تعدادی از رویدادهای عناصر فرم (عناصری که با برچسب <input> ایجاد می‌شوند) آشنا شوید. رویدادها، اعمالی در صفحه وب هستند که می‌توانند توسط جاوا اسکریپت شناسایی گردند. با استفاده از رویدادهای عناصر درون صفحه می‌توان محیطی کاملاً پویا و تعاملی را در صفحه وب ایجاد نمود. در جدول زیر، رویدادهای عناصر فرم را مشاهده می‌کنید.

نام رویداد	زمان وقوع
onchange	مقدار عنصر تغییر کند
onfocus	تمرکز فرم به عنصر منتقل شود
onblur	تمرکز فرم از روی عنصر برداشته شود
onclick	روی عنصر کلیک شود
ondblclick	روی عنصر دوبار کلیک شود
onmouseover	اشاره‌گر ماوس از روی عنصر رد شود.
onmousedown	دکمه ماوس فشار داده شود
onkeydown	دکمه‌ای از صفحه‌کلید فشار داده شود.
onkeypress	دکمه‌ای از صفحه‌کلید فشار داده شود و رها گردد
onkeyup	دکمه‌ای از صفحه‌کلید پس از فشار داده شدن، رها گردد
onselect	عنصری انتخاب شود

در جدول فوق با مفهومی به نام تمرکز[1] مواجه شدید که نیاز به توضیح دارد. وقتی با استفاده از کلید Tab میان عناصر موجود در فرم جابه‌جا می‌شوید یا با کلیک روی عنصر موردنظر، در صدد تغییر مقدار آن برمی‌آید، در واقع تمرکز فرم را به آن عنصر منتقل کرده‌اید. برای مثال وقتی می‌خواهید نام کاربری را در فرم طراحی شده وارد نمایید، درون آن کلیک می‌کنید؛ در این حالت، تمرکز به آن فرم منتقل شده و رویداد onfocus فرم به وقوع می‌پیوندد. هنگامی هم که با استفاده از کلید Tab به سراغ عنصر دیگری می‌روید یا با اشاره‌گر ماوس روی عنصر دیگری کلیک می‌کنید و در واقع آن عنصر را ترک می‌نمایید، رویداد onblur آن عنصر رخ می‌دهد.

فرم HTML را به گونه‌ای تغییر دهید که اعتبارسنجی کادر نام کاربری پس از خروج مکان‌نما از آن صورت گیرد.

1 . Focus

```
function checkUsername()
{
    var username=document.frm1.txtUsername.value;
    if (username.length<6 || username.length>12)
    {
        window.alert("نام کاربری باید ۶ تا ۱۲ نویسه داشته باشد");
        document.frm1.txtUsername.focus();
    }
}
```

ضمناً رویداد onblur این عنصر را با نام تابع مقداردهی کنید.

```
<input type="text" name="txtUsername" id="txtUsername" onblur="checkUsername()" />
```

بررسی کد:

- در این تابع، طول نام کاربری وارد شده مورد بررسی قرار می‌گیرد و چنان‌چه در محدوده مجاز نباشد، پیغام مناسب نمایش داده می‌شود.
- شاید این پرسش برای شما پیش بیاید که آخرین سطر تابع چه عملی را انجام می‌دهد؟
- چنان‌چه کاربر عبارتی مثلاً ۵ حرفی را درون کادر متنی نام کاربری وارد نموده و با فشـار دادن کلید Tab یا کلیک درون کادر متنی بعدی، از آن خارج شـود، پیغام مناسـب نمایش داده می‌شود اما وقتی پیغام را تأیید می‌کند، مکان‌نما به درون کادر بعدی منتقل می‌شود؛ در صورتی‌که کاربر باید نام کاربری را ویرایش نماید.
- با اسـتفاده از متد ()focus، مکان‌نما را در همان عنصری که ترک آن باعث بروز پیغام خطا شـده نگه می‌داریم.

کدی بنویسید که تعداد نویسه‌های وارد شده درون یک کادر متنی را هم‌زمان با تایپ نمایش دهد.

```
<html><head>
<script type="text/javascript">
function CountChar()
{
document.getElementById("charShow").innerHTML = document.
getElementById("txtTextBox").value.length;
}
```

```
</script>
</head>
<body>
<input type="text" name="txtTextBox" id="txtTextBox" onKeyUp="CountChar()" />
<p id="charShow"></p>
</body></html>
```

بررسی کد:

با اســتفاده از متــد getElementById و خصوصیت length، طول عبارت وارد شــده درون کادر متنی txtTextBox محاسبه می‌شود.

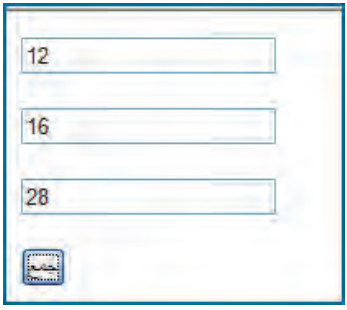
خصوصیــت innerHTML از پاراگــراف charShow با این عدد مقداردهی می‌شــود. با اســتفاده از این خصوصیت که برای اغلب عناصر HTML وجود دارد می‌توانید مقداری را بین برچسب شروع و پایان یک عنصر درج کنید.

تابع تعریف شده، با هر بار وقوع رویداد onKeyUp فراخوانی می‌گردد و طول عبارت را درون پاراگراف می‌نویسد.

اگر به‌جای رویداد onKeyUp از رویداد onkeydown یا onkeypress استفاده کنیم چه اتفاقی می‌افتد؟

صفحــه‌ای حاوی ســه کادر متنی و یک دکمه ایجاد کنید تــا وقتی کاربر دو عدد را درون کادر متنی اول و دوم وارد و روی دکمه کلیک می‌کند، حاصل‌جمع اعداد در کادر متنی سوم نمایش داده شود.

```
<html>
<head>
<script type="text/javascript">
function add()
{
	var n1 =Number(document.getElementById("txt1").value);
	var n2 =Number(document.getElementById("txt2").value);
	document.getElementById("txt3").value =  n1+n2;
}
</script>
</head>
<body>
<input type="text" name="txt1" id="txt1" />
<br/><br/>
<input type="text" name="txt2" id="txt2" />
<br/><br/>
<input type="text" name="txt3" id="txt3" />
<br/><br/>
<input type="button" value="جمع" onClick="add()" />
</body>
</html>
```



اگر در این مثال از تابع تبدیل Number() استفاده نکنیم، مقدار کادرهای متنی به صورت رشته‌ای خوانده می‌شود و لذا به جای عدد ۲۸، رشته 1216 در کادر متنی سوم درج می‌گردد.