

# ParsBook.Org

پارس بوک، بزرگترین کتابخانه الکترونیکی فارسی زبان

# ParsBook.Org



The Best Persian Book Library

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِیْمِ

آموزش

*JavaScript*

## گام اول

کاش در دهکده عشق فراوانی بود  
توی بازار صداقت کمی ارزانی بود

کاش اگر گاه کمی لطف به هم میکردیم  
مختصر بود ولی ساده و پنهانی بود

گام اول، ذکر چند مرجع است. یعنی چند سایت، کتاب و سی دی آموزشی در این رشته رو معرفی می‌کنم. این مراجع، به طور کلی می‌توانند شروع و استارت خوبی برای ما باشند. در راهی که دیگران رفتند، تجربه کردند و مطمئن هست، حرکت می‌کنیم.

گام دوم، معرفی یک کتاب، به عنوان مدار اصلی درس، است. یعنی ما، حول همین کتاب و طبق فصول و ابوابش، پیش می‌رویم. از مطالب خوبش استفاده کرده و اگر کم و کاستی داشت، با هم کاملش می‌کنیم. گام سوم، مطالعه و تحقیق شما.

گام چهارم، ممکنه در ضمن مطالعه، برایتان سوالهایی پیش بیاد، که من در خدمتان هستم. البته فکر نکنم به راحتی و حاضر و آماده، بشود جواب سوالی را از من بگیرید. اگر برای اولین بارست که با این مساله روبرو می‌شوید، شما را راهنمایی می‌کنم به منابعی که در دسترس است. اگر قبلا آن را خواندید و فراموش کردید، سعی می‌کنم یادآوری کنم. با این کار، خود شما، جواب سواتون را به دست می‌آورید و لذتی که می‌برید، بیشتر از خوردن یک غذای کنسروی و آماده است. بله، اگر در نهایت، متوجه پاسخ نشودید یا اشتباهی در ضمن کار به وجود آمد، ...

داخل پرانتز: این کار، چون از راه دور و غیر حضوری است، خسته کننده، طولانی و گاهی اوقات ناامید کننده می‌شود. تنها چیزی که می‌تواند در این مواقع به یاری شما بیاد، فراموش نکردن هدف اصلی، و صبر و استقامت است. پس با نام خدا، آستین همت، بالا زده و شروع می‌کنیم. چند مرجع (سایت) :

- ایران JavaScript: فارسی. از سایت‌های فارسی زبان که تخصصا درباره این زبان، فعالیت می‌کند. قسمتی از آن به آموزش، و قسمتی به بیان مثال، می‌پردازد.
- استاد آنلاین: فارسی. علوم مختلف رایانه‌ای را به شما آموزش می‌دهد که جاوا یکی از آنهاست.

• **Learn JavaScript**: انگلیسی. سایت **w3schools**، مرجع بسیار جامع و جالبی است برای بسیاری از علوم رایانه‌ای. یکی از امکانات جالب آن، ایجاد محیطی برای تمرین آنلاین است. یعنی در همین سایت، می‌توانید هر درسی را تمرین، اجراء و نتیجه آنرا ببینید.

• **JS Workshop**: انگلیسی. یک مخزن عالی با مقادیر زیادی اسکریپت و مسأله‌های فراوان برای یادگیری جاوا اسکریپت.

• **Pagere Source**: انگلیسی. هرآنچه را که برای طراحی وب، لازم است، در چند بخش، به شما آموزش می‌دهد.

۱- HTML Tutorials

۲- JavaScript Tutorials

۳- DHTML/Style Sheets

۴- CGI/Perl

۵- Web Design Articles

در هر بخش، با مسائل طبقه بندی شده روبرو می‌شوید.

## آشنایی با برنامه نویسی

اساسا کاربران، برای منتقل کردن دستورات خود به رایانه، از زبانهای برنامه نویسی (C، ویژوال بیسیک، دلفی، جاوا اسکریپت)، استفاده می کنند. مثلا برای نوشتن و یک تایپ ساده، باید برنامه‌ای مانند Word، توسط یک برنامه نویس، نوشته و در اختیارشان قرار گیرد که بتواند حرکات انگشت شما را بر روی صفحه کلید، ترجمه و پردازش کرده، تبدیل به حروف کند، و به شما نمایش دهد. به همین صورت است دیدن و طراحی عکس، شنیدن صوت، دیدن فیلم و هزاران کار دیگر، که در رایانه به راحتی برای شما میسر شده است.

• پس زبانهای برنامه نویسی، وسیله‌ای هستند برای ساختن برنامه و نرم افزار مورد نیاز ما. البته این زبانها با هم تفاوت‌هایی دارند:

۱- در سبک نوشتن دستورات

۲- در قدرت و محدوده فعالیتشان

۳- در محیط اجراء برنامه‌هایی که ساخته‌اند.

اما تفاوت سوم: اکثر برنامه‌ها، به راحتی، در محیط رایانه خانگی شما و در سیستم عامل ویندوز، قابل اجراء هستند؛ ولی ممکن است در سیستم عاملهای دیگر، مانند لینوکس، به این صورت و به این راحتی نباشد. یا در همین ویندوز خودمان، هر متنی قابل دیدن است؛ مانند doc, txt, pdf, htm؛ ولی در محیط اینترنت، پرونده‌هایی از نوع doc و pdf قابل اجراء نیستند، و باید اول آنها را دانلود کنیم و بعد در رایانه ببینیم. تنها برنامه و پرونده‌ای قابل اجراء و استفاده است که با زبان html همخوانی و سازگاری داشته باشد.

داخل پرانتز: تا همین چند وقت پیش، در محیط وب، حتی عکس هم قابل دیدن نبود؛ ولی الانه با قدرتمند شدن برآزرها و جستجوگرهای اینترنت، حتی می توانیم بعضی از انواع فیلم را در اینترنت ببینیم.

• پس به طور کلی می توانیم برنامه‌های خود را به دو گروه، تقسیم کنیم:

۱- برنامه‌هایی که در محیط رایانه، استفاده می شوند.

۲- برنامه‌هایی که در محیط وب و اینترنت قابل استفاده هستند.

● با جاوا اسکریپت، می‌توانیم برنامه‌ای بنویسیم که هم در رایانه و هم در محیط اینترنت، قابل اجرا باشد. البته زادگاه اصلی آن، اینترنت است.

پاورقی: در اینترنت، زبانهای مختلفی به وجود آمده‌اند که همین تفاوت (محیط اجراء) را با هم دارند:

۱- بعضی از آنها در طرف کاربر و کلاینت (Client) - یعنی شما که از اینترنت استفاده می‌کنید - اجراء می‌شود. JavaScript, JScript, VBScript

۲- بعضی از آنها در طرف سرور (Sserver) - یعنی سایتی که شما از آن استفاده می‌کنید - اجراء می‌شود. asp, php

● جاوا اسکریپت، از آن دسته از زبانهاست که در طرف کاربر، عمل می‌کند. با دو مثال، موضوع را برای شما روشنتر می‌کنم.

۱- یک ساعت را تصور کنید. اگر با جاوا اسکریپت، نوشته شود، زمان را طبق ساعت رایانه شما نشان خواهد داد؛ ولی اگر با برنامه‌ای مانند asp - که زبان سروری است - نوشته شود، زمان رایانه سرور را نشان خواهد داد که ممکن است با زمان شما، تفاوت داشته باشد.

۲- یک فرم نظرخواهی را تصور کنید. در رابطه با این فرم، دو کار انجام می‌شود.

- اطلاعات از شما گرفته، و برای سایت مورد نظر فرستاده شود. این بخش از کار را، جاوا انجام می‌دهد.

- اطلاعات فرستاده شده، در یک بانک، ذخیره شود. این بخش از کار، به عهده زبان سروری می‌باشد.

● بنابراین جاوا اسکریپت، یک زبان برنامه نویسی می‌باشد که به وسیله آن می‌توان بین کاربر و سایت ارتباط برقرار نمود.

توجه: استفاده از جاوا اسکریپت در صفحات وب، نیازی به نوشتن اسکریپتها نداشته، و می‌توانید اسکریپتهای موجود در شبکه اینترنت را کپی و در مکان مورد نظر، با ایجاد تغییرات مورد نیاز، استفاده کنید؛ اما در صورتی که یک برنامه نویس باشید، خواهید توانست اسکریپتهای مورد نظرتان را بنویسید.

## ارتباط جاوا را با زبان html

جلسه قبل، مطالب کلی درباره برنامه نویسی، و انواع آن، گفته شده؛ مثل ارتباط زبان جاوا اسکریپت، با زبان سروری. اینبار میخواهم ارتباط جاوا را با زبان html بیان کنم.

• **html** یک زبان برنامه نویسی نیست؛ بلکه زبان طراحی است. شالوده هر صفحه‌ای که در اینترنت می‌بینید، با این زبان نوشته شده است. مثلاً این وبلاگ، و همین کلماتی که من می‌نویسم و شما می‌بینید. اگر بخواهم کلمه‌ای را بنویسم، بزرگ شود، رنگی باشد، لینک داشته باشد، یا یک عکس وارد کنم، یا یک جدول طراحی کنم؛ همه و همه با این زبان است. **متن، کیفیت متن (نوع، رنگ، اندازه)، عکس، جدول، فرم، دکمه؛ به وسیله زبان html نوشته می‌شوند.**

### کار با html

• نوشتن این زبان، احتیاج به محیط خاصی ندارد. مثلاً زبان ویژوال بیسیک را باید برنامه ویژوال بیسیک را نصب، و در محیط آن بنویسیم؛ ولی زبانهای وب، چنین محدودیتی ندارند. مثلاً شما در یک متن ساده (Text)، می‌توانید به راحتی، هم HTML بنویسید و هم Java و هم PHP.

• در رابطه با طراحی وب، برنامه‌هایی ساخته شده‌اند که ما را در هرچه راحتتر، سریعتر، و دقیقتر نوشتن، کمک می‌کنند. مانند "فرانت پیج" (FrontPage) از مایکروسافت، و "دریم ویور" (Dreamweaver) از ماکرومدیا، و برنامه‌های دیگر. در این نوع نرم‌افزارها، امکاناتی تدارک دیده شده که کار با html را آسان و مطمئن کرده‌اند.

۱- ویرایشگر قوی: امکان برگشت (Undo)، تا چندین مرحله. رنگی کردن قسمتهای مختلف کُد، برای بهتر مشخص شدن دستورات.

۲- ویژوالی شدن: ویژوال یعنی بصری، دیدنی، قابل دیدن. یعنی به جای کدنویسی در یک محیط ساده، برای ساخت یک دکمه، از دکمه‌های آماده در این برنامه استفاده می‌کنید. این نرم‌افزار، عناصر مختلف را به صورت آماده، در اختیار ما قرار داده است. کافی است شما آنها را انتخاب کنید.

۳- کدهای آماده: بعضی از دستورات متداول را به صورت آماده، ارائه می کنند.

۴- غلط یاب: در صورت اشتباه نوشتن کد، اخطار می دهد.

مثال: برای ضخیم کردن یک کلمه در **html**، باید این دستور را نوشت:

**<b>**مثال**</b>**

ولی در فرانت پیج، کافی است کلمه مورد نظر را انتخاب کنید، و در نوار ابزار، آیکون **B** را کلیک کنید.

**داخل پراگمات:** در اینجا نه قصد دارم فرانت پیج یا هر نرم افزار دیگری را آموزش دهم، و نه استفاده از آنها را به دوستان تازه کار، توصیه می کنم؛ چون ما را از هدفمان که یادگیری برنامه نویسی است دور کرده و باعث تنبلی می شود. تنها یک نرم افزار را پیشنهاد می کنم که از مدتها قبل، همراه و یار من در یادگیری و برنامه نویسی بوده. "1st Page 2000". این برنامه رایگان را می توانید از این آدرس (<http://www.evrsoft.com/download.shtml>) دریافت کنید.

## ساختار HTML

یک صفحه وب، از دو قسمت سر (Head) و بدنه (Body) تشکیل شده، که به این صورت نوشته و مشخص می شود.

**<html>** شروع صفحه

**<head>** شروع سر

**<Script>**

دستورات جاوا

**</Script>**

**</head>** پایان سر

**<body>** شروع بدنه

محتویات صفحه: متن، عکس

پایان بدنه `</body>`  
پایان صفحه `</html>`

اسکلت اصلی یک صفحه وب، به این صورت است.

## واژه نامه:

**کُد (Code):** دستور و فرمانی که به زبان برنامه نویسی، نوشته شده.  
**تَگ (Tag):** کوچکترین جزء یک کد. مثلا تگ سر = `<head>`; تگ بدنه = `<Body>`; تگ اسکریپت = `<Scirpt>`. هر تگ باید در داخل این دو علامت `<`، `>` قرار گیرد.

در جلسه آینده، به امید خدا، با انواع تگ، آشنا خواهیم شد.

## اولین دستور

### انواع تگ:

۱. تگ ساده. بعضی از انواع تگ، یک قسمتی هستند؛ مانند تگ خط (لاین) `<hr>`.
۲. تگ مرکب. تگهایی که از دو قسمت شروع و پایان، ساخته شده‌اند؛ مانند تگ خط و متن `<Font> </Font>`. اکثر تگها، به این شکل هستند. این نوع تگ هم به دو دسته تقسیم می‌شود.
  ۱. تگی که می‌توان انتهای آنرا نوشت؛ به اصطلاح، آنرا باز بگذاریم. مانند تگ پاراگراف `<p>`
  ۲. تگی که باید انتهای آن مشخص و بسته شود. مانند تگ اسکریپت `<Script> </Script>`

**داخل پرانتز:** چون این تقسیم بندی، جدید و تجربی است، ممکن است جامع و دافع نباشد، و بشود بهتر از این تعریف کرد.

متوجه شدیم که تگ جاوا اسکریپت، از نوعی است که باید ابتداء و انتهای آن مشخص شود. ابتداء تگ جاوا را به این شکل `<Script>` می‌نویسیم و انتهایش را به این صورت `</Script>`. دستورات جاوا هم ما بین این دو تگ نوشته می‌شود.

اکسپلورر، با دیدن تگ اسکریپت، متوجه می‌شود که این یک دستور برنامه نویسی است و نباید آنرا نشان دهد و فقط آنرا اجرا می‌کند. ولی اگر این تگ را ننویسیم، یا اشتباه بنویسیم، دستورمان اجرا نشده و هر چه نوشته‌ایم، در صفحه وب، نمایش داده می‌شود.

### اولین دستور:

حالا می‌خواهیم اولین دستور جاوا اسکریپت را بنویسیم. اولین چیزی که با هم تمرین می‌کنیم، دستور پیام (alert) است. ابتداء یک تگ کامل اسکریپت را نوشته؛ و بعد داخل آن بنویسید: alert ("سلام")

```
<Script>  
alert ("سلام")  
</Script>
```

بعد از ذخیره کردن، آنرا اجرا کنید. یک کادر، نمایش داده می‌شود که بر روی آن نوشته: سلام. این اولین سلام، از طرف جاوا اسکریپت به شماست؛ و شاید از طرف دنیای برنامه نویسی. حتما متوجه شده‌اید که کلمه سلام، به عنوان مثال بود و شما می‌توانید هر متن کوتاه یا بلندی را جایگزین آن کنید.

**تذکر:** در برنامه نویسی، به کوچک و بزرگ بودن حروف، دقت کنید.

## پرسش و پاسخ:

### ? پرسش

سلام. خسته نباشید. می‌خواستم بدونم که دستوره‌ای جاوا، در قسمت بدنه هم به کار میره یا فقط در قسمت سر هستش. و اگر میشه استفاده کرد می‌خوام بدونم چه تعدادی از دستوره‌ای جاوا میشه در یه صفحه وب استفاده کرد. ممنون میشم.

### ● پاسخ

۱. معمولا قسمت سر، برای نوشتن کدهای مختلف، مانند جاوا، و قسمت بدنه، برای نوشتن متن، به کار می‌رود. برای اشتباه نشدن و دسترسی آسان، بهتر است شما هم همین کار را کنید؛ اما در اکثر اوقات، الزامی برای این کار نیست. البته گاهی در برنامه نویسی، مواردی پیش آمده که باید در قسمت بدنه، دستور جاوا را می‌نوشتیم.  
۲. ظاهرا که محدودیتی در تعداد دستورات جاوا، وجود ندارد.

## واژه نامه:

**پارامتر (Parameter):** مقداری که برای تگ، نوشته می‌شود. مثلا برای تگ خط، معین می‌کنیم که از چه نوعی، به چه اندازه‌ای و چه رنگی باشد. مثال: کلمه **Color**، در تگ **Font**، پارامتری است که رنگ متن را مشخص می‌کند.

<Font color="#0000FF">سلام</Font>

پارامتر را شاید بتوان این گونه تعریف کرد که **قید حالت است برای تگ**. تگ اسکریپت، هم دارای پارامترهایی است که در جلسه بعد به آن، می‌پردازیم.

### معرفی کتاب:

• نام: خود آموز جاوا اسکریپت در ۲۴ ساعت

## Teach yourself Java Script in 24 hours

نویسنده: مایکل مانکر

مترجم: مهندس علی ناصح

ناشر: موسسه فرهنگی هنری دیباگران تهران

نوبت چاپ: اول

قیمت: ۳۳۰۰ تومان

تلفن: ۲۰۹۰۰۰۱ الی ۵

پایگاه اینترنتی: <http://www.jsworkshop.com>

این سایت، دارای نکات آموزنده و جالب در علوم مختلف طراحی وب، بالاخص جاوا اسکریپت می‌باشد. می‌توانید مثالهای کتاب را در این آدرس (<http://www.jsworkshop.com/js3examples.html>) با توضیح دیده و از این آدرس (<http://www.jsworkshop.com/js3e/js3e.zip>) دانلود کنید.

• نام: خود آموز سریع جاوا اسکریپت

## Java Script For The World Wide Web 2003

نویسندگان: تام نگرینو و دوری اسمیت

مترجم: مهندس حمید رضا ذوقی

انتشارات: ذوقی

نوبت چاپ: اول

قیمت: ۳۹۵۰ تومان

پایگاه اینترنتی: <http://www.javascriptworld.com>

در این پایگاه، علاوه بر مثالهای کتاب، پایگاه‌های کمکی برای زبانهای وب، نیز معرفی شده است. می‌توانید مثالهای کتاب را در این

آدرس (<http://www.javascriptworld.com/scripts/index.html>) با توضیح دیده و از  
این آدرس (<http://www.javascriptworld.com/scripts/JSVQS5eScripts.zip>)  
دانلود کنید.

## پارامتر

### پارامترهای اسکریپت:

- تگ اسکریپت، پارامترهایی دارد که در این جلسه با آنها آشنا می‌شویم. به این خط دقت کنید:

```
<script language="JavaScript" src="">
```

### پارامتر زبان (language)

- اولین پارامتری که به چشم می‌خورد، زبان است. اسکریپت، یک زبان عمومی است که انواع و اقسامی دارد. یکی از آنها جاوا اسکریپت است که معروف و متداولترین آنهاست. JScript، LiveScript و VBScript، انواع دیگر آن است. این زبانها، با هم تفاوت دارند. بنابراین در ابتدای کار، باید برای اکسپلورر، مشخص کنیم که به چه زبانی می‌خواهیم برنامه نویسی کنیم، تا طبق آن عمل کند.

**داخل پراتنز:** چون جاوا اسکریپت، اولین و مشهورترین زبان اسکریپتی است، احتیاجی نیست که این پارامتر را برایش بنویسیم. به عبارت دیگر، هرگاه که نوع زبان اسکریپتی، مشخص نشود، اکسپلورر، فرض را بر جاوا اسکریپت می‌گیرد و طبق آن عمل می‌کند.

### پارامتر مکان (src)

دومین پارامتر، مکان است؛ یعنی محلی که کدهای جاوا در آن نوشته شده است.

### توضیح:

- معمولاً، کدهای جاوا را در خود صفحه html می‌نویسیم و اجرا می‌کنیم. ولی می‌توان تمام کدهایی را که نیاز داریم، در یک صفحه مستقل یا پرونده متنی، ذخیره کرده، و فقط آدرس آن پرونده را بنویسیم. در این صورت، اکسپلورر، با رسیدن به این پارامتر، آدرس مورد نظر را خوانده، به آن مراجعه کرده، و کد مورد نظر را اجرا می‌کند. این پرونده را - که یک بانک دستورات جاواست - با پسوند JS ذخیره می‌کنند. وقتی یک صفحه

اینترنتی را ذخیره می‌کنیم، عکس و دیگر متعلقات آن، در یک پوشه به همین نام، ذخیره می‌شود. اگر دقت کرده باشید، گاهی پرونده‌هایی با همین پسوند (JS) در آن وجود دارد.

بنابر آنچه که گفته شد، وقتی از این پارامتر استفاده می‌شود که دستورات جاوا را در جای دیگری ذخیره کرده‌ایم.

**مثال:** اگر این پرونده را به نام `java`، در پوشه‌ای به نام `image`، و در کنار صفحه `html` خودمان، ذخیره کرده‌ایم، اینگونه آدرس آنرا می‌نویسیم:

```
src="image\java.js"
```

#### داخل پراتنز:

- اسکریپت، پارامترهای دیگری هم دارد که این دو پارامتر، مهمترین و کاربردی‌ترین آنهاست.
- اگر قرار است جاوا بنویسید، و در همان صفحه باشد، نه در جای دیگر، کافی است بنویسید: `<script>`

## جاوای باهوش

• بعد از نوشتن اولین دستور برنامه نویسی، بهتر است با مفهوم این دستورات آشنا شویم. همانطور که ملاحظه کردید، میان یک دستور جاوا با اچ تی ام ال، تفاوت است. اچ تی ام ال، یک صفحه ساده و ثابت را به ما می دهد؛ و جاوا، یک صفحه متحرک و هوشمند. با اچ تی ام ال، بدنه و اسکلت یک صفحه (متن، عکس، جدول و فرم) را می سازیم و با جاوا به این بدن و اسکلت، روح می دمیم و آنرا زنده می کنیم؛ تا آن حد که نسبت به هر عملی، واکنش و عکس العمل مناسبی را نشان می دهد.

**مثال:** یک فرم ارسال نامه را تصور کنید. اگر مثلاً نام خود یا آدرس گیرنده را اشتباهاً ننویسید، برنامه، با پیامی این موضوع را به شما یادآوری می کند. یعنی اینکه متوجه عمل شما شده و یک واکنش مناسب، نشان می دهد. در این جلسه با این مقوله، یعنی **هوشمند بودن جاوا**، آشنا می شویم.

### شرط

### تعریف شرط

• یکی از ابزار و دستورات مهم هر زبان برنامه نویسی، شرط است. آیا می دانید شرط چیست؟ برای درک مفهوم آن، احتیاجی نیست به خود زحمت بدهید؛ چون این دستور، در زندگی من و شما به وفور یافت می شود. اگر خورشید درآید، هوا روشن می شود. اگر درس بخوانی، قبول می شوی.

اگر دلی را به ناله آری، ز برق آهش امان نداری  
بلا در افتد به هر چه داری، که چوب یزدان صدا ندارد

اگر عمری گنه کردی، مشو نومید از رحمت  
تو نام توبه را بنویس، امضا کردنش با من

به تمام اینها، جمله شرطی گفته می شود.

## ادات شرط

۱. if

۲. switch ... case

۳. variable = (condition) ? true : false

معروفترین ادات شرط، if است.

## ساختمان شرط if

- جمله شرطی، از سه قسمت اصلی، تشکیل شده است.
  - ۱- ادات شرط: کلمه‌ای که شرط را می‌رساند؛ مانند اگر.
  - ۲- شرط: جمله‌ای که شرط انجام کاری، واقع شده.
  - ۳- مشروط یا جزا: جمله‌ای که جزای شرط و در جواب آن آمده؛ یعنی در صورت تحقق شرط، آن نیز به وقوع می‌پیوندد.
- در زبان برنامه‌نویسی هم مثل زبان محاوره‌ای، این سه رکن، رعایت شده. با این عناصر و طریقه نوشتن آنها، آشنا می‌شویم.

## عناصر شرط، در جاوا:

۱- ادات شرط: if

۲- شرط: داخل پرانتز (...)

۳- مشروط: داخل کروشه {...}

( شرط ) if

{ جزا };

**توجه:** اگر شرط، تنها شامل یک مشروط باشد، نیازی نیست که آن را درون علامت کروشه { } قرار دهیم.

## اهمیت شرط

• اهمیت استفاده از شرط در برنامه‌نویسی، به فایده آن، برمی‌گردد. هر چقدر که استفاده از آن، منفعت داشته باشد، اهمیت دارد. این موضوع را با مثال، توضیح می‌دهم.

به اولین دستوری که نوشتید دقت کنید. آنرا در داخل یک صفحه، نوشته‌اید که به محض ورود به آن، پیام دیده می‌شود. این خوب است ولی کامل نیست. تصور کنید هر دستور جاوایی که می‌نویسیم به محض ورود، نمایش داده شده یا اجرا شود. مثلا شما می‌خواهید اگر کاربر، آدرس گیرنده را ننوشت، اختطاری نشان داده شده و مانع ادامه کار گردد. این دستور باید کجا قرار گیرد؟ چگونه باید آنرا کنترل کرد که هر جا شما خواستید یا کاربر اشتباهی کرد، اجرا شود؟ اینجاست که باید دست به دامن شرط شد. پس از شرط، برای کنترل و تنظیم فرامین، استفاده می‌کنیم.

حالا همان مثال اولی، یعنی نمایش پیام را می‌خواهیم با کمی کنترل، اجرا کنیم. مثلا اگر رنگ پس زمینه صفحه ما، سفید بود، بنویسد سفید:

```
if (document.bgColor == "#ffffff")  
    {alert("سفید")  
    };
```

حالا این پیام، در صورتی نمایش داده می‌شود که شرط آن، یعنی سفید بودن پس زمینه، محقق شده باشد. البته این یک مثال بود که الان به ذهنم رسید. شما می‌توانید شرط یا جزای دیگری را در نظر بگیرید.

در جلسه آینده، به امید خدا، با اقسام شرط آشنا شده و تسلط بیشتری بر دستورات خود، پیدا خواهیم کرد.

## واژه نامه:

**document** = سند، صفحه جاری، پنجره‌ای که در آن کار می‌کنید  
**document.bgColor** = رنگ پس زمینه سند

## فصل هفتم

### شرط (۲)

#### ۲. شرط switch

• **switch** از ادات دیگر شرط است که کارایی بالایی دارد، و مانند **if** عمل می‌کند؛ هر چند از لحاظ ترکیب، تفاوت دارد. این شرط، زمانی به کار می‌آید که مجبور باشیم از چندین شرط، به صورت پی در پی استفاده کنیم. در جلسه قبل، با اشکال مختلف شرط **if** آشنا شدید. **switch** می‌تواند به جای انواع پیچیده آن، به کار گرفته شود.

#### ساختمان switch

switch (متغیر)

```
{  
    case 1: مشروط ; break;  
    case 2: مشروط ; break;  
    case 3: مشروط ; break;  
    case 4: مشروط ; break;  
    case 5: مشروط ; break;  
    default: break;  
}
```

تمرین

می‌خواهیم برنامه‌ای بسازیم که در هر روز از هفته، یک پیام دهد. این برنامه، بسیار مهم و پرکاربرد است. با آن می‌توانید مثلاً برای خود یک تقویم بسازید که مناسبت‌های هر روز را به شما نشان دهد؛ یا یک برنامه که هر روز، یک حدیث، شعر یا جملات زیبا را بنویسید.

**کد:**

```
var Now = new Date()
var Week = Now.getDay()
switch (Week)
{
    case 0 : alert("یکشنبه");
    break;
    case 1 : alert("دوشنبه");
    break;
    case 2 : alert("سه‌شنبه");
    break;
    case 3 : alert("چهارشنبه");
    break;
    case 4 : alert("پنجشنبه");
    break;
    case 5 : alert("جمعه");
    break;
    case 6 : alert("شنبه");
    break;
    default : break;
}
```

**تحلیل**

**switch**: ادات شرط است. داخل پرانتز، نام متغیر نوشته می‌شود. **caes**: بعد از این کلمه، شرط ما نوشته شده است. در این مثال، شرط، همان عددی است که نمایانگر روز هفته می‌باشد.

**break**: این کلمه کلیدی، در انتهای هر مشروط، نوشته شود.

**default**: این کلمه کلیدی، برای جایی است که خارج از شرط است؛ یعنی شاید جایی باشد که ما در نظر نگرفتیم و داخل شرطهای ما نباشد. در این صورت، شرط، به حالت پیش فرض (**default**)، مراجعه می کند. این شرط، این معنی را دارد: روز هفته، اگر ۱ بود، پیام بده "دوشنبه"؛ اگر ۲ بود، پیام بده "سه شنبه"؛ الی آخر.

### ۳. شرط سوم

برای این نوع شرط، هنوز اسمی پیدا نکرده ام؛ چون از هیچ کلمه یا حرفی تشکیل نشده است. می شود از آن به جای شرط (**if ... else**) استفاده کرد.

```
variable = (condition)?(true):(false);
```

```
value = (a == 1) ? 1 : 0;
```

این شرط را به این شکل هم می شود نوشت:

```
if (a == 1) {value = 1};  
else {value = 0};
```

با مقایسه این دو شرط، متوجه معنای آن خواهید شد. سعی کنید یک مثال عملی با این شرط، بنویسید.

### خصوصیات

از خصوصیات و فواید این نوع شرط، مختصر و کوتاه بودن آن است؛ و از معایبش این است که همه جا نمی شود مورد استفاده قرار بگیرد؛ به عبارت دیگر، قدرت و توانایی **if** را ندارد.

## Object (شی)

- هر چیزی که در صفحه ما وجود دارد، یک شی حساب می‌شود و می‌توان از آن استفاده کرد و تغییرش داد. اولین شی موجود، خود صفحه و سند (document) است، که قبلا با آن آشنا شده‌اید.

## Properties (خصوصیات)

- هر شی، یک مشخصات و خصوصياتی دارد که بیانگر کمیت و کیفیت آنست. مثلا یک سند، دارای این خصوصیات است: پس زمینه، عنوان (Title)، نوار وضعیت (Status)، نوار پیمایش (Scroll)، متن (Text)، لینک (Link). بعضی از مشخصات سند را در جدول پایین آورده‌ام. شما این جدول را برای خود داشته باشید و هرگاه که با خصوصیت جدیدی آشنا شدید، به آن اضافه کنید.
- طریقه نوشته: ابتدا نام شی را نوشته، و بعد از یک نقطه، نام خصوصیت مورد نظر را وارد کنید. گاهی اوقات، آن خصوصیت هم دارای چند خصوصیت دیگر است که باید بعد از یک نقطه، نوشته شود.

خصوصیات سند	
document.title	عنوان
document.URL	آدرس
document.location	آدرس
document.bgColor	رنگ پس زمینه
document.fgColor	رنگ متن
document.linkColor	رنگ لینک
document.alinkColor	رنگ لینک فشرده شده
document.vlinkColor	رنگ لینک دیده شده

جدول شماره ۲

## Methods (متد)

- کاری که اشیاء می‌توانند انجام دهند، متد نامیده می‌شود؛ مانند باز کردن یا بستن پنجره. شالوده و اصل یک زبان برنامه نویسی، متدهای آن برنامه است، که ما باید آنها را به خوبی یاد گرفته و در جای مناسب استفاده

نماییم. متدها، توسط سازندگان زبان برنامه نویسی، نوشته و تهیه می‌شود و ممکن است در نسخه‌های جدید آن زبان، تغییر داده شده، اضافه و تکمیل شود.  
مثال: متد بسته شدن صفحه:

window.close()

## Event (رویداد)

• رویداد، عملیاتی است که کاربر، در یک صفحه انجام می‌دهد. بعضی از رویدادها مشترک و عمومی هستند و بعضی مخصوص یک شی. بعضی از این رویدادها را برای شما دسته بندی کرده‌ام که حتما ذخیره نموده و به مرور آنرا کامل کنید.  
طریقه نوشتن: رویدادها را با کلمه on شروع می‌کنند و به این صورت می‌خوانند: onClick = هنگام کلیک کردن.

رویدادها					
صفحه کلید		موس		عمومی	
پایین آمدن کلید	onKeyDown	حرکت موس	onMouseMove	باز شدن	onLoad
فشاردن کلید	onKeyPress	ورود موس	onMouseover	بسته شدن	onUnload
بالا آمدن کلید	onKeyUp	خروج موس	onMouseout	انصراف	onAbort
		کلیک	onClick	فوکوس، زوم	onFocus
		دوبار کلیک	onDblclick	برگشت از فوکوس	onBlur
		کشیدن	onDragstart	خطا	OnError

جدول شماره ۳

## پرسش و پاسخ

### ? پرسش:

سلام آموزشت روان و قابل فهمه؛ اما اگر ممکنه مثالهای عملی بکار ببر؛ یعنی فرمان واقعی قابل اجرا مثال بزن. سوالی هم دارم آیا تمام دستوراتی که در جاوا اسکریپت قابل اجراست - مانند اچ تی ام ال - دستورات مخصوص خودش را دارد؟ موفق باشی.

● پاسخ:

سلام. تشکر از پیشنهاد شما. چشم، حتی المقدور درس را با مثال عملی، همراه می‌کنم. اما سواتان، مسلماً اینطور است که شما گفتید. جاوا اسکریپت یک زبان برنامه نویسی دقیق و ظریف هست که دستورات مخصوص به خودش را دارد. البته این دستورات به صورت کلی، با زبانهای دیگر مشترک است - مانند دستور if که در همه زبانها، موجود است - ولی در نوشتن، تفاوتی دارد.

## ثابت و متغیر

• جلسه امروز ما، درباره دو مفهوم مهم و پرکاربرد در برنامه نویسی است؛ یعنی ثابت و متغیر؛ اما قبل از آنکه وارد بحث شویم، باید کمی به حرفای "من درآوردی" گوش کنید. از این جهت من درآوردی که شاید در کتابی نشود به این صورت، ساده و دسته بندی شده پیدا کرد.

### عملیات برنامه نویسی

• یک عملیات برنامه نویسی، دارای سه بخش، است.

۱. دریافت داده. در این قسمت، اسکریپت، اطلاعاتی را که نیاز دارد، می‌گیرد.
۲. پردازش داده. در این قسمت، اطلاعاتی که تهیه کرده، تجزیه و تحلیل می‌کند.
۳. نمایش داده. در این قسمت، اطلاعات پردازش شده را، ارائه می‌کند.

**داخل پراتنز:** بعضی از عملیات، آنقدر ساده و بسیط هستند که نیازی به پردازش ندارند؛ بنابراین، اسکریپت، به محض دریافت اطلاعات لازم، آنرا ارائه می‌کند؛ بر خلاف بعضی از عملیات، که بسیار پیچیده هستند.

**مثال:** یک ماشین حساب را تصور کنید. طرز کارش چگونه است؟ شما دو عدد را به او می‌دهد، نوع عملیات را برایش مشخص می‌کنید، مثلاً جمع؛ ماشین حساب آن عدد و عملیات را پردازش می‌کند، و نتیجه را در صفحه نمایش به شما نشان می‌دهد. به امید خدا در چند جلسه آینده می‌توانید به راحتی یک ماشین حساب بنویسید.

### راه دریافت داده

به نظر شما، یک اسکریپت، از چه راهی اطلاعات خود را به دست می‌آورد؟ به تصور من که از ۳ روش.

۱. برنامه نویسنده معمولاً شما به عنوان برنامه نویسنده، اطلاعاتی را به او می‌دهید. مثلاً یک کلمه عبور و پسورد را ذخیره می‌کنید، که اگر کلمه‌ای که کاربر وارد کرده با آن، یکی بود، اجازه ورود به او می‌دهد؛ وگرنه با یک پیام خطا، مانع ورود او می‌شود.
۲. کاربر. بعضی از اوقات، کاربر، اطلاعاتی را به برنامه شما می‌دهد؛ مانند مثال قبلی، که کلمه‌ای را به عنوان کلمه عبور، وارد کرده.
۳. سیستم. در بسیاری از مواقع، این سیستم و رایانه شماست که منبع اطلاعاتی برنامه است. مثلاً برای نمایش زمان و تاریخ، با یک متد ساده، تاریخ مورد نظر را از رایانه، به دست آورده و نمایش می‌دهید.

## انواع داده (ثابت و متغیر)

بعضی از اطلاعاتی که یک برنامه، به دست می‌آورد، ثابت (Constant) است، و بعضی، متغیر (Variable). متغیر به این معنی که یک چیز ثابت، مشخص و از پیش تعریف شده نیست. آنچه که برنامه نویس، بیشتر با او سر و کار دارد، متغیر است.

### مثال:

به مثال قبلی دقت کنید. به نظر شما کدام قسمت از داده‌ها، ثابت و مشخص است، و کدام، متغیر و تعریف نشده؟ کلمه‌ای که برنامه نویس نوشته، مشخص و ثابت است، و کلمه‌ای که کاربر، ممکن است وارد کند، نامشخص و متغیر. زمان و تاریخی که از رایانه دریافت می‌شود، چطور؟ ثابت است یا متغیر؟

امیدوارم مطلب، برای شما واضح و ملموس شده باشد.

## معرفی متغیر و ثابت

بعد از این مقدمه، باید یاد بگیرید که چگونه می‌شود یک متغیر، به زبان برنامه نویسی نوشت. تا اینجای درس، میان تمام زبانهای برنامه نویسی مشترک بوده؛ ولی نوشتن و مشخص کردن ثابت و متغیر، در زبانها، مختلف است. مثلاً در زبان ویژوال بیسیک و VB Script با Dim، یک متغیر را معرفی می‌کنند و با Const، یک ثابت را. در حالی که در زبان دلفی، PHP و جاوا اسکریپت با Var، متغیر معرفی می‌شود.

**پاورقی:** خوشبختانه جاوا اسکریپت در این زمینه بسیار راحت و ساده برخورد کرده و لازم نیست مانند زبانهای دیگر، تشریفات خاصی را رعایت کرد.

## متغیر در جاوا اسکریپت

۱. نام گذاری متغیر

به این جمله دقت کنید و همیشه به خاطر داشته باشید: "می‌توانیم برای هر چیزی نامی انتخاب کنیم - مثلاً برای اطلاعات، یا متدهای جاوا - و به جای آنکه مرتباً اسم بلند و طویل، و گاهی مشکل آن متد را بنویسیم، نامی که خودمان برایش انتخاب کردیم را صدا می‌کنیم."

- یک متغیر را در جاوا، بعد از کلمه کلیدی var و به این صورت می‌نویسند: var name.

- به جای `name`، هر اسمی که دوست دارید بنویسید. البته بهتر است اسم انتخابی شما، مناسب با نوع متغیر باشد.

- هر کجا که `var` نوشته شده، مشخص است که متغیری را می‌خواهند تعریف کنند.

۲. مقدار دهی متغیر

- مقدار دهی یعنی مشخص کنیم که این متغیر، برابر با چه داده و اطلاعاتی است.  
- این کار را با علامت = انجام می‌دهیم. اول، نام متغیر را نوشته، و بعد از آن مساوی، و سپس داده یا متد مورد نظر را یادداشت می‌کنیم. به این صورت:

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

**مثال:** قرار است - با توجه به مطالب این درس و درس قبل - برنامه‌ای بنویسیم که زمان را به ما نمایش دهد.

**وسایل مورد نیاز:**

```
<script></script> کد اسکریپت  
new Date (), toLocaleString() متد زمان  
document.write ( ) متد نوشتن
```

با استفاده از این وسایل و از روشی که گفته شد - دریافت داده - پیش می‌رویم.

```
<script>  
var D = new Date (); // اولین متغیر - متد زمان  
var L = D.getHours (); // دومین متغیر - متد ساعت - دریافت زمان از سیستم  
document.write (L); // متغیر خود را درون پرانتز می‌نویسیم - نمایش داده  
</script>
```

## واژه نامه

**new Date**: متدِ زمان به صورت کلی و مطلق است. بعد از آن باید نوع زمان خود را مشخص کنید؛ یعنی چه قسمتی از زمان را می‌خواهید بدانید؛ سال، ماه، هفته، روز و ... به تنهایی چیزی را نشان نمی‌دهد.

**toLocaleString**: متدی برای به دست آوردن زمان کامل (تاریخ و ساعت) است که به عنوان خصوصیت، برای زمان کلی استفاده می‌شود.

**document.write**: متدی که برای نوشتن متن در صفحه استفاده می‌شود.

**document.writeln**: مانند دستور قبلی عمل می‌کند؛ با این تفاوت که یک فاصله تا رشته قبلی، ایجاد می‌کند.

: به این علامت، سِمی کالِن گفته می‌شود که در پایان هر دستور، می‌آید؛ هر چند در جاوا، اجباری نیست.

//: به این علامت، کامنت (**Comment**) گفته می‌شود که برای نوشتن توضیح، در میان کدهای برنامه نویسی است. هر عبارتی که بعد از این علامت بیاید، توسط برنامه، خوانده و اجرا نمی‌شود و صرفاً توضیحی است که برنامه نویس، یادداشت کرده.

- این هم یک هدیه، برای شما دوستان. دستور و متدهای زمان. در آینده با آنها بیشتر آشنا می‌شوید.

متدهای زمان		
Fri Feb 3 13:34:04 UTC+0330 2006	زمان	new Date ()
Friday 03 ,February , 2006 01:34:04 ب.ظ	زمان کامل	toLocaleString ()
Friday 03 ,February , 2006	فقط تاریخ	toLocaleDateString ()
01:34:04 ب.ظ	فقط ساعت	toLocaleTimeString ()
Fri, 3 Feb 2006 10:04:04 UTC	زمان بین المللی	toGMTString()
2006	سال	getFullYear()
1	ماه	getMonth()
5	روز هفته	getDay()
3	روز ماه	getDate()

13	ساعت	getHours()
34	دقیقه	getMinutes()
4	ثانیه	getSeconds()

#### جدول شماره ۴

پاورقی: زمان، دارای اجزا مختلف است؛ یعنی انسان، آنرا به بخشهای مختلف تقسیم کرده تا قابل استفاده باشد. قسمتی از آن، تاریخ (سال، ماه، هفته و روز) نام دارد، و قسمتی از آن، به اجزا کوچکتر (ساعت، دقیقه و ثانیه) تقسیم شده، و ساعت، نامیده می‌شود. البته معیار تاریخ در برنامه نویسی، میلادی است. برای تبدیل به شمسی یا قمری، باید از برنامه نویسی استفاده کنید.

#### تمرین

در مثال قبل، می‌توانید به جای `getHours` از متدهای دیگر استفاده کنید. پیشنهاد میکنم هر کدام از آنها را به صورت جداگانه در `document.write` قرار داده و نتیجه‌اش را ببینید.

```
<script>
var Time = new Date (); // اولین متغیر، مدت زمان کلی
document.write (D + "<p>");
document.write (D.toLocaleString() + "<p>");
document.write (D.toLocaleDateString() + "<p>");
document.write (D.toLocaleTimeString() + "<p>");
document.write (D.toGMTString() + "<p>");
document.write (D.getFullYear() + "<p>");
document.write (D.getMonth() + "<p>");
document.write (D.getDay() + "<p>");
document.write (D.getDate() + "<p>");
document.write (D.getHours() + "<p>");
document.write (D.getMinutes() + "<p>");
document.write (D.getSeconds());
</script>
```

## انواع متغیر

با عرض معذرت، فاصله این جلسه کمی طول کشید. هم به این علت که هر چه جلوتر می‌رویم، کار، دقیقتر و مشکلتر می‌شود؛ و هم اینکه سرگرم تغییر دکوراسیون و قالب وبلاگ بودم. امیدوارم که زیبا بوده و استفاده از مطالب، راحتتر شده باشد. بنای من بر این است که در هر هفته، دو جلسه، در خدمت شما باشم. به امید خدا این هفته هم همینطور خواهد بود.

« Date، از اشیاء از پیش تعریف شده در جاوا است که امکان کار با تاریخ و ساعت را در اختیار برنامه‌نویس، قرار می‌دهد. هنگام نیاز به ذخیره یک تاریخ و ساعت بخصوص، می‌توانید از Date استفاده کنید. به کمک متدهای این شی، می‌توانید با تاریخ و ساعت ذخیره شده، کار کنید. شی Date، هیچ خصوصیتی ندارد؛ از این رو، جهت نمایش یا دستیابی به مقادیر ذخیره شده در آن، لازم است تا از متدهای آن استفاده کنید. » کتاب "خود آموز جاوا اسکریپت در ۲۴ ساعت"

## انواع متغیر

پاورقی: « خوش به حال شما که برنامه نویسی را از خوب جایی، و با خوب کسانی شروع کردید. دور و اطرافتان کسانی هستند که از راهنمایی و کمکشان بی‌بهره نیستید. قدر این زمان و مکان را بدانید، استفاده کنید، و مفید باشید.

زمانی که من خواستم شروع کنم، تک و تنها، بدون استاد، و از زبانی بسیار گسترده، پیچیده و مقررانی - یعنی ویژوال بیسیک - شروع کردم. سادگی که در جاوا اسکریپت، وجود دارد، در آن زبان یافت نمی‌شود. آنقدر مقررات و دستورات، که از برنامه نویسی وحشت می‌کردم. یعنی من می‌تونم؟! سوالی که اکثر اوقات برایم پیش می‌آمد. از بخشهای سخت آن زبان، انواع متغیر بود. »

• همانطور که در جلسه قبل آموختید، داده‌ها، مواد خام و اصلی برنامه نویس است که آنها را از راه‌های مختلفی به دست می‌آورد. این اطلاعات، از انواع مختلفی هستند. بعضی حروف و کلمه، و بعضی اعداد و ارقام.

۱- رشته‌ای (String): حروف و کلمه

۲- عددی (Number): اعداد و ارقام (اعداد صحیح و اعشاری)

### ۳- منطقی (Boolean): بله و خیر

این، انواع داده‌ها در جاوا است؛ اما در ویژوال بیسیک، فقط ۷ نوع داده عددی، وجود دارد. در زبانهای دیگر، باید قبل از هر کدنویسی، متغیر، تعریف شود و نوع آن هم مشخص گردد. ولی در جاوا، نه معرفی جداگانه و ابتدایی آن لازم است، و نه مشخص کردن نوعش. این خاصیت، کار را بسیار راحت می‌کند.

### خصوصیات داده‌ها

هر کدام از این انواع، خصوصیات دارند. مثلاً در اعداد، می‌توان از علائم ریاضی استفاده کرد؛ که به آن، عملگرهای ریاضی گفته می‌شود.

عملگرهای ریاضی		
1 + 1	جمع	+
2 - 1	تفریق	-
2 * 2	ضرب	*
4 / 2	تقسیم	/
	خارج قسمت	%
1 ++	یک عدد، اضافه می‌کند	++
10 --	یک عدد، کم می‌کند	--

جدول شماره ۵

مثال: این مثال، مقدمه‌ای است برای ساختن یک ماشین حساب.

### خصوصیات رشته

• برای رشته‌ها یا همان حروف، دستورات و خصوصیات وجود دارد. مثلاً می‌خواهیم بدانیم طول یک رشته چقدر است؛ یعنی چه تعداد حرف، در آن به کار رفته؛ یا فلان حرف یا کلمه، در کجای این رشته قرار دارد؛ یعنی چندمین حرف این رشته است.

پاورقی: « برای دست یافتن به این اطلاعات چه باید کرد؟ برای اینکه اهمیت این موضوع را درک کنید، و ذهنتان نسبت به آن روشنتر شود، مثالی می‌زنم.

می‌دانید که مزیت رایانه نسبت به کتاب در چیست؟ آیا در داشتن اطلاعات است؟ یا در یافتن سریع اطلاعات؟ سرچ و جستجو است که رایانه را بر نسل قبل از خود، یعنی کتاب، برتری داده. جستجو یعنی پیدا کردن یک حرف، کلمه یا جمله، در انبوهی از جملات، کلمات و حروف. برای این کار لازم است برنامه‌ای بنویسید که بتواند موقعیت کلمه مورد نظر شما را دریابد و آنرا به شما نشان دهد. »

### تمرین:

رشته‌ای داریم متشکل از حروف انگلیسی، که نام آنرا `string` می‌گذاریم. به نوبت، هر کدام از خصوصیات آنرا امتحان کنید.

#### خصوصیات رشته

۱۷	طول (تعداد) حروف	<code>string.length</code>
abcdefghijklmnopq	تبدیل حروف بزرگ به کوچک	<code>string.toLowerCase ()</code>
ABCDEFGHIJKLMNOPQ	تبدیل حروف کوچک به بزرگ	<code>string.toUpperCase ()</code>
abcdefghijklmnopq	ضخیم (بُلد) کردن حروف	<code>string.bold ()</code>
<i>abcdefghijklmnopq</i>	کج (ایتالیک) کردن حروف	<code>string.italics ()</code>
abcdefghijklmnopq	تغییر اندازه	<code>string.fontSize (3)</code>
abcdefghijklmnopq	تغییر رنگ	<code>string.fontcolor ("red")</code>
abcdefghijklmnopq	بزرگ کردن	<code>string.big ()</code>
cdef	انتخاب حروف از حرف ۲ تا ۶ حرف بعد از اول	<code>string.substring (2,6)</code>
	انتخاب حروف از حرف ۲ تا ۶ حرف بعد از ۲	<code>string.substr (2,6)</code>

cdefgh	انتخاب يك حرف (حرف چهارم)	string.charAt (4)
e	تعويض دو حرف (حرف b با a)	string.replace ("a","b")
bbcdefghijklmnopq	جستجوي حرف مورد نظر، بعد از عدد معيّن	string.indexOf ("c",1)
2	جستجوي حرف مورد نظر، قبل از عدد معيّن	string.lastIndexOf ("c",1)
-1 17		
abcdefghijklmnopq		
ABCDEFGHIJKLMNO PQ		
abcdefghijklmnopq		
<i>abcdefghijklmnopq</i> abcdefghijklmnopq		
<b>abcdefghijklmnopq</b>		
abcdefghijklmnopq		
cdef		
cdefgh		
e		
bbcdefghijklmnopq		
2		
-1		

جدول شماره ۶

### تذکر:

- دو دستور indexOf و lastIndexOf، شماره حرف مورد نظر را می‌نویسند.
- شمارش آنها از صفر است؛ یعنی شماره حرف اول، صفر است، نه یک.
- در صورتی که کلمه مورد نظر را پیدا نکنند، -۱ می‌نویسند.

## متغیر محلی و سراسری

این تقسیم بندی مهم، چون وابسته به مفهوم "تابع" است، و ممکن است گفتن آن در اینجا، باعث سردرگمی شود، در جلسه آینده و بعد از آشنایی با "تابع"، طرح می‌شود.

## معرفی پایگاه

پردیس

• امروز پایگاهی را به شما معرفی میکنم که حاوی اخبار و آموزش علوم مختلف رایانه‌ای است. یکی از آموزشهای آن، جاوا اسکریپت می‌باشد که خوشبختانه به زبان ساده‌ای، توسط آقای صابر کردستانچی، نوشته شده است. شاید با وجود چنان استادی، نیازی به این وبلاگ نباشد. حتما از آن استفاده کنید.

## پرسش و پاسخ

### ? پرسش:

سلام. خسته نباشید. من منظور شما رو از زمان کلی و زمان کامل متوجه نشدم. میشه برای آنها مثال بزنید؟ ممنون میشم.

و دیگه اینکه اگه براتون امکان داره دستوراتی که می‌گید و از روی همین وبلاگی که هستید براش مثال بیارید که بهتر متوجه بشیم. باز ممنون.

راستش فک کنم یه خورده سخت شده یا اینکه من خیلی کم هوشم. میدونید من می‌خوام تمرین کنم ولی همش واسه دستورایی که نوشتید نمیدونم چیو باید بنویسم اگه مثال باشه می‌فهمم که منظورتون چیه. از زحماتتون سپاسگزارم

### • پاسخ:

سلام

۱. شاید مطالبی که گفتم واضح نبود. بنابراین، تغییراتی در متن درس، دادم. لطفا دوباره بخوانید و اگر مشکلی بود تذکر دهید.

۲. البته من مثالی را آوردم که امیدوارم شما آنرا نوشته و اجراء کرده باشید. آیا این کار را انجام داده‌اید؟ چه نتیجه‌ای داشت؟

۳. نگران نباشید. این طبیعی است؛ به مرور - البته با کمک کتابهایی که معرفی کردم - با مثال و پروژه‌های بیشتری آشنا خواهید شد. شما فعلا همین مثال و تمرینها را به خوبی انجام دهید تا حفظ شوید.

## نام گذاری کدها (تابع)

بحث امروز ما درباره مفهومی است که جایگاه ویژه‌ای در برنامه نویسی دارد؛ تابع. کمتر پیش می‌آید که کدی را بنویسید و از تابع، استفاده نکنید.

### تابع

• تابع یا همان فانکشن (Function)، مجموعه‌ای از یک یا چند کد است که در یک جا جمع شده و دارای نام شده‌اند.

### نام گذاری کد

• اگر یادتان باشد، گفته بودم برای هر چیزی می‌شود نامی انتخاب کرد، و به جای آنکه مدام، کد آنرا تکرار کنیم، نام آنرا صدا می‌زنیم. حالا که تا حدودی با برنامه نویسی آشنا شده‌اید، وقتش شده که با نامگذاری آنها نیز آشنا شوید. برای این کار از تابع استفاده می‌کنیم.

**مثال:** کد پیام را به یاد دارید؟

```
<Script>  
alert ("سلام")  
</Script>
```

اگر بخواهیم برای آن، نامی انتخاب کنیم، ابتدا کد مورد نظر را داخل یک جفت کورشه { } قرار داده، تا از کدهای دیگر جدا شود. بعد، کلمه **function** را در اول آن نوشته و بعد، نامی که انتخاب کرده‌اید را با یک جفت پرانتز ( )، بنویسید. به این صورت:

```
<Script>  
function funAlert()  
{  
alert ("سلام")  
}  
</Script>
```

نامی که من انتخاب کردم `funAlert` است. `fun` مخفف `function`، و `Alert` نوع کد مرا مشخص می‌کند.

آیا می‌دانید کدی که الان درست کرده‌ایم با کد قبلی که بی‌نام بود، چه فرقی دارد؟ در عمل یک فرق کوچک، و آنکه این است که قبلا به محض باز شدن صفحه، اجرا شده و پیام نمایش داده میشد؛ ولی حالا، تا نام آنرا صدا نزنیم، عمل نمی‌کند.

به نظر شما این کار چه فایده‌ای دارد؟ منتظر جواب شما هستم.

## صدا کردن تابع

• حالا باید بدانید چگونه و کجا تابع خود را صدا کنید. در جلسه هشتم، با رویدادها آشنا شدیم. یکی از راه‌های صدا زدن توابع، استفاده از رویدادهاست. اگر می‌خواهید فلان کد، به هنگام باز شدن صفحه، اجرا شود، از رویداد `onLoad` استفاده کنید. اگر می‌خواهید با کلیک کردن بر روی چیزی، برنامه‌ای اجرا شود، از `onClick` استفاده کنید.

**مثال:** می‌خواهم تابع `funAlert`، که در مثال بالا درست کرده‌ام، به هنگام باز شدن صفحه اجرا شود. برای این کار باید به تگ بدنه (`Body = بادی`) رفته و این کد را بنویسم:

```
<Body onLoad="funAlert()" ...>
```

اگر خواستید با کلیک بر روی عکسی، پیامی نمایش داده شود، باید کد زیر را در تگ عکس قرار دهید. به این صورت:

```
<img onClick="funAlert()" ...>
```

نقطه چین، یعنی بقیه کدهای مربوط به عکس.

## خلاصه:

- برای ساخت و به کارگیری یک تابع، سه کار باید کرد.
- ۱- نوشتن کد: کدهایی را که لازم داریم می‌نویسیم.
- ۲- نامگذاری: کدهای نوشته شده را در یک جا، جمع کرده و نامگذاری می‌کنیم.
- ۳- صداکردن: در جای مناسب، تابع را صدا می‌کنیم.

## پاورقی:

- می‌شود در یک صفحه، چند تابع جداگانه بسازیم.
- می‌شود یک تابع را توسط تابع دیگر، صدا زد. این در زمانی است که دو تابع، در امتداد یکدیگر فعالیت می‌کنند.
- گاهی، داخل پرانتزی که بعد از نام تابع قرار می‌گیرد، چیزی نوشته می‌شود به نام آرگومان که یک نوع متغیر است.

## انواع متغیر: متغیر محلی و سراسری

اگر بخواهیم از نظر جنس، به متغیر نگاه کنیم، سه نوع مهم دارد - رشته‌ای، عددی، منطقی - که در جلسه قبل آشنا شدید. اما از لحاظ موقعیت و مکان استفاده، یک تقسیم بندی جدید برای متغیر است.

### ۱- متغیر سراسری، عمومی (Global گلوبال)

اگر متغیر را خارج از تابع بنویسید، عمومی است؛ چون می‌شود در سرتاسر برنامه، آنرا صدا زده و مورد استفاده قرار داد.

### ۲- متغیر محلی، خصوصی (Local لوکال)

اگر متغیری را داخل تابع تعریف کنید، خصوصی است؛ یعنی فقط در همان محل و همان تابع، کاربرد دارد، و بیرون از آن، نه می‌شود صدا زد و نه می‌شود استفاده کرد.

### خصوصیات متغیر محلی و سراسری

- از متغیر سراسری می‌شود در توابع مختلف استفاده کرد؛ اما متغیر محلی، فقط در همان تابع، کاربرد دارد.
- محدوده متغیر محلی، محدوده تابع آن است، و عمر آن، به اندازه عمر تابعش.
- می‌دانید که نمی‌شود برای دو متغیر، یک نام گذاشت؛ اما متغیر محلی می‌تواند با متغیر سراسری، هم‌نام باشد.
- اگر این دو متغیر، هم‌نام شدند، تغییر مقدار یکی از آن دو، تاثیری در دیگری ندارد.

## آرایه

**داخل پراتنز:** همگی خسته نباشید. اجازه بدهید قبل از اینکه درس جدید را شروع کنم، صحبتی داشته باشم. به دوستان جدیدی که به جمع ما پیوسته‌اند، خوش آمد می‌گویم. امیدوارم چیزهای کمی که می‌دانم به درد شما بخورد. به دوستان قدیمی خسته نباشید گفته و یک گلایه هم دارم. سوالی که در جلسه قبل کردم، بی‌جواب ماند.

بیشتر از یک هفته است که درباره موضوعی مطالعه می‌کردم، به نام کوکی (Cookie). با اینکه از مباحث مهم و کاربردی جاوا اسکریپت است، متأسفانه نه در کتاب آموزشی و نه در سایت‌های معروف و مهم، اثری از آن دیدم. طبق معمول، بدون آموزش، از نمونه‌های موجود استفاده کرده و سعی داشتم از آنها بیاموزم که چطور نوشته می‌شود. تقریباً از جستجو، ناامید شده بودم؛ تا اینکه به یکی دو تا پایگاه برخورددم، که منبع بزرگی از کوکی هستند. هرچند آموزشی نیستند، ولی از هیچی بهترند.

خلاصه کلام اینکه اگر خدا بخواهد، این ضعف و نقصان موجود در کتب و پایگاه‌های جاوا را اینجا و با هم جبران می‌کنیم. نمونه‌ای از کاربرد کوکی را می‌توانید در سمت راست همین وبلاگ، در بخش ثبت نام ببینید.

## تعریف آرایه

آرایه، مجموعه‌ای از چند داده (ثابت و متغیر) است که به عنوان یک واحد شناخته شده و اعضای آن، با کد، مشخص شده‌اند. به عبارت دیگر، تعدادی از اطلاعات که در کنار هم قرار گرفته‌اند و یک وجه مشترک داشته و به یک نام خوانده می‌شوند.

**مثال:** به این کلمات دقت کنید: تهران، اصفهان، شیراز. وجه مشترک اینها چیست؟ شهر بودن وجه اشتراکشان است. وقتی که می‌خواهید تلفنی با یکی از این شهرها تماس بگیرید چه می‌کنید؟ کد مخصوص آن شهر را حتما فراموش نکنید. بله، هر یک از شهرها، در شبکه بزرگ مخابرات، دارای کدی است که معرف آن است. بهتر است ما هم به جای اینکه هر کدام از آنها را جداگانه به کار ببریم، برای هر کدام، کد، تعریف کرده و آن را صدا بزنیم.

## نوشتن آرایه

- نامگذاری آرایه

آرایه یک نوع متغیر است که بر خلاف متغیرهای دیگر در جاوا، باید از قبل تعریف شود.

```
var Name = new Array(3);
```

**var**: نشانه متغیر

**Name**: نامی که شما برای آرایه، انتخاب می‌کنید. مناسب است در اینجا **City** را به عنوان نام، برگزینید

**new Array**: کلمه کلیدی که باید به همین صورت، بنویسید

(۳): طول آرایه، تعداد اعضای آرایه

- معرفی اعضای آرایه

بعد از معرفی و نامگذاری آرایه، باید اعضایش را معرفی کنیم؛ که دو روش برای این کار وجود دارد.

- روش اول:

```
Name[0] = "تهران";  
Name[1] = "اصفهان";  
Name[2] = "شیراز";
```

**Name**: نام آرایه که مشترک است

[۰]: کد مخصوص به هر عضو

- روش دوم:

```
var Name = new Array("تهران", "اصفهان", "شیراز");
```

در این روش، کد اعضا، به صورت فرضی، از صفر شروع می‌شود.

## • خواندن آرایه

بعد از نوشتن آرایه، نوبت به خواندن و صدا کردن آن است. در این مثال می‌توانیم از دو دستور `alert` یا `document.write` استفاده کنیم.

```
alert (Name[0])
```

برای صدا کردن هر شهر، کد آنرا بنویسید.

**تمرین:** انتخاب یک گزینه از فهرست. تهران اصفهان شیراز

این برنامه را در قسمت بدنه صفحه خود، کپی کنید:

```
Name="Menu" onChange="funChange()">  
Value="0"> تهران  
Value="1"> اصفهان  
Value="2"> شیراز
```

## معرفی پایگاه

### Useful JavaScript •

این پایگاه انگلیسی زبان، جاوا اسکریپت را در جلسات منظم، ساده و با مثال، آموزش داده است.

### JavaScript Cookies •

Cookie Central (مرکز کوکی) •

اگر مانند من، به دنبال منبعی درباره کوکی‌ها می‌گشتید، این دو پایگاه خارجی را مطالعه نمایید.

## پرسش و پاسخ

### ? پرسش:

سلام، ممنون از زحمات. می‌خواستم لطف کنی در مورد متغیرهای منطقی بیشتر توضیح بدی. آیا انواع گوناگون دارند و در کجاها قابل مصرفند؟ متشکرم. همیشه موفق باشی.

### ● پاسخ:

سلام. قبلاً گفتیم، بولین (بله-خیر)، یک نوع از اطلاعات و داده است. همانطور که در مکالمات روزمره از بله و نه استفاده می‌کنیم، در زبان برنامه نویسی هم، جواب بعضی از سوالات، یک جواب منطقی، یعنی بله و خیر است. مثلاً یک کادر، ظاهر می‌شود که از شما می‌پرسد "آیا مایل به ادامه کار هستید؟" این کادر، دو دکمه دارد. **Ok, Cancel**، که همان بله و خیر است. برای نمایش چنین کادری، از این دستور استفاده کنید.

confirm ("آیا شما مطمئنید؟")

### ? پرسش:

سلام. خسته نباشید. می‌خواستم بدونم که شما چرا اسمای سخت برای کداتون می‌زارید. یعنی آیا میشه اسمای آسون براشون انتخاب کنیم؟ یا حتی به فارسی اسمی رو بنویسیم؟ بعد اینکه می‌خواستم بدونم که مثلاً پیام (سلام). شاید ما بخواهیم، در جاهای مختلف پیامهای گوناگون داشته باشیم. آیا می‌شود، که تابع شماره‌ی ۱ و ۲ و ۳ درست کرد؟ یعنی قبلش تعریف کنیم و بعد با صدا کردن شماره یک پیام مخصوصش بیاید؟ و غیره ... ممنون میشم اگه توضیح بدید. بازم خسته نباشید و بی نهایت سپاسگزارم.

### ● پاسخ:

سلام. هم در نام گذاری اسمی، آزاد هستید و هم در استفاده از پیامهای گوناگون. البته در نام گذاری، بعضی چیزها باید لحاظ بشه؛ مثلاً فارسی نباشه و با عدد شروع نشه؛ اما می‌توانید مثلاً بگویید: **a1, a2, a3**. راهی که گفتید خیلی خوبه؛ یعنی قبلاً آنها را تعریف کنید و بعد صدا بزنید.

### پرسش:

با سلام. متشکریم از این که به ما سر زدید. سوالی داشتم. در مورد **alert** می خواستم بدانم نوع و شکل آن کادری که به دنبال اجرا کردن این دستور باز می شود آیا به نوع سیستم عامل بستگی دارد؟ به عبارتی آن کادر، از سیستم عامل ناشی می شود یا اینکه صرفاً متکی بر سیستم جاوا است. با تشکر و خدانگهدار.

### ● پاسخ:

سلام دوست عزیز! این را به عنوان یک قاعده کلی مد نظر داشته باش. هر کاری که شما به عنوان برنامه نویس می خواهید انجام دهید، مشروط به استعداد و اجازه سیستم عامل است و باید از امکانات موجود استفاده کنید؛ یعنی شما فقط فلان امکان موجود در ویندوز را فراخوانی می کنید. یکی از آن امکانات، پیام است که به اشکال گوناگون وجود دارد.

### پرسش:

با سلام. متشکریم از اینکه جوابی روشن و آشکار به سؤال قبلی ام دادید. سوال دیگری داشتم که اگر جواب دهید ممنون میشوم. چطور میتوان برنامه های جاوا را به هم ربط داد؟ (روشی غیر از فایل های JS\*) زیرا که این روش در واقع استفاده از دو برنامه جدا از هم است اما می خواستم بدانم که اگر در یک صفحه چند برنامه جاوا نوشته باشیم چگونه میتوانیم اینها را به هم ربط دهیم به طور مثال: اگر دو برنامه جاوا (جداگانه) در یک صفحه نوشته شده باشد چطور میتوان از برنامه اولی در برنامه دومی استفاده کرد و آنها را به هم مرتبط کرد. متشکریم. خدا نگهدار

### ● پاسخ:

سلام. اگر منظورتان از ربط دادن، صدا کردن یک تابع در تابع دیگر است، که مشکلی نیست. کافی است نام تابع مورد نظر را در تابع دوم بنویسید. یکی از راههای صدا کردن تابع، همین است. البته اگر خود مثال را می دیدم بهتر بود. در آینده، از این نوع دستورات، یعنی دستورات ترکیبی، مثال خواهم زد.

### پرسش:

سلام. از آموزش خوبتون خیلی خیلی ممنونم. نظیرش را کمتر روی اینترنت دیدم. دو تا سوال ازتون داشتم: ۱- در برنامه انتخاب یک گزینه (که نوشته بودید) چرا وقتی برای اولین بار روی گزینه تهران میریم پیغام نمیده؟ ۲- دستور **OnChange** در اون برنامه کارش چیه؟. من متوجه نشدم. باز هم ممنون. بای بای

● پاسخ:

سلام. جواب سوال اول شما، در سوال دوم است. دستور OnChange (که یک نوع رویدادست و من انتخاب کرده‌ام)، به معنای "هنگام تغییر" است. یعنی وقتی گزینه جدید را انتخاب کنید و قبلی، تغییر کند. چون اولین گزینه، به صورت پیش فرض، تهران است و انتخاب شده، هیچ پیامی داده نمی‌شود. امیدوارم متوجه شده باشید. اگر جوابم واضح نبود، تذکر بدهید.

## یک پروژه

### ساعت

#### تغییرات:

- ۱- پاسخ به سوالات جدید در جلسه ۱۲.
- ۲- در جدول شماره ۴، جلسه ۹ تغییراتی داده شد.

حالا که با توابع آشنا شده‌اید، برنامه ساخت یک ساعت را با هم تمرین می‌کنیم. این پروژه را در چند فاز، انجام می‌دهیم که هر فاز، به تنهایی قابل اجراست.

#### مواد مورد نیاز:

. دستورات جاوا را در بخش سر (Head) صفحه خود بنویسید.  
. در قسمت بدنه، یک پاراگراف، درست کرده و نام آنرا Font بگذارید.

```
<p ID="Font"></p>
```

• **فاز اول.** به دست آوردن زمان کلی: متغیری به نام Time می‌سازیم که زمان را به دست می‌آورد.

```
// 1.
```

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

```
document.write (" زمان: " + Time); // با این دستور، زمان به دست آمده، نوشته می‌شود
```

• **فاز دوم.** به دست آوردن اجزاء زمان (ساعت، دقیقه و ثانیه): سه متغیر می‌سازیم برای به دست آوردن سه قسمت از زمان.

```
// 2.
```

```
var H = Time.getHours(); // ساعت
```

```
var M = Time.getMinutes(); // دقیقه
```

```
var S = Time.getSeconds();// ثانیه  
var Clock = H + ":" + M + ":" + S;// متغیری می‌سازیم که این سه عنصر را یک جا  
ذخیره کند  
document.write ("<br>ساعت: " + Clock);// با این دستور، زمان به دست آمده، نوشته  
می‌شود
```

• **فاز سوم.** قرار دادن دستورات در یک تابع: تابعی ساخته به نام `funTime` تا کنترل بیشتری بر کدهای خود داشته باشیم. تمام کدهایی که نوشته‌ایم داخل این تابع قرار می‌دهیم.

```
// 3.  
function funTime()  
{  
var Time = new Date ();  
var H = Time.getHours();  
var M = Time.getMinutes();  
var S = Time.getSeconds();  
var Clock = H + ":" + M + ":" + S;  
Font.innerText = "ساعت: " + Clock;// این، همان پاراگرافی است که قبلا در بدنه نوشته‌ایم  
}
```

در تگ بدنه، این کد را نوشته و با آن، تابعی را که ساختید، صدا کنید.

```
onLoad="funTime()"
```

• **فاز چهارم.** ساخت تایمر: تا اینجا، پروژه ما کامل شده و ساعت را نشان می‌دهد؛ اما فقط زمانی را نشان می‌دهد که برای اولین بار، پنجره را باز کرده‌ایم. به نظر شما چه باید کرد که لحظه به لحظه، زمان را به ما نمایش دهد؟ یک راه، این است که مدام، صفحه را بازخوانی و رفرش کنیم. این کار چندان راحت و معقول نیست. بهترست فقط تابع خود را بازخوانی کنیم. برای این کار از `setTimeout` استفاده می‌کنیم که یک تایمر است.

```
// 4.  
function funTime()
```

```

{
var Time = new Date ();
var H = Time.getHours();
var M = Time.getMinutes();
var S = Time.getSeconds();
var Clock = H + ":" + M + ":" + S;
Font.innerText = "ساعت: " + Clock;
setTimeout ("funTime()", 1000);// با این تایمر، تابع ما، در هر یک ثانیه، دوباره خوانی
می‌شود
}

```

● برای دیدن نمونه عملی آن، اینجا کلیک کنید.

## واژه نامه:

- ID:** آی‌دی، همان نام (Name) شی است؛ البته گاهی در عمل تفاوت‌هایی با هم دارند.
- innerText:** با این دستور، متن یک تگ در قسمت بدنه - مانند محتویات تگ پاراگراف - نمایش داده می‌شود؛ و می‌توانیم آنرا تغییر داده و ویرایش کنیم.
- outerHTML - innerHTML - outerText:** این ۳ متد، شبیه دستور قبلی است؛ البته با کمی تفاوت.
- setTimeout:** دستور تایمر و زمان شمار است. بعد از آن، یک جفت پرانتز، قرار دارد که داخلش، دو پارامتر نوشته می‌شود:
۱. اسم چیزی است که قرار است اجرا شود؛ مانند یک تابع.
  ۲. زمانی که تایمر در آن فعال خواهد شد. واحد زمان، میلی ثانیه است. (۱۰۰۰ = یک ثانیه)
- setInterval:** دستور تایمر و زمان شمار است. مانند دستور قبل، عمل می‌کند.

## جلسه ۱۳

سلام دوباره به همه دوستانی که منو در پایگاه قبلی، همراهی کردند. چه میشه کرد؛ سال جدید و خونه تکونی. امان از درد مستاجری. اونجا صابخونه ما رو جواب کرد. دنبال یک جای مناسب می‌گشتم. اومدم بنگاه بلاگفا که خونه بگیرم؛ اما چشمتون روز بد نبینه. اتاق خالی نداشت. همه پر بودند. البته خالی بود ولی در شان ما نبود. آخه وبلاگ آشپزی به چه کار من میاد. برای همین مجبور شدم، آدرس رو کمی تغییر بدم. به جای **JavaScript** (که فعلا در اختیار دوست عزیزم در پایگاه ایران جاوا اسکریپت است)، شده **JScript** باور کنید این مقاله رو خیلی وقته که نوشتم و آماده کردم؛ ولی امان از دست این آمریکای چیزخوار. چشم نداره که پیشرفت ما رو ببینه. هی میزنه پرشین بلاگ رو خراب میکنه و نمیداره مطلب جدید بنویسم. منم که دیدم این جوریه، گفتم به کوری چشم آمریکا، باید یک جای مناسب پیدا کنم. یکی از جاهایی که مناسب دیدم، همین جاست که الان واستادیم. شاید هم برم جای دیگه. نه به اون روز که هیچ جا نداشتیم که بنویسیم، نه به امروز که ...

### تغییرات:

- ۱- در همین درس، تغییراتی داده شده.
- ۲- واژه نامه در "یک پروژه" کاملتر شد.
- ۳- جلسه ۶، شرط، کاملتر شد.

### حلقه (زنجیره)

• در زندگی، بعضی چیزها تکرار می‌شوند؛ هر روز و هر شب؛ و گاهی هر لحظه؛ مثل نفس کشیدن. کارهایی که حلقه وار انجام می‌شوند یا انتها و عمر محدودی دارند، و یا بی‌انتهای نامحدودند؛ مانند شب و روز. چون برنامه‌نویسی، اقتباسی از زندگی انسان است، و قرارست در خدمت ما باشد، از این قاعده مستثنی نیست.

### تعریف حلقه

• « یکی دیگر از ویژگیهای مفید جاوا اسکریپت و همچنین سایر زبانهای برنامه نویسی، توانایی ایجاد حلقه‌های تکرار، یا اجرای گروهی از دستورالعملها به صورت تکراری است. این، یکی از کارهایی است که رایانه‌ها قادرند تا به خوبی آنها را انجام دهند، و آن چیزی نیست جز انجام کارهای تکراری با دقت بالا. در

برنامه‌های اسکریپت، از حلقه‌های تکرار به عنوان یک ابزار قوی و مهم، استفاده‌های فراوانی می‌شود. « کتاب "خود آموز جاوا اسکریپت در ۲۴ ساعت"

## ادات حلقه

• از روشهای مختلفی برای ایجاد حلقه، استفاده می‌شود.

for .۱

While .۲

do ... while .۳

معروفترین ابزار حلقه، for است.

## ساختمان حلقه for

• ساختمان حلقه، شبیه یک زنجیر است که از تعدادی حلقه، ساخته شده است.

{Code}

for (Counter; Condition Expression; step)

حلقه for، از چهار جزء، تشکیل شده است. همانطور که می‌بینید، دارای سه پارامتر است که داخل پرانتز ( ) قرار گرفته‌اند و با سمی کالن (;) از هم جدا می‌شوند؛ و یک جواب، که حاوی کد مورد نظر ما بوده، و داخل یک کروشه { } قرار می‌گیرد.

۱. ابتدای حلقه:  $a = 0$ . این عدد، شروع و ابتدای حلقه را معین می‌کند.

۲. انتهای حلقه:  $a < 10$ . این عدد، حداکثر و انتهای حلقه - به عبارت دیگر، تعداد تکرار دستورات - را مشخص می‌کند.

۳. افزایش یا کاهش: این پارامتر، مشخص می‌کند که حلقه، در هر بار اجرا، یک واحد زیاد شود یا کم.

۴. جواب حلقه: حلقه، بی‌شبهت به شرط نیست. به عبارت دیگر: اگر متغیر ما (a)، از ۰ تا ۱۰ تداوم داشت، نمایش پیام، ادامه داشته باشد.

نکته: اگر حلقه، تنها شامل یک جواب باشد، - مانند شرط- نیازی نیست که آن را درون علامت کروشه {} قرار دهیم.

**مثال:** دستورالعمل زیر، پیامی را ۱۰ مرتبه، جهت توجه بیشتر کاربر، نمایش می‌دهد:

```
for (a = 0; a < 10; a++)  
{  
  document.write ("پیام" + a); // جواب حلقه  
}
```

**تمرین:** به نظر شما آیا می‌توان این حلقه را به صورت شرطی هم نوشت؟ منتظر پاسخ عملی شما هستیم.

## ساختمان حلقه while

حلقه while از لحاظ اجزاء تشکیل دهنده، فرقی با حلقه for نداری؛ اما از لحاظ ترکیب و محل قرار گرفتن آنها، متفاوت است.

```
var a = 0;  
while (a < 10)  
{  
  document.write ("پیام", a , "<br>");  
  a++;  
}
```

همانطور که می‌بینید، متغیر، قبل از while، تعریف شده، و داخل پرانتز، یک پارامتر قرار گرفته. پارامتر افزایش هم بعد از کد، نوشته شده. به محل قرار گیری این پارامتر دقت کنید؛ چون اگر قبل از دستور، نوشته شود، معنای دیگری می‌دهد و کمی در نتیجه کار، تغییر ایجاد می‌کند.

## ساختمان حلقه do ... while

```
var a = 0;  
do
```

```

{
document.write ("پیام", a );
a++;
}
while (a < 10);

```

توجه دارید که برای درک راحت این مفاهیم، فقط اسکلت و شکل ساده آنها را نوشته‌ام. شما می‌توانید با مراجعه به کتاب و سایتهایی که معرفی شد، با اشکال مختلف آنها آشنا شوید.

### اهمیت حلقه

• اهمیت حلقه را به مرور و در طی برنامه‌های عملی پی خواهید برد. فعلا عرض می‌کنم اگر به شباهت حلقه و شرط دقت کنید، متوجه اهمیت آن نیز خواهید شد. حداقل فایده حلقه، اختصار در کدنویسی است. مثلا به جای اینکه ده مرتبه یک عبارت را بنویسید، اینکار را با دستور ساده حلقه، به راحتی می‌توانید انجام دهید.

### کاردستی ۱: ساخت جدول ضرب، با حلقه

ابتدا لازم است بدانید که چگونه یک جدول (Table) در اچ‌تی‌ام‌ال ساخته می‌شود.

### ساختمان جدول

جدول، تشکیل شده از ستون افقی و ستون عمودی. تگ اصلی جدول به این صورت است: `<table>` `</table>`. مابین این دو، تگ ستون افقی `<tr>` `</tr>` و عمودی `<td>` `</td>` قرار می‌گیرد.

```

<table border="1" bordercolor="#000000" width="150"
height="100" bgcolor="black">
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>

```

```
<td></td>
</tr>
</table>
```

همانطور که می‌بینید، تگ جدول، چند پارامتر دارد که مشخصات آن را (اندازه، حاشیه، رنگ و ...) مشخص می‌کند.

کد:

```
<script>
function funFor()
{
    document.writeln("<Table Border=1>");
    for(a = 1; a <= 10; a++)
    {
        document.writeln("<TR>")
        for(b = 1; b <= 10; b++)
        {
            if ((a+b) % 2)
            {
                document.writeln("<TD bgcolor=blue>" + a * b + "
</TD>")
            }
            else {
                document.writeln("<TD>" + a * b + " </TD>")
            }
        }
        document.writeln("</TR>");
    }
    document.writeln("</Table>");
}
funFor();
</script>
```

## کاردستی ۲: اسکرول صفحه (حرکت صفحه از بالا به پایین)

**شرح عملیات:** ساخت این برنامه، بسیار ساده و راحت است.

. مواد لازم برای آن، یک حلقه، و یک متد اسکرول صفحه است، که در یک تابع قرار می‌دهیم.  
. یک تابع (funDown) برای پایین رفتن، و یک تابع (funUp) برای بالا آمدن. این توابع، به صورت مستقل، اجرا می‌شوند.

. در قسمت بدنه، دو دکمه (یا چیزی شبیه آن) قرار می‌دهیم. یکی برای فعال کردن تابع اول، و دیگری برای تابع دوم. به هنگام کلیک بر روی دکمه‌ها، توابع، اجرا می‌شوند؛ یعنی باید از رویداد onClick استفاده کنیم.

**کد:**

```
<script>
function funDown() // تابعی که صفحه را به طرف پایین می‌برد
{
    for (a = 1; a <= 600; a++) // حلقه‌ای که افزایش دارد
    {
        window.scroll (1, a); // متد اسکرول صفحه
    }
}
function funUp() // تابعی که صفحه را به طرف بالا می‌برد
{
    for (a = 600; a >= 1; a--) // حلقه‌ای که کاهش دارد
    {
        window.scroll (600, a);
    }
}
</script>
```

• برای دیدن نمونه عملی آن، اینجا کلیک کنید.

**واژه نامه:** این اصطلاحات را به خاطر داشته باشید؛ چون در برنامه نویسی، کاربرد دارد.

<table> = جدول.

Row = سطر، ستون افقی. <tr>

Cell = سلول، خانه‌های جدول. <td>

Window = شی صفحه است، که دارای متدهای مختلفی است.

Window.scroll = متد اسکرول صفحه. داخل پرانتز، دو پارامتر دارد که دو عدد هستند که موقعیت صفحه

را نشان می‌دهند. اولی، موقعیت فعلی صفحه؛ دومی، موقعیتی که باید به آنجا برسد.

Window.scrollTo - Window.scrollBy = دو متد دیگر از parent هستند. با تمرین، به دست

آوردید که چه کاری انجام می‌دهند.

Parent = معادل Window است و در مورد اسکرول، می‌تواند به جای آن قرار گیرد.

## حلقه (۲)

یارب! آن نوگل خندان که سپردی به منش  
گرچه از کوی وفا، گشت به صد مرحله دور  
هر که ترسد ز ملال، اندۀ عشقش نه حلال  
می سپارم به تو از چشم حسود چمنش  
دور باد آفتِ دور فلک از جان و تنش  
سر ما و قدمش، یا لب ما و دهنش

### حلقه for ... in

• این حلقه، فقط برای یک منظور استفاده می‌شود؛ به دست آوردن خصوصیات اشیاء. اگر به خاطر داشته باشید، در جلسه هشتم، به مباحث عمومی برنامه نویسی، یعنی متد، خصوصیات و رویداد اشیاء، پرداختیم. حتماً به آن درس، مراجعه کنید.

سوال مهم این است: چگونه خصوصیات و رویداد یک شی را به دست آوریم؟ این پرسش، ممکن است برای هر کسی، پیش آماده باشد. البته یک روش، مراجعه به پایگاه وب و یا کتابهای مرجع است؛ ولی راه راحتتری وجود دارد که امروز به شما می‌گویم. از خود جاوا پرسیم. بله، بهترین و ساده‌ترین راه، این است که از خود جاوا اسکریپت، سوال کنیم که خواص فلان شی \_ مثلا `window` - چیست. به وسیله همین حلقه، از جاوا درخواست می‌کنیم که هرچه درباره یک شی، وجود دارد، به ما بگوید.

### ساختمان حلقه

```
for (var x in object)
{
    document.write("<li>object." + x + " = " + object [x]);
}
```

مشاهده می‌کنید که ساختمان این حلقه، با حلقه `for` که قبلاً دیده‌ایم، تفاوت دارد.

`X` = خصوصیات شی. یک متغیر است که بیانگر تمام خصوصیات شی است.  
`in` = کلمه‌ای کلیدی؛ به این معنی که هرچه خصوصیات (`X`) در این شی است.  
`Object` = به جای این کلمه، نام شی را بنویسید؛ مثلا `window, document`

[x object] = شیئی که نوشته‌ایم، تبدیل به آرایه می‌شود؛ یعنی یک متغیر که دارای چندین مقدار است.

## تمرین

می‌خواهیم، خصوصیات شی سند (window) را به دست آوریم

کد:

```
<script>
document.write("<ol>");
for (var x in window)
{
    document.write("<li>window." + x + " = " + window[x]);
}
</script>
```

چیزی که جاوا به شما نمایش خواهد داد، این است:

1. window.onbeforeunload = null
2. window.onafterprint = null
3. window.top = [object]
4. window.location = file:///F:/Amozesh%20java/14.htm
5. window.parent = [object]
6. window.offscreenBuffering = auto
7. window.frameElement = null
8. window.onerror = null
9. window.screen = [object]
10. window.event = null
11. window.clipboardData = [object]
12. window.onresize = null
13. window.defaultStatus =
14. window.onblur = null
15. window.window = [object]
16. window.onload = null

17. window.onscroll = null
18. window.screenTop = 112
19. window.onfocus = null
20. window.Option = [object]
21. window.length = 0
22. window.onbeforeprint = null
23. window.frames = [object]
24. window.self = [object]
25. window.clientInformation =
26. window.external =
27. window.screenLeft = 0
28. window.opener = undefined
29. window.onunload = function anonymous() { setCookie3() }
30. window.document = [object]
31. window.closed = false
32. window.history = [object]
33. window.Image = [object]
34. window.navigator =
35. window.status =
36. window.onhelp = null
37. window.name =

1. window.onbeforeunload = null
2. window.onafterprint = null
3. window.top = [object]
4. window.location = file:///E:/Web/Weblog/JScript/15.02.htm
5. window.parent = [object]
6. window.offscreenBuffering = auto
7. window.frameElement = null
8. window.onerror = null
9. window.screen = [object]
10. window.event = null
11. window.clipboardData = [object]

12. window.onresize = null
13. window.defaultStatus =
14. window.onblur = null
15. window.window = [object]
16. window.onload = null
17. window.onscroll = null
18. window.screenTop = 119
19. window.onfocus = null
20. window.Option = [object]
21. window.length = 0
22. window.onbeforeprint = null
23. window.frames = [object]
24. window.self = [object]
25. window.clientInformation =
26. window.external =
27. window.screenLeft = 4
28. window.opener = undefined
29. window.onunload = null
30. window.document = [object]
31. window.closed = false
32. window.history = [object]
33. window.Image = [object]
34. window.navigator =
35. window.status =
36. window.onhelp = null
37. window.name =

## تحلیل

• همانطور که می‌بینید جاوا، ۳۷ خط، نوشته است که مخلوطی از اشیاء، خصوصیات و رویدادهای پنجره (window) است. چرا به شکل درهم و نامنظم نوشته شده، من نمی‌دانم! ولی تشخیص این سه، کار مشکلی نیست.

شی: هر کدام که مقابلش **object** نوشته شده، یک شی دیگری است که زیر مجموعه **window** می باشد؛ مانند **top**.

رویداد: هر کدام که با **on** شروع شده، رویداد (**event**) است؛ مانند **onload**. مقابل رویدادهایی که از آنها در پنجره، استفاده نکرده ایم، **null** نوشته می شود؛ در غیر این صورت، نوع عملیات را می نویسد. خصوصیت: هر چیزی به غیر از اینها، خصوصیات **window** است، که معمولا چیزی مقابلش نوشته شده است؛ مانند **screenTop**، که موقعیت پنجره را از بالا، نشان می دهد. اگر پنجره را تغییر مکان دهید و رفرش کنید، مقدار آن تغییر می کند.

● برای دیدن نمونه های بیشتر، اینجا کلیک کنید.

### پرسش و پاسخ

. با عذرخواهی از کسانی که پاسخشان دیر شد.  
. هر یک از دوستان که جواب مرا کافی یا صحیح ندانست، حتما تذکر دهد.

### ● پاسخ:

تشکر از دوست عزیز، ایرانی، که پاسخ تمرین جلسه قبل را ارسال کرده اند. بله، به نظر می آید نمی شود به غیر از حلقه، این عملیات را انجام داد. منظور من هم همین بود، که متوجه اهمیت و منحصر به فرد بودن حلقه بشوید.

```
var a = 0;
function funFor(){
    if (a < 5)
    {
        a++;
        alert ("پیام" + a);
    }
    setTimeout ("funFor()", 1000);
}
funFor();
```

شاید این حداکثر کاری باشد که می‌توان در تکرار یک دستور انجام داد. مسلماً، بهترین و ساده‌ترین راه، استفاده از حلقه است.

### پرسش:

سلام. اول تشکر کنم به خاطر وبلاگ خوبتون. چطوری میشه خروجی یک کد جاوا اسکریپت را در یک جدول گذاشت؟ یا مثلاً ورودی یک کد جاوا را از یک ادیت (input) گرفت. یک دنیا ممنون.

### پاسخ:

سلام دوست عزیز! تشکر از لطفتان.

۱- یک راه، قرار دادن تگ اسکریپت، درون جدول است. یعنی کل کدها را در بدنه و داخل همان جدول بگذارید و آنگاه از دستور `document.write` برای نوشتن نتیجه، استفاده کنید. برای آشنایی بیشتر، به جلسه ۱۱، "صدا کردن تابع" مراجعه کنید.

روش دیگر این است که از دستور `innerText` استفاده کنید که من در "یک پروژه" به کار برده‌ام.

۲- برای گرفتن داده، از یک شی، به تمرین جلسه ۱۲ مراجعه کنید. همانجا که از `Menu.value` استفاده کرده‌ام. `input` دارای یک `value` است؛ همانجا که متن خود را وارد می‌کنید.

### پرسش:

دوست عزیز! من تازه شروع کردم به یادگیری؛ اما از اونجایی که من با `C, Delphi` به صورت حرفه شغلی کار کردم. این درسهای شما را خوب می‌فهمم. ولی سوالم اینه که با چه ادیتوری باید اینها رو نوشت؟

### پاسخ:

سلام حسین آقا! خوش آمدید.

۱- اگر برایتان ممکن است، از این به بعد پیامتان را در آخرین درس بنویسید، تا زودتر متوجه آن بشوم.

۲- حق با شماست. جاوا اسکریپت چون از لحاظ ساختاری، به `C` و دلفی شبیه است، یادگیری و فهمش برایتان راحت است.

۳- در جلسه سوم در این باره توضیح داده‌ام. "نوشتن این زبان، احتیاج به محیط خاصی ندارد. مثلاً زبان ویژوال بیسیک را، باید برنامه ویژوال بیسیک را نصب، و در محیط آن بنویسیم؛ ولی زبانهای وب، چنین محدودیتی

ندارند. مثلاً شما در یک متن ساده (Text)، می‌توانید به راحتی، هم HTML بنویسید و هم Java و هم PHP. به آن مراجعه کنید.

### پرسش:

لطفاً آموزش این برنامه را بگویید. دادن پیغام در روزهای ۵، ۹، ۱۲ هر ماه. فارسی؛ نه انگلیسی.

### پاسخ:

سلام. برای ساخت این برنامه، شما به چند چیز، احتیاج دارید. یک متد دریافت روزهای ماه (getDate)، و یک پیام. خوشبختانه با تمام این عناصر، در جلسات قبل، آشنا شده‌اید. کافی است تابعی بسازید که در روزهای مقرر، یک پیام بدهد. نتیجه کار را به من اطلاع دهید.

### پاسخ یکی از دوستان:

سلام استاد! در مورد اون برنامه‌ای که دوستانمون از شما سؤال کرده بود، من هم اونو به عنوان یک تمرین انجام دادم. لطفاً آگه راه‌های دیگری هم هست راهنماییم کنید. ممنون.

```
var now = new Date()
var A = now.getDate()
if (A == 6) alert("امروز قرار دارید");
if (A == 12) alert("امروز قرار دارید");
if (A == 26) alert("امروز قرار دارید");
```

• برای جلسه آینده یک بازی زیبا - که با جاوا اسکریپت ساخته شده - آماده کرده‌ام؛ اگر خطوط کُند اینترنتی، مانع نشوند.

## فراخوان

### تغییرات:

- در همین مقاله، تغییراتی داده شده. (پرسش و پاسخ، بازی)
- در پرسش و پاسخ جلسه قبل، تغییراتی داده شده.
- در طراحی این وبلاگ، با کمک جاوا، یک تغییر ظریفی انجام شده. پیدا کنید.

خوشحالم که این شور و شوق و تحرک را در شما می‌بینم. آرزویم این بوده که پایگاهی وجود داشته باشد که مرجع و منبعی برای علاقمندان باشد. من که فاصله دارم با این هدف؛ ولی به شما دوستان امیدوارم. به قول پیر مردا: "چشم امیدم به شماست".

پیشنهاد می‌کنم به منظور رسیدن به این هدف، این جمع دوستانه و صمیمی را حفظ کنید. هر مطلب جالبی، مقاله جدید، روش آموزشی نوین، یا کد زیبایی به چشمتان خورد، به نام خودتان در این وبلاگ، قرار بدهید.

به امید خدا، یا جای بهتر و بزرگتری پیدا می‌کنیم؛ یا شعبه‌هایی در همین پایگاه، ایجاد می‌کنیم که به صورت زنجیره‌ای فعالیت کنند. مثلاً:

وبلاگ آموزشی

وبلاگی مخصوص کد و برنامه‌های آماده

وبلاگ اخبار و تازه‌های جاوا اسکریپت، فناوری و نرم افزارهای کمکی

تقریباً در هر سه مورد، کارهایی انجام داده‌ام، و بانکی از کدها، و نرم افزارهای این زبان را تهیه کرده‌ام.

● روش من، همانطور که از اول گفته‌ام، این است که شما به صورت خودکار دست به کار شوید؛ و به جوابتان برسید. یک علت تاخیر نوشتن مقاله جدید یا در پاسخ به سوالها، همین است؛ که فرصت تمرین بیشتر داشته باشید و کسانی که به هر دلیلی از درس عقب مانده‌اند، خودشان را برسانند.

داخل پراتنز: یکی از چیزهایی که باعث ناامیدی بعضی از دوستان تازه کار می‌شود، این است که از دیگران عقب مانده‌اند و یا دیرتر درس را یاد می‌گیرند. منتظر می‌مانم تا برسند. دوستان دیگری که از هوش و استعداد، یا فرصت بیشتری برخوردارند، می‌توانند با مراجعه به پایگاه‌ها و کتابهایی که معرفی شده، پیش بروند.

به قول اون سرباز زخمی و فداکار: شما برید؛ من اینجا می‌مونم.  
و به قول فرمانده همون سرباز: یک فرمانده شجاع، هیچ وقت سربازش رو تنها نمیذاره.

به حرفهایی که گفتم فکر کرده و راهنمایی‌های خود را از من دریغ نفرمایید.

## پرسش و پاسخ:

### پرسش:

سلام. یه سوال: اگه خواستیم مثلا با کلیک روی یک دکمه یه پاراگراف جدید درست بشه چکار باید کرد؟

### ● پاسخ:

سلام

۱- یک خواهش داشتم. سوالت را واضح بنویس؛ مثلا بگو در کجا می‌خواهی این کد را استفاده کنی. با این کار، مخاطب شما، بهتر منظورتان را بهتر درک می‌کند.

۲- با تشکر از ایرانی که لطف کردند و پاسخ دادند. آفرین به شما.

۳- راه حلی که من دارم خدمتان می‌نویسم.

```
<body>
<input type="button" value="پاراگراف" onclick = "funAdd()" >
<script>
function funAdd()
{
    document.body.innerHTML += "<li><p>پاراگراف"
}
</script>
```

</body>

### ● پاسخ:

سلام دوست عزیز mreza! نامه شما به دستم رسید. ظاهراً شما تمایل دارید با قدرت و کارایی این زبان آشنا شوید. نمی‌دانم قبلاً با چه زبانی کار کرده‌اید؛ اما نظر شما را به این موضوع جلب می‌کنم که: عناصر و اشیاء موجود در یک صفحه، ماهیت و کارایی مختلفی دارند. بعضی از آنها قابل ویرایش نیستند و به اصطلاح ReadOnly (فقط خواندنی) هستند؛ و بعضی اجازه تغییر و اصلاح و یا نوشتن متن را به شما می‌دهند. مانند جعبه متن (Textbox). توجه به این نکته، شاید شما را در رسیدن به هدفتان، کمک کند.

### ● پاسخ:

آقا حامد سلام. نامه‌تان به دستم رسید. تشکر از صبر و همتتان. ببخشید اگر جسارتی شد. صلاح مملکت خویش، خسروان دانند. تنها یک نکته. فرمودید که متأسفانه بازدید کننده کم است. عرض می‌کنم که چه کمی؟ کم کمی، یا کم کیفی؟ مطالب و بلاگ شما، تخصصی است. یک نفر بازدید کننده از چنین وبلاگی، برابری می‌کند با ۱۰ نفر بازدید کننده وبلاگ یا سایتی که پر از شعر و عکس و ترانه و جک است. علاوه بر اینکه موافق نظرتان نیستم، و امیدوارم که وبلاگتان، به مرور علاقمندان خاص خود را پیدا کند؛ شما را به رعایت نکاتی که عرض کردم و همچنین تبلیغ مناسب از وبلاگتان، توصیه می‌کنم. خدانگهدار

### ? پرسش:

با سلام. شما می‌توانید میلی ثانیه را با دستور زیر به دست بیاورید؟

getMilliseconds()

### ● پاسخ:

سلام. تا آنجا که من اطلاع دارم، این دستور هنوز ساخته نشده است. تا آن زمان، می‌توانید از دستور ثانیه استفاده کنید؛ و اگر خواستید آنرا تقسیم به ده، صد، هزار یا میلیون نمایید.

### بازی:

بازی که قولش را داده بودم، امروز آماده کرده‌ام. اما چون اینجا، وبلاگ است و محیط وبلاگ - به قول آقا حامد - محدودیتهای فراوانی دارد، آماده سازی این بازی، یا پروژه‌های عملی دیگر، طول می‌کشد. اگر با این

مشکل آشنا باشید و آنرا در نظر بگیرید، متوجه خواهید شد که با چه زحمتی و با چه ترفندهای مختلفی، یک صفحه خارجی برای شما درست می‌شود.

## فرم

اینبار به مقوله‌ای می‌پردازیم که هم کاربردی است و هم محلی برای تمرین دروس قبلی.

اگر به خاطر داشته باشید، گفته شد که قسمتی از برنامه نویسی، با داده‌هایی سر و کار دارد که از کاربران دریافت می‌شود. دریافت اطلاعات، راه‌های مختلفی دارد که مهمترین و معروفترین آنها، استفاده از فرم است.

روزانه، فرمهای مختلفی را پر می‌کنیم. فرم ثبت نام در یک اداره، فرم باز کردن حساب در بانک، فرم ثبت نام در قرعه‌کشی، و ... حتما در اینترنت هم با امثال این فرمها، روبرو شده‌اید. مثلا فرم عضویت در یک پایگاه، فرم ثبت یک ایمیل در یاهو، فرم ارسال نامه، یا مثلا فرم نظرخواهی همین وبلاگ. آیا تا به حال، به طرز کار این فرمها فکر کرده‌اید؟ برای شما که یک برنامه‌نویس هستید، توجه به ساختمان و ماهیت فرم - و اشیاء دیگر - لازم است.

«پایگاههای وب، از فرمها برای منظوره‌های مختلفی استفاده می‌کنند. از قبیل: ثبت نام کردن کاربران، و یا ورود آنها به فضاهای خاص وب (Login)، دریافت نظرات و فرمهای نظرسنجی، و یا خرید و فروش آنلاین.»

### ساختمان فرم

هر فرمی، عملیات خود را - که دریافت و ارسال داده باشد - در سه بخش، انجام می‌دهد.

اسکلت و ظاهر: این بخش - که تشکیل شده از تعدادی جعبه متن (فیلد) و دکمه - توسط اچ‌تی‌ام‌ال ساخته می‌شود. البته با نرم‌افزارهایی - که قبلا نام بردیم - این کار، سریعتر و دقیقتر انجام می‌شود. این قسمت، مکان نوشتن اطلاعات، توسط کاربر است.

خطایاب: این بخش، تخصص جاوا اسکریپت است. یک حد میانی بین اچ‌تی‌ام‌ال و زبان سروری. در این قسمت، اطلاعات وارد شده توسط کاربر، کنترل می‌شود. مثلا تمامی فیلدهایی که باید نوشته شود، پر شده یا نه؟ و آیا ترکیب ایمیلی که وارد شده، درست است؟ در صورت وجود خطا، با اخطار به کاربر، مانع از ارسال مطالب به صورت ناقص یا نادرست می‌شود.

ارسال و ذخیره اطلاعات: این قسمت، به عهده یک زبان سروری مانند asp و php است، و از حیطة مسؤولیت جاوا، خارج است.

با توجه به این تقسیم بندی، وظیفه جاوا، مشخص شده و دقیقا می دانیم که از او چه می خواهیم.

### داخل پرانتز:

با توجه به مطالبی که گفته شد، بدیهی است که از بعضی از فرمها نمی شود در وبلاگ استفاده کرد؛ یا حتی در سایتهایی که فضایشان را به صورت رایگان، تهیه کرده اند؛ چون اینگونه فضاها، اجازه نوشتن و اجرای زبان سروری را نمی دهند.

### اجزای فرم

فرمها، بسته به نیاز ما، می توانند ساده یا پیچیده باشند. کمترین حد فرم، یک جعبه متن و یک دکمه ارسال (Submit) است؛ همانند فرم ثبت نامی که در همین وبلاگ، قرار داده ام.

دکمه Reset که یک دکمه اختیاری است، برای پاک کردن فرم است. این ساده ترین شکل فرم است.

### اسکلت فرم (کد HTML)

```
<form method="post" action="test.htm" name="Form1"
onSubmit="return funForm()"><!-- شروع فرم -->
نام:
<input type="text" value="" name="Name"><!-- جعبه متن -->
<input type="submit" value="Submit" name="B1"><!-- دکمه ارسال -->
<input type="reset" value="Reset" name="B2"><!-- دکمه پاک کردن -->
</form><!-- پایان فرم -->
```

method = نوع ارسال اطلاعات را معین می کند، که دو نوع است: **post, get**. این دو روش با تفاوتیهایی، عملیات ارسال را انجام می دهند. روش پُست، عمومی تر است و از لحاظ امنیتی، برتری دارد. اگر متد را ننویسید، از پُست، استفاده می شود.

**action** = محلی را نشان می‌دهد که اطلاعات، برای پردازش، به آنجا ارسال خواهد شد؛ یعنی آدرس یک صفحه است، که در اینجا برای مثال، **test.htm** را انتخاب کرده‌ام.  
**name** = نام فرم.  
**onSubmit** = یک رویداد است. تابعی را که در جاوا اسکریپت، نوشته‌اید را صدا می‌کند.  
فراموش نکنید که برای هر جزء، نامی بگذارید. تا اینجای کار، اسکلت و ظاهر فرم، ساخته شده است.

### خطایابی (کد جاوا)

- از این به بعد، به بخش خطایابی می‌پردازیم. اول باید در نظر بگیریم که خطا از نظر ما چیست.
    - مثلاً در یک فرم ارسال ایمیل، نباید آدرس ایمیل غلط باشد.
    - یا مثلاً متنی که نوشته می‌شود، نباید بیشتر یا کمتر از فلان حد باشد.
    - یا موردی که باید نوشته و مشخص شود، خالی مانده.
- پس باید طوری برنامه نویسی کنیم که مانع انجام این اشتباهات بشویم.
- در فرم ساده ما، خطا، زمانی است که اسمی نوشته نشود، و فرم خالی ارسال شود. پس باید با کمک جاوا بنویسیم که اگر فیلد نام، خالی بود، مانع ارسال فرم شده و پیام بدهد.

```
<script>
function funForm()
{
    if (Form1.Name.value == "")
    {
        alert ('!نام خود را بنویسید');
    }
}
```

```
Form1.Name.focus();  
return false;  
}  
}  
</script>
```

## ارسال و ذخیره اطلاعات

بعد از دریافت صحیح اطلاعات، باید آن را به جایی که قرار است ذخیره شود، بفرستیم. در اینترنت، به چند صورت می‌شود اطلاعات را ذخیره نمود.

ذخیره در کوکی: اطلاعات کوتاه و ضروری را در آن ذخیره می‌کنند. به امید خدا در جلسات آتی، به آن خواهیم پرداخت.

ذخیره در پرونده‌های معمولی: مثلاً در پرونده متنی (Text). نسبت به کوکی، قادر به ذخیره اطلاعات بیشتری است؛ اما به لحاظ اینکه برای ساخت، تغییر و حذف یک پرونده یا پوشه، احتیاج به مجوز از طرف سیستم شماست، و این کار، به خاطر جلوگیری از نفوذ ویروس و خرابکاران (هکر)، به سختی انجام می‌پذیرد، مقرون به صرفه نیست.

ذخیره در بانک اطلاعاتی: بهترین روش ذخیره اطلاعات است؛ هم از لحاظ ذخیره حجم بالای اطلاعات، هم ترتیب و نظم، هم خواندن راحت و سریع آنها. اساس کار بانک اطلاعات، استفاده از جدول است؛ جدولی که تشکیل شده از ستون و سطرهاى مختلف؛ مانند جداولی که در بسیاری از جاها، دیده‌اید.

استفاده از بانک اطلاعاتی، مناسبترین روش است که معمولاً اطلاعات فرمها، به آن ارسال می‌شود؛ اما بانکها، زبان ما - جاوا اسکریپت - را نمی‌فهمند، و زبان خاص خودشان را دارند؛ یعنی زبانهای سروری که قبلاً گفته شد.

وبلاگها و پایگههایی که رایگان تهیه شده‌اند، اجازه اجرا این گونه زبانها را نمی‌دهند.

- به امید خدا بعد از پایان درس جاوا اسکریپت، اگر دوستان علاقمندی باشند، به بحث درباره یکی از زبانهای سروری (asp, php) خواهیم پرداخت؛ حداقل تا جایی که بتوانید خودتان یک بانک را ساخته و اطلاعاتان را در آن ذخیره کنید.

- ظاهراً بحث فرم - تا آنجا که به جاوا مربوط می‌شده - در اینجا به پایان رسیده است. این جلسه، هدیه‌ای به دوستان عزیزی که تقاضای درس عملی و کاربردی را داشته‌اند. در پایان، شما را با عناصر دیگری که ممکن است در فرم، استفاده شوند، آشنا می‌کنم.

## واژه‌نامه

**Get - Post** = تفاوت بین این دو، در چگونگی ارسال اطلاعات از فرم، به اسکریپت پردازشگر است. اگر از روش گت، استفاده کنید، مقدار اطلاعاتی که می‌توانید ارسال کنید، محدود است.

با استفاده از روش **get**، اطلاعات به دست آمده از فرم، به عنوان بخشی از یک **URL**، انتقال می‌یابد؛ اما در روش **post**، اطلاعات، در حقیقت به صورت نامرئی، منتقل می‌شود. اگر برای مثال، در روش گت، کاربر، یک رمز عبور را در فیلد پسورد، بنویسد، و فرد دیگری نیز در حال مشاهده صفحه نمایش کاربر باشد، به راحتی می‌تواند از رمز عبور کاربر، آگاه شود. به عبارت دیگر، این روش، در سطح امنیتی پایین‌تری از روش پست می‌باشد.

صفحه‌ای که حاوی فرمی با مشخصه گت است، می‌تواند نشانه گذاری (**Bookmark**) شود؛ در حالی که در روش پست، این کار، ممکن نیست.

## عناصر فرم

- همیشه پاسخها، به صورت تشریحی نیست. گاهی اوقات، تستی بوده و جوابهای از پیش تعریف شده دارد. مثل سوال از جنسیت کاربر، یا شهر محل سکونت. در این موارد، باید از عناصر دیگه فرم استفاده نمود؛ مانند چک باکس، یا دکمه رادیویی.

گروه

دکمه

Button

Button2

کد این دو دکمه، با هم تفاوت دارد

Submit

دکمه ارسال

Reset

دکمه پاک کردن



Checkbox - چک باکس

دکمه رادیویی



مرد



ززن

گزینه 1



Go

ComboBox-JumpMenu - فهرست آبخاري

گزینه 1

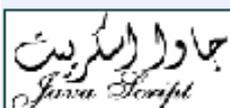
گزینه 2

ListBox - لیست باکس

گزینه 1

گزینه 2

ListBox-Multiple - لیست باکس چند انتخابی



Button-Image - دکمه عکسی

**File** - دریافت فایل

**Text** - متن

**Password** - پسورد

**Hidden** - متن مخفی

**TextArea** - متن زیاد

## جلسه ۱۶

### دستور زبان (آیین نگارش)

تعجب نکنید. نه شما اشتباه آمدید، و نه من اشتباه نوشتم. موضوع، همان برنامه‌نویسی است. این جلسه، اختصاص دارد به یک بحث مختصر و مفید، که از ظرافت بالایی برخوردار است؛ همان ظرافتی که یک نویسنده، در نوشتن مقالات و داستانهای خود، به خرج می‌دهد. چه از لحاظ دسته بندی و مرتب بودن فصول و ابواب، و چه از لحاظ فهرست بندی، و چه از لحاظ مراعات قواعد دستوری و آیین نگارش.

بعد از این همه تلاش و تمرین، لازم است بدانیم چگونه برنامه‌ای بنویسیم که کمترین عیب و ایراد را داشته باشد. این مبحث - جز موارد خاصی - در میان زبانهای برنامه‌نویسی، عمومی بوده و اختصاصی به جاوا اسکریپت ندارد.

#### ۱- غلط املائی

هنگام ملاحظه تمرینات دوستان، که برایم ارسال کرده‌اند، متوجه این نکته شدم که قسمتی از اشتباه آنان، مربوط به غلط املائی می‌شود. دستورات، مرتب و صحیح، نوشته شده بود، اما وجود یکی دو غلط املائی، مانع اجرای صحیح برنامه می‌شد. مثلاً

در نوشتن window، دقت کنید. ویندوز، به صورت مفرد است. با نام سیستم عامل ویندوز، اشتباه نگیرید. یا document و Mouse، جای حروف o و u را اشتباه نکنید. یا کلماتی که از لحاظ املائی، نزدیک هم هستند؛ مانند white (سفید) with (با) While (حلقه).

#### ۲- کلمات رزرو شده و کلیدی

خیلی از کلمات برنامه‌نویسی، کلمات رزرو شده هستند؛ یعنی کلماتی که از قبل در آن زبان، مورد استفاده قرار گرفته و دارای معنای خاصی هستند. از این کلمات نمی‌شود به عنوان نام چیز دیگری، مانند تابع، استفاده کرد. در صورت عدم مراعات این قاعده، برنامه شما، دچار مشکل می‌شود.

### ۳- کلمات کلیدی

در نوشتن این کلمات، خیلی دقت کنید؛ مخصوصا در بزرگ و کوچک بودن حروف. این قاعده کلی را در مورد جاوا اسکریپت، به یادگار داشته باشید: همه متد و دستورها، در جاوا اسکریپت، با حرف کوچک، نوشته و شروع می‌شود؛ الا موارد خاصی مانند **Array**. البته شاید یک کلمه ترکیبی باشد؛ چون همیشه اینگونه نوشته می‌شود **.new Array**

### ۴- دستورات ترکیبی

یعنی دستوراتی که از دو یا سه کلمه، تشکیل شده‌اند. مانند **setTimeout, onMouseover** که از سه کلمه، ترکیب شده‌اند. اولین کلمه این دستورات، با حرف کوچک شروع شده، و دومین کلمه، با حرف بزرگ.

### ۵- نام گذاری

در زبانهایی مانند ویژوال بیسیک، در نامگذاری اشیاء، دقتی به خرج داده می‌شود، که در زبانهای وب، ندیده‌ام. به روشی که عرض می‌کنم توجه نموده و از آن استفاده کنید.

اسم بی‌مسمی شنیده‌اید؟ اسمی که با شخص یا شیئی نامیده شده، سنخیتی ندارد؛ مثل اینکه به یک سگ سیاه، بگوییم سفید برفی. اسمی که برای اشیا یا دستورات خود، انتخاب می‌کنید، باید جنس و کار آن را نشان دهد.

مثلا می‌خواهم تابعی بنویسم که مطالب یک فرم را ارسال کند. نام دکمه ارسال را **btnSubmit** گذاشته، و نام تابع را **funSubmit** می‌گذارم.

**btn** = مخفف **button** (دکمه)

**fun** = مخفف **function** (تابع)

استفاده از مخفف نوع اشیا، در راهنمایی شما یا کسانی که از برنامه‌یتان بازدید می‌کنند، بسیار کمک می‌کند. مطمئن باشید در آینده نه چندان دور، که به برنامه خود مراجعه می‌کنید، حافظه، شما را یاری نخواهد نمود تا جزئیات را به خاطر بیاورید. آنگاه مجبور خواهید شد مانند غریبه‌ها، از اول، کل برنامه را مطالعه کنید.

## ۶- مرتب نویسی

کدها باید طوری نوشته شود که خوانا و واضح باشد. شاید در برنامه‌های کوتاه، ضرورتی برای این کار دیده نشود؛ اما در کدهای عریض و طویل و پیچیده که از دهها تابع ساخته شده‌اند، این امر، ضروری است.

به ترتیب و چینش این کدها دقت کنید:

```
<script>
function funForm()
{
    if (Form1.Name.value == "");
    {
        alert ('پیام');
        Form1.Name.focus();
        return false;
    }
}
</script>
```

رعایت سر شاخه و زیر مجموعه، در این برنامه، نمایان است. یک مجموعه داریم که تابع ما است. داخل آن تابع، یک شرط نوشته شده، و داخل آن شرط، چند دستور. با فاصله دادن آنها از کناره سمت چپ صفحه، آنها را از هم مجزا می‌کنیم. به قرار گرفتن گروه‌ها، دقت کنید.

## ۷- پایان دستور

در پایان هر دستور، از نقطه ویرگول (;) استفاده کنید. هرچند در جاوا اسکریپت، این کار ضرورتی ندارد؛ اما مراعات آن، ضروری ندارد. نه تنها پایان هر دستوری را مشخص می‌کند، بلکه ما را در زبانهای دیگر - مانند php که این کار لازم است - آماده می‌کند.

## معرفی نرم‌افزار

### JavaScript Code Improver 1.00 - ۱

این برنامه، کدهای نوشته شده جاوا اسکریپت را مرتب می‌کند. همانطور که در نکته ۶ گفته شد، مرتب نویسی، باعث خوانا شدن کدها می‌شود؛ اما همیشه از کدهای خود استفاده نمی‌کنیم؛ بلکه در بسیاری مواقع، از برنامه‌های آماده در اینترنت یا نوشته شده توسط دوستان خود، بهره‌برداری می‌کنیم. در این صورت، برای اینکه وقت زیادی را صرف مرتب کردن آنها نکنید، بهتر است این نرم‌افزار را به کار برده تا ظرف چند ثانیه، کدها را شسته رفته، به شما تحویل دهد.

این برنامه را - که به زبان فارسی ترجمه شده است - می‌توانید از این آدرس:  
(<http://www.javaniran.com/service/download/JavaScriptCodeImprover.exe>)

دریافت کنید.

## ۲- JavaScript Maker 2.7

با این برنامه، می‌توانید کدهای نوشته شده در `htm` را تبدیل به `js` کنید. به این صورت که تمام کدهای شما، صحیح و بدون هیچگونه خطا، داخل یک تابع قرار می‌گیرد. برای اجرا صفحه وب تبدیل شده، کافی است آن تابع را صدا کنید.

این روش، در جایی به کار می‌آید که فضایی برای ذخیره یک صفحه مستقل ندارید؛ مانند وبلاگ. در همین وبلاگ، بارها از این روش، برای ارایه مثالها، استفاده شده است.

«دانلود» (<http://www.byte-sizesoftware.com/jscp27.zip>)

## ۳- JavaScript Animator Express v1.10

با این برنامه، می‌توانید به راحتی چند عکس را به صورت اسلاید، نمایش دهید؛ البته با کمک زبان جاوا اسکریپت. برای آشنایی بیشتری و دریافت آن، به این آدرس (<http://www.nirsoft.net/utis/jsae.html>)، مراجعه کنید.

## ۴- HTMLAsText v1.01

تبدیل اچ‌تی‌ام‌ال (`htm`) به متن (`txt`). برای آشنایی بیشتر و دریافت آن، به این آدرس (<http://www.nirsoft.net/utis/htmlastext.html>)، مراجعه کنید.

## جلسه ۱۷

### موس و صفحه کلید

موس و صفحه کلید، به عنوان گیرنده‌های رایانه هستند که به وسیله آنها مطالب را وارد می‌کنیم. مثلاً برای نوشتن این متن، از صفحه کلید استفاده کرده‌ام. حرف ت را فشردم، تا در این صفحه، نوشته شد. البته میان فشردن کلید و نوشته شدن در صفحه - که بلافاصله انجام می‌شود - عملیاتی انجام می‌گردد که از دید ما پنهان است.

هر کلید، یک کد عددی دارد که با فشردن، از صفحه کلید به بخش مربوطه در رایانه، فرستاده می‌شود. آنجا بعد از پردازش، تبدیل به حرف معادل آن کد شده، و در متن، نمایش داده می‌شود. اگر این روند را خوب درک کنیم، می‌توانیم بدون استفاده از صفحه کلید، تایپ کنیم؛ فقط کافی است بدانیم که دستورات لازم آن چیست.

### استخراج کُد کلید (Key Code)

- صفحه کلید

استخراج کُد کلید (Key Code)، کار راحتی است. در زبان جاوا اسکریپت، برای به دست آوردن آن، از این دستور استفاده می‌شود: «`window.event.keyCode`»

### تمرین

برنامه‌ای بنویسیم که کد هر کلید را در یک جعبه متن، نشان دهد.

```
/** صفحه کلید */
function funKeyCode(){
    var key = window.event.keyCode;
    inCode.value = key;
}
```

تنظیم رویداد `document.onkeypress=funKeyCode; //`

در قسمت بدنه، جعبه متنی قرار دهید که محل نمایش کد باشد.

```
<input name="inCode" type="text" value="" size="5" readonly>
```

کد هر کلیدی را که فشار دهید، اینجا نوشته می‌شود:

• موس

موس نیز مانند صفحه کلید عمل می‌کند؛ متنها دارای کلیدهای کمتری است. از این کد `«window.event.button»` می‌توانید برای استخراج کد کلید موس، استفاده کنید.

**تمرین**

برنامه‌ای بنویسیم که کد هر کلید را در یک جعبه متن، نشان دهد.

```
/***/کلیک موس
function funClick()
{
  if (document.all)
  {
    if (window.event.button)
    {
      inButton.value = window.event.button;
      return false;
    }
  }
}
document.onmousedown=funClick; // تنظیم رویداد
```

در قسمت بدنه، جعبه متنی قرار دهید که محل نمایش کد باشد.

<input name="inButton" type="text" value="" size="5" readonly>

کُد کلیک موس :

#### • اشاره گر

اشاره گر (Cursor) - که کاربران معمولاً و به اشتباه، به آن موس هم می گویند - همان شکلی است که با حرکت موس، در مانیتور، حرکت می کند و نمادی از انگشت و دست شماست برای اشاره به اشیاء داخل صفحه.

تاریخ تولد اشاره گر و موس، به ابتدایی ترین سیستم عامل ویندوز برمی گردد. قبل از آن، سیستم عامل داس بود و آن محیط سیاه و مخوف. نه اثری از رنگ بود، نه صوت و نه موس.

اشاره گر، دارای شکلهای مختلفی است، و در حالات گوناگون، تغییر شکل می دهد. مثلاً گاهی به شکل ساعت شنی، و گاهی به صورت یک فلش دو سر یا چهار سر، یا علامت سوال، تبدیل می شود.

این تغییر شکلهای، همه به دست توانای شمای برنامه نویس، میسر می شود و بس. خوشبختانه در وب نیز می شود شکل دلخواه و مورد نیاز به اشاره گر داد؛ کافی است که کد هر شکل را بنویسید.

#### کُد

قبل از نوشتن کد، توضیحی باید عرض کنم، و آن این است که تغییر شکل اشاره گر، به صورت مستقیم به جاوا اسکریپت مربوط نمی شود؛ بلکه مربوط به مبحث استایل (Style) است؛ اما می شود در جاوا از آن به دلخواه استفاده نمود.

#### استایل

استایل (Style)، تگی است مانند اسکریپت (Script) که درون سر (Head) نوشته می شود. همانند جاوا، دو قسمت دارد: ۱- نوشتن کد ۲- فراخوانی کد.

۱- نوشتن کد. کد را داخل تگ استایل و درون سر، می نویسیم.

```
<html>  
<head>
```

```

<style>
#Auto {Cursor: auto}
#Crosshair {Cursor: crosshair}
#Default {Cursor: default}
#Pointer {Cursor: pointer}
#Hand {Cursor: hand}
#Move {Cursor: move}
#e-resize {Cursor: e-resize}
#ne-resize {Cursor: ne-resize}
#nw-resize {Cursor: nw-resize}
#n-resize {Cursor: n-resize}
#se-resize {Cursor: se-resize}
#sw-resize {Cursor: sw-resize}
#s-resize {Cursor: s-resize}
#w-resize {Cursor: }
#text {Cursor: text}
#wait {Cursor: wait}
#help {Cursor: help}
</style>
</head>

```

**تحلیل کد: #Hand = نام کد، که به دلخواه شماست. { } = کد، داخل این علامت، نوشته می شود.**  
**Cursor: hand = کد شکل اشاره گر، که در اینجا به صورت دست است.**

۲- فراخوانی کد. شی مورد نظر را، که قرار است اشاره گر، به محض رفتن بر روی آن، تغییر شکل دهد، داخل بدنه می نویسیم. آن شی، ممکن است یک عکس باشد یا یک متن یا چیز دیگر. در این مثال، متنی را در نظر گرفته ام که درون تگ (Span) نوشته ام.

```

<BODY>
<SPAN ID="Auto">Auto</SPAN><BR>
<SPAN ID="Crosshair">Crosshair</SPAN><BR>
<SPAN ID="Default">Default</SPAN><BR>
<SPAN ID="Pointer">Pointer</SPAN><BR>

```

```
<SPAN ID="Hand">Hand</SPAN><BR>
<SPAN ID="Move">Move</SPAN><BR>
<SPAN ID="e-resize">e-resize</SPAN><BR>
<SPAN ID="ne-resize">ne-resize</SPAN><BR>
<SPAN ID="nw-resize">nw-resize</SPAN><BR>
<SPAN ID="n-resize">n-resize</SPAN><BR>
<SPAN ID="se-resize">se-resize</SPAN><BR>
<SPAN ID="sw-resize">sw-resize</SPAN><BR>
<SPAN ID="s-resize">s-resize</SPAN><BR>
<SPAN ID="w-resize">w-resize</SPAN><BR>
<SPAN ID="text">text</SPAN><BR>
<SPAN ID="wait">wait</SPAN><BR>
<SPAN ID="help">help</SPAN><BR>
</BODY>
</HTML>
```

ID = آی‌دی، همان نامی است که در بالا نوشتیم؛ مانند (Hand).

**نمونه عملی:** اشاره‌گر خود را بر روی نوشته‌ها ببرید.

Auto  
Crosshair  
Default  
Pointer  
Hand  
Move  
e-resize  
ne-resize  
nw-resize  
n-resize  
se-resize  
sw-resize  
s-resize  
w-resize

text  
wait  
help

اسم هر کد، نمایانگر حالت و نوع آن است.

### اشاره گر دست ساز

اینها، شکل‌های پیش فرض و موجود در ویندوز است. اما آیا می‌شود شکل اشاره گر را آنگونه که خودمان می‌خواهیم، تغییر دهیم؛ یعنی به شکلی که خودمان طراحی کردیم؟ جواب این سوال، مثبت است. کافی است یک اشاره گر - که با پسوند CUR است - انتخاب کرده و آدرس آنرا بنویسید. به این صورت:

```
#Test {Cursor: url (test.cur); }
```

### داخل پراتنز:

هر تغییر شکلی، باید به خاطر نیاز و مناسب با زمان و مکان باشد؛ وگرنه کار شایسته‌ای نیست.

## کتابخانه وب

خدمت همه دوستان و تازه واردان عزیز، سلام و خسته نباشید، و خوش آمد عرض می‌کنم. از کسانی که از مطالب وبلاگ، خوششان آمده و ابراز لطف نموده‌اند تشکر می‌کنم.

قرار بود این جلسه را اختصاص دهم به معرفی چند نرم افزار تخصصی و مفید، که مسلماً شما را در نوشتن و عیب‌یابی کدها، کمک می‌کند. اما همانطور که قبلاً گفتم، آرزویم، یک بانک و مرجع عمومی و کامل است.

این طرح بزرگ، یک مشکل بزرگ داشت، و آن کوچکی من است. هم اطلاعاتم محدود است، و هم اینکه همیشه در بین شما نیستم که این کار را انجام دهم. احتیاج به همکاری علاقمندان بیشتر است. برای همین، بانکی را تهیه کردم که باز است و هرکسی می‌تواند نرم‌افزاری را که می‌شناسد معرفی کند. نام این بانک را "کتابخانه وب" گذاشتم که لینکش در سمت چپ همین وبلاگ قرار دارد.

اختصاص به زبانی ندارد. هرآنچه که مربوط به طراحی و برنامه نویسی وب است. این پروژه، به صورت آزمایشی راه‌اندازی شده است. البته این به نظر قاصر و ناقص راهنماست. برای کامل شدن این پروژه، احتیاج به راهنمایی و نظر شماست.

## جلسه ۱۸

### حرکت ۱

#### حرکت پنجره

مفهوم حرکت در رایانه و به خصوص وب، بحث جذاب و کاربردی است که هم باعث زیبایی صفحات می‌شود، هم توجه بیننده را به خود جلب می‌کند و هم در بهتر ارزیابی کردن مطالب، دست ما را باز می‌گذارد.

برای دو مورد اول، مثالهای فراوانی در پایگاه و وبلاگها وجود دارد. مثل اشکال و نوشته‌هایی که به دنبال موس حرکت می‌کنند؛ یا ستاره‌هایی که در صفحه، پخش می‌شوند؛ یا نوشته‌هایی که در نوار عنوان یا نوار وضعیت، در حال حرکت هستند.

همانطور که بهره‌برداری به جا از این ابزار، باعث زیبایی و کارایی می‌شود، استفاده بی‌مورد یا زیادی - که متأسفانه در بعضی از وبلاگها مشاهده می‌شود - باعث اذیت و خسته شدن بیننده خواهد شد؛ مخصوصاً در دو نوار عنوان و وضعیت. به یاد داشته باشید که این دو مکان، ساخته شده‌اند برای کار مهمتری؛ نه برای بازی و حرکت بی‌مورد نوشته‌ها.

#### مفهوم حرکت

حرکت، نسبی است. کم و بیش با این جمله آشنا هستید. نظریه "نسبیت حرکت"، می‌گوید که حرکت شی، نسبت به اطراف او سنجیده می‌شود. تصور کنید در قطار هستید. شما نسبت به اشیا بیرون از قطار، در حال حرکت هستید. از بعضی دور می‌شوید و به بعضی نزدیک؛ که این بستگی به "جهت حرکت" شما دارد. اما نسبت به خود قطار، حرکتی ندارید؛ مگر اینکه در همان قطار، شروع به راه رفتن کنید.

کره زمین برای ما ثابت و بی‌حرکت است؛ اما نسبت به کهکشان، جرم کوچکی است در حال حرکت. اشیاء داخل صفحه وب نیز همین حالت را دارند؛ که بزرگترین و مهمترین آن، خود پنجره (window) است.

#### انواع حرکت

به یک صفحه، خوب نگاه کنید. به نظر شما چه چیزهایی می‌توانند حرکت کنند؟ آیا همه حرکتها، از یک نوع است؟ عامل اصلی حرکت چیست؟

ظاهرا می شود گفت هر چیزی که در صفحه وجود دارد، قابل حرکت است. ظاهرا حرکتها از یک نوع نیستند؛ چون "نیروی محرکه" آنها تفاوت دارد. چه چیز باعث حرکت اشاره گر موس می شود؟ خود پنجره قابل حرکت است. می توانید با کلیک بر روی نوار عنوان، آنرا گرفته و به هر نقطه که تمایل دارید بکشید. اگر تغییر اندازه را هم نوعی حرکت بدانیم، با نوع متفاوتی از حرکت روبرو خواهیم شد.

## قالب حرکت

در رایانه، قالبی وجود دارد که موقعیت اشیا، نسبت به آن سنجیده می شود. بالا و پایین، چپ و راست. این چهار جهت اصلی، قالب ما هستند و موقعیت یک آیکن، نوشته، عکس یا پنجره، نسبت به آنها سنجیده می شود. چرا می گوئیم این پنجره از آن پنجره بالاتر است؟ چون نسبت به بالا، نزدیکتر است.

اگر این بخش را هم خوب درک کرده باشید، متوجه می شوید که برای شروع کار، لازم است "موقعیت جغرافیایی" یا همان "طول و عرض جغرافیایی" را به دست بیاوریم. خوشبختانه در زبان جاوا اسکریپت، دستورات لازم برای به دست آوردن آن، وجود دارد.

از چهار جهت اصلی، دو جهت، معیار هستند؛ بالا و چپ. البته ما فارسی زبانها، چون از راست به چپ می نویسیم و آرایش صفحات ما از راست به چپ است، بهتر است تا آنجا که ممکن است، بالا و راست را معیار خود قرار دهیم.

طول جغرافیایی - فاصله از بالا - را با  $Y$  (ایگرگ)، و عرض جغرافیایی - فاصله از چپ - را با  $X$  (ایکس) نشان می دهند.

## ● حرکت پنجره

برای حرکت پنجره، دو دستور وجود دارد.

### 1- window.moveBy (x, y)

با این دستور، می توان پنجره را یکباره به منطقه ای که با عدد مشخص ( $X, Y$ ) می شود، منتقل کرد.

### 2- window.moveTo (x, y)

این دستور مانند دستور قبل عمل می‌کند؛ با این تفاوت که اگر آنرا در یک حلقه قرار دهیم، به آهستگی پنجره را حرکت می‌دهد.

**کد:** کافی است این کد را داخل یک اسکریپت قرار داده و صفحه را باز کنید.

```
for(a = 1; a <= 250; a++) {window.moveTo (250, a)};
```

### ● تغییر اندازه پنجره

برای تغییر اندازه پنجره، دو دستور وجود دارد.

1- window.resizeTo (x, y)

با این دستور می‌توان اندازه پنجره را تغییر داد و اگر آنرا در یک حلقه قرار دهیم، به آهستگی اندازه پنجره را تغییر می‌دهد.

**کد:**

```
for(a = 1; a <= 500; a++) {window.resizeTo (100, a)};  
for(b = 1; b <= 500; b++) {window.resizeTo (b, 500)};
```

2- window.resizeBy (x, y)

این دستور هم برای تغییر اندازه است؛ اما به دقت و قدرت دستور قبلی نیست.

## جلسه ۱۹

### حرکت ۲

فرارسیدن ماه مبارک رمضان را به شما دوستان عزیز! تبریک می‌گوییم. امیدوارم که این ماه برای همه، پر از خیر و برکت باشد.

تغییرات:

- بحث "حرکت دادن اشیا" به این جلسه اضافه شده است. (جمعه ۱۵ مهر)

بحث ما حرکت است. در جلسه قبل، مشاهده کردید که حرکت پنجره به چه راحتی انجام پذیر است. در این ساعت، خواهی دید که حرکت اشیا به دنبال اشاره‌گر موس نیز به سهولت امکان دارد.

• تعقیب اشاره‌گر

• برای این عملیات، ابتدا موقعیت اشاره‌گر را به دست می‌آوریم. همانطور که در تمرین جلسه قبل، اجرا شد، با این دو دستور (`event.clientX - event.clientY`) می‌شود موقعیت اشاره‌گر را گرفت.

نمونه:

موقعیت موس از بالا - `clientY`:

موقعیت موس از چپ - `clientX`:

• حالا باید چیزی را که می‌خواهید به دنبال اشاره‌گر، حرکت کند مشخص نمایید. این شی می‌تواند از لحاظ ظاهر، متن یا عکس یا هر چیز دیگر، و از لحاظ محتوا، ثابت یا متغیر باشد.  
- سه متن بنویسید؛ یک متن برای حرکت عمودی، و یک متن برای حرکت افقی، و یکی هم برای حرکت کامل.

- هر کدام را داخل یک لایه قرار دهید.

- لایه‌ها را به این ترتیب نامگذاری نمایید: (`div1 - divY - divX`)

- استیل آنها را به این صورت تعریف کنید: (`style=position: absolute`). این کد، به شی مورد نظر اجازه حرکت می‌دهد.

• کد HTML:

```
<p><div id="div1" style="position: absolute;">متن</div></p>  
<p><div id="divY" style="position: absolute;">عمودی</div></p>  
<p><div id="divX" style="position: absolute;">افقی</div></p>
```

• کد JavaScript:

```
function cursorPos() // تابع  
{  
  // حرکت عمودی و افقی  
  div1.style.top = event.clientY;  
  div1.style.left = event.clientX;  
  // حرکت عمودی  
  divY.style.top = event.clientY;  
  // حرکت افقی  
  divX.style.left = event.clientX;  
}  
document.onmousemove=cursorPos;
```

برای دیدن نمونه، اینجا کلیک کنید.

• حرکت دادن اشیاء

در این مبحث، خواهیم آموخت که چگونه به وسیله موس، اشیا را جابجا کنیم؛ همانگونه که در ویندوز به راحتی انجام می‌شود.

• مراحل عملیات:

- ساختن شی مورد نظر
- به دست آوردن موقعیت اشاره‌گر

- به دست آوردن موقعیت شی مورد نظر
- تعریف کردن رویداد کلیک و کشیدن

• شیئی که برای این کار در نظر گرفته‌ام، یک جدول است که شبیه کادرهای محاوره‌ای در ویندوز می‌باشد.

- این جدول، دارای دو سلول است. سلول اول، برای عنوان کادر و محل کشیدن و جابجایی، و سلول پایین، محل نوشته محتوی است.

- داخل سلول اول، یک لایه (div) قرار داده‌ام به نام (moveMe1). جدول را هم (moveMe)، نامگذاری نموده‌ام.

- استایل جدول را به این صورت تعریف کنید: (style=position: relative). این کد، به شی مورد نظر اجازه حرکت می‌دهد.

#### • کد HTML:

```
<table id="moveMe" style="position: relative;" border="0"
width="200" bgcolor="#FBF9FB">
<tr>
<td bgcolor="#F6F4F6" align="center">
<div id="moveMe1">&nbsp;</div></td>
</tr>
<tr>
<td>&nbsp;</td>
</tr>
</table>
```

#### • کد JavaScript:

```
mouseover = true
function coordinates()
{
if (!moveMe)
{
```

```

return
}
if (event.srcElement.id == "moveMe1")
{
mouseover = true
// موقعیت جدول
pleft = moveMe.style.pixelLeft
ptop = moveMe.style.pixelTop
// موقعیت اشاره گر
xcoor = event.clientX
ycoor = event.clientY
document.onmousemove = funmove
}
}
function funmove()
{
if (mouseover && event.button == 1) // اگر کلیک چپ شد
{
moveMe.style.pixelLeft = pleft + event.clientX-xcoor
moveMe.style.pixelTop = ptop + event.clientY-ycoor
moveMe1.style.cursor = "move"; // تغییر شکل اشاره گر
moveMe.style.filter = "alpha(opacity=50)"; // استفاده از فیلتر آلفا
return false
}
}
function mouseup()
{
moveMe1.style.cursor = ""; // شکل عادی اشاره گر
moveMe.style.filter = "alpha(opacity=100)"; // از کار انداختن فیلتر آلفا
mouseover = false
}
document.onmousedown=coordinates

```

document.onmouseup=mouseup

• شرح کد:

- `moveMe1.style.cursor = move` = در جلسه ۱۷، خواندیم که شکل اشاره‌گر، قابل تغییر است. در اینجا، شکل "حرکت" را انتخاب می‌کنیم.

- `event.button == 1` = در جلسه ۱۷، دیده شد که کد کلیک چپ، ۱ است. با به دست آوردن این کد، می‌فهمیم که موس، در حالت کلیک باقی مانده؛ یعنی همان کشیدن به وسیله موس.

- `(moveMe.style.filter = "alpha(opacity=50` = به وسیله این کد، که از زبان HTML است، به جدول خود، فیلتر آلفا، یا همان کمرنگ شدن در حالت جابجایی می‌دهیم.

برای دیدن نمونه، اینجا کلیک کنید.

همانطور که مشاهده کردید، این کار زیبا به سادگی امکان پذیر است. شما هم می‌توانید در پایگاه و حتی وبلاگ خود، از کادرهای متحرک استفاده کنید. کادرهایی که کاربر می‌تواند آنها را جابجا نماید. با به کارگیری کمی ذوق و سلیقه، کادرهای زیباتر، طبیعی‌تر و دارای قابلیت بیشتری - مانند امکان کمینه (مینی مایز) کردن و یا بستن کادر - خواهید ساخت.

## جلسه ۲۰

### مفاهیم و واژه‌نامه

این جلسه، مجموعه‌ای از مفاهیم و واژه‌هایی است که در جلسات مختلف، بیان شده است. سعی شده که مرتب و بدون دخل و تصرف نوشته شود؛ الا بعضی موارد که نیاز بوده. این جلسه را هدیه می‌کنم به دوستان علاقمند، صادق و استوار و بلاگ جاوا اسکریپت.

« مفاهیم »

برنامه نویسی

زبان برنامه نویسی، وسیله‌ای است برای ساختن برنامه و نرم‌افزار مورد نیاز ما.

به طور کلی می‌توانیم برنامه‌های خود را به دو گروه، تقسیم کنیم:

۱- برنامه‌هایی که در محیط رایانه، استفاده می‌شوند.

۲- برنامه‌هایی که در محیط وب و اینترنت قابل استفاده هستند.

زبان کاربری

بعضی از زبانهای برنامه نویسی تحت وب، در طرف کاربر و کلاینت (Client) - یعنی شما که از اینترنت استفاده می‌کنید - اجراء می‌شود. JavaScript, JScript, VBScript

زبان سروری

زبانهای برنامه نویسی تحت وبی که در طرف سرور (Sserver) - یعنی سایتی که شما از آن استفاده می‌کنید - اجراء می‌شود. asp, php

جاوا اسکریپت

با جاوا اسکریپت، می‌توانیم برنامه‌ای بنویسیم که هم در رایانه و هم در محیط اینترنت، قابل اجرا باشد. البته زادگاه اصلی آن، اینترنت است. جاوا اسکریپت، از آن دسته از زبانهاست که در طرف کاربر، عمل می‌کند.

## Object (شی)

هر چیزی که در صفحه ما وجود دارد، یک شی حساب می‌شود و می‌توان از آن استفاده کرد و تغییرش داد. اولین شی موجود، خود صفحه و سند (document) است، که قبلا با آن آشنا شده‌اید.

## Properties (خصوصیات)

هر شی، یک مشخصات و خصوصیتی دارد که بیانگر کمیت و کیفیت آنست. مثلا یک سند، دارای این خصوصیات است: پس زمینه، عنوان (Title)، نوار وضعیت (Status)، نوار پیمایش (Scroll)، متن (Text)، لینک (Link).

## Methods (متد)

کاری که اشیاء می‌توانند انجام دهند، متد نامیده می‌شود؛ مانند باز کردن یا بستن پنجره. شالوده و اصل یک زبان برنامه نویسی، متدهای آن برنامه است، که ما باید آنها را به خوبی یاد گرفته و در جای مناسب استفاده نماییم. متدها، توسط سازندگان زبان برنامه نویسی، نوشته و تهیه می‌شود و ممکن است در نسخه‌های جدید آن زبان، تغییر داده شده، اضافه و تکمیل شود.

## Event (رویداد)

رویداد، عملیاتی است که کاربر، در یک صفحه انجام می‌دهد. بعضی از رویدادها مشترک و عمومی هستند و بعضی مخصوص یک شی. بعضی از این رویدادها را برای شما دسته بندی کرده‌ام که حتما ذخیره نموده و به مرور آنها کامل کنید.

## شرط

یکی از ابزار و دستورات مهم هر زبان برنامه نویسی، شرط است. آیا می‌دانید شرط چیست؟ برای درک مفهوم آن، احتیاجی نیست به خود زحمت بدهید؛ چون این دستور، در زندگی من و شما به وفور یافت می‌شود. اگر خورشید درآید، هوا روشن می‌شود. اگر درس بخوانی، قبول می‌شوی.

## ادات شرط

۱. if

۲. switch ... case

۳. variable = (condition) ? true : false

معروفترین اادات شرط، if است.

## ساختمان شرط

- جمله شرطی، از سه قسمت اصلی، تشکیل شده است.
  - ۱- اادات شرط: کلمه‌ای که شرط را می‌رساند؛ مانند اگر.
  - ۲- شرط: جمله‌ای که شرط انجام کاری، واقع شده.
  - ۳- مشروط یا جزا: جمله‌ای که جزای شرط و در جواب آن آمده؛ یعنی در صورت تحقق شرط، آن نیز به وقوع می‌پیوندد.

## Operators (عملگرها)

در برنامه نویسی، از علامت و نشانه‌هایی استفاده می‌شود که به آن عملگر گفته می‌شود. یکی از انواع عملگرها، عملگر شرطی می‌باشد.

عملیات برنامه نویسی

- یک عملیات برنامه نویسی، دارای سه بخش، است.
- ۱. دریافت داده. در این قسمت، اسکرپیت، اطلاعاتی را که نیاز دارد، می‌گیرد.
- ۲. پردازش داده. در این قسمت، اطلاعاتی که تهیه کرده، تجزیه و تحلیل می‌کند.
- ۳. نمایش داده. در این قسمت، اطلاعات پردازش شده را، ارائه می‌کند.

#### ثابت و متغیر

بعضی از اطلاعاتی که یک برنامه، به دست می‌آورد، ثابت (Constant) است، و بعضی، متغیر (Variable). متغیر به این معنی که یک چیز ثابت، مشخص و از پیش تعریف شده نیست. آنچه که برنامه نویس، بیشتر با او سر و کار دارد، متغیر است.

#### انواع متغیر

- ۱- رشته‌ای (String): حروف و کلمه
- ۲- عددی (Number): اعداد و ارقام (اعداد صحیح و اعشاری)
- ۳- منطقی (Boolean): بله و خیر

#### متغیر سراسری، عمومی (Global گلوبال)

اگر متغیر را خارج از تابع بنویسید، عمومی است؛ چون می‌شود در سرتاسر برنامه، آنرا صدا زده و مورد استفاده قرار داد.

#### متغیر محلی، خصوصی (Local لوکال)

اگر متغیری را داخل تابع تعریف کنید، خصوصی است؛ یعنی فقط در همان محل و همان تابع، کاربرد دارد، و بیرون از آن، نه می‌شود صدا زد و نه می‌شود استفاده کرد.

#### زمان

زمان، دارای اجزا مختلف است؛ یعنی انسان، آنرا به بخشهای مختلف تقسیم کرده تا قابل استفاده باشد. قسمتی از آن، تاریخ (سال، ماه، هفته و روز) نام دارد، و قسمتی از آن، به اجزا کوچکتر (ساعت، دقیقه و ثانیه) تقسیم

شده، و ساعت، نامیده می‌شود. البته معیار تاریخ در برنامه نویسی، میلادی است. برای تبدیل به شمسی یا قمری، باید از برنامه نویسی استفاده کنید.

#### تابع

تابع یا همان فانکشن (Function)، مجموعه‌ای از یک یا چند کد است که در یک جا جمع شده و دارای نام شده‌اند. آرگومان: گاهی، داخل پرانتزی که بعد از نام تابع قرار می‌گیرد، چیزی نوشته می‌شود به نام آرگومان که یک نوع متغیر است.

#### آرایه

آرایه، مجموعه‌ای از چند داده (ثابت و متغیر) است که به عنوان یک واحد شناخته شده و اعضای آن، با کد، مشخص شده‌اند. به عبارت دیگر، تعدادی از اطلاعات که در کنار هم قرار گرفته‌اند و یک وجه مشترک داشته و به یک نام خوانده می‌شوند.

#### حلقه

یکی دیگر از ویژگیهای مفید جاوا اسکریپت و همچنین سایر زبانهای برنامه نویسی، توانایی ایجاد حلقه‌های تکرار، یا اجرای گروهی از دستورالعملها به صورت تکراری است. این، یکی از کارهایی است که رایانه‌ها قادرند تا به خوبی آنها را انجام دهند، و آن چیزی نیست جز انجام کارهای تکراری با دقت بالا. در برنامه‌های اسکریپت، از حلقه‌های تکرار به عنوان یک ابزار قوی و مهم، استفاده‌های فراوانی می‌شود.

#### ادوات حلقه

- از روشهای مختلفی برای ایجاد حلقه، استفاده می‌شود.

for .۱

## While ۲ do ... while ۳

معروفترین ابزار حلقه، **for** است.

### ساختمان حلقه

۱. ابتدای حلقه:  $a = 0$ . این عدد، شروع و ابتدای حلقه را معین می‌کند.
۲. انتهای حلقه:  $a < 10$ . این عدد، حداکثر و انتهای حلقه - به عبارت دیگر، تعداد تکرار دستورات - را مشخص می‌کند.
۳. افزایش یا کاهش: این پارامتر، مشخص می‌کند که حلقه، در هر بار اجرا، یک واحد زیاد شود یا کم.
۴. جواب حلقه: حلقه، بی شباهت به شرط نیست. به عبارت دیگر: اگر متغیر ما ( $a$ )، از ۰ تا ۱۰ تداوم داشت، نمایش پیام، ادامه داشته باشد.

فرم

فرم وسیله‌ای است برای دریافت اطلاعات از کاربر.

**method** = نوع ارسال اطلاعات را معین می‌کند، که دو نوع است: **post, get**. این دو روش با تفاوت‌هایی، عملیات ارسال را انجام می‌دهند. روش پُست، عمومی‌تر است و از لحاظ امنیتی، برتری دارد. اگر متد را ننویسید، از پُست، استفاده می‌شود.

**action** = محلی را نشان می‌دهد که اطلاعات، برای پردازش، به آنجا ارسال خواهد شد؛ یعنی آدرس یک صفحه است، که در اینجا برای مثال، **test.htm** را انتخاب کرده‌ام.

**name** = نام فرم.

**onSubmit** = یک رویداد است. تابعی را که در جاوا اسکریپت، نوشته‌اید را صدا می‌کند.

**Get - Post** = تفاوت بین این دو، در چگونگی ارسال اطلاعات از فرم، به اسکریپت پردازشگر است.

اگر از روش گت، استفاده کنید، مقدار اطلاعاتی که می‌توانید ارسال کنید، محدود است.

با استفاده از روش `get`، اطلاعات به دست آمده از فرم، به عنوان بخشی از یک `URL`، انتقال می‌یابد؛ اما در روش `post`، اطلاعات، در حقیقت به صورت نامرئی، منتقل می‌شود. اگر برای مثال، در روش `get`، کاربر، یک رمز عبور را در فیلد پسورد، بنویسد، و فرد دیگری نیز در حال مشاهده صفحه نمایش کاربر باشد، به راحتی می‌تواند از رمز عبور کاربر، آگاه شود. به عبارت دیگر، این روش، در سطح امنیتی پایین‌تری از روش پست می‌باشد.

صفحه‌ای که حاوی فرمی با مشخصه `get` است، می‌تواند نشانه گذاری (`Bookmark`) شود؛ در حالی که در روش پست، این کار، ممکن نیست.  
اشاره‌گر

اشاره‌گر (`Cursor`) - که کاربران معمولاً و به اشتباه، به آن موس هم می‌گویند - همان شکلی است که با حرکت موس، در مانیتور، حرکت می‌کند و نمادی از انگشت و دست شماست برای اشاره به اشیاء داخل صفحه.

« واژه نامه »

کُد (`Code`): دستور و فرمانی که به زبان برنامه نویسی، نوشته شده.  
تگ (`Tag`): کوچکترین جزء یک کد. مثلاً تگ سر = `<head>`؛ تگ بدنه = `<Body>`؛ تگ اسکریپت = `<Script>`. هر تگ باید در داخل این دو علامت `< >`، قرار گیرد.

پارامتر (`Parameter`): مقداری که برای تگ، نوشته می‌شود. مثلاً برای تگ خط، معین می‌کنیم که از چه نوعی، به چه اندازه‌ای و چه رنگی باشد. مثال: کلمه `Color`، در تگ `Font`، پارامتری است که رنگ متن را مشخص می‌کند. پارامتر را شاید بتوان این گونه تعریف کرد که "قید حالت است برای تگ". تگ اسکریپت، هم دارای پارامترهایی است.

`document` = سند، صفحه جاری، پنجره‌ای که در آن کار می‌کنید

`document.bgColor` = رنگ پس زمینه سند

`new Date`: متد زمان به صورت کلی و مطلق است. بعد از آن باید نوع زمان خود را مشخص کنید؛ یعنی چه قسمتی از زمان را می‌خواهید بدانید؛ سال، ماه، هفته، روز و ... به تنهایی چیزی را نشان نمی‌دهد.

`toLocaleString`: متدی برای به دست آوردن زمان کامل (تاریخ و ساعت) است که به عنوان خصوصیت، برای زمان کلی استفاده می‌شود.

`document.write`: متدی که برای نوشتن متن در صفحه استفاده می‌شود.

`document.writeln`: مانند دستور قبلی عمل می‌کند؛ با این تفاوت که یک فاصله تا رشته قبلی، ایجاد می‌کند.

؛ : به این علامت، سِمی کالِن گفته می‌شود که در پایان هر دستور، می‌آید؛ هر چند در جاوا، اجباری نیست. // : به این علامت، کامنت (`Comment`) گفته می‌شود که برای نوشتن توضیح، در میان کدهای برنامه نویسی است. هر عبارتی که بعد از این علامت بیاید، توسط برنامه، خوانده و اجرا نمی‌شود و صرفاً توضیحی است که برنامه نویس، یادداشت کرده.

`ID`: آی‌دی، همان نام (`Name`) شی است؛ البته گاهی در عمل تفاوت‌هایی با هم دارند.  
`innerHTML`: با این دستور، متن یک تگ در قسمت بدنه - مانند محتویات تگ پاراگراف - نمایش داده می‌شود؛ و می‌توانیم آنرا تغییر داده و ویرایش کنیم.  
`outerHTML - innerHTML - outerText`: این ۳ متد، شبیه دستور قبلی است؛ البته با کمی تفاوت.

`setTimeout`: دستور تایمر و زمان شمار است. بعد از آن، یک جفت پرانتز، قرار دارد که داخلش، دو پارامتر نوشته می‌شود:

۱. اسم چیزی است که قرار است اجرا شود؛ مانند یک تابع.

۲. زمانی که تایمر در آن فعال خواهد شد. واحد زمان، میلی ثانیه است. (۱۰۰۰ = یک ثانیه)

`setInterval`: دستور تایمر و زمان شمار است. مانند دستور قبل، عمل می‌کند.

`<table>` = جدول.

`<tr>` = سطر، ستون افقی.

`<td>` = سلول، خانه‌های جدول.

`window` = شی صفحه است، که دارای متدهای مختلفی است.

`window.scroll` = متد اسکرول صفحه. داخل پرانتز، دو پارامتر دارد که دو عدد هستند که موقعیت صفحه را نشان می‌دهند. اولی، موقعیت فعلی صفحه؛ دومی، موقعیتی که باید به آنجا برسد.

`window.scrollTo - window.scrollBy` = دو متد دیگر از `parent` هستند. با تمرین، به دست آوردید که چه کاری انجام می‌دهند.

`parent` = معادل `window` است و در مورد اسکرول، می‌تواند به جای آن قرار گیرد.

## گام آخر

کاش در دهکده عشق فراوانی بود  
توی بازار صداقت، کمی ارزانی بود

کاش اگر گاه، کمی لطف به هم می کردیم  
مختصر بود ولی ساده و پنهانی بود

این روزا باید هممون برای هم سایه باشیم  
شبا یکم دلواپس کودک همسایه باشیم

اگه به هم کمک کنیم زندگی دیدنی میشه  
بر سر پیمان می موندن دوستای خوب تا همیشه

# ParsBook.Org

پارس بوک، بزرگترین کتابخانه الکترونیکی فارسی زبان

# ParsBook.Org



The Best Persian Book library