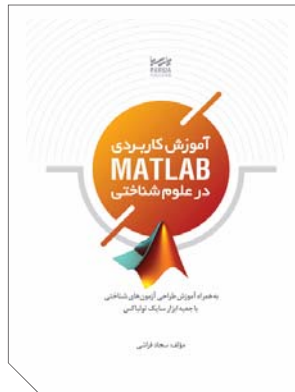




## آموزش کاربردی MATLAB در علوم شناختی

به همراه آموزش طراحی آزمون‌های شناختی  
با جعبه ابزار سایک تولباکس



مؤلف:  
سجاد فراشی



سرشناسه:  
عنوان و نام پدیدآور:

مشخصات نشر:

مشخصات ظاهری:

شابک:

وضعیت فهرست نویسی:

موضوع:

موضوع:

موضوع:

موضوع:

رده بندی کنگره:

رده بندی دیویی:

شماره کتابشناسی ملی:

فراشی، سجاد، ۱۳۶۲ -

آموزش کاربردی MATLAB در علوم شناختی به همراه آموزش طراحی  
آزمون‌های شناختی با جعبه ابزار سایک تولباکس / مولف سجاد فراشی.

تهران: پارسیا، ۱۳۹۷.

۳۳۶ص.

۹۷۸-۶۲۲-۹۹۶۱۳-۱-۵

فپیا

متلب

MATLAB

شناخت (علم)

Cognitive science

۱۳۹۷/۸۳۴/۸۳۴۹۷

۰۲۸۵۵۳۶/۵۱۸

۵۳۴۵۴۳۲

## آموزش کاربردی MATLAB

در علوم شناختی

**پارسیا**  
نشر پارسیا

مؤلف: سجاد فراشی

ناشر: پارسیا

شمارگان: ۵۰۰ نسخه

نوبت چاپ: اول - ۱۳۹۷

شابک: ۹۷۸-۶۲۲-۹۹۶۱۳-۱-۵

قیمت: ۳۷۰۰۰ تومان

مرکز پخش:

تهران، خیابان انقلاب، خیابان فخررازی، خیابان شهدای  
ژاندارمرینرسیده به خیابان دانشگاه ساختمان ایرانیان، پلاک ۵۸  
طبقه دوم، واحد ۶ تلفن: ۹۲-۶۶۴۸۴۱۹۱، www.noavarpub.com

کلیه حقوق چاپ و نشر این کتاب مطابق با قانون حقوق مؤلفان و مصنفان  
مصوب سال ۱۳۴۸ برای ناشر محفوظ و منحصراً متعلق به نشر پارسیا  
می‌باشد. لذا هر گونه استفاده از کل یا قسمتی از این کتاب (از قبیل هر  
نوع چاپ، فتوکپی، اسکن، عکس برداری، نشر الکترونیکی، هر نوع انتشار به  
صورت اینترنتی، سی‌دی، دی‌وی‌دی، فیلم فایل صوتی یا تصویری و غیره)  
بدون اجازه کتبی از نشر پارسیا ممنوع بوده و شرعاً حرام است و متخلفین  
تحت پیگرد قانونی قرار می‌گیرند.

## فهرست مطالب

۱۳	فصل اول: شروع کار با متلب
۱۳	آنچه در این فصل فرا خواهیم گرفت:
۱۳	۱-۱. مقدمه‌ای در مورد برنامه‌نویسی
۱۶	۱-۲. پیش‌نیاز برنامه‌نویسی در متلب
۱۷	۱-۳. شروع کار با نرم‌افزار متلب
۱۷	۱-۳-۱. متلب چیست؟
۱۷	۱-۳-۲. تاریخچه توسعه نرم‌افزار متلب
۱۸	۱-۳-۳. اجزا و قسمت‌های نرم‌افزار متلب
۲۱	۱-۳-۴. نگاهی جزئی‌تر به نوار ابزار متلب
۳۲	فصل دوم: متغیرها در متلب
۳۲	آنچه در این فصل خواهیم آموخت:
۳۳	۲-۱. تعریف متغیرها در متلب
۳۴	۲-۲. قواعد تعریف نام متغیر در متلب
۳۵	۲-۳. انواع متغیرها
۳۵	۲-۳-۱. متغیر عددی
۳۶	۲-۳-۲. متغیر منطقی
۳۸	۲-۳-۳. متغیر رشته‌ای
۳۹	۲-۳-۴. متغیر سمبولیک
۴۰	۲-۴. ماتریس
۴۱	۲-۴-۱. دستوره‌ای پرکاربرد برای ماتریس‌ها
۴۲	۲-۴-۲. انجام عملیات روی ماتریس‌ها
۴۳	۲-۵. داده‌های ساختاری در متلب
۴۵	۲-۶. داده‌ی سلولی در متلب
۴۶	۲-۶-۱. دستوره‌ای مرتبط با داده‌ی سلولی در متلب
۴۷	۲-۷. محیط کار در متلب
۴۹	فصل سوم: ترسیم شکل‌ها و نمودارها در متلب
۴۹	۳-۱. دستور plot در متلب
۵۲	۳-۲. ویرایش نمایش نمودار
۵۲	۳-۲-۱. برجسب‌زنی نمودار
۵۳	۳-۲-۲. تفکیک پنجره‌ی نمایش به چندین بخش با دستور subplot
۵۴	۳-۲-۳. برجسب‌زنی محورها
۵۴	۳-۲-۴. تنظیم محدوده‌ی نمایش محورها با دستور axis
۵۵	۳-۳. جلوه‌های گرافیکی در رسم نمودار
۵۷	۳-۴. شبکه‌بندی پنجره‌ی نمایش با دستور grid
۵۸	۳-۵. اضافه کردن علائم و اختصارات به شکل با دستور legend
۶۰	۳-۶. رسم چندین شکل در یک نمودار با دستور hold on
۶۱	۳-۷. اضافه نمودن خط و نوشته به نمودار
۶۲	۳-۸. نمودارهای لگاریتمی
۶۴	۳-۹. رسم نمودارهای میله‌ای با دستور bar
۶۷	۳-۱۰. رسم نمودارهای گسسته با دستور stem
۶۸	۳-۱۱. رسم هیستوگرام

۷۰	رسم نمودار دایره‌ای (پای چارت).....	۳-۱۲
۷۲	رسم نمودارهای سه‌بعدی در متلب.....	۳-۱۳
۷۲	دستور plot <sup>۳</sup> .....	۳-۱۳-۱
۷۳	دستور mesh.....	۳-۱۳-۲
<b>۷۶</b>	<b>فصل چهارم: کدنویسی در متلب.....</b>	
۷۶	آنچه در این فصل خواهیم آموخت:.....	
۷۸	۴-۱. عبارتهای شرطی.....	
۷۸	۴-۱-۱. ساختار شرطی if.....	
۷۹	۴-۱-۲. ساختار شرطی switch-case.....	
۸۰	۴-۲. حلقه‌های تکرار.....	
۸۰	۴-۲-۱. حلقه تکرار for.....	
۸۲	۴-۲-۲. حلقه تکرار while.....	
۸۲	۴-۳. دستور break.....	
۸۳	۴-۴. ایجاد توابع در متلب.....	
۸۶	۴-۵. تعریف توابع به صورت تو در تو.....	
۸۷	۴-۶. فراخوانی توابع با استفاده از مفهوم هندل تابع.....	
<b>۸۹</b>	<b>فصل پنجم: تولید صوت در متلب.....</b>	
۸۹	آنچه در این فصل خواهیم آموخت:.....	
۸۹	۵-۱. مقدمه‌ای در مورد صوت.....	
۹۲	۵-۲. شروع کار با صوت در متلب.....	
۹۴	۵-۳. پخش فایل‌های صوتی در متلب.....	
۹۴	۵-۳-۱. دستور audiinfo.....	
۹۵	۵-۳-۲. دستور audioread.....	
۹۶	۵-۳-۳. دستور wavread.....	
۹۶	۵-۳-۴. دستور sound.....	
۹۶	۵-۳-۵. دستورهای audioplayer و play.....	
۹۶	۵-۳-۶. دستور beep.....	
۹۷	۵-۳-۷. دستور audiorecorder.....	
۹۷	۵-۴. نکاتی در مورد تولید صوت در متلب.....	
۹۷	۵-۴-۱. بریده‌شدن داده‌ی صوتی.....	
۹۸	۵-۴-۲. همزمانی و غیر همزمانی اجرای اصوات.....	
<b>۱۰۰</b>	<b>فصل ششم: تولید و پردازش تصاویر در متلب.....</b>	
۱۰۰	آنچه در این فصل خواهیم آموخت:.....	
۱۰۲	۶-۱. واردسازی تصاویر به متلب.....	
۱۰۲	۶-۲. دستور imwrite.....	
۱۰۳	۶-۳. نمایش تصویر در متلب.....	
۱۰۳	۶-۳-۱. دستور image.....	
۱۰۳	۶-۳-۲. دستور imshow.....	
۱۰۵	۶-۴. عملیات بر روی تصاویر در متلب.....	
۱۰۵	۶-۴-۱. تغییر شدت پیکسل‌های تصویر.....	
۱۰۷	۶-۵. دستور imtool.....	

<b>فصل هفتم: ذخیره‌سازی و بارگذاری داده‌ها در متلب</b> .....	<b>۱۰۸</b>
۷-۱. ذخیره‌سازی نتایج در فرمت اکسل .....	۱۰۸
۷-۲. ذخیره‌سازی داده‌های محیط‌کار متلب با استفاده از دستور save .....	۱۱۰
۷-۳. بارگذاری داده‌ها در متلب با استفاده از دستور load .....	۱۱۱
۷-۴. خواندن فایل‌های متلب بدون بارگذاری آنها .....	۱۱۲
<b>فصل هشتم: ایجاد واسط گرافیکی کاربری در متلب</b> .....	<b>۱۱۳</b>
آنچه در این فصل خواهیم آموخت: .....	۱۱۳
۸-۱. ساختار اصلی یک GUI .....	۱۱۴
۸-۲. ایجاد واسط‌های گرافیکی کاربردی .....	۱۱۷
<b>فصل نهم: خطاهای مصطلح در برنامه‌نویسی متلب</b> .....	<b>۱۳۱</b>
آنچه در این فصل خواهیم آموخت: .....	۱۳۱
۹-۱. خطاهای مصطلح در متلب .....	۱۳۱
۹-۱-۱. خطای [ , {, or} Error: Expression or statement is incorrect--possibly unbalanced .....	۱۳۱
۹-۱-۲. خطاهای Matrix dimensions must agree .....	۱۳۱
۹-۱-۳. خطای ضرب داخلی دو ماتریس: Inner matrix dimensions must agree .....	۱۳۲
۹-۱-۴. خطای اندیس آرایه: Index exceeds matrix dimensions .....	۱۳۲
۹-۱-۵. خطای اندیس آرایه: Subscript indices must either be real positive integers or logicals .....	۱۳۲
۹-۱-۶. خطای Subscripted assignment dimension mismatch ??? .....	۱۳۲
۹-۱-۷. خطای Dimensions of matrices being concatenated are not consistent .....	۱۳۳
۹-۱-۸. خطای Attempt to execute SCRIPT as a function .....	۱۳۳
۹-۱-۹. خطای Error: Unexpected MATLAB operator .....	۱۳۳
۹-۱-۱۰. خطای Undefined function or method 'x' for input arguments of type 'double' .....	۱۳۴
۹-۱-۱۱. خطای Invalid (object) property: (property) .....	۱۳۴
۹-۱-۱۲. خطای تعداد آرگومان‌های ورودی/خروجی .....	۱۳۵
۹-۱-۱۳. خطای Undefined function or variable .....	۱۳۵
<b>فصل دهم: روش‌های آماری در متلب</b> .....	<b>۱۳۷</b>
آنچه در این فصل خواهیم آموخت: .....	۱۳۷
۱۰-۱. اصطلاحات آماری .....	۱۳۷
۱۰-۱-۱. احتمال .....	۱۳۷
۱۰-۱-۲. متغیر تصادفی و تابع توزیع احتمال .....	۱۳۸
۱۰-۱-۳. تابع توزیع تجمعی (CDF) .....	۱۴۰
۱۰-۱-۴. جامعه و نمونه آماری .....	۱۴۱
۱۰-۱-۵. فرضیه و نقیض آن در مطالعات .....	۱۴۱
۱۰-۱-۶. مقدار p (p-value) .....	۱۴۲
۱۰-۱-۷. سطح معناداری .....	۱۴۳
۱۰-۱-۸. جدول توزیع آماری .....	۱۴۴
۱۰-۱-۹. آزمون فرض آماری .....	۱۴۶
۱۰-۲. آزمون فرض یک‌نمونه‌ای .....	۱۴۷
۱۰-۲-۱. نحوه‌ی انتخاب نوع توزیع در آزمون‌های یک‌نمونه‌ای .....	۱۴۹
۱۰-۳. آزمون t-test یک‌نمونه‌ای .....	۱۵۰
۱۰-۴. آزمون‌های دو نمونه‌ای (Two-sample) .....	۱۵۱

۱۵۱.....	۱۰-۴-۱. آزمون فرض برای میانگین دو جامعه با اندازه نمونه بزرگ.....
۱۵۴.....	۱۰-۴-۲. آزمون فرض برای میانگین دو جامعه با اندازه نمونه کوچک.....
۱۵۵.....	۱۰-۴-۳. آزمون فرض برای نسبت دو جامعه (اندازه‌ی نمونه بزرگ).....
۱۵۷.....	۱۰-۵. آزمون فرض برای نمونه‌های وابسته.....
۱۵۹.....	۱۰-۶. آزمون‌های فرض با تکیه بر واریانس نمونه‌ها.....
۱۵۹.....	۱۰-۶-۱. آزمون فرض برای واریانس دو جامعه.....
۱۶۲.....	۱۰-۷. آنالیز واریانس (ANOVA).....
۱۶۴.....	۱۰-۸. آزمون فرض مبتنی بر مقدار میانی (مدین) Sign Test.....
۱۶۸.....	۱۰-۹. آزمون Wilcoxon Signed-Rank.....
۱۷۰.....	۱۰-۱۰. آزمون Wilcoxon rank-sum test.....
۱۷۱.....	۱۰-۱۱. آزمون Kruskal-Wallis.....
۱۷۲.....	۱۰-۱۲. واریانس، همبستگی و کوواریانس.....
۱۷۷.....	۱۰-۱۳. رگرسیون داده‌ها.....
۱۸۱.....	۱۰-۱۴. معرفی جعبه‌ابزار Statistics متلب.....
۱۸۱.....	۱۰-۱۴-۱. قراردعی داده‌ها در ساختارهای جدولی.....
۱۸۲.....	۱۰-۱۴-۲. اندازه‌گیری پخش‌شدگی در داده.....
۱۸۵.....	۱۰-۱۴-۳. منحنی‌های نمایش توزیع‌ها.....
۱۹۱.....	۱۰-۱۴-۴. واحدهای استاندارد یا Z-score.....
۱۹۱.....	۱۰-۱۴-۵. مومان‌ها.....
۱۹۲.....	۱۰-۱۴-۶. توابع توزیع پارامتریک و غیر پارامتریک.....
۱۹۴.....	۱۰-۱۴-۷. تخمین ماکزیمم لایکلیهود.....
۱۹۶.....	۱۰-۱۴-۸. آزمون‌های فرض در متلب.....

## فصل یازدهم: معرفی جعبه‌ابزار Psychtoolbox (PTB) / نمایش بصری..... ۲۰۶

۲۰۶.....	۱۱-۱. صفحه نمایش رایانه.....
۲۰۹.....	۱۱-۲. الحاق Psychtoolbox به متلب.....
۲۱۰.....	۱۱-۳. دستورهای PTB برای ایجاد تحریک‌های بصری.....
۲۱۰.....	۱۱-۳-۱. دستور PsychDefaultSetup(featureLevel).....
۲۱۰.....	۱۱-۳-۲. دستور Screen.....
۲۱۳.....	۱۱-۳-۳. دستور Flip.....
۲۱۴.....	۱۱-۳-۴. دستور بستن پنجره نمایش.....
۲۱۵.....	۱۱-۳-۵. دستور نوشتن متن در پنجره‌ی نمایش با استفاده از دستور DrawFormattedText.....
۲۱۸.....	۱۱-۳-۶. رسم خط با استفاده از دستور Screen.....
۲۱۹.....	۱۱-۳-۷. رسم کمان با استفاده از دستور Screen.....
۲۲۱.....	۱۱-۳-۸. رسم کمان با استفاده از دستور Screen.....
۲۲۱.....	۱۱-۳-۹. ترسیم مستطیل با استفاده از دستور Screen.....
۲۲۲.....	۱۱-۳-۱۰. رسم چارچوب مستطیلی با استفاده از دستور Screen.....
۲۲۳.....	۱۱-۳-۱۱. رسم بیضی با استفاده از دستور Screen.....
۲۲۳.....	۱۱-۳-۱۲. رسم چارچوب بیضی‌شکل با استفاده از دستور Screen.....
۲۲۴.....	۱۱-۳-۱۳. دستور رسم چندضلعی توسط دستور Screen.....
۲۲۴.....	۱۱-۳-۱۴. رسم چارچوب چندضلعی با استفاده از دستور Screen.....
۲۲۴.....	۱۱-۳-۱۵. دستور کپی نمودن بخشی از صفحه نمایش در یک ماتریس.....
۲۲۶.....	۱۱-۳-۱۶. دستور انتقال اطلاعات ماتریس تصویر به صفحه نمایش.....
۲۲۷.....	۱۱-۳-۱۷. دستور توقف رسم اشیا در صفحه نمایش.....

۲۲۸.....	۱۱-۳-۱۸. توقف اجرای برنامه با استفاده از دستور Screen.....
۲۲۸.....	۱۱-۳-۱۹. گرفتن اطلاعات صفحه‌نمایش با استفاده از دستور Screen.....
۲۲۹.....	۱۱-۳-۲۰. تخمین زمان اجرای دستور Flip.....
۲۲۹.....	۱۱-۳-۲۱. محاسبه محدوده‌ی صفحه نمایش.....
۲۳۰.....	۱۱-۳-۲۲. محاسبه عمق بیت پیکسل‌ها.....
۲۳۰.....	۱۱-۳-۲۳. محاسبه طول و عرض پنجره نمایش.....
۲۳۰.....	۱۱-۳-۲۴. محاسبه بردار زمانی اجرای توابع PTB.....
۲۳۰.....	۱۱-۳-۲۵. دستورات دریافت اطلاعات در مورد رایانه.....
۲۳۰.....	۱۱-۳-۲۶. دستورات ایجاد و نمایش تصاویر در PTB.....
۲۳۴.....	۱۱-۴. توابع مربوط به استفاده از فیلم‌ها و آزمون‌های چندرسانه‌ای.....
۲۳۴.....	۱۱-۴-۱. باز کردن یک فایل ویدیویی جهت نمایش.....
۲۳۷.....	۱۱-۴-۲. بستن فایل ویدیویی.....
۲۳۷.....	۱۱-۴-۳. تصویربرداری از فایل ویدیویی.....
۲۳۸.....	۱۱-۴-۴. نمایش فایل ویدیویی.....
۲۳۸.....	۱۱-۴-۵. اندیس زمانی نمایش.....
۲۳۹.....	۱۱-۴-۶. ایجاد فایل ویدیویی جدید.....
۲۴۰.....	۱۱-۴-۷. اضافه کردن داده صوتی به فیلم.....
۲۵۱.....	۱۱-۵. ساخت انیمیشن با استفاده از تصاویر.....

### فصل دوازدهم: استفاده از PTB در تولید صوت ..... ۲۵۷

۲۵۷.....	آنچه در این فصل فرا خواهیم گرفت:.....
۲۵۸.....	۱۲-۱. استفاده از PTB برای تولید صوت.....

### فصل سیزدهم: ارتباط با موس و صفحه کلید از طریق سایک تولباکس ..... ۲۷۶

۲۷۶.....	آنچه در این فصل خواهیم آموخت:.....
۲۷۶.....	۱۳-۱. صفحه کلید، موس و کدگذاری کاراکتر.....
۲۷۸.....	۱۳-۲. معرفی دستورات PTB برای کار با صفحه کلید.....
۲۷۸.....	۱۳-۲-۱. بررسی وضعیت کلیدهای صفحه کلید با دستور KbCheck.....
۲۷۹.....	۱۳-۲-۲. دستور GetChar.....
۲۸۰.....	۱۳-۲-۳. دستور تبدیل کاراکتر به کد اسکی و بالعکس.....
۲۸۱.....	۱۳-۳. تعریف رخداد کیبورد و موس در PTB.....
۲۸۳.....	۱۳-۴. دستور توقف کار با صفحه کلید.....
۲۸۵.....	۱۳-۵. کنترل موس با استفاده از PTB.....
۲۸۵.....	۱۳-۶. بررسی همزمان همه‌ی ابزارهای متصل به رایانه.....
۲۸۶.....	۱۳-۷. مثالهایی از کار با دستورات PTB در متلب.....

### فصل چهاردهم: طراحی آزمون متغیرهای توجه (Tova) ..... ۲۹۴

۲۹۴.....	آنچه در این فصل خواهیم آموخت:.....
۲۹۴.....	۱۴-۱. آزمون متغیرهای توجه.....
۲۹۵.....	۱۴-۲. مراحل انجام تست Tova.....

### فصل پانزدهم: طراحی آزمون استروپ با استفاده از Psychtoolbox ..... ۳۱۰

۳۱۰.....	آنچه در این فصل خواهیم آموخت:.....
۳۱۰.....	۱۵-۱. آزمون استروپ چیست؟.....

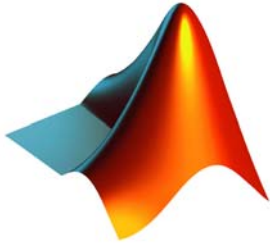
- ۱۵-۲. طراحی یک نمونه از آزمون stroop با استفاده از PTB ..... ۳۱۱
- فصل شانزدهم: طراحی آزمون جستجوی بصری (Visual Search) ..... ۳۲۴
- فصل هفدهم: طراحی آزمون حافظه فعال ..... ۳۳۲
- ۱۷-۱. آزمون حافظه فعال چیست؟ ..... ۳۳۲
- ۱۷-۲. طراحی یک آزمون نمونه برای سنجش حافظه فعال / آزمون Corsi ..... ۳۳۳





## فصل اول

### شروع کار با متلب



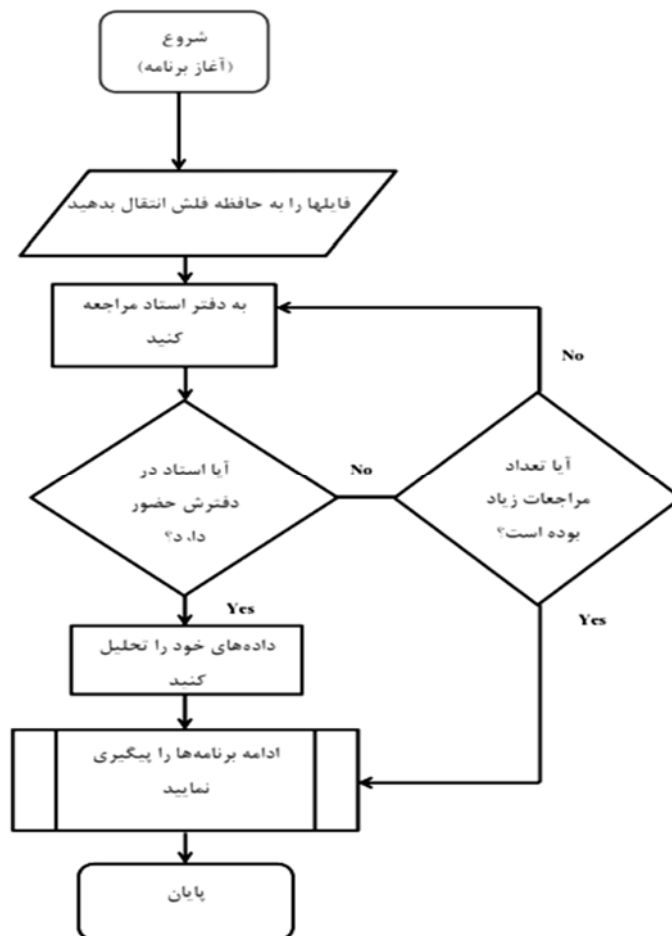
#### آنچه در این فصل فرا خواهیم گرفت:

- ✓ برنامه‌نویسی چیست؟
- ✓ آشنایی با نرم‌افزار متلب و قسمت‌های مختلف آن
- ✓ آشنایی با نوار ابزار متلب

#### ۱-۱. مقدمه‌ای در مورد برنامه‌نویسی

هدف اصلی این کتاب آموزش برنامه‌نویسی با استفاده از نرم‌افزار متلب است بنابراین بهتر است که مباحث کتاب را از تعریف برنامه‌نویسی شروع کنیم. فرض کنید که قصد دارید برنامه یک روز کاری خود را تنظیم کنید. ممکن است که در آغاز روز قصد مراجعه به دانشگاه خود را داشته باشید تا با استاد خود ملاقات کنید و آخرین پیشرفت‌های پایان‌نامه‌ی خود را با او در میان بگذارید. برای این منظور احتمالاً در ابتدا تمامی فایل‌های مربوط به پایان‌نامه را به حافظه‌ی فلش خود انتقال می‌دهید. ممکن است در برنامه‌ریزی زمانی خود این احتمال را در نظر بگیرید که در صورتی که استادتان در دفتر کارش حضور نداشت به سالن مطالعه دانشگاه بروید و یا وقت خود را با دوستانتان بگذرانید. به عبارت دیگر بخشی از برنامه روزمره شما مشروط به رخدادهای حضور استادتان در دفتر کارش خواهد بود. شاید بخواهید در صورت عدم حضور استادتان در اتاقش چندین بار به دفتر کارش مراجعه کنید و حضور یا عدم حضور او را مجدداً بررسی کنید در این صورت شما مجبور به تکرار کردن مجموعه‌ای از عمل‌ها هستید. احتمالاً در صورت عدم حضور استادتان ترجیح خواهید داد که ادامه برنامه‌ای را که برای آن روزتان در نظر گرفته‌اید پیگیری کنید مثلاً برای صرف ناهار به رستوران دانشگاه بروید. در نهایت ممکن است که برای پایان روز کاری خود نیز مراجعه به بازار و یک خرید خاص را برنامه‌ریزی

کنید و در نهایت برنامه تعیین شده‌ی شما به پایان خواهد رسید. برنامه‌ای که در بالا و به عنوان یک مثال مدنظر قرار داده شد را اصطلاحاً الگوریتم می‌گویند. الگوریتم مجموعه‌ای از دستورات عمل‌ها است که به ترتیب به خصوصی و جهت حل یک مسأله‌ی خاص اجرا می‌شوند. شما می‌توانید نحوه‌ی اجرای این دستورات را به صورت شماتیکی نیز ترسیم کنید. شماتیکی که ترسیم می‌شود روند اجرای الگوریتم را نشان می‌دهد و می‌تواند روند اجرای الگوریتم را به شکلی قابل فهم‌تر نشان دهد به همین دلیل به آن روند نما، چارت انجام کارت و یا فلوچارت گفته می‌شود. چنانچه قرار باشد فلوچارت شما برای دیگران نیز قابل فهم باشد باید در ترسیم آن از علامتها و نشانه‌هایی استفاده کنید که برای دیگران نیز بامعنی باشند. مثلاً می‌توان قرارداد کرد که اگر قرار باشد که اجرای عملی به صورت شرطی انجام شود چنین عملی درون نماد لوزی نوشته شود و یا اینکه قرار داد نماییم که مسیر اجرای برنامه از طریق سیس‌های رابط دنبال شود. فلوچارت مثال بالا را می‌توان به صورت زیر ترسیم کرد.



شکل ۱-۱: فلوچارت (روند نما)، مراحل اجرای الگوریتم را نشان می‌دهد.

همانگونه که از مثال بالا می‌توان استنباط کرد همه ما در زندگی روزمره‌ی خود به صورت دائم در حال برنامه‌ریزی و مدیریت زندگی خود هستیم از این منظر همه‌ی ما ذاتاً موجوداتی برنامه‌نویس هستیم! با

این وجود هنگامی که از برنامه‌نویسی صحبت می‌شود ناخودآگاه به یاد افرادی باهوش خواهیم افتاد که پشت رایانه‌های خود نشسته‌اند و در حال استفاده از نرم‌افزارهای خاص و تایپ نمودن واژگان عجیب و غریب در آن نرم‌افزارها می‌باشند. با این حال واقعیت این است که برنامه‌نویسی رایانه‌ای نیز از همان اصول برنامه‌ریزی معمولی تبعیت می‌کند یعنی در آن لازم است که برنامه‌نویس پیش از نوشتن دستورات در محیط برنامه‌نویسی مد نظر خود، الگوریتم و یا ترتیب اجرای دستورات را مشخص کند و احتمالاً فلوچارتی را جهت اجرای دستورات تهیه کند با این تفاوت که بر خلاف برنامه‌ریزی متعارفی که ما در زندگی روزمره انجام می‌دهیم و دستورات را بر روی کاغذ و یا در ذهنمان می‌نویسیم در برنامه‌نویسی رایانه‌ای، برنامه‌نویس دستورات را بر اساس قواعد نگارشی خاص و در یک محیط برنامه‌نویسی ویژه می‌نویسد. در نهایت برنامه‌نویسی به کمک رایانه به معنای نوشتن یک توالی خاص از دستورات است که قرار است توسط رایانه اجرا شوند. قدرت و سرعت پردازشی بالای رایانه‌های امروزی این امکان را فراهم می‌کند که برنامه‌های پیچیده با سرعت و قابلیت اطمینان بالا پیاده‌سازی و اجرا شوند و به همین دلیل رایانه و برنامه‌نویسی مبتنی بر رایانه به مساله‌ای مورد علاقه برای محققان تبدیل شده‌است. سوال مهم دیگری که ممکن است ذهن شما را درگیر کرده باشد این است که پس از نوشتن مجموعه‌ای از دستورات در رایانه، این دستورات چگونه توسط یک ماشین بدون درک و فهم تشخیص داده‌شده و اجرا می‌شوند. هدف از نوشتن یک برنامه در رایانه، اجرای مجموعه‌ای از دستورات توسط رایانه و تولید یک خروجی توسط آن است. نرم‌افزارهای رایانه‌ای که امکان نوشتن دستورها در آنها فراهم است دارای یک ویرایشگر متنی هستند. ویرایشگر متنی، محیطی است که برنامه‌نویس می‌تواند دستورات مد نظر خود را در آن تایپ کند. اما رایانه چگونه دستوراتی را که در ویرایشگر متن یک نرم‌افزار نوشته می‌شوند تفسیر می‌کند؟ پیش از پاسخ دادن به این سوال دانستن پاره‌ای از مفاهیم ضروری است. رایانه‌هایی که امروزه مورد استفاده قرار می‌گیرند دارای دو بخش مجزای نرم‌افزار و سخت افزار هستند. بخش‌هایی نظیر برنامه‌ها که غیرقابل لمس کردن هستند، یعنی نمی‌توان با دست آنها را لمس کرد، افزار نرم و یا اصطلاحاً نرم افزار گرفته می‌شوند و اجزایی که قابل لمس کردن هستند ابزار سخت و یا اصطلاحاً سخت افزار گفته می‌شوند و کنترل آنها بوسیله‌ی نرم‌افزار انجام خواهد گرفت. به صورت مشابه آنچه که به عنوان فکر در ذهن انسان وجود دارد شبیه نرم‌افزار و اعضای بدن انسان که توسط افکار او کنترل می‌شوند مشابه سخت افزار می‌باشند. سخت افزار رایانه از تجهیزات الکترونیکی تشکیل شده‌اند که جهت راه‌اندازی و استفاده از آنها از ولتاژ الکتریکی استفاده می‌شود به همین دلیل است که جهت استفاده از رایانه باید آن را به برق شهر متصل کرد. مدارات الکترونیکی به‌کار رفته در رایانه‌ها غالباً با استفاده از تکنولوژی نیمه‌هادی ساخته شده‌اند که در آنها اعمال نیرو به بارهای الکتریکی توسط ولتاژ اعمال شده، سبب جابجایی بار الکتریکی می‌شود. از این طریق می‌توان ذخیره‌گاه‌های بار الکتریکی را در نیمه هادی پر و یا خالی کرد. تغییر بار الکتریکی این ذخیره‌گاه‌ها می‌تواند نوع عملکرد مدار الکتریکی را تغییر دهد. جهت فعال شدن یک جزء سخت افزاری میزان ولتاژ و یا بار الکتریکی اختصاص داده‌شده به آن باید از حدی (حد آستانه) بیشتر شود در حالی که اگر ولتاژ اعمالی از حدی کمتر باشد جزء سخت‌افزاری غیر فعال خواهد بود. از لحاظ عددی چنانچه یک جزء سخت‌افزاری فعال باشد مقدار آن یک و در صورتی که غیر فعال باشد مقدار آن صفر در نظر گرفته خواهد شد به همین دلیل رایانه تنها دو حالت فعال و غیر فعال و یا دو حالت صفر و یک را تشخیص

می‌دهد. به عنوان یک مثال فرض کنید هدف ذخیره‌سازی عدد ۱۳ در حافظه رایانه است. حافظه‌ی رایانه را می‌توان به صورت عناصر الکترونیکی (سلولهای) در نظر گرفت که در کنار یکدیگر قرار گرفته‌اند و ارزش آنها از چپ به راست افزایش می‌یابد به نحوی که اگر اولین سلول ارزشی برابر یک داشته باشد سلول دوم ارزشی برابر ۲، سلول سوم ارزشی برابر ۴ و سلول چهارم ارزشی برابر ۸ خواهد داشت. عدد ۱۳ را می‌توان به صورت مجموع سه عدد ۱، ۴ و ۸ در نظر گرفت بنابراین اگر ولتاژ اعمال شده به سلولهای ۱، ۳ و ۴ از حد آستانه بیشتر شود می‌توان عدد ۱۳ را در حافظه ذخیره نمود. حافظه‌های متعارف از بسته‌های ۸ تایی از سلولها تشکیل شده‌اند. از آنجا که هر عملی در رایانه بوسیله‌ی فعال یا غیر فعال بودن (به ترتیب وضعیت یک و صفر) مشخص می‌شود لذا اصطلاحاً گفته می‌شود که رایانه یک سیستم دودویی (باینری) است. همانطور که گفته شد رایانه تنها اعداد صفر و یک را می‌شناسد این در حالی است که برنامه‌نویس برنامه‌ی مورد نظر خود را عمدتاً به شکلی که به زبان انسان نزدیک است می‌نویسد بدین معنی که مثلاً برای مطرح کردن یک شرط در غالب زبانهای برنامه‌نویسی از واژه‌ی if و برای پرش به بخشی از برنامه از دستور jump و یا goto استفاده می‌شود. چنین زبانی را یک زبان سطح بالا (زبان نزدیک به زبان انسان) و یا زبان مبدا نیز می‌نامند. دستورهای سطح بالا باید به زبانی قابل فهم برای رایانه (زبان ماشین و یا زبان سطح پایین) یعنی به صورت مجموعه‌ای از صفر و یکها تبدیل شوند. یک توالی مشخص از یکها و صفرها می‌تواند به ترتیب اجزای رایانه را روشن و خاموش کند و از این طریق امکان اجرای دستور خاصی امکان‌پذیر خواهد شد. وظیفه‌ی تبدیل زبان سطح بالا به زبان ماشین بر عهده‌ی یک مترجم خاص است که اصطلاحاً کامپایلر یا مفسر گفته می‌شود. بنابراین به صورت خلاصه، هنگامی که برنامه‌نویس برنامه‌ی مد نظر خود را در ویرایشگر یک نرم‌افزار می‌نویسد، کامپایلر مربوط به آن نرم‌افزار دستورهای نوشته شده را به مجموعه‌ای از صفر و یکها تبدیل خواهد کرد. بر اساس توالی صفر و یک تولید شده اجزای سخت‌افزاری فعال و یا غیر فعال خواهند شد و در نتیجه یک عمل خاص انجام خواهد شد و از این طریق یک برنامه توسط رایانه اجرا خواهد شد.

## ۲-۱. پیش نیاز برنامه‌نویسی در متلب

حال فرض کنید که به عنوان مثال می‌خواهید شروع به نوشتن یک برنامه در نرم‌افزار متلب کنید. پیش نیازهای انجام این برنامه‌نویسی چه خواهد بود؟ قطع به یقین نخستین گام جهت نوشتن برنامه، آشنایی با چگونگی نوشتن برنامه در محیط متلب است به عبارت دیگر برنامه‌نویس باید بداند یک عبارت شرطی و یا یک عمل تکرارشونده چگونه در نرم‌افزار متلب نوشته می‌شوند. به مجموعه قواعد نوشتاری اصطلاحاً syntax گفته می‌شود. بخش مهمی از هدف این کتاب آموزش قواعد نوشتاری در نرم‌افزار متلب است. توجه داشته باشید که هر نرم‌افزار برنامه‌نویسی دارای قواعد نوشتاری مخصوص خودش است بنابراین اگر شما طریقه برنامه‌نویسی در متلب را آموختید برای نوشتن برنامه در محیطی دیگر، برای مثال در ویژوال C، لازم است که قواعد نوشتاری آن محیط را هم یاد بگیرید هرچند که معمولاً نقاط اشتراک بسیار زیادی بین آنها وجود خواهد داشت. پس از یادگیری قواعد نوشتاری و پیش از نوشتن برنامه در محیط متلب باید مراحل نوشتن برنامه خود را از طریق کشیدن فلوجارت تعیین کنید. لازم به ذکر است که برنامه‌ها در متلب به صورت خط به خط اجرا می‌شوند بنابراین هر خط از برنامه یک عمل خاص را انجام می‌دهد.

هنگامی که فردی با قواعد نوشتاری و چگونگی تعیین روال اجرای برنامه از طریق کشیدن یک فلوجارت آشنا باشد به راحتی می‌تواند اقدام به نوشتن برنامه مورد نظر خود در متلب کند.

### ۳-۱. شروع کار با نرم‌افزار متلب

#### ۳-۱-۱. متلب چیست؟

نرم‌افزاری که در این کتاب به معرفی آن خواهیم پرداخت با نام متلب (Matlab) شناخته می‌شود. عنوان متلب از کلمات آغازین دو واژه ماتریس (Matrix) و آزمایشگاه (Laboratory) اقتباس شده است. به عبارت دیگر این نرم‌افزار یک آزمایشگاه ماتریس است و همانطور که در ادامه‌ی این کتاب خواهید دید ماتریس‌ها نقشی کلیدی را در نرم‌افزار متلب ایفا خواهند کرد. متلب محصولی از کمپانی آمریکایی MathWorks است که تخصص اصلی آن ایجاد و توسعه‌ی نرم‌افزارهایی است که در حوزه‌ی محاسبات عددی به کار برده می‌شوند. با استفاده از متلب می‌توان بر روی داده‌ها طیف وسیعی از عملیات‌ها را انجام داد، توابع و داده‌ها را رسم کرد، واسط‌های کاربری بسیار حرفه‌ای ایجاد نمود تا حتی کاربرانی که فاقد اطلاعات در مورد متلب هستند بتوانند از برنامه‌های نوشته شده استفاده نمایند و حتی برنامه‌های نوشته شده بوسیله‌ی دیگر زبانهای برنامه‌نویسی نظیر C، C++، جاوا، فورترن و یا پایتون را فراخوانی کنند. اغلب نرم‌افزارهایی که تاکنون ایجاد شده‌اند و یا در حال ایجاد می‌باشند عمدتاً تک منظوره هستند به این معنی که تنها توسط طیف خاصی از متخصصین قابل استفاده می‌باشند این در حالی است که متلب به واسطه‌ی کاربردها و انعطاف‌پذیری‌های منحصر به فردی که دارد به نرم‌افزاری کاربردی در بسیاری از حوزه‌های علمی تبدیل شده‌است. شما چه یک مهندس باشید، چه یک روانشناس و یا یک پزشک و چه اینکه فردی باشید که در حوزه‌های علوم انسانی فعالیت می‌کنید قطع به یقین در امور پژوهشی خود با داده‌ها و پردازش آنها سروکار خواهید داشت بنابراین می‌توانید به نرم‌افزار متلب به عنوان یک ابزار قدرتمند در مدیریت داده‌های خود نگاه کنید.

#### ۳-۱-۲. تاریخچه توسعه نرم‌افزار متلب

اولین قدم‌ها در به وجود آمدن نرم‌افزاری که امروزه به عنوان متلب شناخته می‌شود در سال ۱۹۷۰ برداشته شد جایی که Cleve Moler رئیس دانشکده علوم رایانه دانشگاه نیومکزیکو به فکر ایجاد نرم‌افزاری افتاد تا دانشجویانش بتوانند با استفاده از آن از دو نرم‌افزار LINPACK و EISPACK که دو نرم‌افزار مورد استفاده در محاسبات جبر خطی بودند استفاده کنند. این دو نرم‌افزار بر اساس زبان برنامه‌نویسی فورترن نوشته شده بودند و دانشجویی که قصد استفاده از آنها را داشت می‌بایست در ابتدا زبان برنامه‌نویسی فورترن را یاد می‌گرفت. این ایده جالب به سرعت در دیگر دانشکده‌ها و بویژه بین ریاضیدانان طرفداران فراوانی پیدا کرد. در سال ۱۹۸۴ و با اضافه شدن یک مهندس با نام Jack Little به Moler دومین گام اساسی در توسعه نرم‌افزار متلب برداشته شد و این بار این نرم‌افزار با زبان برنامه‌نویسی C بازنویسی شد. گام سوم در توسعه متلب در سال ۲۰۰۰ برداشته شد که در آن مجموعه‌ی جدیدی از کتابخانه‌ها به متلب اضافه شد و از این طریق به این نرم‌افزار این قابلیت را داد که بتواند محاسبات عددی را برای متخصصین در حوزه‌های متعددی پیاده‌سازی کند.