



دانشگاه شاهرود

دانشکده فنی

گروه مهندسی کامپیوتر

عنوان گزارش:

آشنایی مقدماتی با ابزار ویرایش متن $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$

نگارش:

علی احمدیان رمکی

آبان ۱۳۹۲

فهرست مطالب

۳	۱ آشنایی و نصب
۳	۱.۱ تاریخچه مختصری از صنعت چاپ و حروفچینی
۴	۲.۱ T _E X چیست؟
۵	۳.۱ L _A T _E X چیست؟
۶	۴.۱ برخی از مزیت‌ها
۶	۵.۱ چه کسی به L _A T _E X نیاز دارد؟
۸	۶.۱ نصب L _A T _E X
۱۲	۲ مقدمات کار با L _A T _E X
۱۲	۱.۲ ویرایشگر
۱۳	۲.۲ شروع کار
۱۳	۱.۲.۲ خطوط کلیدی مورد نیاز
۱۶	۳.۲ بسته‌های اضافی
۱۷	۳ ساختاردهی به متن
۱۷	۱.۳ ایجاد فصل، بخش، زیربخش، زیرزیر بخش و فهرست مطالب
۱۸	۲.۳ شماره‌گذاری و مورد بندی
۲۱	۴ فرمول نویسی
۲۱	۱.۴ مقدمات و آشنایی با علائم
۲۴	۲.۴ دستورات اولیه
۲۶	۳.۴ نحوه ارجاع دادن به رابطه‌ها
۲۸	۵ درج تصویر
۲۸	۱.۵ دستورات اولیه
۳۱	۲.۵ ارجاع دادن به تصاویر و caption گذاری
۳۲	۶ درج جدول
۳۲	۱.۶ دستورات اولیه
۳۳	۲.۶ قالب‌دهی به جدول و تعریف آن به صورت شناور
۳۳	۳.۶ تعریف جدول به صورت شناور و نحوه ارجاع دادن به آن

۳۵

۷ آشنایی با بسته geometry

۳۸

۸ فارسی نویسی در \LaTeX

۱ آشنایی و نصب

۱.۱ تاریخچه مختصری از صنعت چاپ و حروفچینی

در دوره پیش از صنعت چاپ و حروفچینی، یک کتاب با نسخه برداری به طور دستی پدید می آمد و انتشار می یافت. کار نسخه برداری از یک کتاب، کاری سخت، دشوار و بسیار زمان گیر بود. از سوی دیگر نمونه های آماده شده نیز با یکدیگر بسیار متفاوت بودند. این وضعیت تا اختراع گوتنبرگ، در سال ۱۴۴۰ میلادی، ادامه داشت. تکامل در صنعت چاپ و حروفچینی را می توان به چهار دوره تقسیم کرد.

- دوره گوتنبرگ، شروع ۱۴۵۰ میلادی

اساس اختراع گوتنبرگ این گونه بود که ابتدا نقش حروف بر اجسام مناسب کنده کاری می شد و سپس با ریختن فلز مذاب در این قالب های کنده کاری شده، حروف فلزی ایجاد می شد. سپس حروف ایجاد شده کنار هم قرار می گرفت تا کلمات و پاراگراف ها شکل گیرد و یک صفحه کامل پدید آید. صفحه پدید آمده با حروف فلزی را به جوهر آغشته می کردند و با قرار دادن آن روی برگ کاغذ، یک صفحه از کتاب چاپ می شد. از آن زمان عمل چیدن حروف کنار هم، برای پدید آوردن واژه ها و پاراگراف ها، عمل حروفچینی نام گرفت.

- دوره انقلاب صنعتی، شروع ۱۸۷۰ میلادی

انقلاب صنعتی نوآوری بزرگی در صنعت چاپ و حروفچینی پدید آورد. نیروی بخار و حرکت دورانی جایگزین دست های کارگران شد. ماشین بخار همان کاری را که کارگران انجام می دادند، با کاهش ۸۵ درصدی در زمان، به پایان می رساند.

- دوره چاپ به کمک عکس برداری، شروع ۱۹۶۰ میلادی

در سال ۱۹۶۰ میلادی جهش قابل توجهی در پیشرفت صنعت چاپ و حروفچینی انجام گرفت و آن چاپ با کمک عکاسی بود. اساس کار این گونه بود که ابتدا، به وسیله دوربین عکاسی، تصویری از صفحه مورد نظر، با اندازه دلخواه، فراهم می شد. فیلم حاصل را روی یک صفحه فلزی مناسب قرار می دادند و حاصل کار را درون یک مایع شیمیایی مخصوص غوطه ور می کردند. آن قسمت از صفحه فلزی که تصویر حروف و کلمات قرار داشت، برجای می ماند و سایر قسمت های صفحه، توسط مایع شیمیایی، اصطلاحاً خورده می شد. در نتیجه سطحی بدست می آمد که در آن نوشته مورد نظر، به صورت برجسته، نمایان بود. سطح حاصل را به جوهر آغشته می کردند و برای چاپ استفاده می نمودند. این فرایند حروفچینی سرد نام داشت، زیرا در آن از فلز مذاب و داغ استفاده نمی شد. یکی از مزایای این روش آن بود که به آسانی، بزرگنمایی حروف و کلمات قابل تغییر بود و حتی می توانستند حروف و کلمات را روی هم چاپ کنند.

• دوره حروف چینی رایانه‌ای، شروع ۱۹۷۰ میلادی

در دهه‌ی ۱۹۷۰ میلادی حروف چینی بر اساس سامانه‌های رایانه‌ای ظاهر گشت.

۲.۱ TeX چیست؟

در حدود سی سال پیش، حروف چینی نوشته‌هایی که دربردارنده فرمول‌ها و علائم ریاضی بود، کاری سخت و حتی غیر ممکن بود. ابتدا دست نوشته‌ها، به وسیله یک ماشین حروف چینی، آماده می‌شد و سپس فرمول‌ها و علائم ریاضی، با صرف وقت و زمان زیاد، به طور دستی، درون متن گنجانده می‌شد و حاصل کار به چاپ‌خانه ارسال می‌گشت. قبل از چاپ نهایی، کتاب حروف چینی شده در اختیار نویسنده قرار می‌گرفت و نویسنده، برای رفع اشکالات احتمالی، آن را بررسی می‌کرد و در صورت لزوم نکاتی در حاشیه‌ها می‌نوشت و آن را جهت اصلاح برای ناشر می‌فرستاد.

هیچ‌گونه ارتباطی بین نویسنده و مسئول حروف چینی وجود نداشت. در آن هنگام حروف چینی متون ریاضی، گاهی به عنوان یک جریمه برای کارکنان چاپ‌خانه به حساب می‌آمد. در مدت زمانی که شخص جریمه شده به سختی یک صفحه از متن ریاضی را حروف چینی می‌کرد، همکار وی می‌توانست بیش از ده صفحه از متن معمولی را حروف چینی کند. به علاوه، از آنجا که ریاضی‌دانان افرادی دقیق هستند، نوشتار حروف چینی شده تا چند بار، برای انجام اصلاحات، بین نویسنده و حروفچین، در رفت و برگشت بود. تا اینکه TeX در سال ۱۹۸۲ میلادی، توسط دونالد کنوث، از دانشگاه استنفورد، منتشر شد.

اما داستان TeX ریشه در سال‌های پیش از آن دارد. در سال ۱۹۶۹ میلادی، جلد اول از یک کتاب با عنوان The Art of Computer Programming نوشته شده توسط دونالد کنوث، که یک شاه‌کار به حساب می‌آید، به وسیله‌ی حروف فلزی و با شیوه قدیمی، توسط یک ماشین چاپ مربوط به قرن ۱۹ میلادی، چاپ و منتشر شد. اما وقتی ویرایش دوم از جلد دوم همان کتاب، در سال ۱۹۷۶ میلادی، با کمک فناوری عکاسی و با بکارگیری قلم‌های جدید، حروف چینی گردید، لازم شد همه کتاب دوباره حروف چینی و چاپ گردد، زیرا دیگر قلم‌های قدیمی در دسترس نبود.

هنگامی که کتاب حروفچینی شده، برای بررسی نهایی، به دست کنوث رسید، حاصل کار برای وی ناپسند آمد. در همین روزها بود که کنوث با نمونه‌هایی از حروف چینی رایانه‌ای برخورد کرد و علاقه‌مند شد تا خود برنامه‌ای رایانه‌ای، برای حروف چینی متون علمی، با کیفیت خروجی بالا، پدید آورد. وی در ماه می سال ۱۹۷۷ میلادی، شروع به کار روی یک سامانه پردازش متنی کرد، که اکنون با نام TeX شناخته می‌شود.

TeX یک سامانه جدید رایانه‌ای است که برای حروف چینی کتاب‌ها و به ویژه کتاب‌هایی که در بردارنده عبارت‌های ریاضی زیادی هستند، به طور زیبا و ماهرانه، مناسب است. کارآمدی TeX، هنگامی خودنمایی می‌کند که متن مورد نظر دارای فرمول‌های ریاضی فراوان و پیچیده است. با فراگیری اندکی از قواعد فرمول‌نویسی می‌توان هر عبارت ریاضی پیچیده را به آسانی و به زیبایی حروف چینی کرد. TeX از توانایی حمل پذیری بالایی برخوردار است. به این معنی که TeX برگستره پهناوری از سامانه‌های رایانه‌ای قابل اجرا است و رفتار و نتیجه کار، در همه‌ی این سامانه‌ها، یکسان است. این واقعیتی

است که در ارتباطات علمی و فنی بسیار اهمیت دارد. به علاوه TeX یک زبان برنامه‌نویسی هم هست و با فراگیری این زبان می‌توان به توسعه و پیشرفت آن کمک کرد. TeX بالقوه یکی از مهمترین اختراعات در زمینه حروف‌چینی نوشته‌ها در قرن حاضر به حساب می‌آید. TeX معرف یک زبان استاندارد در زمینه حروف‌چینی رایانه‌ای می‌باشد که از لحاظ اهمیت در سطح اختراع گوتنبرگ است.

۳.۱ TeX چیست؟

تک با همه مزایایی که دارد، برای استفاده گسترده و کاربرپسند دارای یک مشکل اساسی است. تک یک زبان برنامه‌نویسی واقعی، گسترده و مشکل است که یادگیری و به کارگرفتن آن برای کاربران عادی پرزحمت و غیراقتصادی است. LaTeX بخوانید: (لی تک) که در سال ۱۹۸۴ توسط Leslie Lamport به وجود آمد، در واقع مکمل تک بود. با همه چیزهایی که لازم بود به آن اضافه شود تا به محصولی قابل استفاده برای عموم تبدیل گردد. تعداد زیادی امکانات امنیتی و پیغام‌های خطا، همچنین قالب‌های متن مختلف (کتاب، نامه، گزارش و...)، امکانات فراوان برای ایجاد فصل‌ها، بخش‌ها، فهرست مطالب، فهرست راهنما (Index)، فهرست منابع (Bibliographic Index) و ایجاد پیوندهای مورد نیاز برای ساختن این فهرست‌ها در متن سند، از جمله امکاناتی هستند که در کنار سیستم حروف‌چینی و صفحه‌بندی تک، لی تک را به وجود می‌آورند. از زمان ارائه لی تک، استفاده از آن، به شیوه اصلی بهره‌گیری از سیستم تک برای تولید اسناد تبدیل شده است. اهمیت تک به عنوان قلب اصلی سیستم، همچنان محفوظ است، اما کاربران عموماً با لی تک کار می‌کنند. تک و لی تک هر دو با مجوزهای نرم افزار آزاد منتشر شده اند و کد منبع آن‌ها در دسترس همگان قرار دارد. کاربر هنگام کار با LaTeX، به جای پرداختن به جزئیات مربوط به قالب ظاهری نوشتار، روی ساختار منطقی آن متمرکز می‌شود.

مثال: نقش نویسنده، طراح کتاب، و حروف‌چین

در فرایند انتشار یک کتاب ابتدا نویسنده نوشته‌ی خود را به موسسات انتشاراتی می‌دهد. در مرحله بعدی طراح کتاب در مورد سبک نوشته (عرض ستون، قلم، فاصله قبل و بعد از سربرگ و...) تصمیم می‌گیرد. در گام بعدی طراح کتاب راهنمایی لازم را به حروف‌چین می‌کند تا کتاب را بر طبق آن حروف‌چینی کند. طراح کتاب سعی می‌کند بفهمد خواست نویسنده هنگام نوشتن کتاب چه بوده است. او در مورد سربرگ فصل‌ها، ارجاع‌ها، مثال‌ها، فرمول‌ها، و غیره بر اساس اطلاعات حرفه‌ای خود و اطلاعات در مورد محتوای نوشته تصمیم می‌گیرد.

در محیط لاتک، لاتک نقش طراح کتاب را برعهده می‌گیرد و از تک به عنوان حروف‌چین استفاده می‌کند. اما لاتک تنها یک برنامه است و بنابراین نیاز به راهنمایی دارد. نویسنده باید اطلاعات کافی در مورد ساختار منطقی کارش را به لاتک بدهد. این اطلاعات در متن به صورت فرمان‌های لاتک وارد می‌شوند. این کار کاملاً با روش WYSIWYG (What you see is what you get) تفاوت دارد که بسیاری از پردازشگرهای متنی مانند MS Word و یا Corel WordPerfect از آن پیروی می‌کنند. در این نرم افزارها، نویسنده سبک نوشتار را به صورت مستقیم هنگام نوشتن آن مشخص می‌کند. در این نرم افزارها شکل خروجی را، همزمان که نوشتار را تایپ می‌کنید، به صورت مستقیم می‌توان بر روی

صفحه نمایش دید. وقتی که از لاتک استفاده می‌کنید به طور معمول نمی‌توانید همزمان با تایپ متن شکل خروجی را ببینید، اما می‌توانید آن را بعد از پردازش توسط لاتک مشاهده کنید. در این صورت تصحیح آنرا می‌توان قبل از فرستادن نوشته به چاپگر انجام داد.

۴.۱ برخی از مزیت‌ها

- قالب‌های زیبای حرفه‌ای موجودند که متن را آن گونه طراحی می‌کنند که واقعاً باید چاپ شود.
- حروف چینی فرمول‌های ریاضی به بهترین شکل حمایت می‌شود.
- کاربر تنها کافی است تعدادی فرمان آسان را یاد بگیرد تا ساختار منطقی نوشته‌اش را طراحی کند. معمولاً لازم نیست در مورد ساختار واقعی متن نگران باشید.
- ساختارهای پیچیده مانند پانوش‌ها، ارجاع‌ها، فهرست مطالب، و کتاب‌نامه به راحتی قابل تولید هستند.
- بسته‌های اضافی مجانی بسیاری برای کارهایی که لاتک انجام نمی‌دهد وجود دارند. به عنوان مثال بسته‌های پُست اسکریپت برای گرافیک یا بسته‌هایی برای قرار دادن ارجاع‌ها به شکل استاندارد وجود دارند.
- تک، موتور لاتک، بسیار قابل انعطاف و مجانی است و روی هر سیستم عاملی کار می‌کند.

۵.۱ چه کسی به \LaTeX نیاز دارد؟

اگر متنی که شما درصدد تبدیل کردن آن به یک سند هستید، یک یا چند مورد از خصوصیات ذیل را دارد، احتمالاً برای ایجاد ساختار آن به ابزاری توانمند مانند \LaTeX نیازمندید:

- متن‌های طولانی که به فصل‌ها و بخش‌های مختلف تقسیم‌بندی می‌شود.
 - ساختارهای تکراری فراوانی که از نظر مفهومی یا ظاهری پیچیده است.
 - یک متن رسمی که نیازمند دقت بالا است. برای مثال کتاب، مقاله، تز یا یک فایل راهنما
- استفاده از یک ویرایشگر متن (حتی از نوع ساده آن) برای نوشتن ساختار سند و نهایتاً سپردن کار تولید خروجی نهایی به تک، بهبود عظیمی در روند کاری و چگونگی تفکر شما، نسبت به حالتی که از یک واژه پرداز استفاده نمایید، پدید می‌آورد.
- هنگامی که سند خود را «برنامه‌نویسی» می‌کنید، از وقفه‌های ذهنی و حرکتی متوالی در جریان کارتان بی‌نیاز می‌شوید. چرا که دیگر لازم نیست مانند وقتی که با واژه‌پرداز کار می‌کردید، دائماً به تغییر دادن اندازه، نوع و آرایش فونت‌ها، تغییر کادر، و انجام کارهای مختلف دیگر با استفاده از ماوس بپردازید. شما در میان متن‌تان پیغامی برای موتور تک می‌گذارید

که مثلاً، این یک عنوان است یا عکس را در اینجا وارد کن. تک هنگام تولید سند نهایی، این کار را به سازگارترین و شکیل‌ترین گونه ممکن انجام خواهد داد.

مسئله مهم دیگری که شاید در حین کار با واژه‌پردازها متوجه آن شده باشید این است که اساساً تعیین کردن نسبت اندازه فونت‌ها، فواصل، چگونگی قرارگیری کادرها و مسائلی از این دست در یک سند، به گونه‌ای که ظاهر آن چه روی صفحه نمایشگر و چه بعد از چاپ روی کاغذ، زیبا و حرفه‌ای باشد، به هیچ عنوان کار ساده و سریعی نیست. اگر سند شما حاوی ساختارهای پیچیده‌ای مانند فرمول‌های ریاضی، نمودار، جدول و مواد مشابه باشد، کار باز هم سخت‌تر می‌شود. شاید برای شما هم پیش آمده باشد که وقت زیادی برای تهیه یک سند و آراستن آن به شیوه‌ای که دوست دارید (یا فکر می‌کنید درست است) صرف کنید و نهایتاً وقتی آن را چاپ می‌کنید، از حاصل کار دل‌سرد شوید. مثلاً ممکن است عناوین فصل‌ها را برای خوانایی بهتر زیاده از حد بزرگ کنید، به گونه‌ای که بعد از چاپ، اندازه آن مناسب نباشد. یا این که فونت‌های مختلف را به گونه‌ای نازیبا با هم به کار برده باشید. واقعیت آن است که اگرچه دسترسی به ابزارهای نشر رومیزی امروزه بسیار آسان شده است، برای تهیه سندی که حروف‌چینی و صفحه‌آرایی زیبا و متناسبی داشته باشد، فقط دانستن شیوه کار با یک واژه‌پرداز کافی نیست و یا شاید اصلاً به کار نیاید. چنین جزئیاتی در حیطه تخصص و تجربه حرفه‌ای‌های گرافیک، حروف‌چینی و صفحه‌بندی قرار دارند.

مجله‌ها و ناشران مختلف برای تدوین اختصاصی اندازه‌ها و نسبت فواصل در صفحات نشریات خود افراد زبده‌ای را به خدمت می‌گیرند. یکی از مهم‌ترین مزیت‌های تک این است که امکان تنظیم بسیار دقیق اندازه‌ها و فواصل را فراهم می‌کند. سیستم داخلی اندازه‌گیری تک فواصل را دقیق‌تر از طول موج نور مرئی، اندازه‌گیری و تعیین می‌نماید. اما چنین امکاناتی فقط به درد حرفه‌ای‌ها می‌خورد.

خبر خوب آن است که تنظیمات پیش‌فرض در تک و مخصوصاً در قالب مجموعه لی‌تک بر اساس تجربه‌های چندصدساله حروفچین‌ها و صفحه‌بندهای کامپیوتری و غیرکامپیوتری ایجاد شده و خروجی‌های بسیار زیبایی را برای شما به ارمغان می‌آورند. در عین حال، مجبور نیستید با این پیش‌فرض‌ها کار کنید. می‌توانید اندازه‌ها و فواصل را در سند خودتان با دستورات متنوعی که در اختیار دارید، تغییر دهید، یا فایل‌های Style یا پکیج‌های مختلف را از اینترنت دریافت نمایید و آن‌ها را بر سند خود اعمال کنید یا حتی فایل‌های Style خود را بسازید، ذخیره نمایید یا منتشر کنید. همه چیز در اختیار شماست.

به طور خاص، تک به عنوان بهترین گزینه برای حروف‌چینی متونی که حاوی فرمول‌های ریاضی متعدد هستند، شناخته می‌شود و در واقع یکی از مهم‌ترین اهداف اولیه، ایجاد سیستمی بود که فرمول‌های ریاضی را به شکلی دقیق، صحیح و البته شکیل و زیبا حروفچینی نماید. اگر تاکنون تجربه وارد کردن یک فرمول ریاضی را در متن خود در برنامه‌ای مانند Microsoft Office Word داشته‌اید، حتماً به دشواری انجام دادن این کار واقف هستید و در ضمن برای این کار لازم است پکیج Microsoft Equation را روی سیستم خود نصب کرده باشید تا از امکانات فرمول‌نویسی Word استفاده نمایید. حال فرض کنید که یک سند Word را که حاوی فرمول‌های ریاضی است، روی کامپیوتر خودتان تهیه، ویرایش و ذخیره کرده‌اید.

اگر این سند را روی کامپیوتر دیگری که Word مایکروسافت روی آن نصب شده است باز کنید، ممکن است متوجه شوید که شکل ظاهری فرمول‌ها نسبت به آن چه که روی کامپیوتر خودتان دیده می‌شد تغییر کرده یا اصلاً بعضی از حروف و علائم ریاضی که شما وارد کرده‌اید، روی این دستگاه جدید جای خود را به علائم دیگری داده‌اند. در مقابل، شما می‌توانید هر نوع فرمول و علائم ریاضی را با آرایش دلخواه‌تان توسط تک بنویسید و فایل خروجی نهایی روی همه کامپیوترها به شکل یکسانی تولید می‌شود و به نمایش در می‌آید.

در محیط‌هایی مانند تک، شما سند خود را برنامه‌نویسی می‌کنید و نهایتاً برای دیدن سند به شکلی که بعد از چاپ دیده خواهد شد، باید کد کامل شده را کامپایل کنید. فرآیند کامپایل کردن و مشاهده خروجی نهایی همزمان با وارد کردن متن به ویرایشگر انجام نمی‌شود. بنابراین محیط‌های لی تک غالباً به طور مستقیم و همزمان WYSIWYG نیستند. اگرچه این رهیافت ممکن است برای کاربرانی که به استفاده از ابزارهای بصری WYSIWYG هم زمان عادت کرده‌اند خوشایند نباشد، باز هم مزیت‌های بسیاری دارد. مهم‌ترین آن‌ها همانگونه که در ابتدا گفتیم آن است که روال کاری و فکری کاربر در حین تایپ کردن سند خود به طور مداوم به هم نمی‌خورد. در ضمن تغییراتی که در ظاهر سندتان می‌دهد چون باید به شکل کد بیان شوند، دقیق و منسجم اعمال می‌شوند و مدیریت آن‌ها هم ساده‌تر می‌شود.

۶.۱ نصب LaTeX

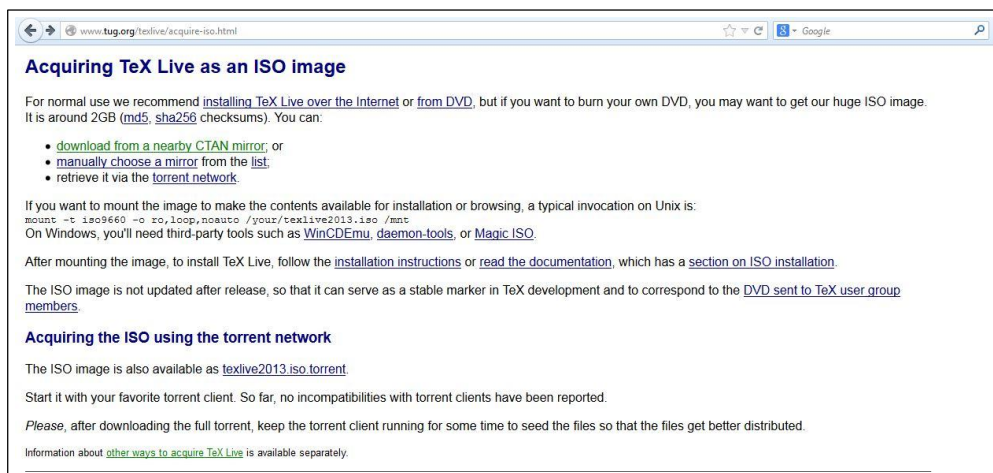
توزیع‌های مختلفی از LaTeX وجود دارند که به راحتی قابل دریافت هستند. از جمله محبوب‌ترین توزیع‌ها می‌توان به MikTeX و LiveTeX اشاره کرد. در این دوره از توزیع Livetex استفاده خواهیم کرد. هر دو این توزیع‌ها اکثر package های مورد نیاز را به همراه دارند. برای نصب Livetex ابتدا نسخه نهایی آنرا تهیه کنید. برای اینکار می‌توانید به آدرس زیر مراجعه کنید.

<http://www.tug.org/texlive/acquire-iso.html>

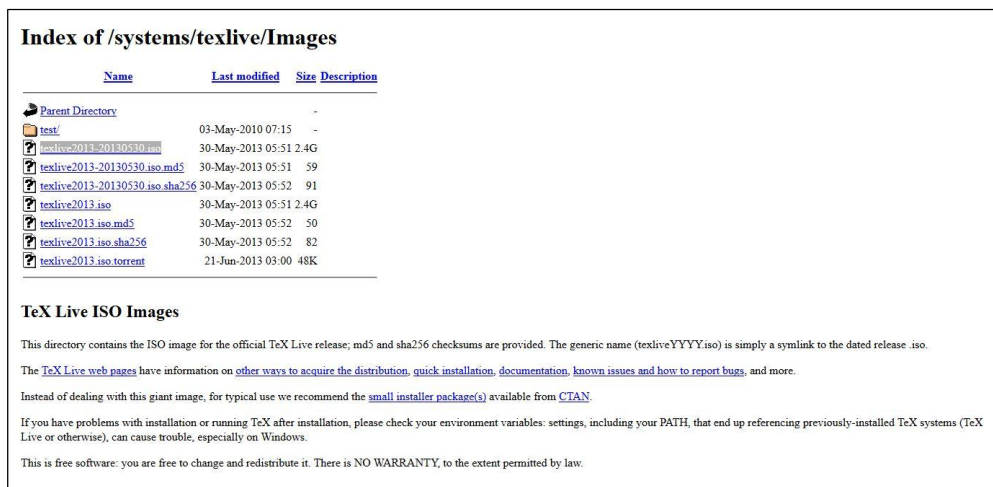
صفحه اصلی آدرس بالا را می‌توانید در شکل ۱ مشاهده کنید. در این صفحه بر روی لینک که به رنگ سبز مشخص شده کلیک کنید تا به نزدیک ترین سرور منتقل شوید. نتیجه کار در شکل ۲ آمده است. در این صفحه بر روی گزینه مشخص شده کلیک کنید تا فایل ISO نرم افزار را دریافت کنید. اگر در محل دانشکده قرار دارید می‌توانید از آدرس زیر اقدام به دریافت نرم افزار نمایید (شکل ۳).

<ftp://ftp.sharif.edu/Engineering/LaTeX/>

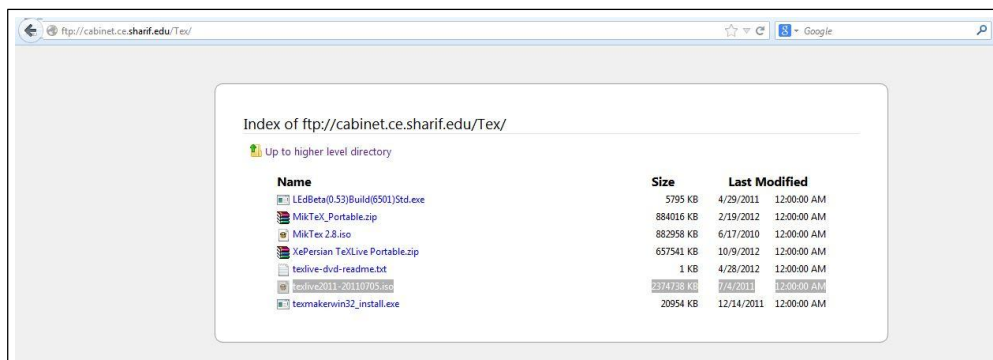
برای نصب ابتدا لازم است نسخه ISO را بر روی دیسک dvd قرار دهید و یا با استفاده از نرم‌افزارهای مربوطه همچون virtual CloneDrive فایل مربوطه را باز کنید. در این مرحله بر روی فایل اجرایی install-tl.bat که در شکل ۴ برجسته شده است، کلیک کنید. (بهرتر است قبل از شروع مراحل نصب، اقدام به غیر فعال نمودن Anti virus نمایید.) در مرحله بعدی گزینه‌های next را انتخاب کنید و براحتی نرم افزار را نصب نمایید (شکل‌های ۵، ۶ و ۷).



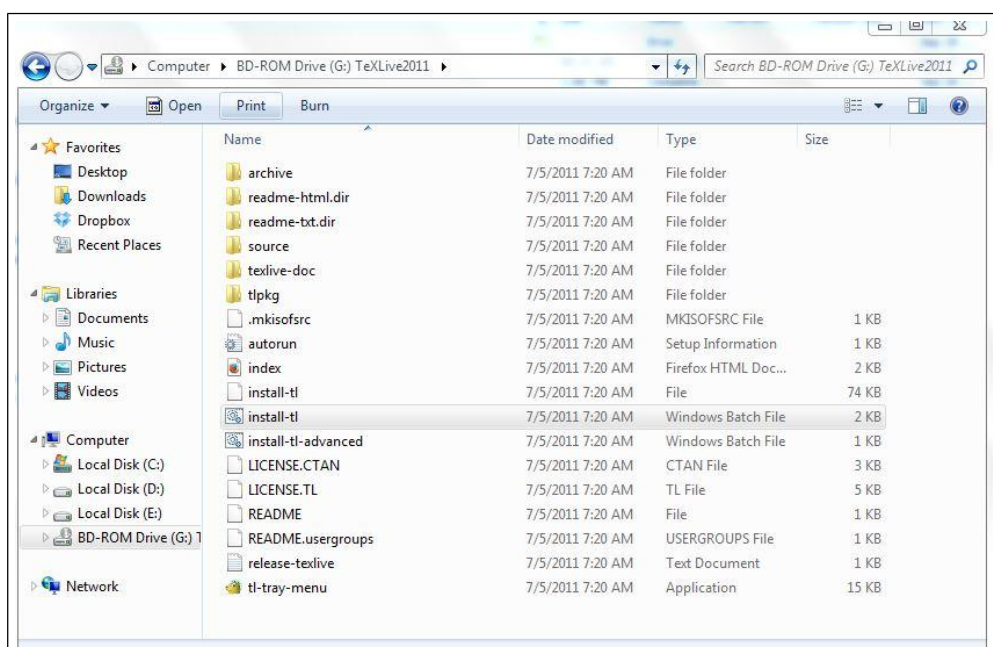
شکل ۱: صفحه اصلی دریافت آخرین نسخه Livetex



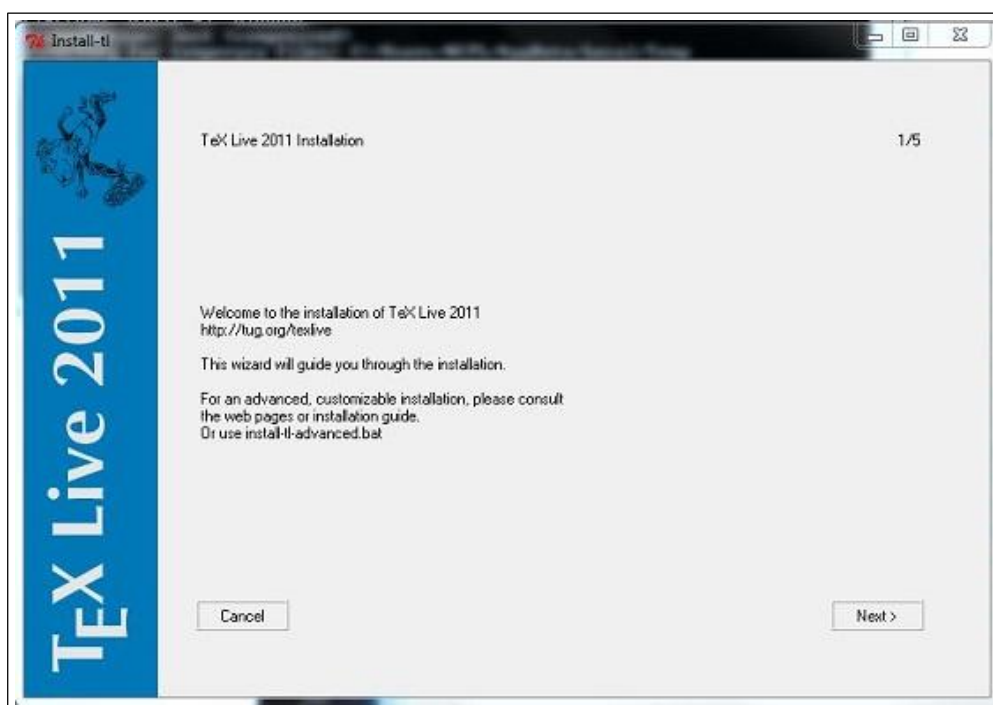
شکل ۲: لینک دریافت آخرین نسخه Livetex



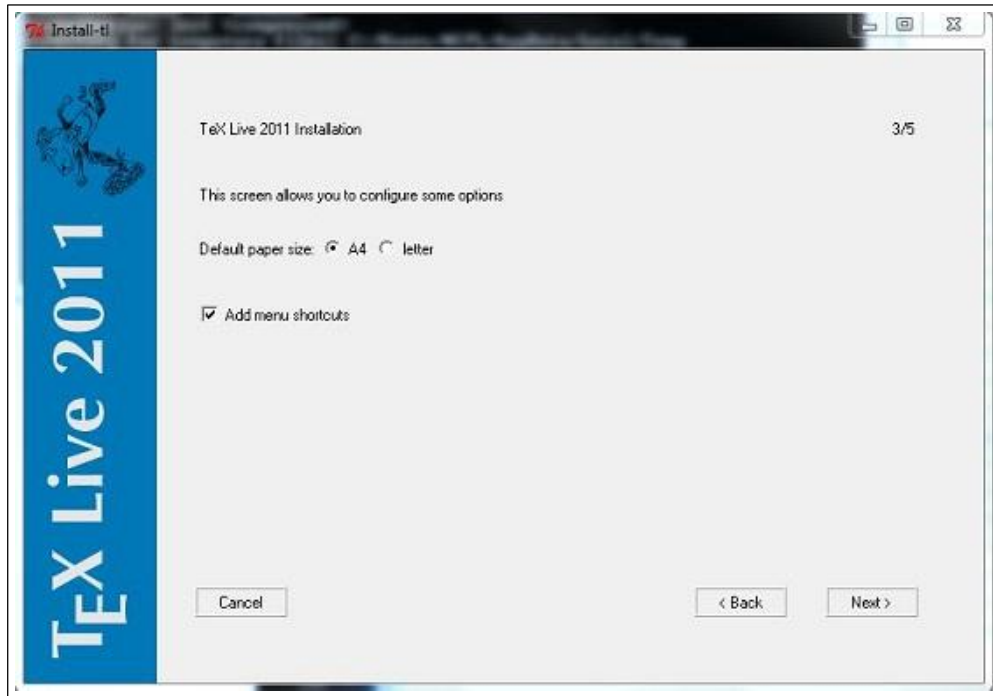
شکل ۳: دریافت Livetex از سرور دانشکده



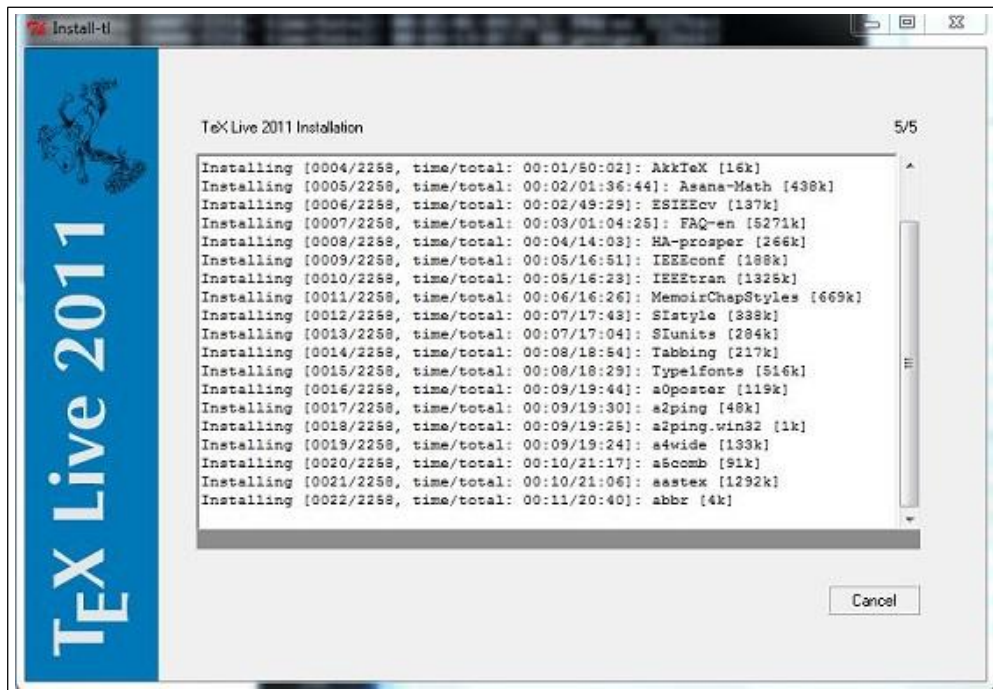
شکل ۴: انتخاب فایل نصب Livetex



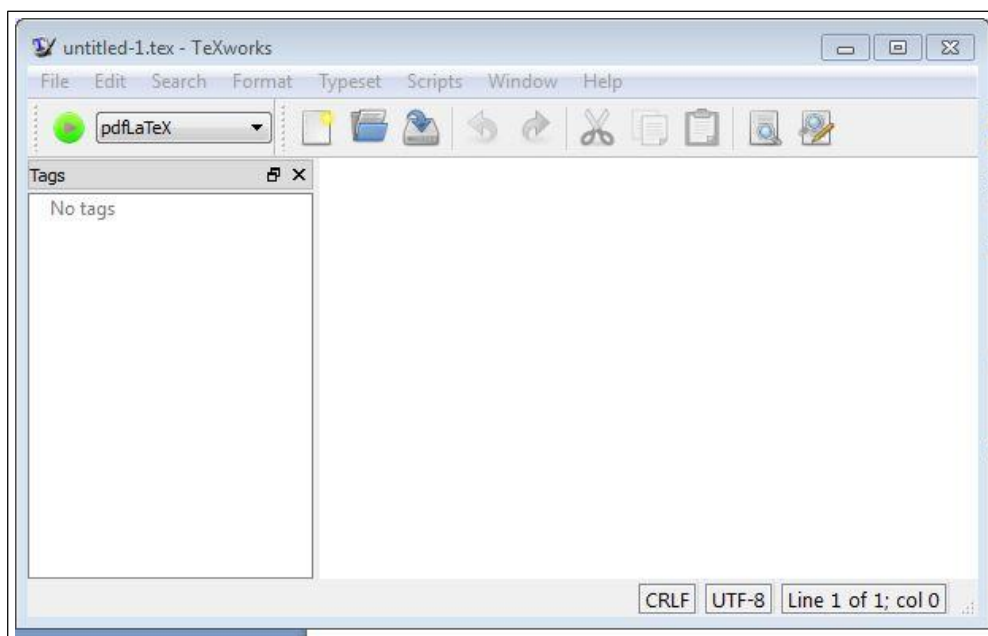
شکل ۵: نصب Livetex



شکل ۶: نصب Livetex



شکل ۷: نصب Livetex



شکل ۸: ویرایشگر TeXworks editor

۲ مقدمات کار با \LaTeX

۱.۲ ویرایشگر

در این مرحله باید از یک ویرایشگر استفاده کنید، می‌توانید از ویرایشگرهای عمومی چون ++notepad نیز استفاده کنید. به طور کلی ما دو نوع ورودی خواهیم داشت. یک ورودی متن مورد نظرمان و ورودی دیگر دستورات latex می‌باشد. هر دو این ورودی‌ها در کنار هم نوشته می‌شوند و در مرحله compile خود \LaTeX با استفاده از دستورات وارد شده، فرمت مورد نظر ما را روی متن اعمال می‌کند. توصیه می‌شود از ویرایشگری استفاده کنید که بتواند خطاهای تایپی مربوط به دستورات \LaTeX را تشخیص دهد. برای این کار می‌توانید از ویرایشگری که خود توزیع texlive در اختیار ما می‌گذارد استفاده کنیم. نام این ویرایشگر TeXworks editor می‌باشد. در شکل ۸ تصویر کلی از این ویرایشگر آمده است. لازم به ذکر است که ویرایشگرهای دیگری نیز که دارای امکانات بیشتر نیز هستند، موجود است و می‌توانید از آنها استفاده کنید. نکته مهم این اسن که تحت هر شرایط شما باید خود Texlive یا MikTeX را نصب کرده باشید. از جمله معروف‌ترین این ویرایشگرها می‌توان به موارد زیر اشاره کرد.

• TeXstudio

• TeXmaker

• LyX

• AUCTeX

• TeX BaKoMa

همانطور که پیش‌تر نیز گفته شد، ما در این دوره از ویرایشگر خود Texlive یعنی ویرایشگر TeXworks editor استفاده خواهیم کرد.

۲.۲ شروع کار

در ابتدا لازم است توجه شما را به چند نکته مهم جلب نمایم.

نکته ۱ فایل‌هایی که به هنگام *compile* مورد پردازش قرار می‌گیرند، فایل‌های دارای پسوند *.tex* می‌باشند.

نکته ۲ دستورات در *LaTeX* با \backslash شروع می‌شوند و آرگومان‌های دستورات در $\{ \}$ قرار می‌گیرند.

نکته ۳ *LaTeX* به حروف کوچک و بزرگ حساس است.

۱.۲.۲ خطوط کلیدی مورد نیاز

همه فایل‌های *LaTeX* حتماً باید سه خط زیر را داشته باشند. در صورت عدم وارد کردن این خطوط چیزی پردازش نخواهد شد (شکل ۹).

۱. مشخص کردن قالب (style) کلی متن. برای مثال، book report و ...

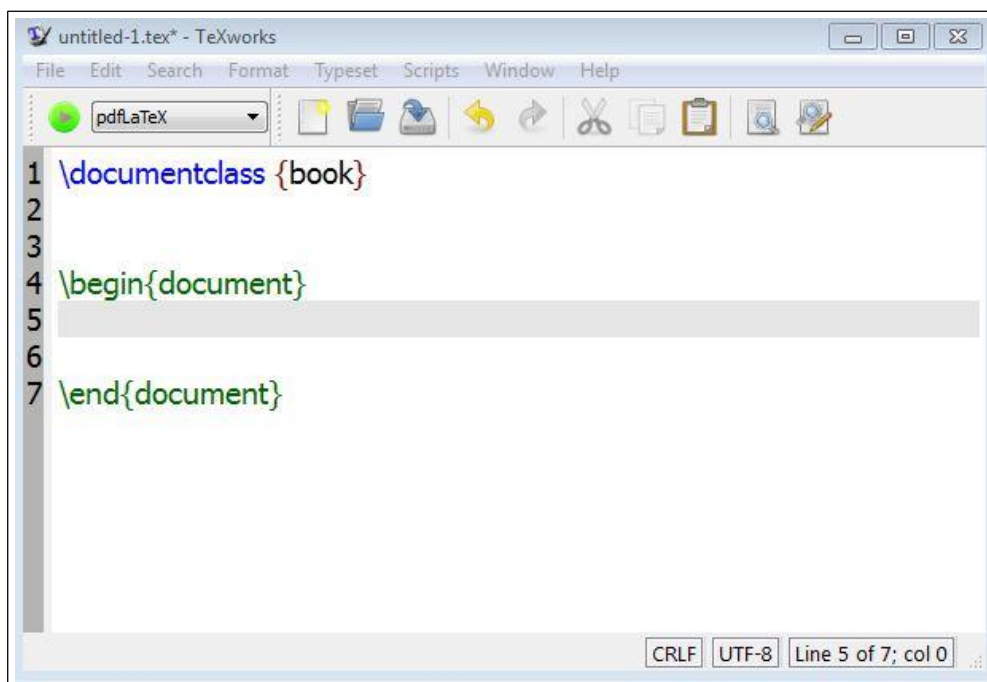
```
\documentclass{book}
```

۲. شروع ناحیه مربوط به متن

```
\begin{document}
```

۳. اتمام ناحیه مربوط به متن

```
\end{document}
```

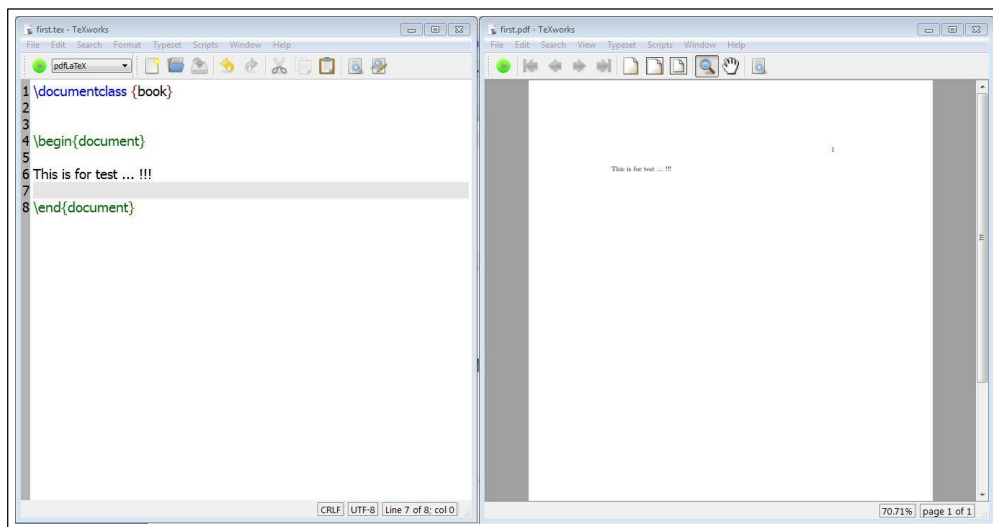


شکل ۹: خطوط کلیدی

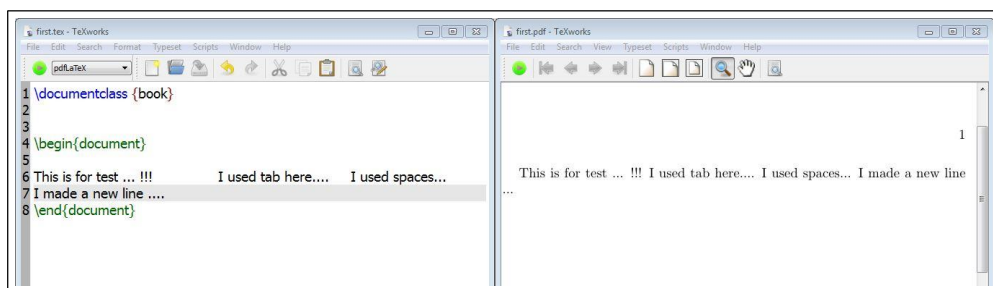
کلیه متن مربوطه باید در داخل قسمت document قرار گیرد. از سوی دیگر باید بدانیم که compiler بعد از $\end{document}$ چیزی را پردازش نخواهد کرد. پردازش متن در واقع به معنای حروف چینی متن با توجه به تنظیمات مد نظر است.

برای امتحان کردن ابتدا یک متن انگلیسی در داخل قسمت begin و end می‌نویسیم. سپس چون متن انگلیسی وارد کرده‌ایم باید یک کامپایلر متناسب انتخاب کنیم. جهت این کار کامپایلر pdflatex را انتخاب می‌کنیم و سپس کل فایل را کامپایل می‌کنیم. بعد از پردازش‌های لازم نسخه pdf نمایش داده می‌شود (شکل ۱۰). چون نوع documentclass را از نوع book انتخاب کرده‌ایم. برخی تنظیمات چون شماره صفحه و فاصله متن از بالا و سمت راست به صورت اتوماتیک اعمال می‌شود.

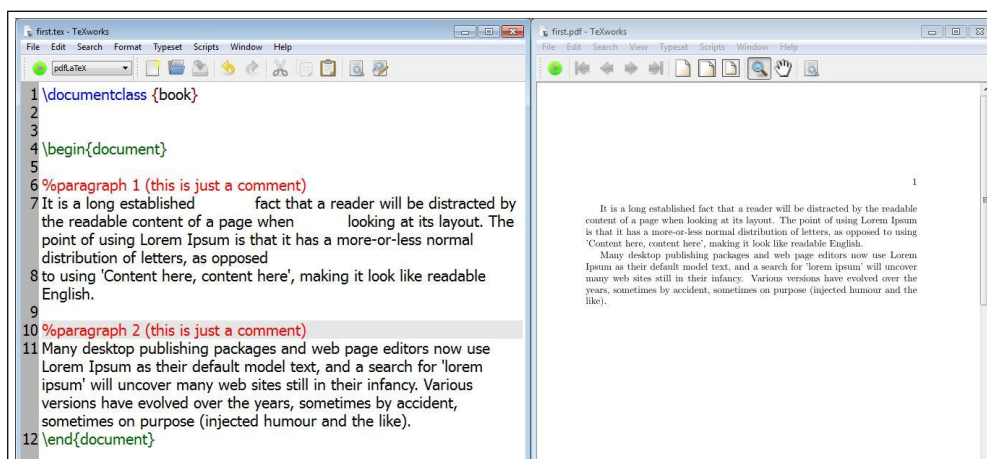
نکته ۴ ایجاد فاصله و خط جدید (بدون استفاده از دستورات تعریف شده) کار خاصی انجام نمی‌دهد (شکل ۱۱). به عبارت دیگر در $LaTeX$ ایجاد خط جدید و ایجاد فاصله صرفاً با وارد کردن دستورات معین قابل اعمال است.



شکل ۱۰: کامپایل و نمایش خروجی



شکل ۱۱: عدم تاثیر فاصله و خط جدید در خروجی



شکل ۱۲: کامنت گذاری و ایجاد پاراگراف جدید

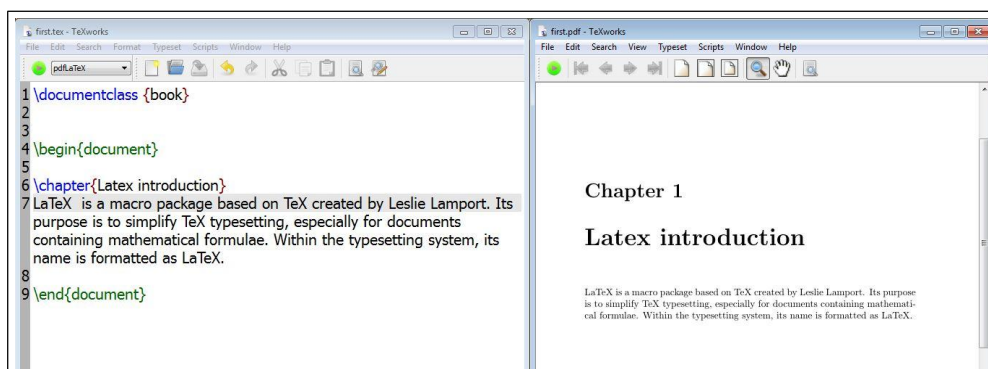
نکته ۵ برای ایجاد *comment* از % استفاده می کنیم.

نکته ۶ همانطور که در نکات قبلی گفته شد، وجود فاصله و یا ایجاد خط جدید هیچ معنی خاصی ندارد و \LaTeX کل متن وارد شده را به صورت یک پاراگراف پیوسته کامپایل می کند. برای ایجاد پاراگراف جدید کافی است یک خط خالی کامل ایجاد کنیم (شکل ۱۲).

۳.۲ بسته های اضافی

خود \LaTeX در حالت کلی برای حروف چینی استفاده می شود و برخی قابلیت ها به صورت پیش فرض در آن تعبیه نشده است. از جمله این قابلیت ها می توان به نحوه مدیریت تصاویر و اشکال، پشتیبانی از زبان های مختلف چون فارسی و ... اشاره کرد. در حالت کلی برای رفع این کمبودها می توان به صورت `package` این قابلیت ها را به \LaTeX افزود. بسته ها باید قبل از دستور `\begin{document}` تعریف شوند.

```
\usepackage{Package name}
```



شکل ۱۳: ایجاد فصل

۳ ساختار دهی به متن

۱.۳ ایجاد فصل، بخش، زیربخش، زیر زیر بخش و فهرست مطالب

برای ایجاد فصل باید از دستور زیر استفاده کنیم.

`\chapter{NAME}`

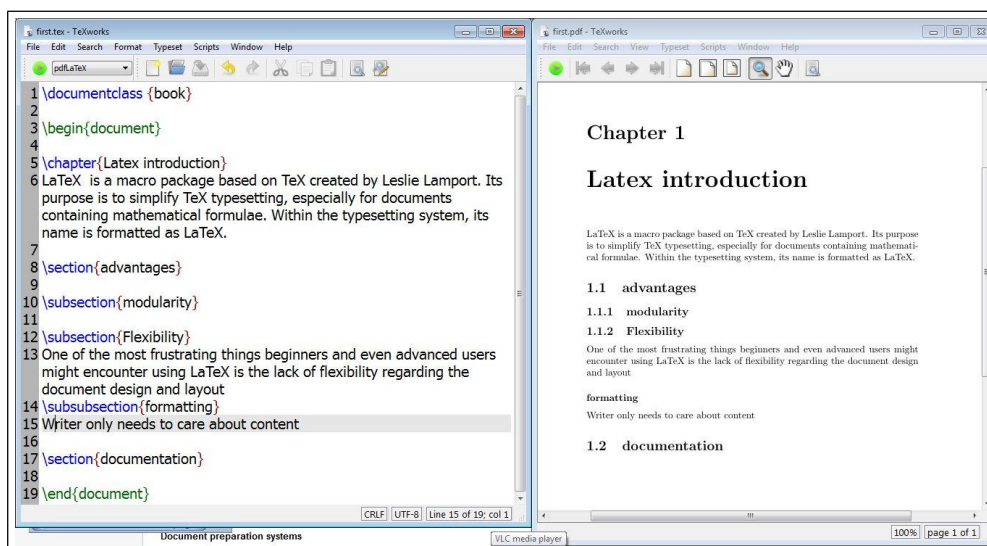
پس از استفاده از این دستور خود \LaTeX با توجه به نوع قالب انتخابی فاصله های لازم را در مکان های لازم اعمال می کند (شکل ۱۳). برای ایجاد بخش و زیر بخش ها، دستورات زیر در \LaTeX موجود است (مثال در شکل ۱۴ آمده است).

- `\section{title}`
- `\subsection{title}`
- `\subsubsection{title}`

دستور `\subsubsection` جزء نهایی آخر می باشد و در واقع پایین ترین سطح تقسیم کننده متن می باشد.

نکته ۷ اگر در قالب کتاب اقدام به ایجاد فصل نمایید، شاید در بین فصل ها شاهد این باشید که صفحه خالی وجود دارد. علت این امر آن است که فصل جدید باید از یک صفحه فرد شروع شود. در واقع خود \LaTeX به صورت اتوماتیک این کار را انجام می دهد.

با استفاده از دستور زیر می توان فهرست مطالب ساخت. فقط نکته مهم آن است که برای ظاهر شدن صحیح فهرست حتما باید دوبار اقدام به کامپایل متن نمایید.



شکل ۱۴: ایجاد بخش و زیر بخش‌ها

نکته ۸ با استفاده از دستور `tableofcontents` می‌توان فهرست مطالب ساخت (شکل ۱۵). فقط نکته مهم آن است که برای ظاهر شدن صحیح فهرست حتماً باید دوبار اقدام به کامپایل متن نمایید.

۲.۳ شماره‌گذاری و مورد بندی

برای شماره‌گذاری آیتم‌ها (`enumerate`) از فرمت زیر استفاده می‌کنیم (شکل ۱۶).

```
\begin{enumerate}
```

```
\item
```

write item 1 here

```
\item
```

write item 2 here

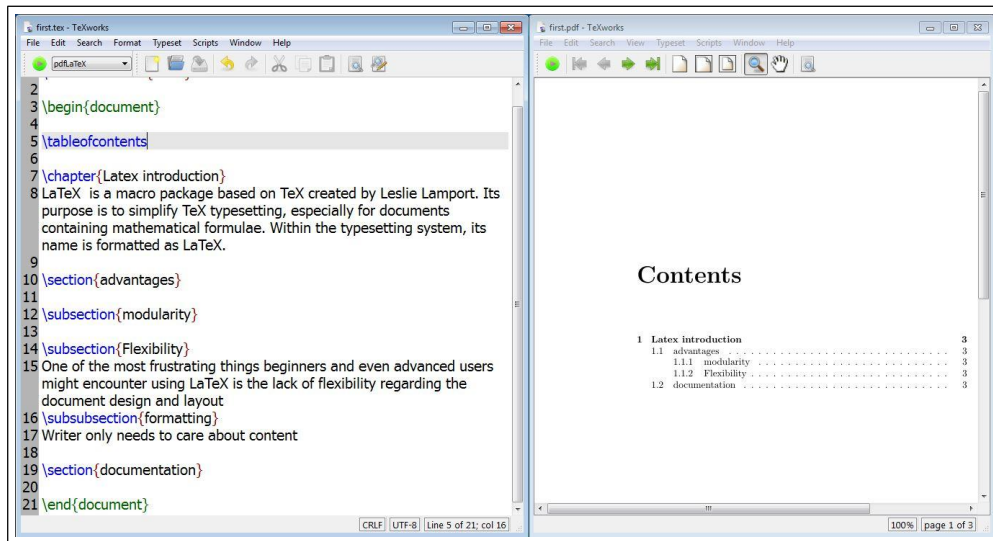
```
\item
```

write item 3 here

```
\end{enumerate}
```

برای مورد بندی آیتم‌ها (`itemize`) از فرمت زیر استفاده می‌کنیم (شکل ۱۶).

```
\begin{itemize}
```



شکل ۱۵: ایجاد فهرست مطالب

`\item`

write item 1 here

`\item`

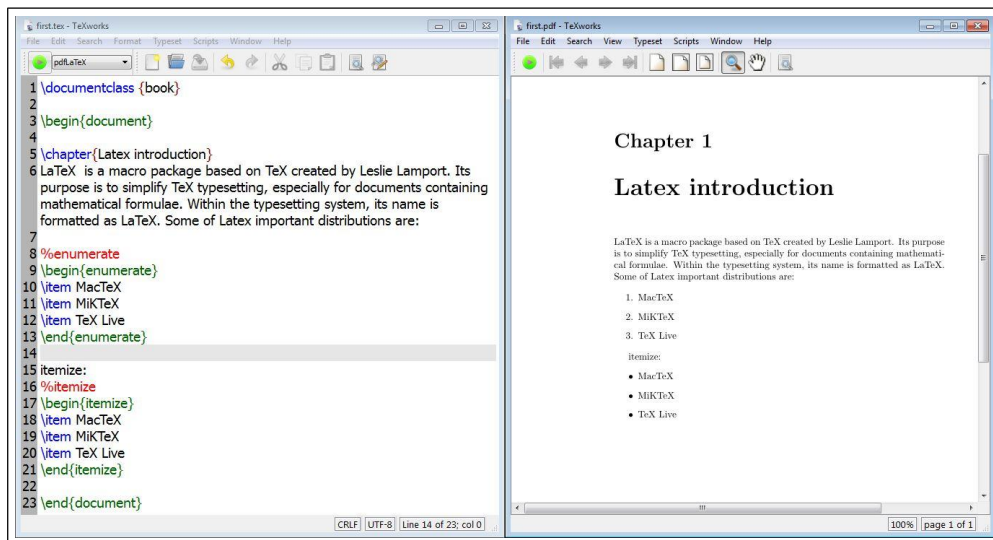
write item 2 here

`\item`

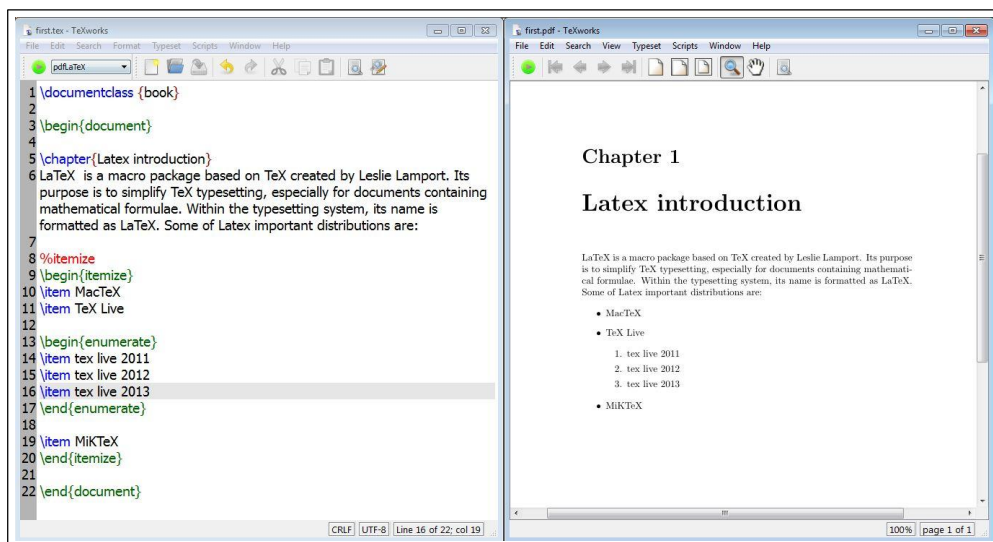
write item 3 here

`\end{itemize}`

نکته ۹ `enumerate` و `itemize` را می توان به صورت تو در تو نوشت (شکل ۱۷).



شکل ۱۶: شماره‌گذاری و مورد بندی



شکل ۱۷: شماره‌گذاری و مورد بندی تو در تو

α	<code>\alpha</code>	θ	<code>\theta</code>	o	<code>o</code>	τ	<code>\tau</code>
β	<code>\beta</code>	ϑ	<code>\vartheta</code>	π	<code>\pi</code>	υ	<code>\upsilon</code>
γ	<code>\gamma</code>	γ	<code>\gamma</code>	ϖ	<code>\varpi</code>	ϕ	<code>\phi</code>
δ	<code>\delta</code>	κ	<code>\kappa</code>	ρ	<code>\rho</code>	φ	<code>\varphi</code>
ϵ	<code>\epsilon</code>	λ	<code>\lambda</code>	ϱ	<code>\varrho</code>	χ	<code>\chi</code>
ε	<code>\varepsilon</code>	μ	<code>\mu</code>	σ	<code>\sigma</code>	ψ	<code>\psi</code>
ζ	<code>\zeta</code>	ν	<code>\nu</code>	ς	<code>\varsigma</code>	ω	<code>\omega</code>
η	<code>\eta</code>	ξ	<code>\xi</code>				
Γ	<code>\Gamma</code>	Λ	<code>\Lambda</code>	Σ	<code>\Sigma</code>	Ψ	<code>\Psi</code>
Δ	<code>\Delta</code>	Ξ	<code>\Xi</code>	Υ	<code>\Upsilon</code>	Ω	<code>\Omega</code>
Θ	<code>\Theta</code>	Π	<code>\Pi</code>	Φ	<code>\Phi</code>		

Table 1: Greek Letters

شکل ۱۸: برخی از علائم مهم در فرمول نویسی به همراه دستورات

<code>\arccos</code>	<code>\cos</code>	<code>\csc</code>	<code>\exp</code>	<code>\ker</code>	<code>\limsup</code>	<code>\min</code>	<code>\sinh</code>
<code>\arcsin</code>	<code>\cosh</code>	<code>\deg</code>	<code>\gcd</code>	<code>\lg</code>	<code>\ln</code>	<code>\Pr</code>	<code>\sup</code>
<code>\arctan</code>	<code>\cot</code>	<code>\det</code>	<code>\hom</code>	<code>\lim</code>	<code>\log</code>	<code>\sec</code>	<code>\tan</code>
<code>\arg</code>	<code>\coth</code>	<code>\dim</code>	<code>\inf</code>	<code>\liminf</code>	<code>\max</code>	<code>\sin</code>	<code>\tanh</code>

Table 8: Log-like Symbols

شکل ۱۹: برخی از علائم مهم در فرمول نویسی به همراه دستورات

۴ فرمول نویسی

۱.۴ مقدمات و آشنایی با علائم

برای فرمول نویسی ابتدا باید بدانیم که کارکترهای مورد نیاز هر کدام دارای یک دستور منحصر بفرد هستند. برای دسترسی به لیست کامل این دستورات می توانید به آدرس زیر مراجعه کنید.

<http://web.ift.uib.no/Teori/KURS/WRK/TeX/symALL.html>

در ادامه و در شکل های ۱۸، ۱۹، ۲۰، ۲۱، ۲۲، ۲۳، ۲۴ برخی از مهمترین و پرکارترین دستورات آمده است.

\widetilde{abc}	<code>\widetilde{abc}</code>	\widehat{abc}	<code>\widehat{abc}</code>
\overleftarrow{abc}	<code>\overleftarrow{abc}</code>	\overrightarrow{abc}	<code>\overrightarrow{abc}</code>
\overline{abc}	<code>\overline{abc}</code>	\underline{abc}	<code>\underline{abc}</code>
\overbrace{abc}	<code>\overbrace{abc}</code>	\underbrace{abc}	<code>\underbrace{abc}</code>
\sqrt{abc}	<code>\sqrt{abc}</code>	$\sqrt[n]{abc}$	<code>\sqrt[n]{abc}</code>
f'	<code>f'</code>	$\frac{abc}{xyz}$	<code>\frac{abc}{xyz}</code>

Table 12: Some other constructions

شکل ۲۰: برخی از علائم مهم در فرمول نویسی به همراه دستورات

\leq	<code>\leq</code>	\geq	<code>\geq</code>	\equiv	<code>\equiv</code>	\models	<code>\models</code>
\prec	<code>\prec</code>	\succ	<code>\succ</code>	\sim	<code>\sim</code>	\perp	<code>\perp</code>
\preceq	<code>\preceq</code>	\succeq	<code>\succeq</code>	\simeq	<code>\simeq</code>	\mid	<code>\mid</code>
\ll	<code>\ll</code>	\gg	<code>\gg</code>	\asymp	<code>\asymp</code>	\parallel	<code>\parallel</code>
\subset	<code>\subset</code>	\supset	<code>\supset</code>	\approx	<code>\approx</code>	\bowtie	<code>\bowtie</code>
\subseteq	<code>\subseteq</code>	\supseteq	<code>\supseteq</code>	\cong	<code>\cong</code>	\Join^b	<code>\Join^b</code>
\sqsubset^b	<code>\sqsubset^b</code>	\sqsupset^b	<code>\sqsupset^b</code>	\neq	<code>\neq</code>	\smile	<code>\smile</code>
\sqsubseteq	<code>\sqsubseteq</code>	\sqsupseteq	<code>\sqsupseteq</code>	\doteq	<code>\doteq</code>	\frown	<code>\frown</code>
\in	<code>\in</code>	\ni	<code>\ni</code>	\propto	<code>\propto</code>	$=$	<code>=</code>
\vdash	<code>\vdash</code>	\dashv	<code>\dashv</code>	$<$	<code><</code>	$>$	<code>></code>
$:$	<code>:</code>						

شکل ۲۱: برخی از علائم مهم در فرمول نویسی به همراه دستورات

\sum	<code>\sum</code>	\bigcap	<code>\bigcap</code>	\bigodot	<code>\bigodot</code>
\prod	<code>\prod</code>	\bigcup	<code>\bigcup</code>	\bigotimes	<code>\bigotimes</code>
\coprod	<code>\coprod</code>	\bigsqcup	<code>\bigsqcup</code>	\bigoplus	<code>\bigoplus</code>
\int	<code>\int</code>	\bigvee	<code>\bigvee</code>	\biguplus	<code>\biguplus</code>
\oint	<code>\oint</code>	\bigwedge	<code>\bigwedge</code>		

Table 7: Variable-sized Symbols

شکل ۲۲: برخی از علائم مهم در فرمول نویسی به همراه دستورات

\leftarrow	<code>\leftarrow</code>	\longleftarrow	<code>\longleftarrow</code>	\uparrow	<code>\uparrow</code>
\Leftarrow	<code>\Leftarrow</code>	\Longleftarrow	<code>\Longleftarrow</code>	\Uparrow	<code>\Uparrow</code>
\rightarrow	<code>\rightarrow</code>	\longrightarrow	<code>\longrightarrow</code>	\downarrow	<code>\downarrow</code>
\Rightarrow	<code>\Rightarrow</code>	\Longrightarrow	<code>\Longrightarrow</code>	\Downarrow	<code>\Downarrow</code>
\leftrightarrow	<code>\leftrightarrow</code>	\longleftrightarrow	<code>\longleftrightarrow</code>	\updownarrow	<code>\updownarrow</code>
\Leftrightarrow	<code>\Leftrightarrow</code>	\Longleftrightarrow	<code>\Longleftrightarrow</code>	\Updownarrow	<code>\Updownarrow</code>
\mapsto	<code>\mapsto</code>	\longmapsto	<code>\longmapsto</code>	\nearrow	<code>\nearrow</code>
\hookrightarrow	<code>\hookrightarrow</code>	\hookleftarrow	<code>\hookleftarrow</code>	\searrow	<code>\searrow</code>
\leftharpoonup	<code>\leftharpoonup</code>	\rightharpoonup	<code>\rightharpoonup</code>	\swarrow	<code>\swarrow</code>
\leftharpoondown	<code>\leftharpoondown</code>	\rightharpoondown	<code>\rightharpoondown</code>	\nwarrow	<code>\nwarrow</code>
\rightleftharpoons	<code>\rightleftharpoons</code>	\leadsto	<code>\leadsto</code> ^b		

شکل ۲۳: برخی از علائم مهم در فرمول نویسی به همراه دستورات

\pm	<code>\pm</code>	\cap	<code>\cap</code>	\diamond	<code>\diamond</code>	\oplus	<code>\oplus</code>
\mp	<code>\mp</code>	\cup	<code>\cup</code>	\triangleup	<code>\triangleup</code>	\ominus	<code>\ominus</code>
\times	<code>\times</code>	\uplus	<code>\uplus</code>	\triangledown	<code>\triangledown</code>	\otimes	<code>\otimes</code>
\div	<code>\div</code>	\sqcap	<code>\sqcap</code>	\triangleleft	<code>\triangleleft</code>	\oslash	<code>\oslash</code>
$*$	<code>\ast</code>	\sqcup	<code>\sqcup</code>	\triangleright	<code>\triangleright</code>	\odot	<code>\odot</code>
\star	<code>\star</code>	\vee	<code>\vee</code>	\lhd ^b	<code>\lhd</code> ^b	\bigcirc	<code>\bigcirc</code>
\circ	<code>\circ</code>	\wedge	<code>\wedge</code>	\rhd ^b	<code>\rhd</code> ^b	\dagger	<code>\dagger</code>
\bullet	<code>\bullet</code>	\setminus	<code>\setminus</code>	\unlhd ^b	<code>\unlhd</code> ^b	\ddagger	<code>\ddagger</code>
\cdot	<code>\cdot</code>	\wr	<code>\wr</code>	\unrhd ^b	<code>\unrhd</code> ^b	\amalg	<code>\amalg</code>
$+$	<code>+</code>	$-$	<code>-</code>				

شکل ۲۴: برخی از علائم مهم در فرمول نویسی به همراه دستورات

۲.۴ دستورات اولیه

برای فرمول‌نویسی، لازم در محیط مربوطه این کار را انجام دهید. محیط مخصوص این کار `equation` نام دارد و به صورت زیر استفاده می‌شود.

```
\begin{equation}
content...
\end{equation}
```

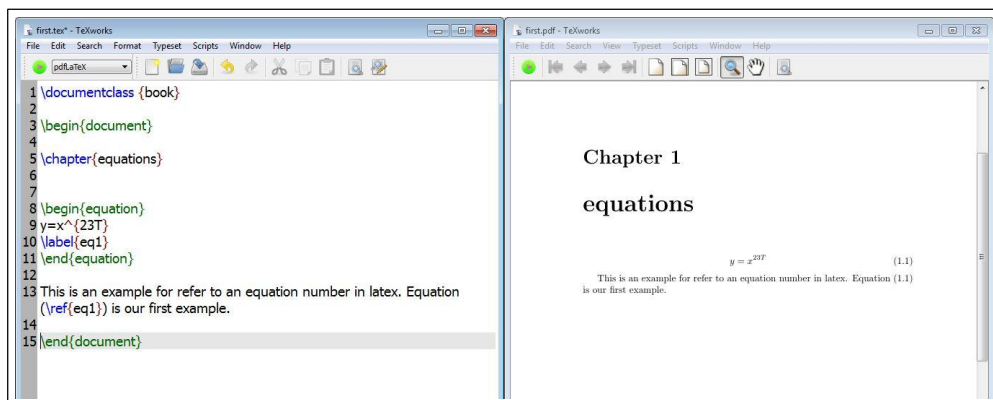
در شکل ۲۵ سعی شده است تا لیست کامل و متنوعی از فرمول‌ها به همراه دستور کامل آنها آورده شود. برای منابع بیشتر به وب سایت‌های زیر مراجعه نمایید.

- <http://www.personal.ceu.hu/tex/cookbook.html>
- <http://elevatorlady.ca/doc/refcard/expressions.html>
- <http://en.wikibooks.org/wiki/LaTeX/Mathematics>
- <http://www.codecogs.com/latex/eqneditor.php>

نکته ۱۰ معادله‌های ریاضی (*equation*) دارای شماره‌گذاری خودکار هستند و در این شماره‌گذاری از شماره فصل پیروی می‌کنند.

ردیف	عنوان	دستورات Latex	خروجی
1	Simple sum	$\begin{equation}$ $y=x^2$ $\end{equation}$	$y = x^2$
		$\begin{equation}$ $y=x^{23T}$ $\end{equation}$	$y = x^{23T}$
		$\begin{equation}$ $y=x^{\{23T\}}$ $\end{equation}$	$y = x^{23T}$
2	square root	$\begin{equation}$ $\sqrt{x+\sqrt{y}}$ $\end{equation}$	$\sqrt{x + \sqrt{y}}$
3	higher order roots	$\sqrt[mn]{x+y} \quad \sqrt[3]{2}$	$\sqrt[mn]{x+y} \quad \sqrt[3]{2}$
4	fractions	$\frac{a+b}{x+\log\frac{Y}{Z}}$	$\frac{a+b}{x+\log\frac{Y}{Z}}$
5	continued fraction	$1+\cfrac{2}{3+\cfrac{4}{5+\cfrac{6}{7+\dotsb}}}= \cfrac{1}{\sqrt{e}-1}$	$1+\cfrac{2}{3+\cfrac{4}{5+\cfrac{6}{7+\dotsb}}}= \cfrac{1}{\sqrt{e}-1}$
6	prime	$y''+y'+y=u$	$y''+y'+y=u$
7	sum	$\sum_{i=1}^{+\infty}$	$\sum_{i=1}^{+\infty}$
8	Integral	$\int_{x_0}^{x_1}$	$\int_{x_0}^{x_1}$
9	piecewise functions/cases	$a_k = \begin{cases} k & \text{for } k \leq n/2 \\ n & \text{for } k = n/2 \\ k-1 & \text{otherwise} \end{cases}$	$a_k = \begin{cases} k & \text{for } k \leq n/2 \\ n & \text{for } k = n/2 \\ k-1 & \text{otherwise} \end{cases}$

شکل ۲۵: مثال‌هایی از فرمول‌نویسی در L^AT_EX



شکل ۲۶: ارجاع دادن به رابطه‌ها

۳.۴ نحوه ارجاع دادن به رابطه‌ها

برای ارجاع دادن به معادله‌های ریاضی ابتدا باید label تعریف شود.

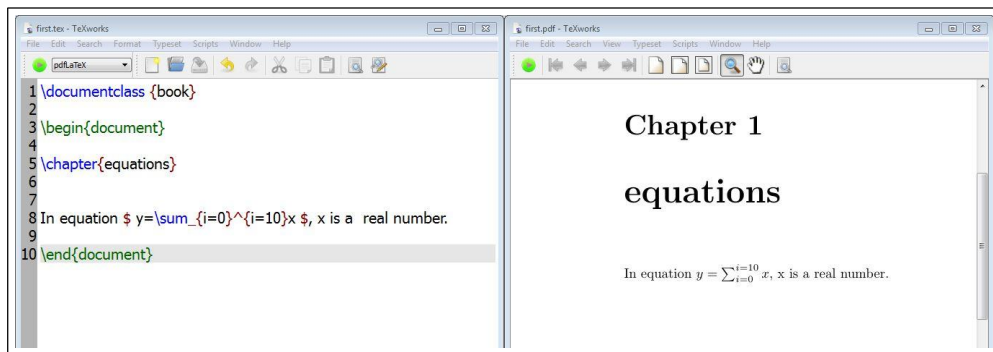
```
\begin{equation}
your equation
\label{LABEL_NAME}
\end{equation}
```

سپس کافی است در هر کجای متن با استفاده از دستور زیر

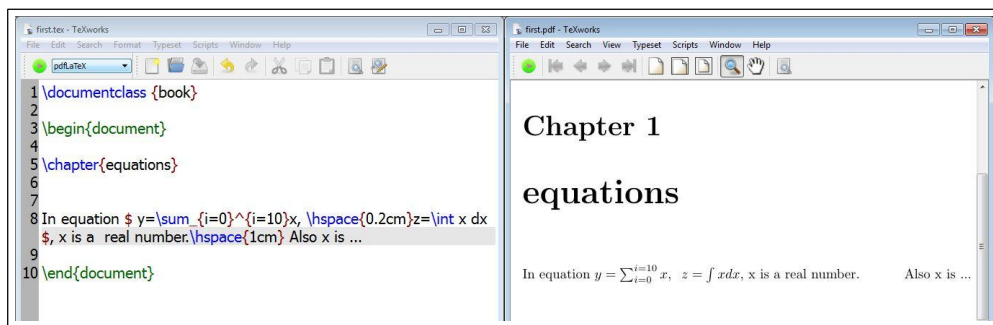
```
\ref{LABEL_NAME}
```

به شماره معادله بالا که به صورت اتوماتیک توسط \LaTeX اختصاص می‌یابد، دسترسی پیدا کنید. نکته مهم آن است که برای اجرای صحیح این دستور نیز باید دو بار کامپایل انجام گیرد (شکل ۲۶).

نکته ۱۱ برای نوشتن یک معادله ریاضی در داخل یک متن (*inline*) از نماد $\$$ استفاده می‌کنیم (شکل ۲۷).



شکل ۲۷: نوشتن رابطه به صورت inline



شکل ۲۸: اعمال فاصله افقی با استفاده از دستور `hspace`

نکته ۱۲ برای اعمال فاصله افقی در داخل یک خط به صورت دلخواه از دستور زیر استفاده می‌کنیم.

`\hspace{length}`

این دستور در داخل معادله نویسی *inline* نیز قابل استفاده است (شکل ۲۸).

۵ درج تصویر

۱.۵ دستورات اولیه

برای درج تصویر در $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ مراحل زیر را انجام دهید.

۱. نیاز به بسته `graphicx` داریم. `\usepackage{graphicx}`

۲. بهتر است کلیه تصاویر مورد نیاز در یک پوشه و در کنار فایل `.tex` قرار گیرند. این امر موجب راحتی در آدرس دهی به

تصاویر می‌شود. فرمت کلی دستورات به صورت `\includegraphics[option]{filename}` می‌باشد. اگر از این دستور

به همین ترتیب استفاده شود، شکل به صورت `Inline` چاپ می‌شود.

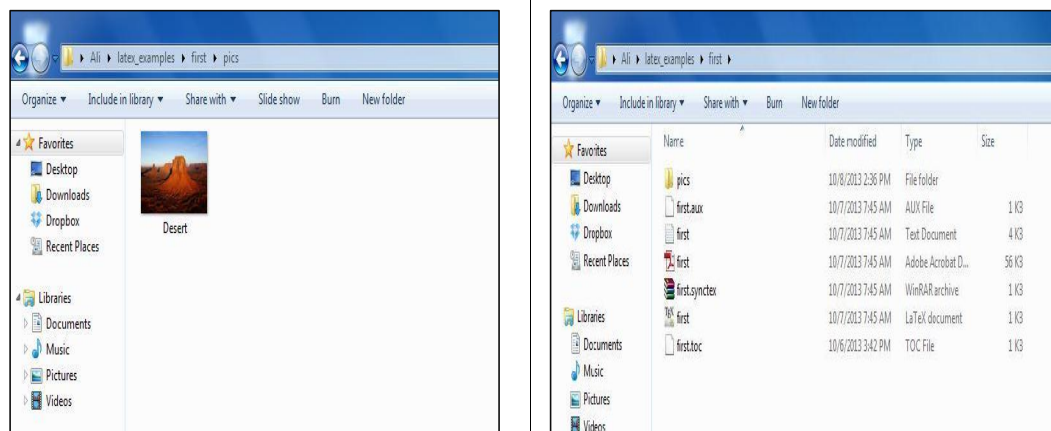
۳. یکی از `option` های قابل استفاده، عرض و ارتفاع شکل می‌باشد.

`\includegraphics[width = ۴cm, height = ۳cm]{filename}`

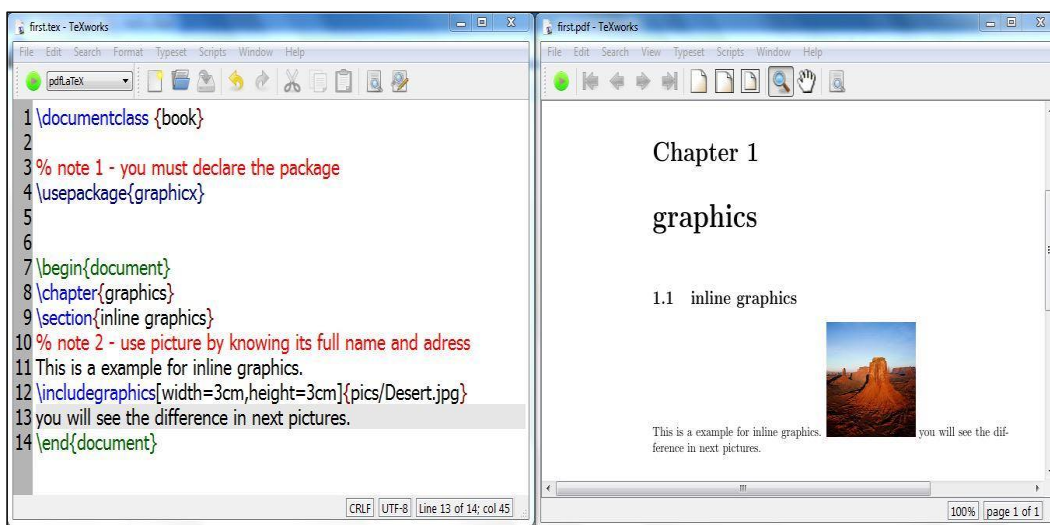
قسمت `option` اختیاری می‌باشد.

در شکل ۲۹، یک مثال کامل را می‌بینید.

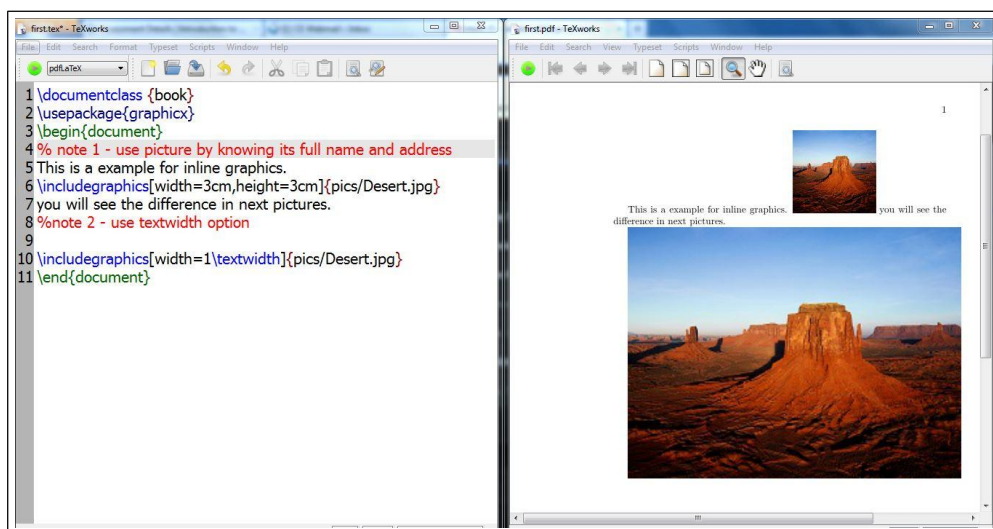
مرحله یک - ایجاد یک پوشه در محل فایل اصلی و قرار دادن تصاویر در آن



مرحله دو - ارجاع به عکس و استفاده از دستورات مربوطه

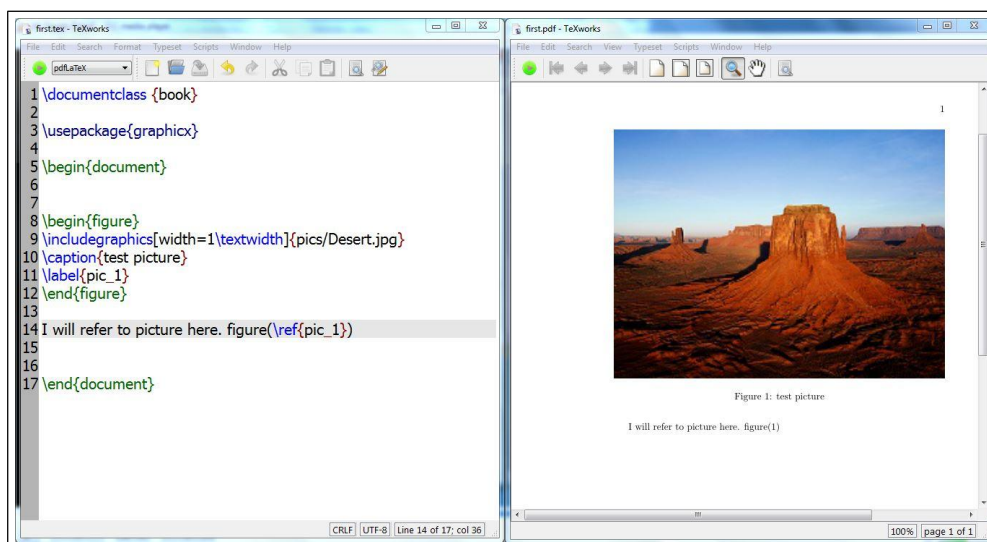


شکل ۲۹: درج تصویر در \LaTeX



شکل ۳۰: تعیین اندازه تصویر نسبت به عرض صفحه

نکته ۱۳ برای معین کردن اندازه یک تصویر روش‌های نسبی نیز وجود دارد. برای مثال می‌توانید عرض یک تصویر را به اندازه عرض صفحه معین کنید (شکل ۳۰). $\backslash includegraphics[width = 1\textwidth]{filename}$



شکل ۳۱: caption گذاری و ارجاع دادن به تصویر

۲.۵ ارجاع دادن به تصاویر و caption گذاری

برای reference دادن به تصاویر باید از فرمان‌های زیر استفاده کنیم و به نکات زیر توجه نماییم.

۱. تصویر را در داخل محیط زیر تعریف کنید.

```
\begin{figure}
\includegraphics[options]{name}
\end{figure}
```

۲. برای Caption گذاری تصاویر، در محیط بالا از دستور مقابل استفاده می‌کنیم. $\backslash caption\{yourCaption\}$

۳. برای ارجاع دادن ابتدا باید label مشخص کنیم و سپس در داخل متن به آن ارجاع نماییم.

$\backslash label\{yourLabel\}$

۴. برای ارجاع دادن در داخل متن از دستور زیر استفاده می‌کنیم. $\backslash ref\{definedLabel\}$

۵. برای نمایش صحیح نتایج باید حتماً دو بار کامپایل کنیم.

در شکل ۳۱ می‌توانید نحوه caption گذاری و ارجاع دادن به یک تصویر را مشاهده نمایید.

نکته ۱۴ محیط figure از محیط‌های شناور است و خود $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ در مورد محل قرارگیری آن در صفحه تصمیم می‌گیرد. به عبارت دیگر بسته به اندازه تصویر و caption تصویر خود $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ تصمیم می‌گیرد که بهترین مکان برای قرارگیری تصویر در صفحه جاری کجاست.

۶ درج جدول

۱.۶ دستورات اولیه

برای تعریف یک جدول ساده به مراحل و نکات زیر توجه کنید.

۱. جدول در داخل محیط tabular ساخته می‌شود.

```
\begin{tabular}
```

```
\end{tabular}
```

۲. با استفاده از آرگومان دوم دستور tabular ستون‌ها و جهت قرار گیری آنها را مشخص می‌کنیم. $\begin{tabular}{crl}$ این دستور به آن معناست که ما یک جدول با سه ستون داریم که در آن ستون اول به صورت center، ستون دوم به صورت right و ستون سوم به صورت left مرتب می‌شوند.

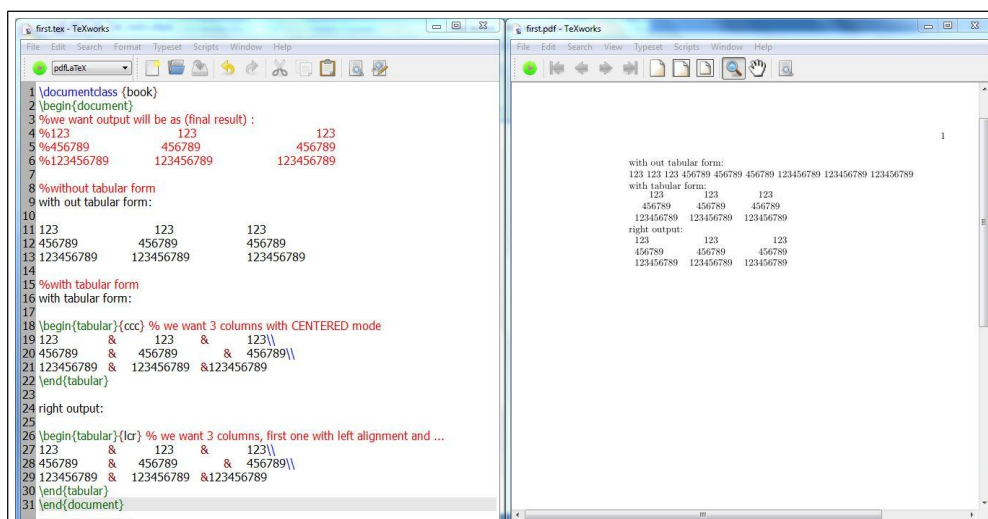
۳. در داخل متن، برای جدا کردن ستون‌ها (کدام قسمت متن در کدام ستون قرار گیرد) از کاراکتر & استفاده می‌شود.

۴. برای جدا کردن سطرها نیز کافی است از \backslashbackslash استفاده کنیم. لازم به ذکر است که نباید برای سطر آخر از این کاراکتر استفاده کنیم.

۵. پهنای هر ستون متناسب با پهنای بزرگترین سطر آن ستون است.

۶. لازم به ذکر است که اگر جدول را در محیط tabular تعریف کنیم، صرفاً به صورت یک کاراکتر با جدول برخورد می‌شود و به صورت inline نمایش داده می‌شود.

در شکل ۳۲ می‌توانید نتیجه دستورات بالا را مشاهده کنید.



شکل ۳۲: ساخت جدول اولیه

۲.۶ قالب‌دهی به جدول و تعریف آن به صورت شناور

برای رسم خطوط جدا کننده در بین ستون‌ها یا سطرها از فرمان‌ها و نکات زیر استفاده می‌کنیم.

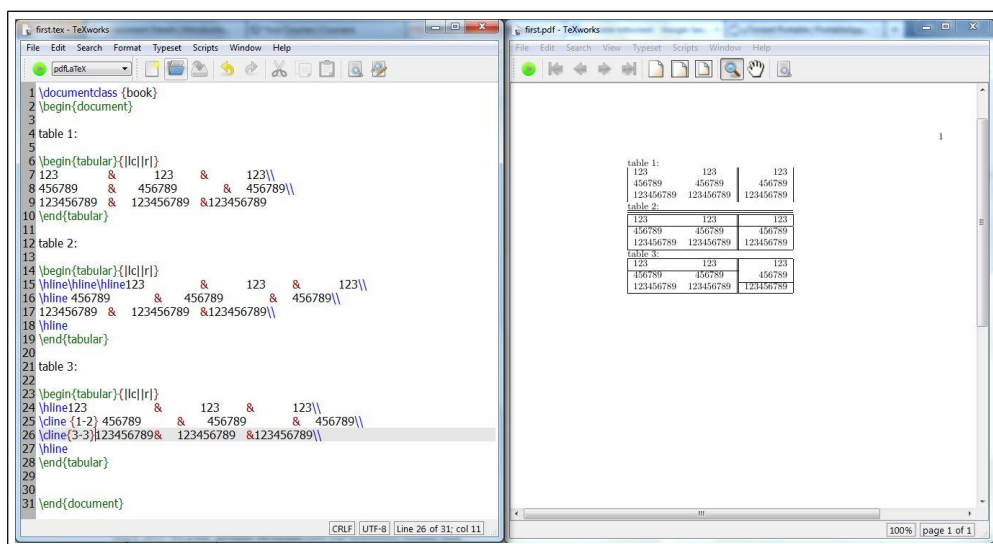
- اگر به هنگام تعریف محیط `tabular` در آرگومان دوم بین ستون‌ها از کاراکتر `|` (کاراکتر پایپ) استفاده کنید، به همان ترتیب و به همان تعداد بین ستون‌ها خط ایجاد می‌شود.
- برای تعریف خطوط بین سطرها باید در مکان مناسب از فرمان `\hline` استفاده کنید. به تعداد دلخواه می‌توانید از این فرمان استفاده کنید.
- اگر می‌خواهید در پایان جدول نیز خط افقی رسم کنید، برخلاف گفته قبلی این بار باید در انتهای سطر آخر نیز از کاراکترهای `\\` استفاده کنید.
- اگر می‌خواهید به جای رسم یک خط کامل بین سطرها، از خطوط ناقص استفاده کنید، باید از فرمان زیر استفاده کنید.

`\cline{column#1 - column#N}`

در شکل ۳۳ می‌توانید نتیجه دستورات بالا را مشاهده کنید.

۳.۶ تعریف جدول به صورت شناور و نحوه ارجاع دادن به آن

برای تعریف جدول به صورت شناور و همچنین قابلیت ارجاع‌دهی و `caption` گذاری، باید به جدول به عنوان یک موجودیت مستقل نگاه کنید. برای اینکار باید آنرا در داخل یک محیط دیگر بنام محیط `table` تعریف کنید.



شکل ۳۳: قالب‌دهی به جدول

```
\begin{table}

\begin{tabular}

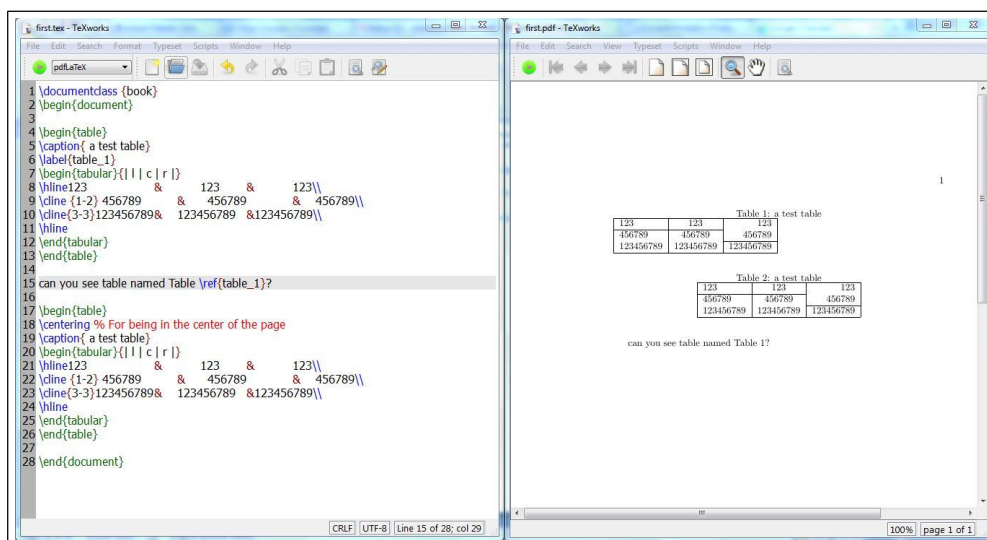
\end{tabular}

\end{table}
```

نکته ۱۵ محیط *table* نیز همانند محیط *figure* از محیط‌های شناور است و خود *LaTeX* در مورد محل قرارگیری آن در صفحه تصمیم می‌گیرد.

نکته ۱۶ با استفاده از دستور `\caption{name}` می‌توان به جدول *caption* اختصاص داد.

نکته ۱۷ برای ارجاع دادن به یک جدول همانند تصویر، ابتدا باید یک *label* به آن اختصاص دهید و در ادامه با استفاده از آن *label* به آن ارجاع دهید.



شکل ۳۴: تعریف جدول به صورت شناور

نکته ۱۸ برای وسط چین شدن نسبت به متن در محیط *table* از دستور *centering* باید استفاده کنیم.

در شکل ۳۴ می‌توانید نتیجه دستورات بالا را مشاهده کنید.

۷ آشنایی با بسته *geometry*

همانطور که قبلاً گفته شده است، همه تنظیمات مربوط به سند، همانند اندازه حاشیه‌های صفحه، شماره‌گذاری‌ها، نحوه قرارگیری قسمت‌های مختلف و موارد بسیار دیگر توسط *LaTeX* و به صورت اتوماتیک انجام می‌گیرد. در واقع با مشخص کردن قالب کلی متن در قسمت `\documentclass{}` این تنظیمات انتخاب می‌شود. در صورتی که بخواهید برخی از این تنظیمات را تغییر دهید، از بسته به نام *geometry* می‌توان استفاده کرد. در شکل ۳۵ برخی از تنظیمات قابل اعمال توسط این بسته آمده است.

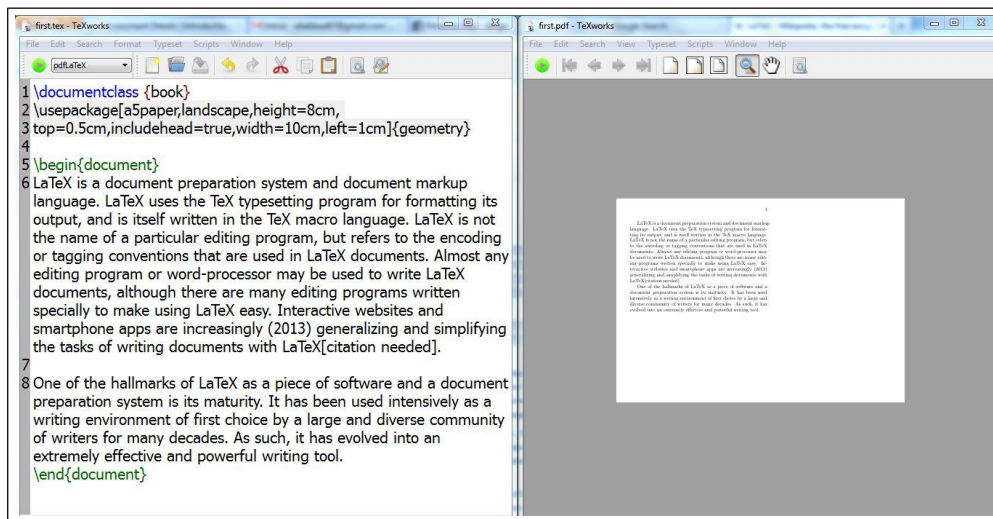
نکته ۱۹ در حالت پیش‌فرض اگر به صفحه حاشیه بدهید، این فاصله از انتهای متن محاسبه خواهد شد و مواردی چون شماره صفحه در داخل حاشیه قرار می‌گیرند. اگر می‌خواهید برای محاسبه حاشیه، از انتهای آخرین مورد استفاده شود (برای مثال این بار از خود شماره صفحه تا انتهای صفحه فاصله مورد نظر وجود داشته باشد) باید در داخل *option* های بسته *geometry* از دستورات مقابل استفاده شود. *Includehead = true* و *Includefoot = true*

عنوان	توضیح - مثال
تغییر اندازه برگ- برای مثال a4, a5 و...	<code>\usepackage[a5paper]{geometry}</code>
تغییر صفحه از portrait به landscape	<code>\usepackage[landscape]{geometry}</code>
تغییر پهنای صفحه	<code>\usepackage[width=15cm]{geometry}</code>
تغییر ارتفاع صفحه	<code>\usepackage[height=15cm]{geometry}</code>
اعمال فاصله و قرار گیری متن در نقاط مختلف صفحه	<code>\usepackage[top=15cm]{geometry}</code> <code>\usepackage[left=15cm]{geometry}</code>

شکل ۳۵: برخی از تنظیمات قابل اعمال توسط بسته geometry

یک مثال کلی: خروجی تنظیمات زیر به چه صورت خواهد شد؟ (پاسخ در شکل ۳۶ آمده است.)

```
\usepackage[a5paper,landscape,height=8cm,top=2.5cm,includehead=true,width=15cm]{geometry}
```



شکل ۳۶: تنظیمات اعمال شده توسط بسته geometry

۸ فارسی نویسی در L^AT_EX

توجه: این بخش خارج از سرفصل دوره می باشد و صرفاً جهت شروع کار آورده شده است.

برای نوشتن در latex به زبان فارسی به نکات زیر توجه کنید.

- در حالت کلی، کلیه دستورات گفته شده برای زبان انگلیسی برای فارسی نیز قابل استفاده است.
- باید بسته مربوط به زبان فارسی را حتما اضافه نمایید. (دقت کنید این بسته حتما باید آخرین بسته اضافه شده باشد.)

`\usepackage{xepersian}`

- باید فونت های مربوطه را اضافه نمایید. لازم به ذکر است کلیه فونت های قابل استفاده در ویندوز، قابلیت استفاده در L^AT_EX را دارند. ولی در برخی موارد دارای نقص هایی می باشند. به همین دلیل بهتر است فونت های سری XB را

دانلود کرده و پس از نصب استفاده نمایید. `\settextfont{XBZar}`

- حتما باید از compiler دیگری بنام Xe_latex برای پردازش متن استفاده نمایید.
- اگر می خواهید کلیه اعداد حتی اعداد استفاده شده در عبارات ریاضی نیز به صورت فارسی نمایش داده شوند، باید از

دستور زیر استفاده نمایید. `\setdigitfont{XBZar}`

پایان
