

جلسه اول مجازی
طراحی وب
کاردانی نرم افزار ۲

مدرس:

فاطمه دهقانی فیروزآبادی

کاربردهای نرم افزار دریم ویور + معرفی معایب آن

دریم ویور این امکان را به شما می دهد که بدون کدنویسی، یک وبسایت را ایجاد کنید. این برنامه برای افراد مبتدی و حرفه ای در زمینه طراحی وب بسیار مفید است. با این نرم افزار می توانید سایت های زیادی را ایجاد کنید، اما این برنامه محدودیت هایی نیز دارد، برای اینکه بتوانید وبسایت ایده آلی طراحی کنید نیاز دارید که روی کدنویسی HTML , CSS مسلط باشید. در این نوشته تعدادی از نقاط ضعف و قوت دریم ویور را بررسی می کنیم.

*ساده سازی طراحی وب

یکی از ویژگی های کاربردی دریم ویور، ساده سازی طراحی وب است. می توانید با استفاده از خصوصیت گرفتن و کشیدن در زمانی کوتاه سایت خود را طراحی کنید.

*زمینه هایی که دریم ویور می تواند در آنها مفید باشد

این برنامه برای افرادی که کدنویسی می کنند مزایای زیادی به همراه دارد. باعث برجسته شدن کد می شود و به شما کمک می کند تا در زمان کوتاهی کد را بخوانید و خطاهای احتمالی را بررسی کنید. از دیگر مزایای این برنامه برای کدنویسان می توان به وجود ابزار مدیریت فایل و اعتبارسنجی کد اشاره کرد.

همچنین، دریم ویور پلتفرمی بسیار موثر برای افرادی است که می خواهند سایت های ساده را راه اندازی کنند. تمپلت های موجود در این برنامه به شما اجازه می دهد تا طرح های زیادی را در سایت خود به کار ببرید. در آخرین نسخه این برنامه، ویژگی **Live view** به شما امکان می دهد که ظاهر سایت خود را بصورت آنلاین مشاهده کنید.

یک مزیت دیگر دریم ویور وجود cross-platform است. این پلتفرم به شما این امکان را می‌دهد که بدون دانستن کدنویسی برای وبسایت موبایلی، وبسایت‌های خود را برای اجرا روی تمام دستگاه‌ها بچرخانید. وجود خاصیت Fluid Grid به شما امکان می‌دهد تا وبسایت بصورت تمام صفحه در هر دستگاهی نشان داده شود.

معایب دریم ویور

اگر به ویژگی‌های دریم‌ویور وابسته شوید، شما به یک طراح سایت تنبل تبدیل خواهید شد. استفاده زیاد از دریم‌ویور می‌تواند دانسته‌های شما را محدود به ابزار موجود در این برنامه کند.

از دیگر معایب دریم ویور می‌توان به هزینه‌بر بودن آن اشاره کرد. سیستم‌های مدیریت محتوی بسیاری مثل وورس وجود دارد که می‌توانید بصورت رایگان از آنها استفاده کنید.

گاهی مواقع می‌توانید با استفاده از پلتفرم CMS پروژه بهتری نسبت به دریم‌ویور ارائه دهید. برای اینکه بتوانید از حداکثر توانایی‌های دریم ویور استفاده کنید، به یک (Cloud Subscription هزینه مجازی) نیاز خواهید داشت، که می‌تواند در طول زمان هزینه سنگینی را به شما تحمیل کند.

*کدهای زیاد

یکی از معایب دریم‌ویور این است که کدهای اضافی زیادی را به وبسایت شما اضافه می‌کند و این کار باعث کندی موتور جستجو در زمان افزودن یک لایه جدید به سایت شود. اگر می‌خواهید روی وبسایت خود تسلط داشته باشید و کارایی آن حداکثر باشد، بهتر است کد مخصوص خودتان را بنویسید.

دریم‌ویور به شما امکان کدنویسی در HTML را می‌دهد. با این وجود بسیاری از افراد حرفه‌ای بخاطر کدنویسی‌های ضعیف و کدهای CSS تاریخ مصرف گذشته این برنامه را مورد نکوهش قرار می‌دهند.

- آشنایی با ساختار HTML

- همانطور که گفتیم برای ورود به مرحله طراحی وب سایت در مرحله صفر نیاز به یادگیری دو زبان html و css است . بعد از آن می توانیم به سراغ یادگیری زبان php، جاوا اسکریپت و ... برویم . اما برای ورود به طراحی سایت گام اول html و css است. در این جلسه به مباحث پایه یعنی html چیست ؟ و ساختار html چگونه است؟ می پردازیم.

- HTML چیست؟

- مخفف Hyper Text Markup Language، یک زبان نشانه گذاری برای توزیع صفحات وب می باشد . دقت داشته باشید که HTML زبان برنامه نویسی نیست زبانی است که بر اساس یک سری برچسب یا تگ شناخته می شود. هر تگ html بخش های مختلفی از صفحه مثل عنوان، پاراگراف، جدول و غیره را ایجاد می کند. خب برای اینکه راحت تر با این مفهوم آشنا شویم وارد محیط کدنویسی می شویم.

- نگران یادگیری این مفاهیم نباشید فقط گام به گام با ما همراه باشید و کد بزنیید تا به صورت عملی با مفاهیم آشنا شوید.

ساختار HTML

در ابتدا برای کدنویسی html از ادیتور dream weaver استفاده می کنیم

نوشتن کد در سند html

گام ۱) ادیتور را باز می کنیم و کدهایی که در ادامه نوشتیم را در آن تایپ می کنیم (لطفا فقط این متن ها را نخونین ، دست به کار بشید و با ما تایپ کنید)

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title>  
      اینجا عنوان سایت است  
    </title>  
  </head>  
  <body>  
    اینجا همه محتوای سایت قرار می گیرد  
  </body>  
</html>
```

تفسیر کدها

کدهای بالا یک نمونه خیلی ساده از سند html می باشد.

تفسیر کدها

کدهای بالا یک نمونه خیلی ساده از سند html می باشد.

به هر کدام از `<tag>` ها تگ می گوئیم . تگ باز است و تگ بسته. در تگ باز و بسته یک مقدار نوشته می شود فقط تگ بسته یک اسلش اضافه تر دارد. هر تگی که باز می شود باید حتما بسته شود.(البته جلوتر بریم خواهیم دید که تگ هایی هستند که بسته نمیشن) در بین تگ ها می توانیم محتوا یا تگ های دیگری را قرار دهیم. (ان شاء الله تا آخرین جلسه شما رو با مهمترین و پرکاربردترین تگ های html آشنا خواهیم کرد).

کلا ساختار اینجوری هست که تمام کدهای سایت داخل تگ `<html></html>` قرار می گیرند(بیرون این تگ هر چی بنویسیم به حساب نمی آید) و دو تا تگ اصلی که داخل تگ `<html>` وجود دارد :

- `<head></head>`: اطلاعات کلی از سایت ما و تمام المان های آن را برای مرورگر فراهم می کند. مثل تگ عنوان، استایل سایت، متا، جاوا اسکریپت و ... (در ادامه باهاشون بیشتر آشنا میشیم)
- `<body></body>`: در تگ body محتوای سایت را می نویسیم . در واقع هر چیزی که در یک سایت می بیند در قسمت body قرار می گیرد.

تگ `<title></title>`: عنوان سایت را نشان می دهد و در داخل تگ `<head>` نوشته می شود

توجه: همیشه خط اول `<!DOCTYPE html>` می نویسیم . این خط کد به مرورگر می فهماند که این فایل از نوع html هست.نوشتن این خط در ابتدای فایل قبل از تگ html ضروریه!

نکته: اگر در هر سایتی راست کلیک کنیم و **view page source** رو بزنیم، تمام کدهای html اون سایت رو می بینم . الان همین کارو روی سایت دایژار انجام بدین و اولین خط رو بخونین ! دیدین گفتم DOCTYPE ضروریه) حالا فایده این خط کد چیه که ضروریه؟

- مهم در seo(سئو یا بهینه سازی موتور جستجو)
- استاندارد شدن صفحه و نمایش یکسان در تمام مرورگرها

The screenshot shows a web browser window titled "index.html". The source code is displayed with the following annotations:

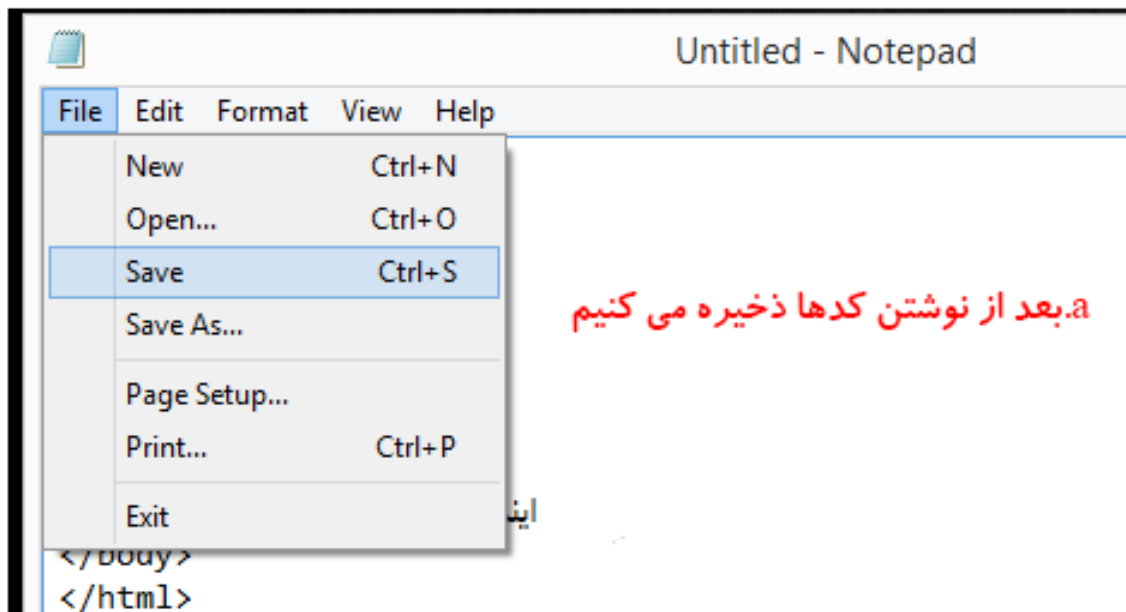
- `<!DOCTYPE html>` is annotated with "اینو حتما بنویسید نشاندهنده این است که فایلتون از نوع html است".
- `<html>` is annotated with "تگ باز html".
- `<head>` is annotated with "تگ باز head".
- `<title>` is annotated with "عنوان سایت" and "داخل تگ head".
- `</title>` is annotated with "اینجا عنوان سایت است".
- `</head>` is annotated with "تگ بسته head".
- `<body>` is annotated with "محتوای سایت" and "داخل تگ body".
- `</body>` is annotated with "اینجا همه محتوای سایت قرار می گیرد".
- `</html>` is annotated with "تگ بسته html".

On the right side of the browser window, a diagram illustrates the structure of the HTML document. It consists of a yellow box labeled "header" at the top, and a larger yellow box labeled "body" below it, representing the main content area.

گام 2) فایل را ذخیره می کنیم ، اینکه کجا و با چه نامی هم مهم است. در یک پوشه به نام مثلا project (که قرار است همه فایل های پروژمونو در این پوشه ذخیره شوند و اصطلاحا root پروژه هست) ذخیره می کنیم. نام این فایل را **index** و پسوند را **html** انتخاب می کنیم. در واقع همیشه **index.html** . موقع ذخیره کردن **Encoding** را روی **UTF-8** قرار می دهیم تا مرورگر سایت های فارسی رو هم پشتیبانی کند.

نکته : چرا باید نام را index باید بنویسیم، چرا؟ چون هر سایتی که شما در دنیای اینترنت می بینید اولین صفحه آن با نام index شناخته می شود. (در واقع در پشت صحنه اجرای سایت ها یک مترجم وجود دارد که به صورت پیش فرض فایل با نام index را صفحه اول سایت می داند و به دنبال فایل با این نام می گردد تا سایت را اجرا کند)

نکته: پسوند رو html انتخاب کنید. توجه داشته باشید که به صورت index.html.txt ذخیره شود نه index.html.txt ! نکته خنده داریه ولی ممکنه پیش بیاد!



آشنایی با تگ p ، تگ h ، تگ br ، تگ pre

تگ p

تگ `<p></p>` به شدت پرکاربرد هست و برای نوشتن متون استفاده می شود. در واقع p نشاندهنده یک پاراگراف هست.

نوشتن کد در سند html

داخل تگ body متنی که نوشته بودیم رو پاک می کنیم تا دو تا پاراگراف اضافه کنیم. پس دو تا پاراگراف باید کجا نوشته شود؟ گفتیم هرچی که بخواهیم کاربر در سایت ببینه باید بین تگ body قرار بگیره.

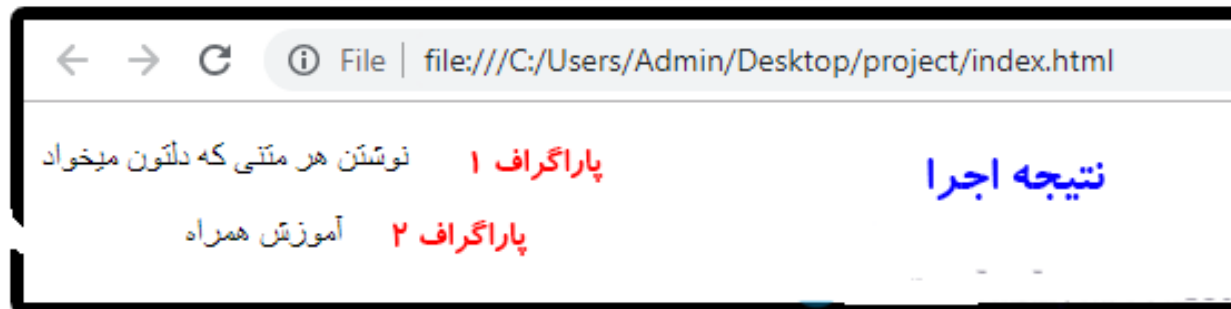
```
<!DOCTYPE html>
<html>
  <head>
    <title>
      اینجا عنوان سایت است
    </title>
  </head>
  <body>
    <p>نوشتن هر متنی که دلتون میخواد</p>
    <p>آموزش همراه</p>
  </body>
</html>
```

سند html را ذخیره می کنیم و خروجی رو می بینیم.

```
<!DOCTYPE html>
<html>
<head>
<title>
اینجا عنوان سایت است
</title>
</head>
<body>
<p>نوشتن هر متنی که دلتون میخواد </p>
<p>آموزش همراه </p>
</body>
</html>
```

کدهای HTML

دو تا پاراگراف



نکته: جمله ای که در پاراگراف دوم نوشتیم به ادامه جمله اول اضافه نشده است. در واقع پاراگراف بندی شده است.

نکته: مرورگرها در هنگام نمایش صفحه، تمام فضاها را خالی یا خطوط اضافی را حذف می کنند. (با این اوصاف زدن اینتر در خروجی سند html تاثیری ندارد، پس اگر بخواهیم یک خط جا بندازیم چیکار کنیم؟)

تگ br

تگ `
` یک شکست خط را ایجاد می‌کند. اگر بخواهیم بدون ایجاد یک پاراگراف جدید یک خط جا بیندازیم و به خط بعد برویم از تگ `br` استفاده می‌کنیم.

نکته: این هم از اون تگ‌هایی که بسته ندارد.

نکته: البته اگر بخواهیم این شکست در متن به صورت یک خط نمایش داده شود می‌توانیم از تگ `<hr>` (که تگ بسته هم نداره)، استفاده کنیم.

(این مورد رو خودتون تست کنید هر جایی که دوست دارید اینتر بزنید تا به خط برین، یه بار تگ `
` بنویسین و یه بار هم تگ `<hr>`. نتیجه رو ببینید ؛)

تگ pre

متنی که در تگ `<pre></pre>` قرار می‌گیرد در یک عرض و فونت ثابت (معمولا Courier) و با حفظ تمام فضاها و خالی و خطوط جا افتاده نمایش داده می‌شود.

```
index.html
File Edit Format View Help
<!DOCTYPE html>
<html>
<head>
<title>
اینجا عنوان سایت است
</title>
</head>
<body>
<pre>
نوشتن هر متنی که دلتون میخواد
فاصله و خطوط جا افتاده اینجا حفظ می شوند
</pre>
<p>
آموزش همراه
</p>
</body>
</html>
```

کدهای html

به تفاوت این دو متن در خروجی توجه کنید



نتیجه اجرا

فضاهای خالی حفظ شده

تگ h یا عنوان (heading)

حالا اگر بخواهیم در متنی که می نویسیم از یک یا چند تا عنوان استفاده کنیم باید از تگ `<h></h>` استفاده شود.

تگ های `<h></h>` از 1 شروع می شوند تا 6. یعنی `<h1>`، `<h2>`، `<h3>`، `<h4>`، `<h5>`، `<h6>`. فرق این تگ ها در سایز فونت می باشد. هر چی شماره بزرگتر می شود سایز کوچکتر می شود. خب کدش رو بنویسیم تا بیشتر متوجه بشیم.

نوشتن کد در سند html

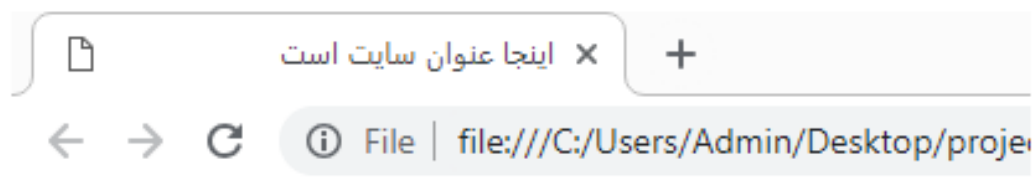
در بین تگ `<body>` ، عناوین را اضافه می کنیم:

```
<!DOCTYPE html>
<html>
  <head>
    <title>
      اینجا عنوان سایت است
    </title>
  </head>
  <body>

    <h1> عنوان </h1>
    <h2> عنوان </h2>
    <h3> عنوان </h3>
    <h4> عنوان </h4>
    <h5> عنوان </h5>
    <h6> عنوان </h6>
    <p>نوشتن هر متنی که دلتون میخواد</p>
    <p>آموزش همراه</p>

  </body>
</html>
```

کد را ذخیره می کنیم و فایل index.html را اجرا می کنیم تا نتیجه را ببینیم. همانطور که در تصویر هم شخص کردیم عناوین از h1 تا h6 کوچکتر می شوند.



h1 عنوان

h2 عنوان

h3 عنوان

h4 عنوان

h5 عنوان

h6 عنوان

نوشتن هر متنی که دل‌تون می‌خواهد

آموزش همراه

به صورت خلاصه با تگ های زیر آشنا شدیم:

تگ	
ایجاد پاراگراف	<p></p>
ایجاد عنوان	<h></h>
ایجاد پاراگراف با عرض و فونت ثابت (معمولا Courier) و حفظ فضاهاى خالى	<pre></pre>
ایجاد یک شکست خط	
ایجاد شکست در متن به صورت خط	<hr>

آشنایی با CSS و استایل دهی به html

CSS چیست؟

مخفف Cascading Style Sheets به معنای برگه های سبک آبشار، یک زبان استایل نویسی است ، که وظیفه اش استایل دادن به کدهای html می باشد. پس CSS مشخص می کند عناصر html با چه ظاهری روی سایت نمایش داده شوند.

این دو زبان در کنار هم مفهوم دارند و بهم معنا می دهند. به عنوان مثال: اگر فرض کنید html نقش اسکلت یک ساختمان را داشته باشد CSS نقش نمای ساختمان را بازی می کند. یعنی زیباسازی صفحات وب توسط زبان استایل دهی CSS انجام می شود.

سه روش برای استایل دهی به html وجود دارد:

1. **روش inline** : برای تگ body و تمام تگ های داخلی آن از صفت style استفاده کنیم و به صورت انحصاری برای هر تگ یک استایل تعریف کنیم.
2. **روش external** : یک فایل جداگانه CSS در پوشه project یا root ایجاد کنیم و با استفاده از تگ link آن را در فایل html خودمان فراخوانی کنیم.
3. **روش internal** : در تگ <head> یک تگ به نام `<style></style>` ایجاد کنیم و تمام کدهای CSS را داخل تگ style بنویسیم (بدون استفاده از فایل جداگانه CSS)

بهترین روش و استانداردترین روش استفاده از روش دوم است یعنی **external** و ما در طی این آموزش از همین روش استفاده می کنیم (برای اینکه گیج نشین، روش های دیگه رو توضیح نمیدیم فقط در آخر آموزش تفاوت دو روش دیگه رو با یک شکل نشون میدیم که نگین گفتین :-).

استایل دهی به html به روش external

گام اول) ایجاد فایل CSS

در پوشه project که جلسه اول ایجاد کردیم و فایل index.html را در آن قرار دادیم ، یک فایل دیگری ایجاد می کنیم با نام style.css .

توجه داشته باشید: به این پوشه project که تمام فایل های اصلی سایت مانند css ، html ، جاوااسکریپت و ... در آن قرار می گیرد root سایت می گویند.

گام دوم) نوشتن کد در سند html

تگ link را به تگ head اضافه می کنیم:

```
<!DOCTYPE html>
<html>
  <head>
    <title>
      اینجا عنوان سایت است
    </title>
  </head>
  <body>
    <p>نوشتن هر متنی که دلتون میخواد</p>
  </body>
</html>
```

گام سوم) برقراری پیوند بین فایل CSS و HTML

فایل css را از طریق تگ **<link>** در فایل html فراخوانی می کنیم.

در سند html، تگ **<link>** رو کجا بنویسیم؟ در داخل تگ **<head></head>**

از الگوی زیر برای لینک دادن فایل css به سند html استفاده می کنیم:

```
<link href="مسیر فایل سی اس اس" rel="stylesheet" type="text/css">
```

همانطور که می بینید تگ link از جمله تگ هایی است که تگ بسته ندارد. در تگ link یک سری چیزایی نوشتیم که به آنها صفت یا attribute می گویند.

نکته: صفت چیست؟

Attribute یا صفت در html اطلاعات کاملتری از یک المان را ارائه می دهد. تمام المان های HTML می توانند یک یا چند خصوصیت داشته باشند. صفت های html همیشه در تگ باز المان مشخص می شود و به صورت جفت نام/مقدار مشخص می شوند. (attribute="value")

تگ <link> سه تا صفت دارد که به صورت زیر می باشد:

- **href** :

این صفت مسیر فایل CSS را مشخص می کند. اگر مستقیماً فایل CSS را داخل روت سایت کنار فایل HTML ایجاد کردیم، کافی است نام فایل CSS را به همراه پسوندش بنویسیم به این شکل: href="style.css". ولی اگر فایل CSS را داخل پوشه ای مثلاً CSS قرار دادیم باید به این شکل بنویسیم: href="css/style.css".

- **rel** :

مخفف relation می باشد که ارتباط بین صفحه مبدا با صفحه مقصد رو مشخص می کند. مقدارش را stylesheet دادیم یعنی ارتباط بین مرورگر با فایلی ما، از نوع برگه استایل هست. دقت کنید درست بنویسید در غیر این صورت ارتباط با CSS برقرار نمی شود.

- **type** :

صفت بعدی type می باشد که مشخص می کند فایل CSS از نوع text بوده است که تبدیل به CSS شده است (اگر ننوشتیم هم مهم نیست)

خب با توجه به توصیفات بالا تگ link را به سند html- بین تگ <head></head> اضافه می کنیم:

```
<!DOCTYPE html>  
<html>  
  <head>  
    <link href="style.css" rel="stylesheet" type="text/css">  
    <title>  
      اینجا عنوان سایت است  
    </title>  
  </head>  
  <body>  
    <p>نوشتن هر متنی که دلتون میخواد </p>  
  </body>  
</html>
```

الان می تونیم بگیم پیوند html و css مبارک lol: . فایل html را ذخیره می کنیم و می بندیم . حالا میرویم سراغ کدهای CSS, تا به کم تو CSS کد بزنیم.

نحوه استایل دهی در CSS

به صورت کلی از الگوی زیر برای کدزندن در CSS استفاده می شود:

```
selector { property:value; property:value; ... }
```

هر قانون استایل در CSS شامل یک selector و یک بلاکی از یک یا چند تعریف (ویژگی و مقدار) می باشد. هر تعریف با "نقطه سمیکالن" از هم جدا می شوند.

در فایل CSS به صورت کلی سه تا **selector** داریم. یعنی به سه روش می توانیم المان یا همون تگ های موجود در فایل html را صدا بزیم تا بهشون استایل بدهیم:

- صدا زدن المان بر حسب نام
- صدا زدن المان بر حسب class (که با نقطه نشان می دهیم)
- صدا زدن المان بر حسب id (که با # نشان می دهیم)

(هر سه روش استفاده می شود که فعلا در این جلسه ما از روش **اول** استفاده می کنیم، یعنی با فراخوانی نام المان به آن استایل می دهیم. در [جلسات بعد](#) با [class](#) و [id](#) آشنا خواهیم شد.)

خب بعد از آشنایی با CSS ، فایل style.css که در گام اول ایجاد کرده بودیم را وارد می کنیم:

باز می کنیم و اولین خط کدهای CSS رو اینجا

```
body{
  background:red;
}
p{
  color:blue;
}
```

توضیح کد CSS:

همانطور که در کدها می بینید از 1 selector استفاده شده است. یعنی نام المان رو صدا زدیم و در داخل {} به آن استایل دادیم. در طول آموزش کم کم با کدهای CSS هم آشنا میشیم! نگران نباشید.

- ویژگی `color`: رنگ متن را مشخص می کند.
- `background-color` یا `background`: هر دو رنگ پس زمینه را مشخص می کنند.

نکته: نقطه سمیکالن آخر هر تعریف مهم است. فراموش نشه!

```
<!DOCTYPE html>
```

فایل html

```
<html>
```

اتصال بین فایل css و html

```
<head>
```

```
<link href="style.css" rel="stylesheet" type="text/css">
```

```
<title>
```

اینجا عنوان سایت است

```
</title>
```

```
</head>
```

```
<body>
```

```
<p>نوشتن هر متنی که دلتون میخواد</p>
```

```
</body>
```

```
</html>
```



style.css - Notepad

File Edit Format View Help

فایل css

```
body{
```

```
background:red;
```

```
}
```

```
p{
```

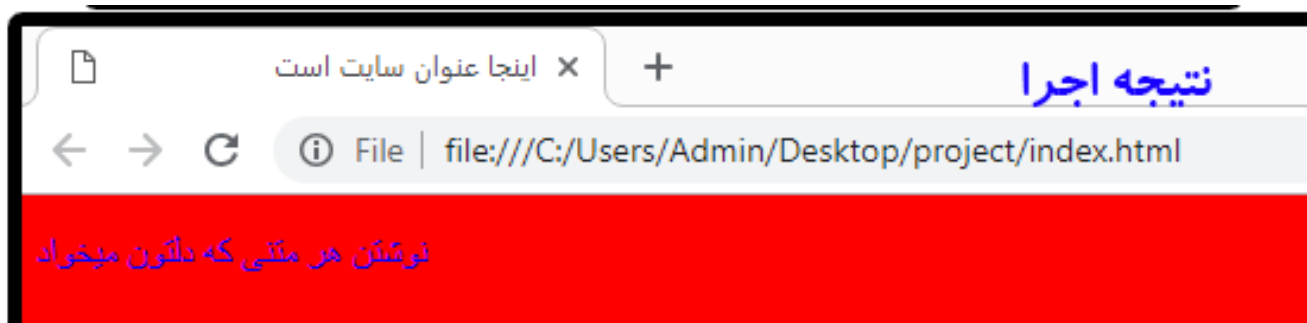
```
color:blue;
```

```
}|
```

کدی که در فایل css

می نویسیم برای تغییر رنگ پس زمینه

تغییر رنگ متن پاراگراف



مقدار رنگ ها در html & css:

حالا که بحث رنگ است این نکته هم بگوییم که به سه شکل می توانیم مقدار رنگ را به امان مورد نظر اعمال کنیم:

1. نام رنگ را بنویسیم (مثلا : red)
2. استفاده از کد هگزا دسیمال با الگوی #RRGGBB که از ترکیب سه رنگ آبی-سبز-قرمز رنگ تولید می شود (هرکدام از 00 شروع میشه تا ff و هر چی مقدار به ff نزدیکتر می شود غلظت رنگ بیشتر می شود، مثلا رنگ #0000ff قرمز هست)
3. به شکل " `rgb(0,0,0)` " مقدار دهیم که ارقام داخل پرانتز به ترتیب از چپ به راست رنگ قرمز، سبز و آبی را مشخص می کند. رنج اعدادی که به هر کدام می دهیم 0 تا 255 می باشد.

توجه داشته باشید روشن_دوم بیشتر استفاده می شود.

تفاوت دو روش inline و internal برای استایل دهی

اون بالا اشاره کردیم که سه روش برای استایل دهی به سند html وجود دارد (internal و inline و external)، که گفتیم روش external از همه بیشتر استفاده می شود. **چرا؟**

جواب: چون در روش inline استایل دهی به صورت انحصاری برای یک تگ خاصی انجام می شود. در روش internal هم، استایل دهی به صورت انحصاری روی یک سند html اعمال می شود ولی اگر فایل css را به صورت external و خارجی تعریف کنیم، می توانیم این استایل را به هر تعداد سند html که می خواهیم اعمال کنیم.

به شکل زیر توجه کنید که نحوه نوشتن کد به صورت inline و internal را نمایش می دهد:

- در روش inline داخل هر المان از صفت style استفاده می کنیم.

<تگ style="property:value";"property:value"; ... ></تگ>

- در روش internal از تگ <style></style> داخل تگ head استفاده می کنیم.

<style>

selector { property:value; property:value; ... }

selector { property:value; property:value; ... }

.

.

.

</style>

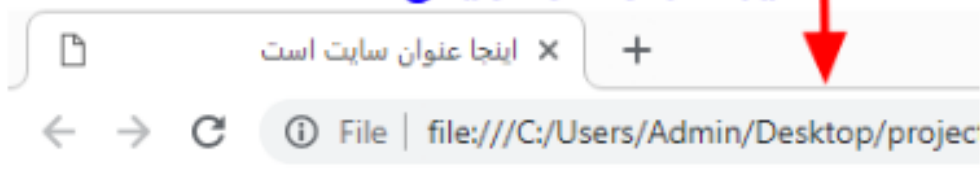
```
<!DOCTYPE html>
<html>
<head>
<title>
اینجا عنوان سایت است
</title>
</head>
<body>
<p style="background:red">نوشتن هر متنی که دلتون میخواد</p>
</body>
</html>
```

روش inline

```
<!DOCTYPE html>
<html>
<head>
<style>
p{
background:red;
}
</style>
<title>
اینجا عنوان سایت است
</title>
</head>
<body>
<p>نوشتن هر متنی که دلتون میخواد</p>
</body>
</html>
```

روش internal

نتیجه اجرا هر دو یکی است



نوشتن هر متنی که دلتون میخواد

به صورت خلاصه با تگ ها و property های زیر آشنا شدیم:

تگ	
برای لینک دادن به CSS با سه صفت (rel.href.type)	<link>
برای استایل دهی (به روش internal)	<style></style>
css در property	
دادن رنگ به متن	color
دادن رنگ به پس زمینه	background

با صفات تگ html و تگ meta آشنا خواهیم شد.

باز هم تاکید می کنم اگر خودتون دست به کار نشین و کد نزنید، خوندن این مطالب هیچ فایده ای نداره!

آشنایی با صفات تگ HTML و تگ META

گفتیم **صفت یا خصوصیت** در تگ ها چی هستند مجدداً به اشاره کوچیکی می کنیم:

- اطلاعات کاملتری از تگ را ارائه می دهند.
- در تگ آغازین قرار می گیرند.
- به صورت جفت نام/مقدار درون تگ نوشته می شوند: "مقدار ویژگی" = نام ویژگی.

خب، روز اول یک تگ مهم به نام <html> رو معرفی کردیم حالا می خواهیم دو تا صفت مهم این تگ رو بگیریم که در تمام پروژه هامون ازش استفاده می کنیم.

```
<html lang="fa" dir="rtl">
```

- **Lang**: مخفف language که زبان سایت را نشان می دهد و مقدار fa را به آن می دهیم چون زبان ما فارسی هست (نکته: می توانیم بنویسیم fa-IR که fa زبان سایت و IR گویش را مشخص می کند)
- **Dir**: مخفف direction، مشخص کننده ساختار سایت است. که دو مقدار می گیرد ، rtl (از راست به چپ) و ltr (از چپ به راست) اگر تا الان توجه کرده باشید محتوای سایت در سمت چپ سایت قرار می گرفتند چون این صفت به صورت پیش فرض مقدار ltr را دارد.

تگ های داخلی تگ <head>

در جلسه اول با تگ مهمی به نام <head></head> آشنا شدیم و گفتیم که این تگ و تگ های داخلی آن اطلاعاتی درباره سند می باشد که نمایش داده نمی شوند و مربوط به موتورهای جستجو می شود. یکی از تگ های مهمی که در داخل تگ head قرار می گیرد. در ادامه تگ های داخلی تگ <head> را معرفی می کنیم که بر حسب نیاز مورد استفاده قرار می گیرند.

تگ <title></title>

در جلسه اول معرفی کردیم که عنوان سایت را در سربرگ مرورگر نشان می دهد.

تگ <link>

برای لینک کردن فایل css به سند html استفاده می شود(استایل دهی به روش external).

تگ <style></style>

در روش استایل دهی internal از این تگ استفاده می شود(آموزش جلسه سوم).البته قرار شد که ما در آموزش ها از روش external برای استایل دهی استفاده کنیم چون استاندارد تره پس نیازی به این تگ نداریم.

تگ <script></script>

برای معرفی کدهای جاوا اسکریپت استفاده می شود که مربوط به مبحث این دوره نمی باشد.

می توانیم بگوییم از مهمترین تگ های داخلی تگ <head> می باشد از این رو که در سئو سایت بسیار تاثیر گذار است. از این تگ برای مشخص کردن محتوای صفحه مثل: توضیحات صفحه، کلمات کلیدی، نویسنده و... استفاده می شود(اطلاعات بیشتر در wikipedia). در ادامه چند تا از خصوصیات و مقادیر مهم این تگ را بررسی می کنیم:

- **charset**: از مهمترین خصوصیت های این تگ charset می باشد که به مرورگر خواهد گفت که صفحه شما از چه ساختاری استفاده می کند و کاراکترهای آن از چه یونیکدی پیروی می کند.(یونیکد: روشی استاندارد که با آن کاراکترها(به فرض فارسی) برای مرورگر قابل فهم و تبدیل به کد می شوند.) از این تگ در ادامه آموزش ها حتما استفاده کنید.

<head>

<meta charset="utf-8">

</head>

- **name**: صفت name خصوصیتی هست که با توجه به مقادیر مختلفی که می گیرد ، عملکردی متفاوت دارد(توجه داشته باشید صفت name همراه با صفت content استفاده می شود):
 - **description**: توضیحات صفحه وب را مشخص می کند. (در صفت content توضیحات را می نویسیم)
 - **keywords**: کلمات کلیدی را مشخص می کند.(در صفت content کلمات کلیدی را می نویسیم)
 - **author**: نویسنده صفحه وب را مشخص می کند.(در صفت content نام نویسنده را می نویسیم)

مثال:

<head>

<meta charset="utf-8">

<meta name="description" content="آموزش html و css">

<meta name="keywords" content="html,css">

<meta name="author" content=" " >

</head>

به صورت خلاصه با تگ ها و صفات زیر آشنا شدیم:

صفات تگ <html>

نام	مقادیر	توضیح
lang	fa (زبان فارسی)	مشخص کننده زبان سایت
dir	ltr یا rtl	مشخص کننده چپ چین یا راست چین بودن سایت

تگ

<meta> برای مشخص کردن محتوای صفحه مثل توضیحات صفحه، کلمات کلیدی و ... برای موتورهای جستجو(باید داخل تگ <head></head> قرار بگیرد)

صفات تگ meta

نام	مقادیر	توضیح
charset	utf_8	مشخص کردن ساختار و یونیکد صفحه
name	description	مشخص کردن توضیحات صفحه(موثر در سئو سایت)
name	keywords	مشخص کردن کلمات کلیدی صفحه
name	author	مشخص کردن نویسنده صفحه

با تگ جدول در [html](#) و استایل دهی آن آشنا خواهیم شد

آشنایی با تگ جدول در html و استایل دهی به آن

تگ های جدول

برای ایجاد جدول و سطرها و ستون هایش از چند تگ استفاده می شود که ابتدا آن ها را نام می بریم و سپس ساختار آن را می بینیم:

<table></table>: جدول را با این تگ تعریف می کنیم.

<tr></tr>: هر یک از ردیف های جدول را با این تگ ایجاد می کنیم.

<th></th>: عناوین سطر اول جدول (یا همون سلول های ردیف اول) را با این تگ ایجاد می کنیم. (به صورت پیش فرض، تیتر جداول به صورت پررنگ می باشد و در مرکز قرار دارند).

<td></td>: و نهایتا هر سلول جدول را با این تگ تعریف می کنیم. (می توانیم هر المانی مثل متن یا تصویر یا لینک یا هر چیز دیگری درون این تگ قرار بدهیم).

خب بریم سراغ یک مثال خیلی ساده.


```
<!DOCTYPE html>
```

```
<html lang="fa" dir="rtl">
```

```
<head>
```

```
<link href="style.css" rel="stylesheet" type="text/css">
```

```
<meta charset="utf-8">
```

```
<title>
```

```
اینجا عنوان سایت است
```

```
</title>
```

```
</head>
```

```
<body>
```

```
<p>نوشتن هر متنی که دلتون میخواد</p>
```

```
<table>
```

```
<tr>
```

```
<th>عنوان اول</th>
```

```
<th>عنوان دوم</th>
```

```
<th>عنوان سوم</th>
```

```
</tr>
```

```
<tr>
```

```
<td>متن اول</td>
```

```
<td>متن دوم</td>
```

```
<td>متن سوم</td>
```

```
</tr>
```

```
<tr>
```

```
<td>متن اول</td>
```

```
<td>متن دوم</td>
```

```
<td>متن سوم</td>
```

```
</tr>
```

```
</table>
```

```
</body>
```

```
</html>
```

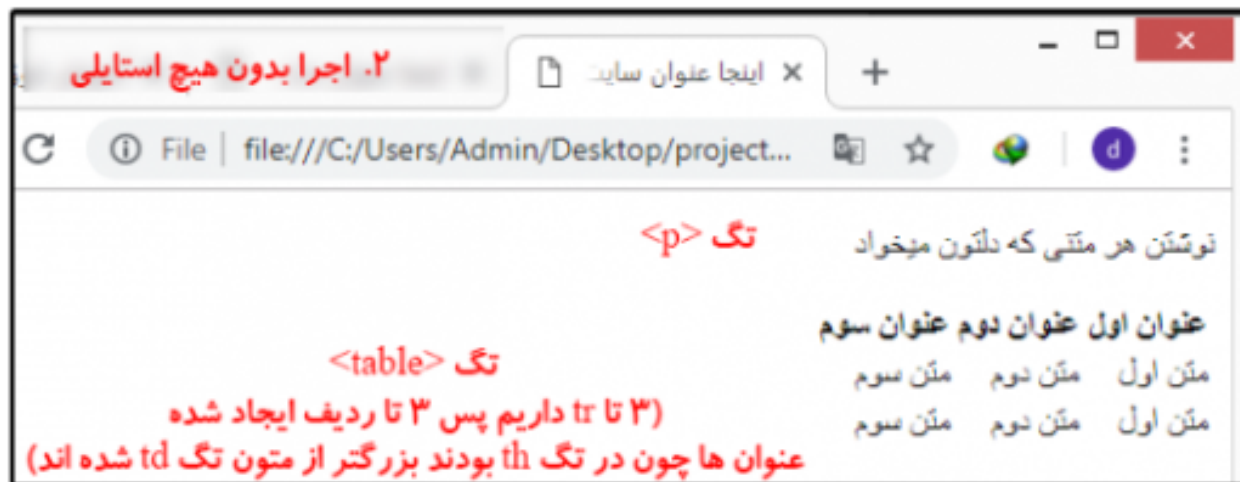
گام اول) نوشتن کدهای html

کدهای زیر در یک سند html وارد می کنیم:

خب فکر نمی‌کنم نیاز توضیح اضافی تر باشد فقط اینکه به هر تعداد <th> داشته باشیم به همان تعداد باید <td> در هر ردیف داشته باشیم
سند را ذخیره می‌کنیم و آن را اجرا می‌کنیم. (توجه داشته باید که هنوز در سند CSS چیزی ننوشتیم)

```
<!DOCTYPE html>
<html lang="fa" dir="rtl">
<head>
<link href="style.css" rel="stylesheet" >
<meta charset="utf-8">
<title>
اینجا عنوان سایت است
</title>
</head>
<body>
<p>نوشتن هر متنی که دلتون میخواد</p>
<table>
<tr>
<th>عنوان اول</th>
<th>عنوان دوم</th>
<th>عنوان سوم</th>
</tr>
<tr>
<td>متن اول</td>
<td>متن دوم</td>
<td>متن سوم</td>
</tr>
<tr>
<td>متن اول</td>
<td>متن دوم</td>
<td>متن سوم</td>
</tr>
</table>
</body>
</html>
```

۱. کدهای html



گام دوم) نوشتن کدهای CSS

خب به بهانه استایل دهی به جدولمون با چند property در CSS آشنا می شویم.

ویژگی border :

برای ایجاد کادر برای هر المانی در CSS از خاصیت **border** (به معنای لبه) استفاده می کنیم. با 3 مقدار به صورت کامل مشخص می شود:

- **مشخصه اول (1px):** ضخامت حاشیه رو مشخص می کند
- **مشخصه دوم:** مشخص می کند که حالت کادر چگونه باشد، (solid) یعنی خط به هم پیوسته است ، (dotted) یعنی کادر به شکل خط ممتد باشد .
- **مشخصه سوم:** رنگ حاشیه را مشخص می کند.

برای استایل دهی به جدولی که در سند html ایجاد کردیم، وارد فایل style.css می شویم و با نوشتن کدهای زیر ، برای سلول های جدول کادر ایجاد می کنیم:

```
th,td{  
    border : 1px solid red;  
}
```

نکته: اگر بخواهیم به دو تا المان یا تگ را یک استایل یکسان بدهیم آن ها را کنار هم می نویسیم و با کاما از هم جدا می کنیم.

نتیجه:

```
File Edit Search View Encoding Language S
index.html style.css
1 th,td{
2   border:1px solid red;
3 }
```

۱. حاشیه پیوسته
(در فایل css)

```
File Edit Search View Encoding Language S
index.html style.css
1 th,td{
2   border:1px dotted red;
3 }
```

۱. حاشیه ممتد



خب اگر بخواهیم کادرهای سلول های جدول بهم بچسبند ، برای المان table ، از خاصیت border-collapse استفاده می کنیم.

- **collapse** : فاصله بین کادرهای جدول حذف می شود.
- **separate** : به صورت پیش فرض این مقدار می باشد و بین کادرها فاصله ایجاد می کند.

```
table{
```

```
border-collapse:collapse;
```

```
}
```

نتیجه:

The image shows a code editor window with the following CSS code:

```

1  th, td{
2      border:1px solid red;
3  }
4  table{
5      border-collapse:collapse;
6  }

```

Below the code editor is a browser window showing the rendered output. The browser title is "اینجا عنوان سایت". The address bar shows "http://localhost/Desktop/project...". The page content includes the text "نوشتن هر متنی که دلتون میخواد" and a table with the following structure:

۲. نتیجه اجرا

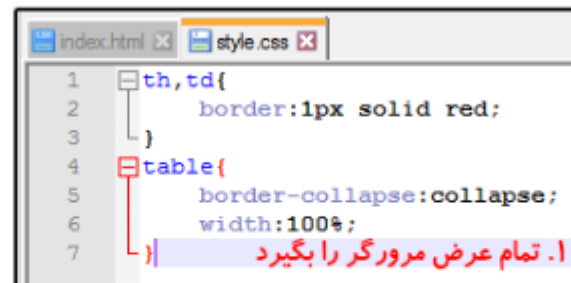
عنوان اول	عنوان دوم	عنوان سوم
متن اول	متن دوم	متن سوم
متن اول	متن دوم	متن سوم

ویژگی width , height

برای اینکه ارتفاع و عرض محتوای یک عنصر را تعیین کنیم از ویژگی های height و width استفاده می شود. این خصوصیت مقادیر مختلفی دارد که معمول ترین آنها دو نوع می باشد:

- به صورت طولی (مثلا 100px)
 - به صورت درصدی: یعنی چند درصد والد خود را فراگیرد.(والد و فرزند رو بعدا در بحث های بعدی متوجه خواهید شد) الان که بنویسیم 100% یعنی جدول به اندازه مرورگر رشد کند.(عرض مرورگر هر مقدار باشد ، جدول به همان اندازه رشد می کند)
- خب حالا می خواهیم امان جدول تمام عرض مرورگر را بگیرد، به صورت زیر مقدار width را 100% می دهیم.

```
table{  
    border-collapse:collapse;  
    width:100%;  
}
```



```
1 th,td{  
2     border:1px solid red;  
3 }  
4 table{  
5     border-collapse:collapse;  
6     width:100%;  
7 }  
    ۱. تمام عرض مرورگر را بگیرد
```

نتیجه:



نوشتن هر متنی که دلتون میخواد

عنوان اول	عنوان دوم	عنوان سوم
متن اول	متن دوم	متن سوم
متن اول	متن دوم	متن سوم

۲. نتیجه اجرا

ویژگی text-align :

با استفاده از این ویژگی می توانیم چیدمان محتوای داخل یک المان (مثل متن ، تصویر یا...) را تعیین کنیم. این خصوصیت مقادیر مختلفی دارد که معمول ترین آنها برای محتوای خطی به شکل زیر می باشد:

- **right**: راست چین
- **left**: چپ چین
- **center**: وسط چین
- **justify**: این مقدار هم مانند مقدار justify در word می باشد، باعث می شود تا برای تمام خطوط به جز خط آخر ، محتوا تمام خط را از لبه سمت راست تا لبه سمت چپ کامل پر کند.

در حال حاضر محتوای داخل المان جدول ، متن می باشد. برای اینکه متن های وسط قرار بگیرند. خاصیت text-align را برای المان table تعریف می کنیم.

```
table{  
border-collapse:collapse;  
width:100%;  
text-align:center;  
}
```



```
1 th,td{  
2     border:1px solid red;  
3 }  
4 table{  
5     border-collapse:collapse;  
6     width:100%;  
7     text-align:center;  
8 }
```

۱. متن های جدول وسط قرار بگیرند

نتیجه:



نوشتن هر متنی که دلگرم میخواد

عنوان اول	عنوان دوم	عنوان سوم
متن اول	متن دوم	متن سوم
متن اول	متن دوم	متن سوم

صفات تگ سلول های جدول:

colspan: چند ستون جدول را با هم merge یا ادغام کنیم.

```
<table>
  <tr>
    <td colspan="2"> دو متن با عرض دو </td>
  </tr>
  <tr>
    <td>متن اول</td>
    <td>متن دوم</td>
  </tr>
</table>
```

rowspan: چند ردیف یک جدول را باهم ادغام کنیم از صفت rowspan استفاده می کنیم.

```
<table>
  <tr>
    <td>متن اول</td>
    <td>متن دوم</td>
  </tr>
  <tr>
    <td rowspan="2"> دو متن با طول دو </td>
  </tr>
</table>
```


این دو مورد هم تست کنید و نتیجه رو ببیند.

استایل هایی که در جلسات قبل هم یاد گرفتید مثل background، color و ... را روی همین جدول اعمال کنید و نتیجه را ببینید. خب آموزش ما اینجا به پایان رسید. به صورت خلاصه با تگ ها و صفات زیر آشنا شدیم ، تا همین حد آشنایی کافی است در صورتی که در آینده نکته ای بود که از قلم افتاده بود حتما اضافه خواهیم کرد.

تگ		
	برای برای ایجاد جدول	<code><table></table></code>
	برای ایجاد سلول های عنوان جدول	<code><th></th></code>
	برای ایجاد سطر	<code><tr></tr></code>
	برای ایجاد سلول	<code><td></td></code>
css در property		
نام	مقادیر	توضیح
border	dotted – solid	مشخص کردن کادر برای المان
border-collapse	collapse-separate	ایجاد یا حذف فاصله بین کادرها
height و width	درصدی یا مقدار طولی	برای مشخص کردن عرض و ارتفاع المان
text-align	right-left-center-justify	مشخص کردن چیدمان محتوای داخل المان
صفات تگ td		
نام	مقادیر	توضیح
colspan	مقدار عددی	ادغام ستون های جدول
rowspan	مقدار عددی	ادغام ردیف های جدول

در آموزش بعدی با [تگ a در html](#) و صفات آن و همچنین نحوه استایل دهی به آن آشنا خواهیم شد.