

جلسه اول مجازی
طراحی وب
کاردانی نرم افزار ۲

مدرس :

فاطمه دهقانی فیروزآبادی

آشنایی با تگ a در html و صفات آن

ساده ترین ساختار تگ a در html

تگ `<a>` با استفاده از این تگ می توانیم لینک ایجاد کنیم ، لینک به کجا؟ به هر جایی ، به مکانی داخل سایت خودمون یا مکانی خارج سایت. یکی از فایده های این تگ بهبود اهداف سئو و بهتر دیده شدن در میان رقبا می باشد(پس توصیه می کنم خوب یاد بگیرید تا ارزش استفاده کنید).

ساده ترین ساختار تگ `<a>` به شکل زیر می باشد:

`` نوشته ای که کاربر با کلیک روی آن به لینک داده شده هدایت می شود ``

مثل:

`` پردازش ``

صفت href:

این صفت آدرس مکانی که می خواهیم به آن لینک بدهیم را مشخص می کند، (تگ a در html با این صفت معنا پیدا می کند. پس در ساده ترین ساختار هم باید نوشته شود).

توجه داشته باشید آدرس باید با پروتکل http یا https باشد.(بدون پروتکل هم تست کنید)

بین تگ `<a>` : متنی است که کاربر با کلیک روی آن به جایی که لینک داده ایم هدایت می شود.(مجبور نیستیم حتما متن بنویسیم بلکه می توانیم هر تگی بین تگ `<a>` بنویسیم. مثلا یک تصویر ایجاد کنیم که کاربر با کلیک روی آن به لینک مورد نظر هدایت شود(تگ تصویر را در جلسات بعد خواهیم گفت)

صفات دیگر تگ a

تگ <a> علاوه بر href چند تا صفت مهم دیگر هم دارد که یکی یکی آن ها را توضیح خواهیم داد:

صفت download:

این صفت فقط برای زمان هایی استفاده می شود که می خواهیم به صورت مستقیم لینک بدهیم تا یک فایل صوتی، تصویری، pdf یا هر چیز دیگری دانلود شود. (لینکی که به href می دهیم به mp4, یا mp3, یا pdf...ختم می شود)

متنی که به کاربر نمایش می دهد

صفت hreflang:

زبان مکانی که به آن لینک داده شده را نشان می دهد.

متنی که به کاربر نمایش می دهد

مثال:

سایت فارسی

صفت target:

این صفت هم اکثر مواقع استفاده می شود. با این صفت مشخص می کنیم که لینک در همین صفحه باز شود یا در صفحه جدید.

صفت target:

این صفت هم اکثر مواقع استفاده می شود. با این صفت مشخص می کنیم که لینک در همین صفحه باز شود یا در صفحه جدید.

این صفت دو مقدار پرکاربرد می تواند داشته باشد:

- **_self** : یعنی عبارت لینک شده در همین پنجره باز شود(به صورت پیش فرض این صفت برابر self_ می باشد پس حتی اگر صفت target را استفاده نکنیم در همین صفحه باز می شود) مثال:

```
<a href="https://dazhyarco.net" target="_self" > دازیار </a>
```

- **_blank** : یعنی عبارت لینک شده در پنجره جدید باز شود.

```
<a href="https://dazhyarco.net" target="_blank" > دازیار </a>
```

البته سه مورد دیگر هم وجود دارد که فقط نام می برم چون مربوط به بحث ما نمی شوند(_parent, _top, framename).

اینا رو همینجوری نخونین و رد بشینا! یکی یکی در سند html خودتون بزنید و نتیجه رو ببینین! :-)

صفت rel:

این صفت مقادیر مختلفی می تواند داشته باشد که از مهمترین آن ها می توانیم به دو مقدار زیر اشاره کنیم که به سئو سایت مربوط می شود:

- **Follow** : با دادن این مقدار به صفت rel ، به ربات های گوگل پیغام می دهیم که من این لینک را تایید می کنم.
- **Nofollow** : با دادن این مقدار به صفت rel ، به ربات های گوگل پیغام می دهیم که من این لینک را تایید نمی کنم.

نحوه استایل دهی به تگ <a>

خب حالا که به صورت کامل با ساختار تگ a در html و صفت های مهم آن آشنا شدیم، میریم سراغ استایل دهی به این تگ.

فایل **style.css** که جلسات قبل ایجاد کرده بودیم را باز می کنیم و کدهای زیر را در آن تایپ می کنیم:

```
a{  
    color : #ff33ff;  
    text-decoration : none;  
}
```

توضیح کد:

در جلسات قبل گفتیم می توانیم به سه روش استایل ها را روی المان ها اعمال کنیم که یکی از آن ها استفاده از نام المان است. الان که نوشتیم { a}، یعنی هر خصوصیتی که در بین این تگ نوشته شده روی تمام تگ های لینک موجود در سند html اعمال می شود.

Color: قبلا هم گفتیم رنگ متن را مشخص می کند.

text-decoration : این خصوصیت مربوط به خطی است که زیر لینک ها کشیده می شود، چهار مقدار می تواند داشته باشد:

- **none**: بدون خط
- **Overline**: خط بالای متن قرار بگیرد.
- **line-through**: خط روی متن قرار بگیرد.
- **underline**: خط زیر متن قرار بگیرد.(به صورت پیش فرض text-decoration برابر این مقدار است)

استایل هایی که در جلسات قبل هم معرفی کردیم روی آن اعمال کنید تا کاملا آشنا شوید (-):

معرفی سلکتورهای رویدادی (اجرای استایل ها موقع رویدادها)

خب در ادامه می خواهیم چهارتا سلکتور معرفی کنیم که استایل ها را موقعی روی المان اعمال می کنند که رویدادی اتفاق بیفتد :

- **hover**: وقتی با موس روی المان برویم استایل اعمال می شود.
- **Focus**: وقتی که رو المانی کلیک شود استایل ها اعمال می شود و تغییرات تا زمانی که روی المان دیگری کلیک نشده باقی می ماند.
- **Visited**: این سلکتور فقط روی تگ `<a>` کار می کند. وقتی روی لینکی کلیک شود و آن لینک باز شود ، استایل های این سلکتور اعمال می شوند.
- **Active**: مشابه سلکتور Focus می باشد با این تفاوت که فقط در لحظه کلیک استایل ها اعمال می شوند و استایل ها باقی نمی ماند.

خب یک مثالی میزنیم همراه با ما تایپ کنید تا مفهوم این چهار تا سلکتور را بهتر درک کنیم، اگر کدهای html را به شکل زیر در نظر بگیرم:

```
<!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>
```

```
<h1><a class="classa" href="https://dazhyarco.net" target="_blank">
```

```
</a></h1>:1
```

```
<h2><a class="classb" href="https://dazhyarco.net" target="_self">
```

```
</a></h2>:2
```

```
<h3><a class="classc" href="https://dazhyarco.net" target="_blank">
```

```
</a></h3>:3
```

```
<h4><a class="classd" href="https://dazhyarco.net" target="_self">
```

```
</a></h4>:4
```

```
</body>
```

```
</html>
```

معرفی دو صفت **id** و **class**

نکته: هر المانی را می توانیم با دو تا صفت **id** و **class** نامگذاری کنیم تا به راحتی در طول طراحی سایت هر جا خواستیم به آن المان اشاره کنیم، از **id** یا **class** استفاده کنیم.

تفاوت **id** و **class**:

- صفت **id** برای هر المان منحصر به فرد است یعنی نباید دو المان **id** یکسانی داشته باشند ولی صفت **class** می تواند تکراری باشد.
- المان با صفت **id** را با علامت **#** و المان با صفت **class** را با **.** فراخوانی می کنیم.

و کدهای CSS را به شکل زیر در فایل style.css وارد می کنیم:

```
a{
color:#ff33ff;
text-decoration:none;
}
.classa:hover{
background: rgba(39,221,152, 0.8);
}
.classb:focus {
background-color: red;
}
.classc:visited {
background-color:yellow;
}
.classd:active {
background-color: #c40951 ;
}
```

نهایتاً اجرا می گیریم و نتیجه را می بینیم.

نکته: در مثال بالا سه مدل مقدار دهی رنگ را استفاده کردیم در مدل `rgba(39221,152,0.8)` سه عدد اول بین 0 تا 255 می باشد و مقدارهای قرمز، سبز، آبی را مشخص می کند و عدد چهارم مقدار شفافیت را نشان می دهد که بین 0 تا 1 می باشد. تست کنید تا بهتر متوجه شوید.

هدایت کاربر به یک المان داخل صفحه با استفاده از تگ `a` (ایجاد بوک مارک)

با استفاده از تگ `a` می توانیم به هر المانی تو سایت لینک شویم، یعنی چی؟ یعنی اگر صفحات وب سایت شما طولانی باشد، با استفاده از بوک مارک که در ادامه ایجاد می کنیم، کاربر با یک کلیک به بخش های بوک مارک شده هدایت می شود. برای این کار کافیست به المانی که می خواهیم کاربر به آن هدایت شود `id` بدهیم و سپس این `id` را به عنوان لینک به صفت `href` تگ `a` بدهیم تا با آن المان لینک شود.

به صورت کلی ساختار آن به صورت زیر است:

ایجاد بوک مارک: اول یک صفت `id` برای المان مورد نظری که می خواهیم کاربر به آن هدایت شود، تعریف می کنیم:

```
<p id="par">می خواهیم کاربر به این المان هدایت شود</p>
```

لینک به بوک مارک: سپس در هر جایی که دوست داریم لینک زیر را می نویسیم تا کاربر با کلیک روی عبارت "برو به ..." به پاراگراف `p` هدایت شود:

```
<a href="#par">... برو به </a>
```

به صورت خلاصه با تگ a در html و صفات آن آشنا شدیم که به صورت خلاصه می توانید ببینید:

تگ

برای ایجاد لینک

<a>

css در property

نام	مقادیر	توضیح
text-decoration	none, Overline, line-through, underline	مشخص کردن خصوصیت خط زیر لینک

صفات تگ a

نام	مقادیر	توضیح
href	آدرس url لینک	مشخص کننده آدرس url لینک
download	-	لینک دهی مستقیم به فایل تصویری، صوتی و ...
hreflang	مثل fa	مشخص کننده زبان مکان هدف
target	_blank و _self	لینک در همین صفحه باز شود یا در صفحه جدید
rel	Follow و Nofollow	لینک فالو شود یا خیر

سلکتورهای رویدادی

نام	مقادیر	توضیح
:hover		وقتی با موس روی المان برویم استایل اعمال می شود
:focus		وقتی که رو المانی کلیک شود استایل ها اعمال می شود و تغییرات تا زمانی که روی المان دیگری کلیک نشده باقی می مانند.
:Visited		این سلکتور فقط روی تگ <a> کار می کند. وقتی روی لینکی کلیک شود و آن لینک باز شود ، استایل های این سلکتور روی لینک اعمال می شوند.
:Active		مشابه سلکتور focus می باشد با این تفاوت که فقط در لحظه کلیک استایل ها اعمال می شوند و استایل ها باقی نمی مانند.

آشنایی با تگ `img` در `html` و صفات آن

ساده ترین ساختار تگ `img` در `html`

تگ ``: تصاویر یکی از مهمترین مباحث ها در طراحی سایت می باشند برای ایجاد تصویر از تگ `` استفاده می کنیم، تگ `img` از اون دسته تگ هاییه که بسته ندارد.

ساده ترین ساختار تگ `` به شکل زیر می باشد:

```

```

صفت `Src`: آدرس تصویر را مشخص می کند.

صفات دیگر تگ img

این تگ چندین صفت مهم دیگر هم دارد :

Alt: این از مهمترین صفاتی هست که توصیه می شود از آن استفاده کنید. در صورتی که عکس به هر دلیلی باز نشود این متن نمایش داده می شود. (در سئو سایت هم بسیار مهم است، یک چیزی داریم میگویم گوگل فقط می خونه و نمی بینه پس با اضافه کردن این صفت برای تصویرتون ، اون رو خوانا کنید)

Title: عنوان تصویر است و زمانی که با موس روی تصویر برویم نمایش می دهد.

Height: ارتفاع تصویر را مشخص می کند.(همیشه واحدش px می باشد)

Width: عرض تصویر را مشخص می کند.(همیشه واحدش px می باشد)

```

```

تصویر به عنوان لینک

اگر بخواهیم تصویر به یک آدرسی لینک شود ، یعنی بتوانیم با کلیک روی آن به آدرس دلخواهی هدایت شویم، کافیه تگ `` را داخل تگ `<a>` قرار بدهیم. به شکل زیر:

```
<a href="https://dazhyarco.net">
```

```
  
```

```
</a>
```

تصویر به عنوان نقشه

تگ `<map></map>` از تگ `<map>` برای تعریف یک نقشه - تصویر (image-map) استفاده می شود. یعنی با این تگ بعضی نقاط تصویر قابلیت کلیک شدن پیدا می کند.

صفت usemap : برای برقراری ارتباط بین تگ `` و تگ `<map></map>` از یک خصوصیت به نام usemap در تگ `` استفاده می کنیم سپس مقدار این خصوصیت را به خصوصیت name در تگ `<map>` می دهیم. (جلوتر خواهید دید)

```

```

تگ `<area>` یکی از مهمترین تگ های داخلی `<map>` تگ area هست که کار این تگ تعیین موقعیت های قابل کلیک در تصویر می باشد. (این تگ هم از اون دسته تگ هایی که بسته ندارد.

صفات این تگ به شرح زیر است:

- **صفت shape** : برای تعیین شکل ناحیه قابل کلیک استفاده می شود مقادیر آن می توانید موارد زیر باشد:
 - مربع و مستطیل : rect , rectangle
 - دایره : circle
 - چندضلعی: poly
- **صفت coords** : با این صفت مختصات نقاط ناحیه مشخص شده را تعیین می کنیم.
 - مربع و مستطیل: 4 عدد - مختصات نقاط 4 گوشه
 - دایره: 3 عدد - دو عدد اول مختصات x و y مرکز دایره و عدد سوم شعاع دایره را مشخص می کند.

- **صفت href** : برای تعیین لینکی که کاربر با کلیک روی تصویر به آن مسیر هدایت شود.

در مثال زیر دو ناحیه در تصویر فرضی مشخص شده است، یکی به شکل مستطیل و دیگری دایره ، که با coords مختصات این نواحی را مشخص کردیم و با href لینک دادیم.

```
  
<map name="dpi">  
<area shape="rect" coords="0,0,60,60" href="https://www.dazhyarco.net">  
<area shape="circle" coords="90,30,30" href="https://www.dazhyarco.net">  
</map>
```

(برای دیدن مثال دقیق تر می توانید به سایت www.w3schools.com مراجعه کنید.)

تگ

برای ایجاد تصویر

صفات تگ img

نام	مقادیر	توضیح
src	آدرس تصویر	مشخص کردن آدرس تصویر
alt	دلخواه	توضیح کوتاه درباره تصویر
title	دلخواه	عنوان برای تصویر
height	مقدار عددی برحسب پیکسل	ارتفاع تصویر
width	مقدار عددی برحسب پیکسل	عرض تصویر
username	#دلخواه	برای برقراری ارتباط بین تگ img و تگ map

تگ

<map></map>	برای تعریف نقشه - تصویر (قابلیت کلیک روی نواحی خاص از تصویر)
<area>	برای تعیین موقعیت های قابل کلیک روی تصویر (تگ داخلی تگ map)

صفات تگ area

نام	مقادیر	توضیح
shape	rect , rectangle,circle,poly	مشخص کردن شکل ناحیه قابل کلیک
coords	4 عدد: مختصات چهارضلعی 3 عدد: مختصات دایره	مختصات نقاط ناحیه
href	دلخواه	تعیین آدرس url دلخواه

تگ ul و ol در html

در html دو نوع لیست داریم :

- **لیست شمارشی یا مرتب:** برای نمایش این لیست از تگ `` استفاده می کنیم.
- **لیست غیر شمارشی یا نامرتب:** برای نمایش این لیست از تگ `` استفاده می کنیم. برای **ایجاد منو** از این تگ استفاده می شود.

داخل هر دو تگ `` و `` از تگ `` برای ایجاد آیتم استفاده می شود. در واقع بدون تگ `` تگ `ol` و `ul` کاربردی ندارند.

```
<ol>  
  <li>شماره 1</li>  
  <li>شماره 2</li>  
  <li>شماره 3</li>  
</ol>
```

```
<ul>  
<li>شماره 1</li>  
<li>شماره 2</li>  
<li>شماره 3</li>  
</ul>
```

استایل دهی به ساختار لیست غیرشمارشی

همانطور که گفتیم از لیست غیرشمارشی یا `` برای **ایجاد منو** استفاده می کنیم اما وجود bullet ها اضافی می باشند برای حذف آن ها فایل `style.css` را باز کنید و کد زیر را وارد کنید:

```
ul li {  
    list-style: none;  
}
```

ویژگی `list-style`: را برابر با **none** قرار می دهیم تا bullet ها حذف شوند.

نکته: (به سلکتور توجه کنید) ما برای اشاره به `li` ها، نوشتیم `{ ul li }` . چرا؟ چون می خواستیم `li` های موجود در تگ `ul` را صدا بزنیم و استایل دهی کنیم . با این اوصاف اگر بخواهیم `li` های موجود در لیست `ol` را استایل دهی کنیم چی باید بنویسیم؟ `{ ol li }`

اگر بخواهیم تمام `li` های موجود ، چه در لیست شمارشی چه در لیست غیر شمارشی را استایل دهی کنیم، فقط می نویسیم : `{ li }`

ویژگی `list-style-type`: به جای ویژگی `list-style` می توان از این ویژگی به صورت کلی تر استفاده کرد:

- **disc:** به شکل دایره توپر
- **circle:** به شکل دایره تو خالی
- **square:** به شکل مربع
- **none:** هیچی نمایش ندهد

تعریف صفت برای تگ `ol` یا لیست شمارشی

در مثال بالا دیدیم که لیست شمارشی با اعداد مرتب شده است، اگر بخواهیم به فرض با حروف انگلیسی مرتب شوند از صفت `type` استفاده می کنیم.

این صفت 4 مقدار می تواند داشته باشد:

- عدد: لیست با اعداد شماره گذاری شود (به صورت پیش فرض با اعداد مرتب می شوند)
- حرف `a`: به ترتیب حروف کوچک الفبای انگلیسی مرتب شود.
- حرف `A`: به ترتیب حروف کوچک الفبای انگلیسی مرتب شود.
- حرف `i`: به ترتیب اعداد لاتین کوچک مرتب شود.
- حرف `I`: به ترتیب اعداد لاتین بزرگ مرتب شود.

```
<ol type="a">  
  <li>شماره 1</li>  
  <li>شماره 2</li>  
</ol>
```

صفت `start`: از این صفت می توانیم مشخص کنیم شمارش از چه عدد یا حرفی شروع شود:

صفت start: از این صفت می توانیم مشخص کنیم شمارش از چه عدد یا حرفی شروع شود:

در مثال زیر از عدد 3 شروع به شماره گذاری می کند:

```
<ol start="3">  
<li>شماره 1</li>  
<li>شماره 2</li>  
</ol>
```

در مثال زیر از حرف c شروع به مرتب کردن می کند:

```
<ol type="a" start="c">  
<li>شماره 1</li>  
<li>شماره 2</li>  
</ol>
```

امیدوارم با تگ ul و ol در html به خوبی آشنا شده باشید. در جلسات آتی با چندین خصوصیت مهم در CSS آشنا خواهیم شد و این قول رو بهتون می دهیم که خیلی زود یک آموزش در رابطه با ایجاد منو با لیست غیر شمارشی تهیه خواهیم کرد.

تگ

ایجاد لیست شمارشی	<code></code>
ایجاد لیست غیر شمارشی	<code></code>
ایجاد آیتم های لیست شمارشی و غیر شمارشی	<code></code>

صفت تگ ol

نام	مقادیر	توضیح
type	عدد, a, A, i, j	مشخص کردن ترتیب براساس حروف یا عدد
start	-	مشخص کردن شروع شمارش

استایل دهی به تگ ul

نام	مقادیر	توضیح
list-style	none	حذف کردن bullet ها
list-style-type	disc, square, circle, none	مشخص کردن شکل bullet ها

آشنایی با انواع سلکتورها یا انتخابگرها در CSS

انتخابگرها

در ادامه با چندین روش انتخاب المان یا selectorها آشنا می شویم:

1. **element**

2. **class**

3. **id**

Selector

نام المان

یا

#id

یا

.class

Declaration

Declaration

```
{ color: blue; background: blue; }
```

↑
Property

↑
Value

↑
Property

↑
Value

که بسیار ساده است و تا الان با آن کار کردیم. در واقع از طریق نام المان آن را فراخوانی می کنیم مثلا در کد زیر تمام المان های a موجود در سند html با رنگ #ff33ff نمایش داده شوند:

```
a{
    color : #ff33ff;
}
```

سلکتور .class

برای اینکه از روش دوم استفاده کنیم لازم است در سند html، یک صفت برای المانی که می خواهیم استایل بدهیم ، تعریف کنیم به نام class و هر مقدار دلخواهی که می خواهیم به آن بدهیم . سپس در فایل CSS، المان را با نام کلاس اش صدا می زنیم ، علامت . (نقطه) قبل از نوشتن مقدار class مهم است.

در سند HTML: فرضاً تگ a با کلاس link داریم:

```
<a class="link" href="#">تگ لینک</a>
```

در فایل CSS : یعنی تمام المان ها با کلاس link با رنگ #ff33ff نمایش داده شوند:

```
.link{  
    color : #ff33ff;  
}
```

ممکن است المان های دیگری با همین کلاس link در سند html وجود داشته باشد. که همگی این استایل را به خود می گیرند.

نکته: یک عنصر می تواند در چندین کلاس قرار بگیرد. مثلا دو کلاس داریم با استایل های مختلف به نام های link و txt. در این صورت باید در سند html به صورت زیر به class مقدار بدهیم. هر دو کلاس را با فاصله کنار هم می نویسیم. این طور استایل دادن پرکاربرد است.

```
<a class="link txt" href="#">تگ لینک</a>
```


سلکتور #id

برای اینکه از روش سوم استفاده کنیم لازم است در سند html، یک صفت برای المان مورد نظر تعریف کنیم به نام id و مقدار منحصر به فرد به آن بدهیم. سپس در فایل CSS، المان را با آی دی اش صدا بزنیم، علامت # (شارپ) قبل از نوشتن مقدار id مهم است. **توجه داشته باشید** که مقدار id نباید تکراری باشد و هیچگاه با عدد آغاز نمی شود. در تصویر زیر می توانید هر سه روش را ببینید:

مثال:

در سند HTML: فرضاً تگ a با id = link داریم:

```
<a id="link" href="#">تگ لینک</a>
```

در فایل CSS: یعنی المان با آی دی link با رنگ #ff33ff نمایش داده شود:

```
#link{  
    color : #ff33ff;  
}
```

چون id منحصر به فرد است فقط یک المان با کلاس link در سند html وجود دارد که این استایل را به خود می گیرد.

ترکیب انتخابگرها

1. `element , element`

2. `element element`

3. و (سلکتورهای دیگه ای از نوع ترکیبی داریم که مربوط به CSS3 می شوند و ان شالله در آینده با آن ها آشنا خواهیم شد.)

سلکتور `element , element`

اگر چندین المان یک استایل خاصی داشته باشند آن ها را با کاما از هم جدا می کنیم و استایل می دهیم.

مثال:

در سند HTML: کدی به شکل زیر داریم:

```
<a href="#">تگ لینک</a>
```

```
<p>پاراگراف</p>
```

در فایل CSS: یعنی تمام المان های a و p در سند html با رنگ #ff33ff نمایش داده شود:

```
p,a {  
    color : #ff33ff;  
}
```

قبل از توضیح در مورد این سلکتور باید بدانیم تگ والد و فرزند چیست؟ به تگی که در دل تگ دیگر قرار بگیرد **فرزند** و به تگی که تگ دیگر را در بر گرفته **والد** می گویند. کد زیر را در جلسه قبل کار کردیم به یاد بیاورید: تگ ul پدر یا والد تگ li و فرزند تگ body می باشد.

```
<body>
<ul>
  <li>شماره 1</li>
  <li>شماره 2</li>
  <li>شماره 3</li>
</ul>
</body>
```

در جلسه قبل برای حذف bullet ها از تک تک عناصر li داخل تگ ul از سلکتور element element استفاده کردیم:

```
ul li {
  list-style: none;
}
```

پس به صورت کلی برای استایل دهی به عناصر فرزند یک والد خاص می توانیم به صورت سلسله مراتبی از این سلکتور استفاده کنیم. (یه تستی بزنید و نتیجه رو ببینید)

الویت استایل دهی در CSS

- **استایل دهی به HTML** به سه روش برای استایل دهی اشاره کردیم: `internal` و `inline` و `external`

در صورتی که به المانی از سه روش برای استایل دهی استفاده کنیم و به فرض مثال استایل `color` را بدهیم، رنگی اعمال می شود که در روش `inline` تعریف شده است پس الویت با روش **inline** می باشد.

- در فایل CSS اگر به المانی دوبار استایل داده باشیم، استایلی اعمال می شود که در خط پایین تری قرار گرفته باشد.
- همیشه سلکتور **Class** الویت بالاتری نسبت به **نام المان** دارد.
- همیشه سلکتور **Class** الویت پایین تری نسبت به **ID** دارد هر چند در خط پایین تری قرار گرفته باشد. مثلا المانی داریم که هم `Class` دارد هم `ID`. استایلی که در `ID` قرار گرفته اعمال می شود.

مثال: زیر را در نظر بگیرد که به روش **internal** استایل داده ایم. بین استایل هایی که به `h1` داده شده استایل `id` یعنی `#int` اعمال خواهد شد. هر چند استایل `class` و نام المان در خط های پایین تری قرار دارند. پس رنگ هدر آبی خواهد شد.

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      #int{background-color: blue;}
      .intro {background-color: yellow;}
      h1 {background-color: red;}
    </style>
  </head>
  <body>
    <h1 id="int" class="intro">This is a heading</h1>
  </body>
</html>
```

تغییر الویت استایل دهی با **!important**

در صورتی که بخواهیم یک استایلی بر تمام استایل های دیگر الویت داشته باشد و همه ی استایل های دیگر را کنار بزند از کلمه کلیدی **!important** استفاده می کنیم.

مثال: در سند HTML کدی به شکل زیر داریم، که یک استایل inline دارد، id و دوتا class هم دارد.

```
<h id="int" class="intro intro2" style="background-color: yellow">This is a heading</p>
```

به صورت اکسترنال هم یک فایل CSS به شکل زیر داریم:

```
<style>
h { background-color: red; }
.intro { background-color: green !important; }
#int { background-color: blue; }
.intro2 { background-color: orange; }
</style>
```

با وجود کلمه کلیدی **!important** استایل رنگ سبز اعمال می شود بدون حتی فکر کردن در مورد استایل های دیگر!

background یا پس زمینه در CSS

یکی از خصوصیات که در CSS مطرح می باشد و پر استفاده، خصوصیت **background** یا **پس زمینه** می باشد. در ابتدا یکی یکی آن ها را توضیح می دهیم سپس مثال می زنیم:

background-color

برای تعیین رنگ پس زمینه هر المان html می توان از این ویژگی استفاده کرد. مقدار رنگ را به سه شکل می توانیم مشخص کنیم:

- **کد هگزادسیمال**: رنگ را می توان با کد هگزادسیمال به شکل **#rrggbb** مشخص کرد که rr معرف رنگ قرمز، gg معرف سبز و bb معرف آبی می باشد. این مقادیر بین 00 تا FF قابل تغییر می باشد.
- **rgb(0-255,0-255,0-1)**: در این شکل از مقادیری پارامتر اول معرف رنگ قرمز، پارامتر دوم معرف سبز و پارامتر سوم معرف آبی و پارامتر چهارم معرف شفافیت می باشد.
- **اسم رنگ**

background-image

این ویژگی تصویر پس زمینه یک المان را مشخص می کند. تصویر را با استفاده از url مشخص می کنیم.

background-position

این ویژگی مکان قرارگیری تصویر پس زمینه یک المان را مشخص می کند. مقدار این ویژگی می تواند به شکل های مختلفی باشد:

- به صورت جهت جهت (left top , left center, left bottom, right top, right center, right bottom, center) : مقدار پیش فرض این خصوصیت top left (بالا-چپ) می باشد.
- به صورت درصدی (%X %Y) : x مقدار افقی و y مقدار عمودی را مشخص می کند. مقدار پیش فرض این خصوصیت 0% 0% می باشد یعنی بالا-چپ صفحه. اگر بنویسیم 100% 100% یعنی پایین-راست صفحه. اگر یک مقدار بنویسیم ، مقدار دیگر به صورت پیش فرض 50% در نظر گرفته می شود.
- به صورت عددی (Xpx Ypx): مثلا 0 0 بالا-چپ صفحه می باشد.

background-size

اندازه تصویر پس زمینه را تنظیم می کند. مقدار این ویژگی می تواند به شکل های مختلفی باشد:

- به صورت عددی: Xpx Ypx
- به صورت درصدی: X% Y%
- مقدار cover: تصویر به اندازه مکان المان گسترش پیدا می کند.
- مقدار Contain: تصویر به اندازه المانی که در داخلش قرار دارد گسترش پیدا می کند البته با حفظ تناسب.
- مقدار auto: یعنی سایز خود تصویر(به صورت پیش فرض هم این مقدار می باشد)

background-repeat

این ویژگی تکرار تصویر پس زمینه را تنظیم می کند. به صورت:

- **Repeat**: تصویر در دو راستای عمودی و افقی تکرار شود. (پیشفرض همین مقدار می باشد)
- **repeat-x**: تصویر در راستای افقی تکرار شود.
- **repeat-y**: تصویر در راستای عمودی تکرار شود.
- **no-repeat**: تکرار نشود.

background-attachment

با این خصوصیت می توانیم مشخص کنیم که مکان تصویر پس زمینه با اسکرول خوردن صفحه حفظ شود یا خیر! مقادیری که می تواند داشته باشد:

- **scroll**: تصویر پس زمینه همراه با عنصر، اسکرول می شود(به صورت پیش فرض این مقدار را دارد)
- **Fixed**: تصویر پس زمینه ثابت می باشد و اسکرول در آن تاثیری ندارد.
- **Local**: تصویر پس زمینه همراه با محتویات عنصر اسکرول می شود.

تمام مقادیری که برای هر کدام از این خصوصیت ها گفته شد را به صورت عملی تست کنید و نتیجه را ببینید . خب در ادامه با یک مثال به صورت خلاصه عملکرد این خصوصیات و بعضی مقادیر آن را خواهیم دید.

مثال:

کد html زیر را در سند خود وارد کنید:

```
<!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>background آشنایی با خصوصیت</p>  
</body>  
</html>
```

کدهای CSS هم به شکل زیر وارد می کنیم. عملکرد خصوصیات گفته شده را روی المان کلی body مشاهده می کنیم:

```
body{
```

```
/*مشخص کردن رنگ پس زمینه*/
```

```
background-color: rgb(100,2,200);
```

```
/*مشخص کردن تصویر پس زمینه*/
```

```
background-image: url('1.jpg');
```

```
/*مشخص کردن حالت تکرار تصویر*/
```

```
background-repeat:no-repeat ;
```

```
/*مشخص کردن مکان تصویر*/
```

```
background-position: center center;
```

```
/*مشخص کردن ثابت بودن یا حرکت کردن تصویر*/
```

```
background-attachment: fixed;
```

```
/*مشخص کردن سایز تصویر*/
```

```
background-size: 500px 500px;
```

```
/*مشخص کردن رنگ نوشته*/
```

```
color: #ffffff;
```

```
/*مشخص کردن سایز و فونت و ارتفاع نوشته*/
```

```
font: 20px/2000px tahoma;
```

```
}
```

css در property

توضیح	مقادیر	نام
تعیین رنگ پس زمینه هر المان html	rgb، کدهگزا دسیمال، اسم رنگ،	background-color
تعیین تصویر پس زمینه یک المان	–	background-image
تعیین مکان قرارگیری تصویر پس زمینه یک المان	به صورت جهت جهت، عددی، درصدی	background-position
تعیین کننده اندازه تصویر پس زمینه	cover، contain، auto، درصدی، عددی، به صورت	background-size
تعیین تکرار تصویر پس زمینه	repeat، repeat-x، repeat-y، no-repeat	background-repeat
تعیین مکان تصویر پس زمینه با اسکرول	scroll، fixed، local	background-attachment

آموزش تگ div در html | آشنایی با خصوصیت margin padding و border و

تگ div در html

برای نظم دادن به ساختار کدهای html و در واقع بلوک بندی سایت از تگ `<div></div>` استفاده می شود. یعنی با استفاده از این تگ می توان سند HTML (یا همون صفحه مرورگر) خود را به بلوک ها یا boxهای جداگانه تقسیم کرد و در هر بلوک محتوا و المان های موردنظر خود را قرار داد. وقتی سایت ما بلوک بندی باشد راحت تر می توانیم استایل های خود را اعمال کنیم.

پس کار تگ div بلوک بندی سند ما می باشد و هر بلوک هم شامل تعدادی المان می باشد. پس می توان گفت تگ div کار گروه بندی المان های یک سند را به عهده دارد. در نتیجه با استایل دادن به div می توان به گروهی از المان ها استایل داد. درسته؟ برای اینکه بهتر متوجه بشیم یک مثال می زنیم.

در مثال زیر ما دو تا div در سند html تعریف کردیم یعنی دو بلوک ایجاد کردیم در هر بلوک محتوایی دلخواه قرار دادیم مثلا در بلوک اول تگ <p> و در بلوک دوم دو تگ <p> و قرار دادیم. همانطور که گفتیم تگ div صرفا برای بلوک بندی و گروه بندی المان ها در سند html می باشد(در کدنویسی های پیشرفته چند بلاک یا div به صورت تو در تو تعریف می شود)

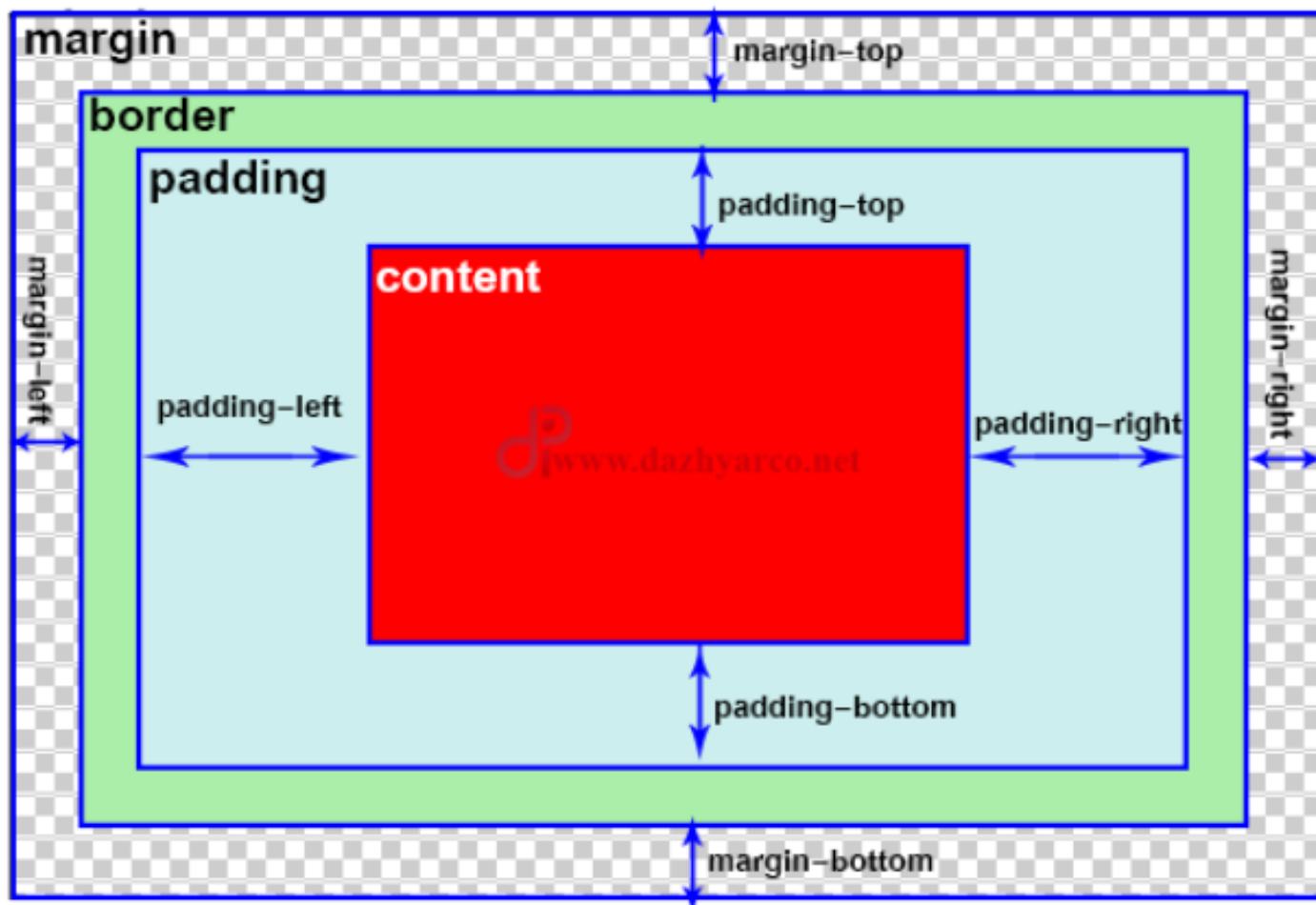
```
<!DOCTYPE html>  
<html lang="fa" dir="rtl">  
  <head>  
    <link href="style.css" rel="stylesheet">  
    <meta charset="utf-8">  
  </head>  
  </body>  
    <div class="a">  
      <p>اول div محتویات </p>  
    </div>  
  
    <div class="b">  
      <p>دوم div محتویات </p>  
        
    </div>  
  </body>  
</html>
```

در سند html به هر کدام از تگ های div یک کلاس دادیم تا در فایل CSS از این کلاس ها استفاده کنیم و به تگ های div استایل دهیم.

```
body{
background:#ffff88;
}
.a{
width:200px;
height:50px;
background:#3d4d3f;
color:#fff;
text-align:center;
}
.b{
width:300px;
height:200px;
background:#3638d7;
color:#fff;
text-align:center;
}
```

CSS در Box Model

گفتیم که محدوده هر المانی در مرورگر به صورت یک جعبه یا باکس نمایش داده می شود . در CSS هر یک از این جعبه ها یا باکس ها با استفاده از یک مدل خاصی به نام Box model توصیف می شوند. که به محدوده هر المان استایل می دهد. در ادامه می خواهیم با چهار تا خصوصیت در CSS آشنا شویم که مرتبط با مدل Box model می باشد. هر المانی می تواند شامل چهار تا خصوصیت باشد که محدوده جعبه المان را مشخص می کند تصویر زیر یک box model را نشان می دهد (



محتویات داخلی یک جعبه می باشد که می تواند شامل پارگراف <p>، تصویر ، لیست یا و یا جدول <table> و ... باشد . اندازه این قسمت را می توان با دو ویژگی **width** و **height** مشخص کرد. اگر اندازه ای به این قسمت ندهیم به اندازه المان و محتوای درونش رشد می کند.

Padding

حاشیه داخلی یک جعبه. دورتا دور content را padding فراگرفته است. پس اگر بخواهیم جعبه از چهار جهت گسترش یابد از این خصوصیت استفاده می کنیم. در واقع فاصله محتوا تا خط دورتا دور المان را می توان با padding تنظیم کرد. به چهار روش می توانیم به این خصوصیت مقدار بدهیم:

- **Padding: top right bottom left** : مثال padding: 10px 20px 30px 40px یعنی 10px از بالا و 20px از راست و 30px از چپ و 40px از پایین گسترش یابد.
- **Padding: top right&left bottom** : مثال padding: 10px 20px 30px 10px یعنی 10px از بالا و 20px از راست و چپ و 30px از پایین گسترش یابد.
- **Padding: top&bottom right&left** : مثال padding: 10px 10px 10px 10px یعنی 10px از بالا و پایین و 10px از راست و چپ گسترش یابد.
- **Padding: top&right&bottom&left** : مثال padding: 10px 10px 10px 10px یعنی 10px از بالا و پایین و راست و چپ گسترش یابد.
- **Padding-right: right** : مثال padding-right:10px یعنی از راست 10px گسترش یابد.
- **Padding-left: left** : مثال padding-left:10px یعنی از چپ 10px گسترش یابد.
- **Padding-top: top** : مثال padding-top:10px یعنی از بالا 10px گسترش یابد.
- **Padding-bottom: bottom** : مثال padding-bottom:10px یعنی از پایین 10px گسترش یابد.

Border

خط دورتا دور یک جعبه. قبلا توضیح دادیم که یک خط دور تا دور المان ایجاد می کند (با ضخامت و رنگی که شما تعیین می کنید)

Margin

حاشیه خارجی یک جعبه. با خصوصیت margin می توانیم فاصله جعبه از المان ها و جعبه های اطرافش از چهار جهت مشخص کنیم. رنگ پس زمینه این فضا شفاف است پس قابل دیدن نمی باشد. مقدار دهی margin هم دقیقا مشابه padding می باشد یعنی به چهار روش می توانیم مقدار بدهیم.

نکته: تا اینجا همه واحدها را به صورت مقداری و از نوع px در نظر گرفتیم در جلسه ای جداگانه در مورد واحدهایی که در CSS استفاده می شود بحث خواهیم کرد.

مثال برای Box Model

خب وقتشه یک مثال بزنینم تا بهتر بفهمیم اینا چی بودند. مثال قبل را در نظر بگیرید. کدهای html را تغییر نمی دهیم . فایل css را باز می کنیم و خصوصیت padding و border و margin را برای دو المان div به روش های مختلف اضافه می کنیم:

```
body{  
background:#ffff88;  
}
```

```
.a{  
width:200px;  
height:50px;  
background:#3d4d3f;  
color:#fff;  
text-align:center;  
/* از بالا 10 پیکسل و از چپ و راست 20 پیکسل و از پایین 5 پیکسل گسترش می دهد */  
padding: 10px 20px 5px;  
/* از بالا 20 از راست 30 از پایین 30 و از چپ 20 پیکسل فاصله می گیرد */  
margin: 20px 30px 30px 20px;  
border: 5px solid #030303;  
}
```

```
.b{  
width:300px;  
height:200px;
```

```
background:#3638d7;  
color:#fff;  
text-align:center
```

```
/* از هر چهارسو 10 پیکسل گسترش می یابد */  
padding: 10px;
```

```
/* از بالا و پایین 20 پیکسل و از راست و چپ 50 پیکسل فاصله می گیرد */  
margin: 20px 50px;
```

```
border: 5px solid #030303;  
}
```

خاصیت Float در CSS و خاصیت Clear

خاصیت Float در CSS

خب این خاصیت float دقیقا چه کاری می‌کنه؟ float یعنی شناور، با این خصوصیت میتوان مکان المان را تغییر داد. وقتی به المانی float می‌دهیم یعنی المان به راست یا چپ هول داده شود. (فقط در راستای افقی و نسبت به مرورگر یا المان والد). این المان می‌تواند مقادیر زیر را بگیرد:

- **right**: المان به سمت راست شناور می‌شود و محتوای بعد آن کنار لبه چپ المان قرار می‌گیرند.
- **left**: المان به سمت چپ شناور می‌شود و محتوای بعد آن کنار لبه راست المان قرار می‌گیرند.
- **none**: حالت پیشفرض می‌باشد انگار float رو ننوشته باشیم.

در ادامه یک مثال براتون می‌زنم و از خصوصیت‌ها و تگ‌هایی که در جلسات قبل یاد گرفتیم هم استفاده می‌کنم تا هم یادآوری بشه هم مفاهیم رو بهتر درک کنیم.

```
1 <!DOCTYPE html>
2 <html dir="rtl" lang="fa">
3 <head>
4 <meta charset="utf-8">
5 <style>
6 *{
7 margin:0;
8 padding:0;
9 border:0;
10 box-sizing: border-box;
11 -webkit-box-sizing: border-box;
12 -moz-box-sizing: border-box;
13 -o-box-sizing: border-box;
14 -ms-box-sizing: border-box;
15 }
16 ul{
17 width:700px;
18 height:400px;
19 background:#025e8c;
20 margin:10px auto;
21 }
22 img{
23 width:100px;
24 height:100px;
25 border:2px solid #039;
26 margin:10px;
27 }
28 li{
29 list-style:none;
30 <!-- float:right; -->
31 }
32 </style>
33 </head>
34 <body>
35 <ul>
36 <li></li>
37 <li></li>
38 <li></li>
39 </ul>
40 </body>
41 </html>
```

توضیح کدهای HTML:

از تگ `ul` برای ایجاد یک لیست غیرشمارشی استفاده شده است سپس داخل هر تگ `li` یک تگ `img` استفاده کردیم. تصاویر را در پوشه `ai` به نام `img` قرار دادیم و مسیر تصاویر را به صفت `src` در تگ `img` دادیم تا تصاویر را نمایش دهد.

توضیح کدهای CSS:

در این مثال استایل دهی به روش `Internal` انجام شده تا مباحث یادآوری بشه.

- اول اینکه با استفاده از سلکتور `*` مقدار `padding` و `margin` و `border` تمام عناصر رو صفر کردیم برای اینکه مرورگر هیچ مقدار پیش فرضی برای این خصوصیات در نظر نگیرد. `box-sizing` رو `border-box` دادیم تا `padding` و `border` روی عرض و ارتفاع المان تأثیر نگذارد. (جلسه یازدهم کامل توضیح دادیم)
- به تگ `ul` عرض و ارتفاع و رنگ پس زمینه دادیم و `margin` را `auto` دادیم تا محدوده المان `ul` وسط قرار گیرد.
- به تگ `img` عرض و ارتفاع و `border` دادیم و یک `margin` که هر تصویر از اطراف 10 پیکسل فاصله بگیرد.
- با سلکتور `li` به خصوصیت `list-style` مقدار `none` دادیم تا `bullet` مربوط به لیست غیرشمارشی حذف شود. (یک خاصیت `float` هم نوشتیم که فعلا کامنت کردیم)

خاصیت Clear در CSS

این خاصیت درست نقطه مقابل خاصیت Float است. همونطور که گفتیم با دادن خاصیت Float به المان X ، این المان شناور می شود و محتویات دیگر در اطراف المان X قرار می گیرند. با خاصیت clear در CSS می توانیم فضای اطراف المان X را خالی کنیم. چهار مقدار می گیرد:

- **right**: در سمت راست المان هیچ محتوایی قرار نگیرد.
- **left**: در سمت چپ المان هیچ محتوایی قرار نگیرد.
- **both**: در دو طرف المان هیچ محتوایی قرار نگیرد.
- **none**: حالت پیشفرض می باشد انگار clear رو ننوشته باشیم.