

فصل دوم CSS

اهداف فصل

در این فصل با بخش اول از مفاهیم و کاربرد **CSS** در توسعه‌ی وب آشنا می‌شویم. یادگیری **CSS** به ما کمک می‌کند تا صفحات وب زیبا، مرتب و مورد علاقه‌ی کاربر ایجاد کنیم.

بعد از گذراندن این فصل موارد زیر را یاد خواهید گرفت:

- ▶ جایگاه زبان **CSS** و آشنایی اولیه با آن
- ▶ انواع روش‌های استایل‌دهی به صفحه
- ▶ چند استایل پایه‌ای
- ▶ استایل‌دهی به متون
- ▶ استایل‌دهی به لینک‌ها و لیست‌ها
- ▶ مدل جعبه‌ای و حاشیه‌ای المان‌ها

تمامی مفاهیم بالا همراه با مثال‌ها و تمرین‌های متعدد برای شما توضیح داده خواهند شد. فراموش نکنید که هر جا با سؤالی مواجه شدید می‌توانید آن را در بخش سؤال خصوصی مطرح کنید.

CSS چیست؟

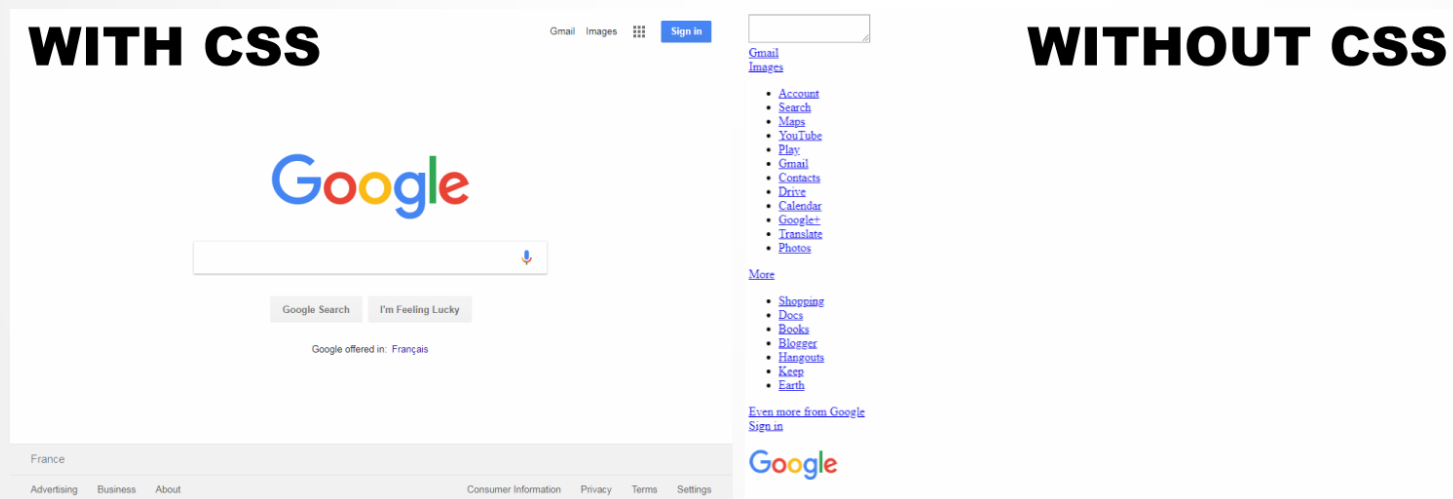
کلمه **CSS** که مخفف **Cascading Style Sheets** است، یک زبان طراحی وب است که برای تعریف چگونگی نمایش المان‌های صفحات وب به کار می‌رود. از **CSS** برای ایجاد طراحی‌های زیبا و ریسپانسیو در وب‌سایت‌ها استفاده می‌شود. در این درسنامه، با مفاهیم اصلی **CSS** و تفاوت‌های آن با **HTML** و اهمیت استفاده از آن آشنا خواهیم شد.

چرا از CSS استفاده می‌کنیم؟

- ▶ کنترل دقیق بر ظاهر وب‌سایت: **CSS** امکانات گسترده‌ای برای تنظیم رنگ‌ها، فونت‌ها، فاصله‌ها، و دیگر جنبه‌های ظاهری وب‌سایت فراهم می‌کند که باعث زیبایی سایت ما می‌شود.
- ▶ ریسپانسیو شدن و سازگاری: با استفاده از **CSS** می‌توانیم طرح‌هایی ایجاد کنیم که در دستگاه‌های مختلف، اعم از تلفن‌های هوشمند، تبلت‌ها، و کامپیوترها، به خوبی نمایش داده شوند.

تفاوت HTML و CSS چیست؟

- ▶ زبان **HTML**: زبانی است که برای ساختار و ایجاد محتوای صفحات وب به کار می‌رود. این زبان به ما اجازه می‌دهد عناوین، پاراگراف‌ها، لینک‌ها، تصاویر و سایر المان‌های محتوایی را تعریف کنیم.
- ▶ زبان **CSS**: برای تعیین استایل و طراحی این المان‌ها استفاده می‌شود. به عبارت دیگر، **HTML** مشخص می‌کند چه محتوایی نمایش داده شود، در حالی که **CSS** نحوه نمایش این محتوا را تعیین می‌کند.



آیا CSS یک زبان برنامه‌نویسی است؟

خیر، CSS به طور کلی یک زبان برنامه‌نویسی به حساب نمی‌آید، چرا که شامل منطق برنامه‌نویسی مانند حلقه‌ها یا شرط‌ها نمی‌شود. CSS یک زبان طراحی وب است که برای تعیین ظاهر و فرمت صفحات وب استفاده می‌شود.

جمع‌بندی

زبان CSS یک ابزار قدرتمند برای طراحان و توسعه‌دهندگان وب است که امکان ایجاد طراحی‌های جذاب و ریسپانسیو را فراهم می‌کند. با یادگیری و به کارگیری CSS می‌توانید تجربه کاربری بهتری را برای بازدیدکنندگان وبسایت خود ایجاد کنید. در درسنامه‌های بعدی، به شیوه‌های پیشرفته‌تر استفاده از CSS و تکنیک‌های طراحی وب می‌پردازیم.

استایل‌های Inline

در این درسنامه، با مفهوم *Inline Style* در *HTML* و *CSS* آشنا می‌شویم. ابتدا با مفهوم استایل‌دهی و انواع آن آشنا می‌شویم و سپس به بررسی *Inline Style* و نحوه استفاده از آن با چند مثال می‌پردازیم. در نهایت، تعدادی از خصوصیت‌هایی که در مثال‌ها استفاده شده‌اند را بررسی خواهیم کرد تا دانش شما در این زمینه تقویت شود.

انواع استایل‌دهی

در *CSS*، سه روش اصلی برای اعمال استایل‌ها وجود دارد:

- ▶ **Inline Styles:** استایل‌دهی مستقیم به تگ‌ها در *HTML* از طریق ویژگی `style`.
- ▶ **Internal Stylesheets:** قرار دادن استایل‌ها درون یک تگ `<style>` در یک سربرگ (`<head>`) سند *HTML*.
- ▶ **External Stylesheets:** استفاده از فایل‌های *CSS* خارجی که با استفاده از تگ `<link>` به *HTML* متصل می‌شوند.

Inline Styles

استفاده از *Inline Styles* ساده‌ترین روش برای اعمال استایل بر روی یک عنصر خاص است. برای این کار، کافی است خصوصیت‌های *CSS* را مستقیماً در تگ *HTML* و به وسیله ویژگی **style** قرار دهیم. این روش برای اعمال استایل‌های سریع و تست طراحی‌ها مفید است اما برای طراحی‌های بزرگ و پیچیده توصیه نمی‌شود زیرا مدیریت استایل‌ها را دشوار می‌کند. به دلیل سادگی این نوع استایل‌دهی در این درسنامه به توضیح این روش می‌پردازیم و در درسنامه‌های بعدی با دیگر روش‌ها آشنا می‌شوید.

مثال

برای مثال، فرض کنید می‌خواهیم برای رنگ متن یک پاراگراف را قرمز و اندازه فونت آن را ۲۰ و همچنین پاراگراف دیگر را به آبی و سایز ۱۵ اعمال کنیم:

```
<p style="color: ■red; font-size: 20px;">این یک نوشته آزمایشی است.</p>  
<p style="color: ■blue; font-size: 15px;">این نیز یک نوشته آزمایشی است.</p>
```

.این یک نوشته آزمایشی است.
.این نیز یک نوشته آزمایشی است.

خصوصیت‌های بالا

color: تعیین رنگ متن ▶

font-size: تعیین اندازه فونت ▶

نتیجه‌گیری

استفاده از *Inline Styles* برای تغییرات سریع و آزمایشی مناسب است اما برای پروژه‌های بزرگ و حرفه‌ای، بهتر است از روش‌های *Internal* یا *External Stylesheet* استفاده کنید تا کدتان سازمان‌یافته‌تر و قابل نگهداری‌تر باشد.

روش‌های دیگر استایل‌دهی

در این درسنامه به دو روش مهم و پرکاربرد در استایل‌دهی وب‌سایت‌ها با استفاده از **CSS** می‌پردازیم. **Internal Stylesheets** و **External Stylesheets**. این دو روش رویکردهای متفاوتی برای جدا کردن طراحی از محتوا ارائه می‌دهند و به توسعه‌دهندگان اجازه می‌دهند که سایت‌های خود را به طور موثرتری طراحی و مدیریت کنند.

Internal Stylesheets

استفاده از **Internal Stylesheet** به این معنی است که کدهای **CSS** را درون همان فایل **HTML**، معمولاً در بخش **<head>** قرار می‌دهیم. این روش برای صفحات وب کوچکتر که نیازمند یک فایل **CSS** جداگانه نیستند، مفید است. با این حال، برای پروژه‌های بزرگتر که نیازمند سازماندهی و نگهداری کدها هستند، استفاده از این روش توصیه نمی‌شود.

نحوه استفاده

برای استفاده از Internal Stylesheets، باید کدهای CSS خود را درون تگ `<style>` و در بخش `<head>` فایل *HTML* قرار دهید:

```
5  <head>
6    <style>
7      body {
8        background-color: rgb(27, 166, 212);
9      }
10
11     p {
12       color: white;
13       font-size: 20px;
14     }
15   </style>
16 </head>
```

در این مثال، رنگ پس‌زمینه بدنه صفحه و رنگ و فاصله پاراگراف‌ها تعریف شده است.

لینک داخلی

External Stylesheets

External Stylesheets امکان جداسازی کامل استایل‌ها از *HTML* را فراهم می‌کند. استایل‌ها در فایل‌های جداگانه با پسوند *CSS* نوشته می‌شوند و سپس با استفاده از تگ `<link>` به فایل *HTML* متصل می‌شوند. این روش برای توسعه وبسایت‌های بزرگ و پیچیده ایده‌آل است، زیرا به مدیریت بهتر و سازماندهی کدهای استایل کمک می‌کند.

نحوه استفاده

برای اتصال یک *External Stylesheet* به *HTML*، اول یک فایل با فرمت *CSS* درست کنید، برای مثال `styles.css` و سپس در تگ `<head>` خود یک تگ `<link>` و `property href` می‌توانید به فایل *CSS* خود وصل شوید، دقت داشته باشید که حتما نیاز است تا `rel` را برابر با `stylesheet` قرار دهید.

```
5 <head>
6   <link rel="stylesheet" href="styles.css">
7 </head>
```

و در فایل styles.css، استایل‌های خود را می‌نویسید:

```
1 body {
2   background-color: #f0f0f0;
3 }
4
5 h1 {
6   color: darkgreen;
7 }
```

لینک خارجی

مزایای استفاده از External Stylesheets

1. نگهداری و به روزرسانی آسان: تغییرات در یک فایل می تواند تأثیری بر تمام صفحات وبسایت داشته باشد.
2. کاهش تکرار: جلوگیری از تکرار کدهای استایل در صفحات مختلف.
3. بهبود عملکرد: مرورگرها می توانند فایل های CSS خارجی را کش کنند، که به سرعت بارگذاری صفحه کمک می کند.

نتیجه گیری

انتخاب بین *Internal* و *External Stylesheets* بستگی به اندازه و پیچیدگی پروژه شما دارد. برای پروژه های کوچک، *Internal Stylesheets* می تواند کارآمد باشد، اما برای توسعه پروژه های بزرگتر، استفاده از *External Stylesheets* به شما اجازه می دهد که استایل های خود را به طور موثرتری مدیریت و سازماندهی کنید.

استایل‌های اولیه

در این درسنامه به بررسی چند مورد از اصول پایه‌ای **CSS** می‌پردازیم، از جمله استایل‌دهی اولیه و انتخاب‌کننده‌ها (**Selectors**) که اجازه می‌دهند ما به طور موثری المان‌های **HTML** را برای اعمال استایل انتخاب کنیم. ابتدا با مفهوم **CSS** و نحوه اضافه کردن آن به صفحات وب آشنا می‌شویم، سپس به سراغ انتخاب‌کننده‌ها و چندین خاصیت مهم و پرکاربرد می‌رویم.

انتخاب کننده‌ها در CSS

انتخاب کننده‌ها (*Selectors*) امکان مشخص کردن المان‌هایی را که می‌خواهیم استایل دهی کنیم، فراهم می‌آورند. انواع مختلفی از انتخاب کننده‌ها وجود دارد:

► **انتخاب کننده‌های عنصر:** برای انتخاب تمام المان‌هایی با نوع مشخص، اسم تگ را نوشته و سپس استایل مدنظر را به آن می‌دهیم.

```
9   p {  
10  |   color: green;  
11  }
```

► **انتخاب کننده‌های کلاس:** با استفاده از نقطه (.) قبل از نام کلاس آن را انتخاب و استایل مدنظر را به آن می‌دهیم.

```
13  .box {  
14  |   border: 2px solid black;  
15  }
```

انتخاب‌کننده‌های **id**: با استفاده از علامت # قبل از آیدی آن را انتخاب و استایل مدنظر را به آن می‌دهیم. ►

```
17  #unique {  
18  |  font-weight: bold;  
19  }
```

این یک متن آزمایشی است.

تفاوت **class** و **id**: تفاوت **class** و **id** در این است که شما فقط یک‌بار می‌توانید از هر **id** در کدتان استفاده کنید ولی برعکس از هر کلاس به هر مقدار که بخواهید می‌توانید استفاده کنید.

برای یادگیری بهتر به مثال زیر توجه کنید:

```
1 #unique {
2   background: #4183c4;
3   color: #eee;
4   padding: 8px 10px;
5   border-radius: 5px;
6   margin-bottom: 10px;
7 }
8
9 .Multiple {
10  padding: 7px 23px;
11  background: #eee;
12  border-radius: 5px;
13  margin: 3px;
14  color: #4183c4;
15 }
```

```
9 <body dir="rtl">
10   <!-- استفاده از آی دی برای عنصر خاص -->
11   <h1 id="unique">رنگ مورد علاقه شما</h1>
12   <!-- استفاده از کلاسی برای موارد عمومی -->
13   <h2 class="Multiple">قرمز</h2>
14   <h2 class="Multiple">آبی</h2>
15   <h2 class="Multiple">سبز</h2>
16 </body>
```



توجه داشته باشید که کاربرد آیدی فقط در استایل‌دهی نیست و در کارهای دیگری مانند انتخاب المان در JS بسیار کاربردی است.

خاصیت‌های پرکاربرد در CSS

چندین خاصیت پرکاربرد در CSS وجود دارد که به طور مکرر استفاده می‌شوند:

► رنگ: **color** برای تعیین رنگ متن و **background-color** برای رنگ پس‌زمینه.

```
17 .text-color {  
18   color: purple;  
19 }  
20  
21 .background {  
22   background-color: lightgrey;  
23 }
```

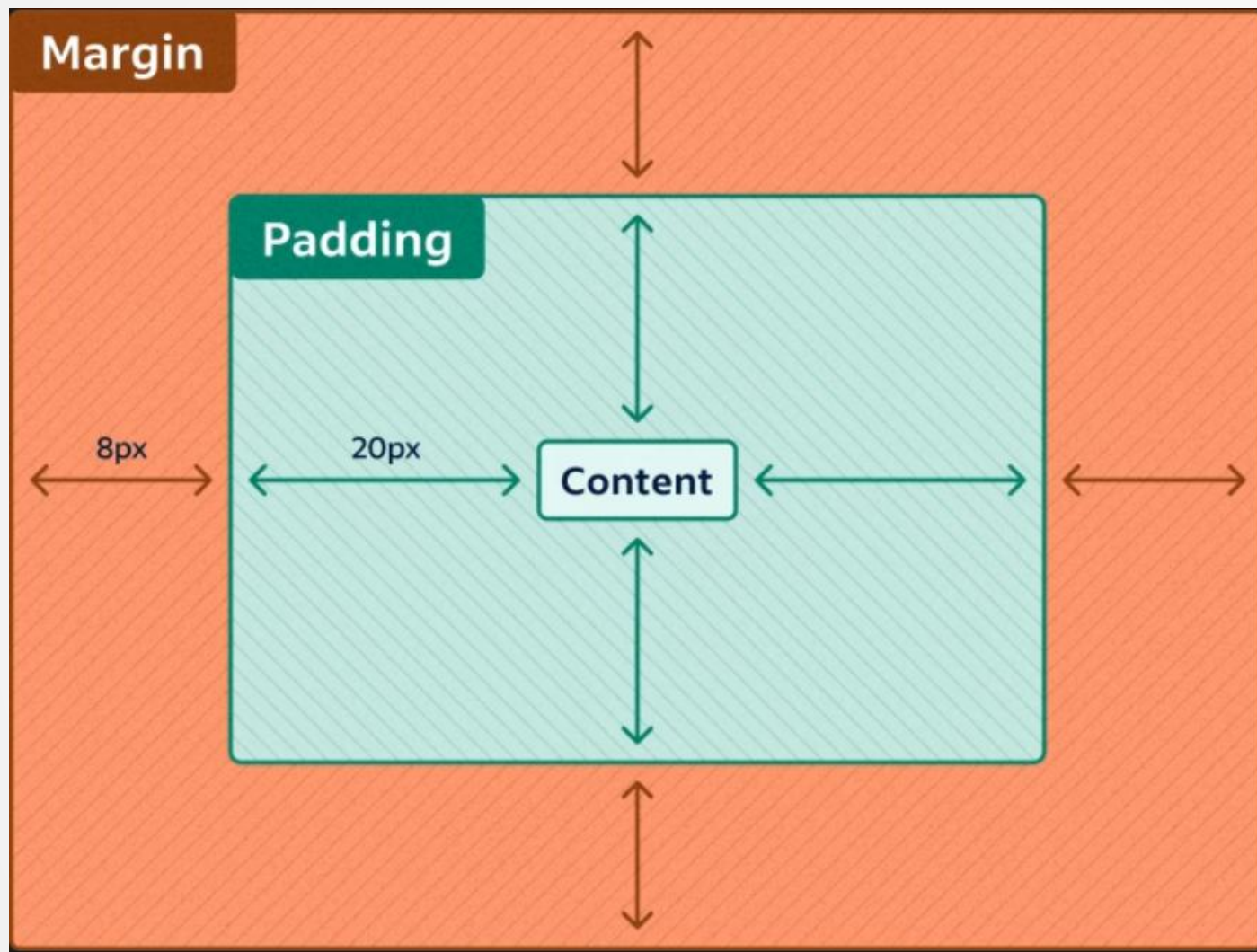
► فونت: **font-family** برای تعیین فونت و **font-size** برای اندازه فونت.

```
25 .font-style {  
26   font-family: Arial, sans-serif;  
27   font-size: 20px;  
28 }
```

▶ فاصله درونی و خارجی: **margin** برای ایجاد فاصله از المان‌های اطراف و **padding** برای فاصله درونی نسبت به محتوای داخلی المان.

```
30 .box {  
31   margin: 10px;  
32   padding: 20px;  
33   border: 1px solid black;  
34 }
```

این یک متن آزمایشی است.



► عرض و ارتفاع: **width** و **height** برای تعیین ابعاد المان‌ها. این خاصیت‌ها به شما امکان می‌دهند اندازه دقیق المان‌هایی مانند تصاویر، جعبه‌ها و دیگر المان‌های صفحه وب را مشخص کنید.

```
36  .box {
37    width: 100px;
38    height: 100px;
39  }
```

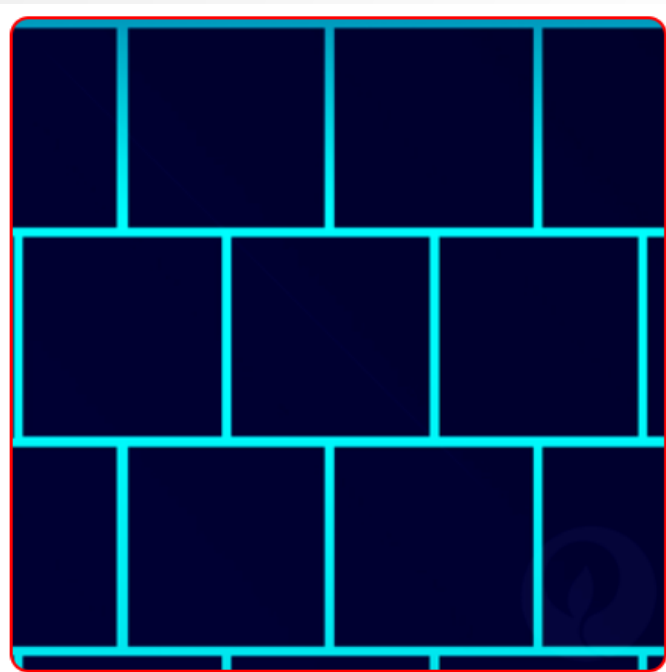
► **border**: برای تعیین حاشیه دور المان‌ها. این خاصیت شامل ضخامت، سبک و رنگ حاشیه است.

```
41  .box {
42    border: 2px solid red; /* ضخامت 2px، سبک solid و رنگ قرمز */
43  }
```

► گوشه‌های گرد: **border-radius** برای ایجاد گوشه‌های گرد در المان‌ها. با استفاده از این خاصیت می‌توانید ظاهری مدرن‌تر به طراحی‌های خود بخشید.

```
45  .rounded-corners {
46    border-radius: 10px; /* گوشه‌های گرد با شعاع 10 پیکسل */
47  }
```

```
49 .background-img {  
50   background-image: url("background.jpg");  
51 }
```



تصویر پس‌زمینه: background-image ►

برای قرار دادن یک تصویر به عنوان پس‌زمینه‌ی المان. این خاصیت امکان استفاده از تصاویر گرافیکی را برای زیباسازی صفحه وب فراهم می‌آورد.


```
52 .text-left {  
53 |   text-align: left;  
54 | }  
55  
56 .text-right {  
57 |   text-align: right;  
58 | }  
59  
60 .text-center {  
61 |   text-align: center;  
62 | }  
63  
64 .text-justify {  
65 |   text-align: justify;  
66 | }
```

ترازبندی متن: **text-align** برای تعیین
چگونگی ترازبندی متن‌ها درون المان‌ها. این خاصیت
امکان تراز کردن متن به چپ، راست، وسط یا
مرتب‌شده را می‌دهد. ▶

چپ چین

راست چین

وسط چین

مرتب شده

رنگ‌ها

برای تغییر رنگ محتوای روی صفحه می‌توان از **color** و **background-color** استفاده کرد. با استفاده از **color** می‌توان رنگ متن را تغییر داد و با **background-color** می‌توان رنگ پس‌زمینه‌ی المان را عوض کرد.

به مثال زیر توجه کنید:

```
1 h1 {  
2   color: red;  
3   background-color: lightpink;  
4 }
```



با کد بالا، رنگ متن المان **h1** به قرمز (**red**) تغییر می‌کند و رنگ پس‌زمینه‌ی آن نیز صورتی روشن (**lightpink**) می‌شود.

برای مقداردهی به ویژگی‌های مربوط به رنگ، می‌توان از چند روش مختلف استفاده کرد که در این بخش به بررسی آن‌ها می‌پردازیم.




رنگ‌های نام‌دار یا Named Colors

همان‌طور که تا اینجا در مثال‌های دوره دیدید، برای مشخص کردن مقدار ویژگی‌های مربوط به رنگ از نام یک رنگ استفاده کردیم. مقادیری مثل **green, yellow, blue** و... در کد بالا هم مقادیر **red** و **lightpink** رنگ‌های نام‌داری بودند که از آنها استفاده کردیم.

لیست کاملی از رنگ‌های نام‌دار را می‌توانید در [این صفحه](#) مشاهده کنید.

همان‌طور که می‌توان در این لیست دید، تعداد این رنگ‌ها محدود است و تعداد رنگ‌های قابل تولید بسیار بیش‌تر از این‌هاست. برای اینکه بتوانیم همه رنگ‌ها را بسازیم می‌توانیم از ۳ روش استفاده کنیم که در ادامه با آنها آشنا می‌شوید.

به مثال زیر توجه کنید:

```
12 .box1 {  
13   background-color:  rgb(255, 99, 71);  
14 }  
15  
16 .box2 {  
17   background-color:  #ff6347;  
18 }  
19  
20 .box3 {  
21   background-color:  hsl(9, 100%, 64%);  
22 }
```

رنگ استفاده شده `rgb(255, 99, 71)`

رنگ استفاده شده `#ff6347`

رنگ استفاده شده `hsl(9, 100%, 64%)`



رشد RGB

در مدلی که بخواهیم از *RGB* استفاده کنیم، به کمک ترکیب سه رنگ قرمز (*red*)، سبز (*green*) و آبی (*blue*) رنگ موردنظر را ایجاد می‌کنیم. شدت هر رنگ بین ۰ تا ۲۵۵ می‌باشد. می‌توانیم به کمک فرمول زیر **rgb** را بهتر بشناسیم:

rgb(red, green, blue)

به عنوان مثال اگر هر سه را به صفر کاهش دهیم، یعنی **rgb(0, 0, 0)**، به رنگ مشکی می‌رسیم و همچنین اگر هر سه را به بیشترین مقدار خود، یعنی **rgb(255, 255, 255)** برسانیم به رنگ سفید می‌رسیم.

به مثال زیر دقت کنید:

```
6  .box1 {
7    background-color:  rgb(0, 0, 0);
8  }
9
10 .box2 {
11  background-color:  rgb(255, 255, 255);
12 }
```

`rgb(0, 0, 0)`

`rgb(255, 255, 255)`

در کد زیر با استفاده از روش *RGB* به المان **h2** رنگ متن و پس‌زمینه داده‌ایم:

```
6 h2 {  
7   color: ■ rgb(44, 98, 222);  
8   background-color: ■ rgb(149, 240, 229);  
9 }
```

متن دوم

اضافه کردن شفافیت رنگ با RGBA

با اضافه شدن یک مقدار دیگر به **rgb** می‌توانیم میزان شفافیت یا **Opacity** را در رنگ مورد نظرمان تغییر دهیم که به این سیستم **rgba** می‌گویند. این مقدار عددی بین ۰ و ۱ می‌باشد که درصد شفافیت را تعیین می‌کند و به آن *آلفا* گفته می‌شود.

واضح است که اگر این مقدار برابر ۰ قرار بگیرد همانند این است که المان مورد نظر هیچ رنگی ندارد و اصطلاحاً **transparent** می‌شود و در صورت برابر قرار گرفتن با ۱ نیز تفاوتی با حالتی که این مقدار را قرار ندهیم (یعنی **rgb** باشد) نمی‌کند. می‌توانیم به کمک فرمول زیر **rgba** را بهتر بشناسیم:

rgba(red, green, blue, alpha)

تصویر روبه‌رو نحوه‌ی نمایش چنین رنگ‌هایی را برای شما نمایش می‌دهد:

```
11 .box1 {  
12   background-color: □ rgba(255, 99, 71, 0);  
13 }  
14  
15 .box2 {  
16   background-color: ■ rgba(255, 99, 71, 0.4);  
17 }  
18  
19 .box3 {  
20   background-color: ■ rgba(255, 99, 71, 0.8);  
21 }
```



`rgba(255, 99, 71, 0)`

`rgba(255, 99, 71, 0.4)`

`rgba(255, 99, 71, 0.8)`

روش HEX

همان طور که برای حالت **RGB** داشتیم، می توانیم به کمک مقادیر هگزادسیمال (مبنای ۱۶) هم رنگ مورد نظر را ایجاد کنیم، به این صورت که دیگر لازم نیست عبارت **rgb** نوشته شود و تنها با گذاشتن علامت **#** و سپس قراردادن مقادیر مورد نظر برای سه رنگ قرمز و سبز و آبی در مبنای **hexadecimal** می توانیم همان کار را انجام دهیم. به هر یک از سه رنگ آبی، قرمز و سبز، دو رقم اختصاص می گیرد که از ۰۰ شروع شده و به **ff** ختم می شود.




برای مثال رنگ قرمز به این صورت نشان داده می شود که مقدار قرمز حداکثر و دو رنگ دیگر حداقل است یعنی

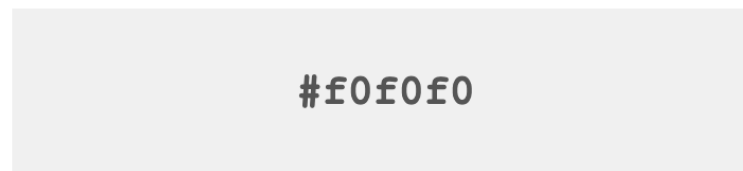
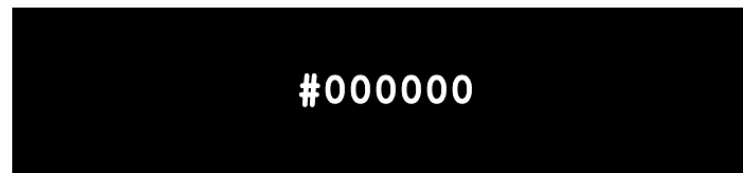
#ff0000

می توانیم به کمک فرمول زیر **HEX** را بهتر بشناسیم:

#rrggbb

نمونه‌هایی از این رنگ‌ها را می‌توانید در تصویر زیر ببینید:

```
11 .box1 {  
12 | background-color:  #000000;  
13 | }  
14  
15 .box2 {  
16 | background-color:  #169ecf;  
17 | }  
18  
19 .box3 {  
20 | background-color:  #f0f0f0;  
21 | }
```



در کد زیر با استفاده از روش *HEX* به المان **h3** رنگ متن و پس‌زمینه داده‌ایم:

```
23 h3 {  
24   color: #056636;  
25   background-color: #76dba9;  
26 }
```

متن سوم


اضافه کردن شفافیت رنگ با HEXA

برای افزودن شفافیت (آلفا)، ساختار به #RRGGBBAA تغییر می کند که:


AA نشان دهنده مقدار شفافیت در دامنه 00 تا FF است. مقدار 00 نشان دهنده کاملاً شفاف (شفافیت ۰٪) و مقدار FF نشان دهنده کاملاً مات (شفافیت ۱۰۰٪) است.

به مثال های زیر دقت کنید:

1. رنگ قرمز با شفافیت ۵۰٪:

```
26 background-color:  #FF000080;
```

2. رنگ آبی با شفافیت ۲۰٪:

```
26 background-color:  #0000FF33;
```

3. رنگ سبز با شفافیت ۷۵٪:

```
26 background-color:  #00ff007f;
```



#00ff0080

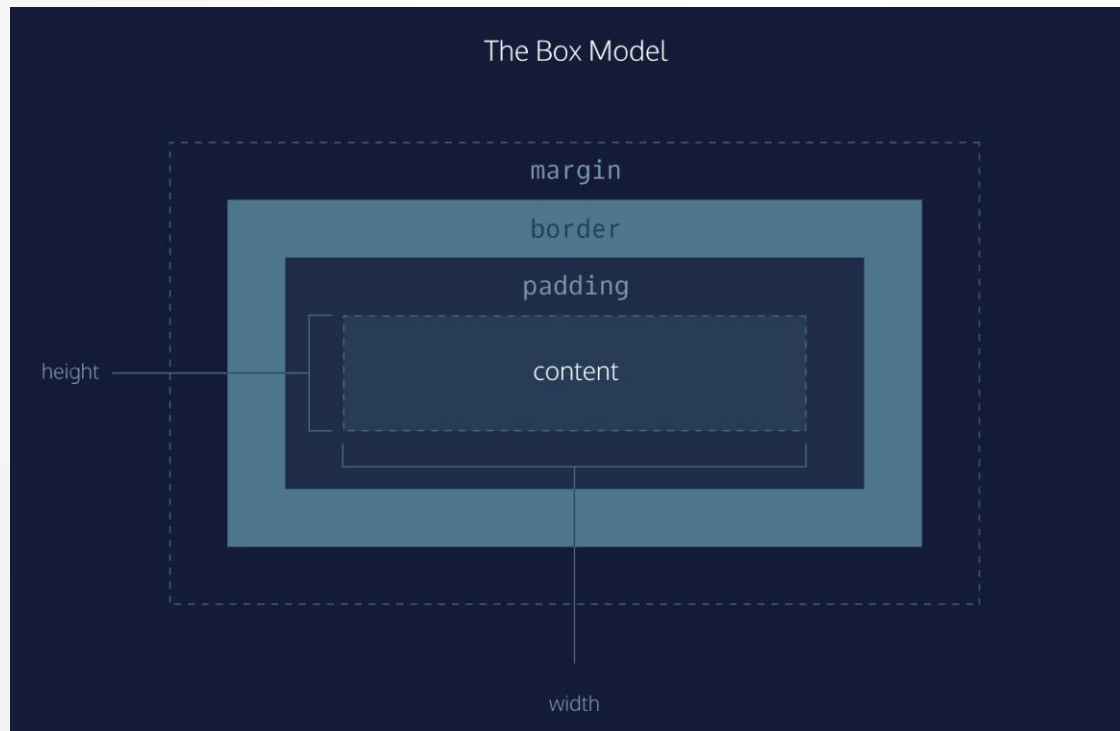
#00ff0033

#00ff007f

برای انتخاب رنگ و استفاده از آن در کد *CSS* می‌توانید *color picker* را جستجو کنید و از نتایج آن استفاده کنید. عموماً *color picker*ها تمامی حالات ساخت یک رنگ از جمله *RGB* و *HEX* را در اختیارتان قرار می‌دهند.

مدل جعبه‌ای

برای این که بفهمیم یک المان چگونه‌ی فضایی از صفحه را به خود اختصاص می‌دهد باید با مدل جعبه‌ای یا همان *Box Model* آشنا شویم. به تصویر زیر دقت کنید:



به صورت پیش فرض برای نمایش یک المان، چهار جعبه یا **Box** تشکیل می شود:

▶ جعبه‌ی محتوا یا **Content Box**: فضایی است که در آن محتوای یک المان نمایش داده می شود. با **property** های **width** و **height** می توان اندازه‌ی این جعبه را تغییر داد. (با این **property** ها در درسنامه‌ی قبلی آشنا شدید.)

▶ جعبه‌ی **Padding** یا **Padding Box**: فضای بین **Content Box** و حاشیه‌ی خارجی المان را تشکیل می دهد. با **property** های مربوط به **padding** می توان این فضای این جعبه را تغییر داد.

▶ جعبه‌ی حاشیه یا **Border Box**: جعبه‌ایست که دو جعبه‌ی قبلی را محدود می کند و آن‌ها را شامل می شود. با **property** های مربوط به **border** می توان این جعبه را تغییر داد. (با این **property** ها در درسنامه‌ی بعدی آشنا می شوید.)

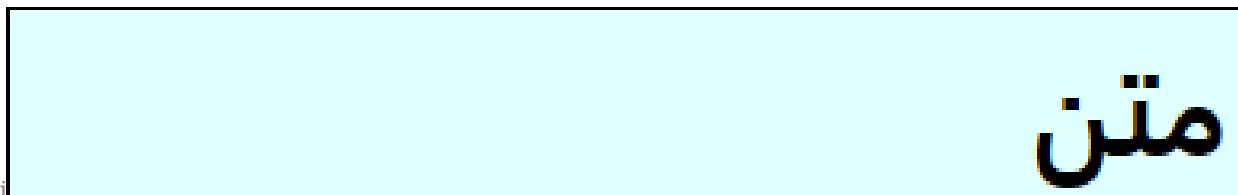
▶ جعبه‌ی **Margin** یا **Margin Box**: فضای بین حاشیه‌ی المان و سایر المان‌ها را اشغال می کند. با **property** های مربوط به **margin** می توان فضای این جعبه را تغییر داد.

در این درسنامه به بررسی دقیق **property** های مربوط به **Padding** و **Margin** خواهیم پرداخت.

جعبه‌ی Padding و تغییر آن

همان‌طور که گفتیم و روی تصویر هم مشاهده کردید، فضای بین محتوای یک المان و حاشیه‌ی المان را **Padding** می‌گوییم. در کد زیر می‌توانید نحوه‌ی استفاده از **property**های مربوط به **padding** را مشاهده کنید:

```
1  h1 {  
2    width: 300px;  
3    background-color: lightcyan;  
4    border: 1px solid black;  
5    padding-left: 4px;  
6    padding-right: 4px;  
7    padding-top: 2px;  
8    padding-bottom: 2px;  
9  }
```

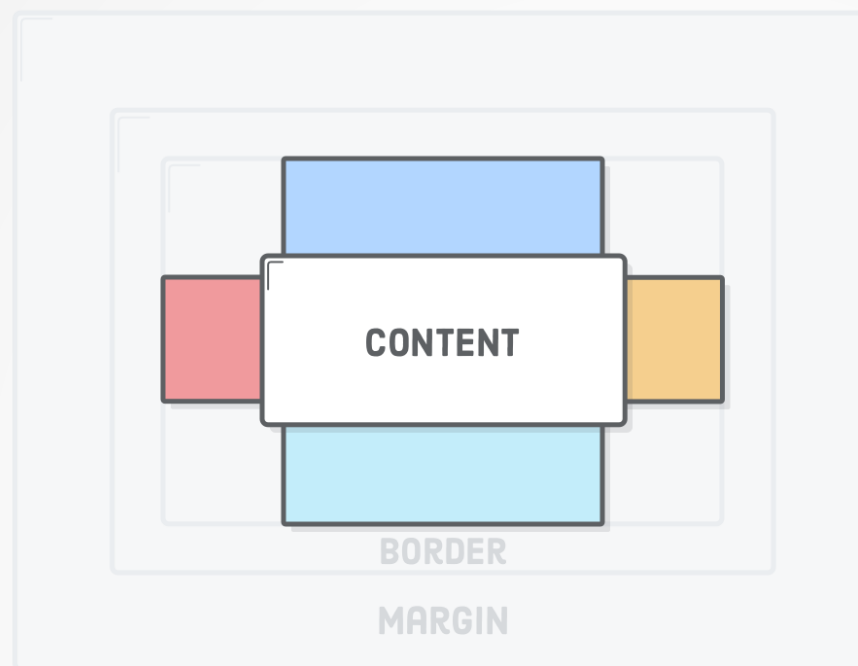


برای مشاهده‌ی بهتر اثرات *padding* های *property*، به این المان **width** دلخواه، یک **border** (که مفصل با آن در درسنامه‌ی بعد آشنا می‌شوید) و رنگ پس زمینه داده شده است.

با استفاده از **padding-left**، **padding-right**، **padding-top** و **padding-bottom** می‌توان در چهار جهت مختلف مقدار فضایی که جعبه‌ی *Padding* اشغال می‌کند را تغییر داد. همچنین می‌توان به صورت زیر نیز این مقادیر را مشخص کرد:

```
10 padding: 2px 4px 2px 4px;
```

با روش بالا به ترتیب مقادیر **padding** در جهت‌های بالا، راست، پایین و چپ به المان داده می‌شود:



PADDING : TOP RIGHT BOTTOM LEFT

همچنین اگر یک مقدار به **padding** داده شود، این مقدار به هر چهار جهت نیز داده می‌شود، مانند کد زیر:

```
10 padding: 4px;
```

جعبه‌ی Margin و تغییر آن

جعبه‌ی *Margin* مربوط به فضای بین حاشیه‌ی یک المان و دیگر المان‌هاست. پس دقت کنید که برای مشاهده‌ی اثرات این جعبه، باید فاصله‌ی المان از سایر المان‌ها را در نظر گرفت. به مثال زیر دقت کنید:

```
10 <h1>مدل جعبه‌ای</h1>
11 <p>
12   یک متن کوتاه
13 </p>
14 <h2>یک تیتر دیگر</h2>
```

```
1 p {
2   border: 1px solid black;
3   margin-left: 24px;
4   margin-right: 24px;
5   margin-top: 60px;
6   margin-bottom: 60px;
7 }
```

مدل جعبه‌ای

یک متن کوتاه

یک تیتر دیگر

در مثال بالا فاصله‌ی حاشیه‌ی المان **p** از سایر المان‌ها را با استفاده از **margin-left**، **margin-right**، **margin-top** و **margin-bottom** از چهار طرف تعیین کرده‌ایم. دقت کنید که به طور پیش فرض نیز میان المان‌ها فاصله‌ای ایجاد می‌شود.

همانند **padding**، با استفاده از **margin** می‌توان این چهار مقدار را تنها به یک **property** داد و آن‌ها را خلاصه کرد.

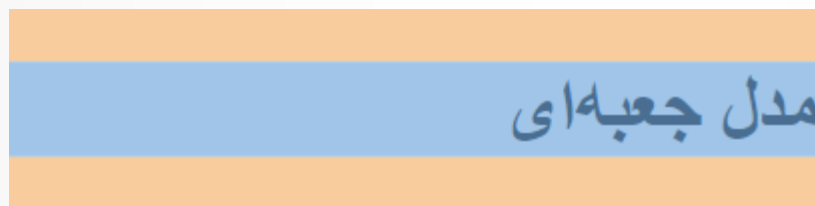
```
8 margin: 60px 24px 60px 24px;
```

با روش بالا به ترتیب مقادیر **margin** در جهت‌های بالا، راست، پایین و چپ به المان داده می‌شود. همچنین اگر یک مقدار به **margin** داده شود، این مقدار به هر چهار جهت نیز داده می‌شود، مانند کد زیر:

```
8 margin: 24px;
```

اعمال استایل روی همه المان ها

گاهی پیش می آید که یک استایل خاص را به همه المان ها نسبت بدهید مثلا زمانی که المانی روی صفحه می گذارید ممکن است به صورت پیش فرض یک استایل مانند **border** یا **padding** و یا **margin** داشته باشد و بخواهید این استایل را حذف و یا تغییر دهید.



برای انجام این کار به شکل زیر عمل می کنیم:

```
1  * {  
2    padding: 0px;  
3    margin: 0px;  
4    border: none;  
5  }
```

با استفاده از علامت * استایل مشترک همه اعمال ها را اعمال می کنید و سپس در ادامه استایل مورد نظر خود را اعمال می کنید.

مدل جعبه ای

- ▶ همچنین به این نکته توجه داشته باشید که وقتی مقدار **padding** یا **margin** را به صورت **10px 30px** وارد کنید مرورگر به صورت خودکار مقدار اول را برای بالا و پایین و مقدار دوم را برای چپ و راست برمی دارد.
- ▶ و یا مقدار **padding** یا **margin** را به صورت **20px 10px 30px** وارد کنید مرورگر به صورت خودکار مقدار آخر را از نظیر آن یعنی **10px** برمی دارد.

حاشیه‌ی المان

همان‌طور که در درسنامه‌ی قبل به آن اشاره کردیم، حاشیه یا همان **Border** یک المان، فضای بین جعبه‌ی **Padding** و جعبه‌ی **Margin** آن المان است و به گونه‌ای جداکننده‌ی آن المان از سایر المان‌ها می‌باشد. برای مشخص کردن حاشیه‌ی یک المان می‌توان از سه **property** استفاده کرد: **border-width**, **border-style** و **border-color** که در این جا به بررسی هر سه می‌پردازیم.

خاصیت border-width

برای تعیین ضخامت حاشیه‌ی یک المان از **border-width** استفاده می‌کنیم، مانند کد زیر:

```
6 border-width: 4px;
```

خاصیت border-style

برای تعیین استایل حاشیه‌ی یک المان از **border-style** استفاده می‌کنیم. از مقادیری که این *property* می‌گیرد، می‌توان به **solid** (ساده)، **dotted** (نقطه‌ای) و **dashed** (خط فاصله‌ای) و ... اشاره کرد. از این مقادیر به صورت زیر می‌توان استفاده کرد:

```
6 border-style: dotted;
```

خاصیت border-color

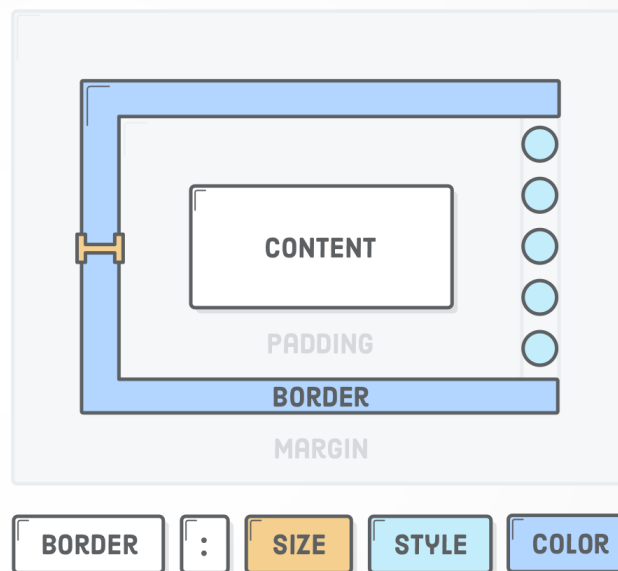
برای تعیین رنگ حاشیه‌ی یک المان از **border-color** استفاده می‌کنیم. از تمامی حالات استفاده از رنگ‌ها که در درسنامه‌ی مربوطه به آن‌ها پرداختیم می‌توان در این‌جا استفاده کرد. نمونه‌ی استفاده از این خاصیت در کد زیر قابل مشاهده است:

```
6 border-color: ■ brown;
```

استفاده از حالت کوتاه با خاصیت border

هر سه خاصیت قبلی را می‌توان در یک خاصیت با نام **border** به شکل زیر خلاصه کرد:

```
6 border: 4px solid ■ brown;
```



همچنین می‌توان با *property* های `border-left`, `border-right`, `border-top` و `border-bottom` حاشیه را در چهار جهت مختلف تعریف کرد.

خمیده کردن گوشه‌ی حاشیه با خاصیت border-radius

با استفاده از `border-radius` می‌توان شعاع گوشه‌های خمیده‌ی حاشیه‌ی یک المان را مشخص کرد. به مثال زیر توجه کنید:

```
3 border-radius: 28px;
```

به مثال زیر توجه کنید:

```
9 <body dir="rtl">
10   <div class="box box1"></div>
11   <div class="box box2"></div>
12   <div class="box box3"></div>
13   <div class="box box4"></div>
14   <div class="box box5"></div>
15   <div class="box box6"></div>
16   <div class="box box7"></div>
17   <div class="box box8"></div>
18   <div class="box box9"></div>
19   <div class="box box10"></div>
20   <div class="box box11"></div>
21   <div class="box box12"></div>
22 </body>
```

```

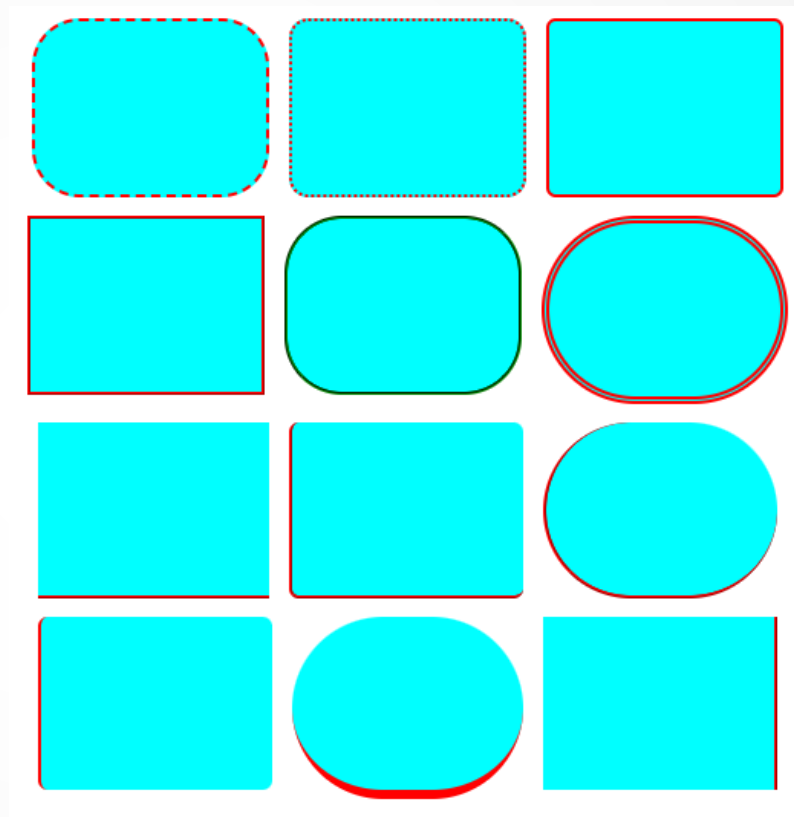
1  body {
2    display: flex;
3    justify-content: center;
4    flex-wrap: wrap;
5  }
6
7  .box {
8    width: 25%;
9    height: 90px;
10   background-color: aqua;
11   margin: 5px;
12 }
13
14 .box1 {
15   border: 2px solid red;
16   border-radius: 5px;
17 }
18
19 .box2 {
20   border: 2px dotted red;
21   border-radius: 10px;
22 }
23
24 .box3 {
25   border: 2px dashed red;
26   border-radius: 25px;
27 }
28
29 .box4 {
30   border: 4px double red;
31   border-radius: 50px;
32 }
33
34 .box5 {
35   border: 2px groove green;
36   border-radius: 30px;
37 }
38

```

```

39 .box6 {
40   border: 2px ridge red;
41   border-radius: 0px;
42 }
43
44 .box7 {
45   border-left: 2px ridge red;
46   border-bottom: 2px ridge red;
47   border-radius: 50px;
48 }
49
50 .box8 {
51   border-left: 2px ridge red;
52   border-bottom: 2px ridge red;
53   border-radius: 5px;
54 }
55
56 .box9 {
57   border-bottom: 2px ridge red;
58   border-radius: 0px;
59 }
60
61 .box10 {
62   border-right: 2px ridge red;
63   border-radius: 0px;
64 }
65
66 .box11 {
67   border-bottom: 5px solid red;
68   border-radius: 50px;
69 }
70
71 .box12 {
72   border-left: 2px solid red;
73   border-radius: 5px;
74 }

```



استایل دادن به متن‌ها

برای معرفی *property* های مربوط به استایل متن، در ابتدا سراغ استایل های مربوط به *font* متن می‌رویم.

نوع فونت با `font-family`

برای مشخص کردن نوع فونت یا همان *typeface*، از `font-family` استفاده می‌کنیم و مقدار آن را برابر با نام فونت موردنظر قرار می‌دهیم. در کد زیر نوع فونت المان‌های `h1` و `h2` به فونت **Tahoma** تغییر کرده است.

```
1 h1,  
2 h2 {  
3   font-family: "Tahoma";  
4 }
```

در مورد **font-family** باید به نکات زیر دقت کنید:

- ▶ فونتی که به عنوان مقدار این **property** مشخص شده باشد، یا باید روی سیستم کاربر نصب شده باشد یا همراه با سند **HTML** و **CSS** برای مرورگر دانلود شود. در غیر این صورت یک فونت پیش فرض به جای فونت مشخص شده نمایش داده می شود.
- ▶ مرورگر باید امکان نمایش فونت را داشته باشد. برخی فونت ها این قابلیت را ندارند.
- ▶ امکان نگذاشتن نام فونت داخل **quote** وجود دارد اما اگر نام فونت دو یا چند قسمتی باشد توصیه می شود که آن را داخل **quote** قرار داد (برای مثال 'Times New Roman').

اندازه فونت با font-size

برای تغییر اندازه فونت متن، از **font-size** استفاده می‌کنیم. برای مثال در کد زیر اندازه‌ی متن المان‌های **p** به **18px** تغییر کرده است.

```
1  p {  
2    font-size: 18px;  
3  }
```

دقت کنید که **px** به معنای پیکسل است و مشخص می‌کند که واحد این اندازه برحسب پیکسل است.

ضخامت فونت با font-weight

برای تعیین میزان **bold** بودن یک متن از **font-weight** استفاده می‌کنیم. دو مقدار اصلی این **property**، حالت **normal** (که پیش‌فرض است) و حالت **bold** (که فونت را ضخیم‌تر می‌کند) هستند. در کد زیر المان‌های **p** به صورت **bold** درآمده‌اند.

```
1  p {  
2    font-weight: bold;  
3  }
```


تراز کردن متن با `text-align`

با توجه به جهت صفحه که آن را با ویژگی `dir` تگ `html` مشخص کردیم، متن از سمت راست یا چپ المانی که در آن قرار دارد شروع می‌شود. مثلاً در کدهای این دوره از `dir="rtl"` استفاده کرده‌ایم در نتیجه متون از راست به چپ نمایش داده می‌شوند پس داخل المان نگه‌دارنده‌ی خود، سمت راست قرار می‌گیرند. برای تغییر این مکان جاگیری یا همان ترازِی متن از `property`ی به نام `text-align` استفاده می‌کنیم. در کد زیر این مکان جاگیری المان `h2` به وسط المان `parent` تغییر کرده است:

```
1  h2 {  
2    | text-align: center;  
3  }
```

از مقادیری که این `property` می‌گیرد می‌توان به `left`، `right` و `center` اشاره کرد.

شما می‌توانید ویژگی‌های زیادی را در متن تغییر دهید که در جدول زیر برخی از این ویژگی‌های مهم را مشاهده می‌فرمایید:

نام	توضیح
text-decoration	تنظیم بعضی ویژگی‌های ظاهری در متن
text-transform	تنظیم <i>Lowercase</i> ، <i>Uppercase</i> و... در متن
line-height	تنظیم فاصله بین خطوط در متن
word-spacing	تنظیم فاصله بین کلمه‌ها در متن
white-space	تنظیم فضاهاى سفید در متن
text-shadow	افزودن و تنظیم سایه‌ها در متن
text-overflow	تنظیم شیوه مدیریت متنی که از اندازه خارج شده
text-indent	تنظیم فرورفتگی‌ها در متن

سعی کنید درباره‌ی *property*‌های بالا جستجو کنید و مقادیری که می‌پذیرند را در کد خود امتحان کنید.

```

9 <body class="" dir="rtl">
10 <a href="/" class="box box1">حذف خط زیر پیش فرض لینک</a>
11 <div class="box box2">capitalize</div>
12 <div class="box box3">فاصله بین کلمات</div>
13 <div class="box box4">متن ضخیم شده</div>
14 <div class="box box5">متن دارای تو رفتگی</div>
15 </body>

```

```

1 .box {
2   padding: 5px;
3   line-height: 30px;
4 }
5
6 .box1 {
7   font-size: 14px;
8   text-decoration: none;
9 }
10
11 .box2 {
12   text-transform: uppercase;
13 }
14
15 .box3 {
16   word-spacing: 25px;
17 }
18
19 .box4 {
20   font-weight: bold;
21 }
22
23 .box5 {
24   text-indent: 20px;
25 }
26
27 div {
28   font-family: "Tahoma";
29 }

```

حذف خط زیر پیش فرض لینک

CAPITALIZE

فاصله بین کلمات

متن ضخیم شده

متن دارای تو رفتگی

اضافه کردن فونت به پروژه

اگر بخواهید که از فونت دلخواهی که جزو فونت‌های پیش فرض سیستم عامل‌ها و مرورگرها نیست استفاده کنید، باید این فونت را به نحوی به پروژه اضافه کنید تا همراه با فایل‌های **HTML** و **CSS** برای مرورگر دانلود شود. برای انجام این کار داخل فایل **CSS** از چنین کدی استفاده می‌کنید:

```
1 @font-face {  
2   font-family: "IranSans";  
3   src: url("fonts/IranSans.woff2") format("woff2"),  
4       url("fonts/IranSans.woff") format("woff");  
5 }
```

نکات کد:

- ▶ تعریف فونت جدید باید داخل بلاکی با نام **@font-face** قرار بگیرد.
- ▶ نام فونت را جلوی **font-family** قرار می‌دهید. در ادامه برای دادن نام فونت به المان‌ها از این مقدار استفاده خواهید کرد.
- ▶ آدرس فایل فونت را جلوی **src** قرار می‌دهید. ممکن است برای این که مطمئن شوید مرورگر فونت را می‌شناسد و استفاده می‌کند، از چندین فایل با فرمت‌های مختلف استفاده کنید. دقت داشته باشید که باید فرمت این فایل‌های مختلف مانند کد بالا مشخص شوند و همچنین این فایل‌ها با " , " از یکدیگر جدا می‌شوند.

استفاده از CDN برای اعمال فونت:

CDN در واقع به این معنی است که از یک لینک خارجی برای اعمال استایل فونت استفاده می کنید. اعمال فونت در HTML:

```
7 <link href="http://cdn.font-store.ir/yekan.css" rel="stylesheet" />
```

اعمال فونت در CSS: در ابتدا فایل CSS فونت را به پروژه اضافه می کنید.

```
1 @import url('http://cdn.font-store.ir/yekan.css');
```

و در نهایت با استفاده از `font-family`، فونت را به پروژه اضافه می کنید:

```
3 h1 {  
4   font-family: yekan;  
5 }
```

دانلود فایل های فونت برای اعمال فونت

ابتدا فونت را از سایت هایی مثل [فونت یاب](#) دانلود می کنید سپس فونت را در کنار فایل های پروژه قرار می دهید و در نهایت به شکل زیر آن را استفاده می کنید.

اگر فایل های پروژه به این شکل باشند:

```
├─ assets
│   └─ Vazir.ttf
├─ index.html
└─ styles.css
```

به شکل زیر عمل می کنید:

```
1  @font-face {
2    font-family: "Vazir.ttf";
3    src: url("../assets/Vazir.ttf");
4  }
```

ابتدا از پوشه فعلی به یک پوشه عقب تر می‌روید سپس وارد پوشه **assets** می‌شوید و فونت مورد نظر را انتخاب می‌کنید. همچنین اگر بیش از یک فونت داشته باشید و بخواهید همه را وارد کنید به شکل زیر عمل می‌کنید:

```
1 @font-face {  
2   font-family: "Vazir.ttf";  
3   src: url("../assets/Vazir.ttf"), url("../assets/Vazir.otf");  
4 }
```