

توضیحات

✓ اغلب المنت های HTML یک تگ باز و یک تگ بسته دارند.

تگ باز یعنی: `<h1>`

و تگ بسته یعنی: `</h1>`

✓ تنها تفاوت بین تگ باز و تگ بسته وجود اسلش (/) در تگ بسته هست، درست بعد از براکت باز تگ بسته.

تمرین

📄 محتوای تگ `h1` را تغییر بدید و داخل آن متن زیر را بنویسید:

Hello World

راهنما

🔥 متن بین تگ `h1` باید Hello World باشد.

توضیحات

- ✓ تگ **h1** یعنی Heading1 و ما از این تگ به منظور نمایش **عناوین** استفاده می‌کنیم.
- ✓ متن رو **درشت و متمایز** می‌کنه.
- ✓ **h1** اولین تگ از مجموعه heading هست و ما level بندی‌های (سطح بندی‌های) بیشتری داریم.
- ✓ **h2** سطح بعدی این تگ هست و اصطلاحاً یک step (قدم) به **h1** اضافه می‌کنه.
- ✓ نکته کنکوری:
- در هر صفحه وب عنوان اصلی رو در **h1** قرار می‌دیم و زیر عنوان‌ها رو با **h2** نمایش می‌دیم.
- اگر زیر عنوان‌ها اضافه بشن (یعنی زیرعنوان زیرعنوان) از تگ‌های **h3, h4, h5** و **h6** استفاده می‌کنیم.
- ✓ پس، از **h1** داریم تا **h6**.
- ✓ البته معمولاً ما از **h1** و **h2** استفاده می‌کنیم.

تمرین

زیر تگ **h1** یک تگ **h2** قرار دهید و داخلش متن CatPhotoApp را بنویسید. 

راهنما

- 🧪 باید یک المنت **h2** داشته باشید.
- 🧪 المنت **h2** باید یک تگ بسته داشته باشد.
- 🧪 المنت **h2** باید حاوی متن «CatPhotoApp» باشد.
- 🧪 المنت **h1** باید حاوی متن «Hello World» باشد.

اگر متن ما طولانی تر بود و یک پاراگراف بود چکار کنیم؟

- ✓ برای این منظور از المنت **p** استفاده می‌کنیم.
- ✓ هرجایی که می‌خوایم متنی رو نمایش بدهیم از این المنت استفاده می‌کنیم.
- ✓ **p** به معنای paragraph (پاراگراف).
- ✓ می‌تونیم یک پاراگراف مثل این داشته باشیم: `<p> I'm a p tag! </p>`

یک المنت **p** زیر المنت **h2** خودتون بسازید و متن Hello Paragraph رو داخلش بنویسید.

✓ نکته: تمام تگ‌های HTML به صورت lowercase یعنی با حروف کوچک انگلیسی نوشته می‌شن.

`<P></P>` غلط

`<p></p>` درست

باید یک المنت معتبر **p** داشته باشید (معتبر به این معنا که با حروف کوچک انگلیسی نوشته شده باشد). 🏠

المنت **p** باید حاوی متن "Hello Paragraph" باشد. 🏠

المنت **p** باید حاوی تگ باز **p** و تگ بسته **p** باشد. 🏠

- ✓ در طراحی صفحات وب زمانی که برای پر کردن یک پاراگراف متن آماده‌ای وجود ندارد یا برای راهنمایی کردن کاربران، از placeholder استفاده می‌کنیم.
- ✓ توسعه دهندگان وب عمدتاً از متن‌های آماده‌ای برای placeholder پاراگراف‌ها استفاده می‌کنند که با آن‌ها lorem ipsum گفته می‌شود (البته این اصطلاح از صنعت چاپ می‌آید).

یک متن دلخواه را جایگزین متن المنت پاراگراف کنید (متن زیر را کپی کنید یا یک متن آماده از اینترنت پیدا کنید).
Kitty ipsum dolor sit amet, shed everywhere shed everywhere stretching attack your ankles chase the red dot, hairball run catnip eat the grass sniff.

راهنما ندارد. 🧪

- ✓ کامنت گذاری یکی از بهترین عادت های یک توسعه دهنده آگاه و حرفه ای می باشد.
- ✓ ما می تونیم در داخل برنامه های خودمون توضیحاتی قرار بدیم تا برای خودمون و دیگران سازوکار اون بخش از برنامه رو توضیح بدیم.
- ✓ یا حتی بخشی از برنامه رو کامنت کنیم تا نادیده گرفته بشه و در خروجی نمایش داده نشه.
- ✓ با این کار می تونیم بدون پاک کردن کدهامون، برنامه رو تست کنیم و نتیجه دلخواه رو بگیریم.
- ✓ کامنت ها در HTML با `<!--!>` شروع می شن و با `<-->` تمام می شن.
- ✓ یعنی هرچی بین این دوتا باشه کامنت محسوب می شه و نادیده گرفته و در خروجی به ما نمایش داده نمی شه.

تمرین

تگ های `h1`، `h2` و `p` را Uncomment کنید. (از حالت کامنت خارج کنید).

راهنما

- 🔥 المنت `h1` بعد از Uncomment کردن باید در صفحه قابل مشاهده باشد.
- 🔥 المنت `h2` بعد از Uncomment کردن باید در صفحه قابل مشاهده باشد.
- 🔥 المنت `p` بعد از Uncomment کردن باید در صفحه قابل مشاهده باشد.
- 🔥 تگ های کامنت گذاری نباید در صفحه وجود داشته باشند (`<!--!>` و `<-->`)

- ✓ کامنت گذاری یکی از بهترین عادت های یک توسعه دهنده آگاه و حرفه ای می باشد.
- ✓ ما می توانیم در داخل برنامه های خودمون توضیحاتی قرار بدیم تا برای خودمون و دیگران سازوکار اون بخش از برنامه رو توضیح بدیم.
- ✓ یا حتی بخشی از برنامه رو کامنت کنیم تا نادیده گرفته بشه و در خروجی نمایش داده نشه.
- ✓ با این کار می توانیم بدون پاک کردن کدهامون، برنامه رو تست کنیم و نتیجه دلخواه رو بگیریم.
- ✓ کامنت ها در HTML با `<!-->` شروع می شن و با `-->` تمام می شن.
- ✓ یعنی هرچی بین این دو تا باشه کامنت محسوب می شه و نادیده گرفته و در خروجی به ما نمایش داده نمی شه.

تمرین

- ✎ فقط `h1` و `p` رو کامنت کنید.
- ✎ یعنی `h2` نباید کامنت باشه.


راهنما

- 🚫 المنت `h1` کامنت می شود و در نتیجه در صفحه نباید مشاهده شود.
- 🚫 المنت `h2` کامنت نمی شود و در نتیجه در صفحه باید مشاهده شود.
- 🚫 المنت `p` کامنت می شود و در نتیجه در صفحه نباید مشاهده شود.
- 🚫 هرکدام از کامنت ها باید با `-->` بسته شوند.
- 🚫 ترتیب المنت های `h2`، `h1` و `p` در کد، نباید تغییر کند.




توضیحات

فرض کنید صفحه ما (مثلا صفحه تلفن همراه ما) فضای عمودی کمی دارد.
✓ پس بیاید المنت های غیرضروری را حذف کنیم تا ساخت **CatPhotoApp** رو شروع کنیم.

تمرین

المنت **h1** رو حذف کنید تا view تمیزتری داشته باشیم. 

راهنما

- المنت **h1** باید حذف شود. 
- المنت **h2** نباید حذف شود و باید همچنان قابل مشاهده باشد. 
- المنت **p** نباید حذف شود و باید همچنان قابل مشاهده باشد. 

✓ در هر صفحه وب، بخش‌های مختلفی داریم:

- بخش اصلی، اسلایدشو و ...
- بخش‌های سمت چپ و راست صفحه اصلی که بهشون می‌گیم سایدبار.
- بخش منوها یا راهنمای نویگیشن.
- بخش پایین یا فوتر.

✓ HTML5 تگ‌های معنادار و بهتری رو برای ما فراهم کرده که می‌تونیم تگ‌های

main, header, footer, nav, video, article, section و ... رو مثال بزنینم.

✓ این تگ‌ها باعث می‌شن ساختار صفحه وب اصطلاحاً توصیفی باشه. یعنی کاملاً مشخصه چی کجاست.

✓ استفاده از این تگ‌ها دوتا مزیت اصلی داره:

- ساختار صفحه بسیار مرتب و خوانا می‌شه.
- بهینه‌سازی برای موتورهای جستجو مثل گوگل.

تمرین

📄 در کد موجود، یک المنت **p** دیگر بسازید و برای محتوای متنی داخل آن از متن زیر(یا متن دلخواه) استفاده کنید:

Purr jump eat the grass rip the couch scratched sunbathe, shed everywhere rip the couch sleep in the sink fluffy fur catnip scratched.

📄 سپس دو پاراگراف را بعنوان فرزند، داخل المنت **main** قرار دهید.

راهنما

🔥 باید ۲ المنت **p** با متن Kitty Ipsum Text یا متن دلخواه خودتون داشته باشید.

🔥 هرکدام از المنت‌های **p** باید دارای تگ بسته **p** باشند (**</p>**).

🔥 کد شما باید دارای المنت **main** باشد.

🔥 المنت **main** باید دو فرزند از نوع پاراگراف (Paragraph) داشته باشد.

🔥 تگ باز المنت **main** باید قبل از تگ باز پاراگراف اول بیاید.

🔥 تگ بسته المنت **main** باید بعد از تگ بسته پاراگراف دوم بیاید.

توضیحات

✓ می‌توانید با استفاده از المنت **img** یک تصویر را به صفحه وب خود اضافه کنید و با استفاده از صفت **src** آدرس تصویر رو مشخص کنید:

```

```

✓ المنت **img** اصطلاحاً **self-closing** می‌باشد و نیازی به تگ بسته ندارد.

✓ تمام المنت‌های **img** حتماً باید دارای صفت یا attribute به نام **alt** باشند.

✓ اگر تصویر دکوری هست و صرفاً در زمان توسعه از اون استفاده می‌کنید، می‌تونید **alt** رو قرار ندید.

✓ در **alt** از کاراکترهای خاص استفاده نکنید و متن خالص وارد کنید.

مثال از صفت alt :

```

```

تمرین

📄 در المنت **main** موجود، قبل از اولین المنت **p**، یک المنت **img** قرار بدید.

برای خاصیت **src** از لینک زیر استفاده کنید:

```
https://chandvajhi.com/relaxing-cat.jpg
```

راهنما

🧪 صفحه شما باید دارای یک المنت **img** باشد.

🧪 المنت تصویر باید دارای صفت **src** باشد که به تصویر گربه اشاره کند.

🧪 صفت **alt** تصویر شما نباید خالی باشد.

✓ برای داشتن لینک به صفحات دیگر (چه صفحات داخلی سایت و چه وبسایتهای دیگر) باید از المنت **a** یا همان **Anchor** استفاده کنیم.

✓ المنت **a** به یک آدرس مقصد نیاز دارد که با صفت **href** این مقصد رو مشخص می‌کنیم.

✓ همچنین به یک *anchor text* یا همون متن لینک نیاز داریم.

مثال :

```
<a href="https://chandvajhi.com">this links to chandvajhi.com</a>
```

مروگر، متن `this links to chandvajhi.com` رو نمایش میده که می‌تونید روش کلیک کنید و وقتی کلیک کردید شمارو به وبسایت چندوجهی می‌بره.

یک المنت **a** با **href** سایت چندوجهی (`https://chandvajhi.com`) و *anchor text* `CHANDVAJHI` بسازید.

🔥 یک المنت **a** بسازید که به وبسایت `https://chandvajhi.com` لینک شده باشد.

🔥 *Anchor text* المنت **a** باید `CHANDVAJHI` باشد.

🔥 المنت **a** باید یک تگ بسته باشد (``).

✓ یکی از کاربردهای المنت **a** برای ساخت لینک به نقاط مختلف داخل یک صفحه وب هست.

✓ برای ساخت یک لینک به بخش داخلی در یک صفحه وب، برای مقدار صفت **href** باید از هَش # به علاوه صفت **id** المنت مقصد استفاده کنیم.

✓ از این حالت معمولا برای مواقعی استفاده می‌کنیم که یک مطلب طولانی چند قسمتی داریم و ابتدای مطلب لیست عناوین رو قرار می‌دیم و کاربر با کلیک بر روی هر کدام از اون عناوین می‌پره به مکان مطلب مورد نظر.

✓ المنت مقصد حتما باید صفت **id** داشته باشد. مثال :

```
<a href="#contacts-header">Contacts</a>
```

...

```
<h2 id="contacts-header">Contacts</h2>
```

وقتی کاربر روی لینک Contacts کلیک کنه به اون قسمت از صفحه می‌ره که المنت **contacts-header** قرار داره.

📄 لینک خارجی موجود رو با تغییر دادن صفت **href** به **#footer** و متن آن از **cat photos** به **Jump to Bottom** به یک لینک داخلی تبدیل کنید.

📄 صفت **target="_blank"** رو از المنت **a** حذف کنید.

📄 سپس یک المنت فوتر **<footer>** بسازید و صفت **id** آن را **footer** قرار دهید.

🚫 تنها یک تگ **a** باید در صفحه موجود باشد.

🚫 تنها یک تگ **footer** باید در صفحه موجود باشد.

🚫 المنت **a** باید دارای صفت **href** با مقدار **#footer** باشد.

🚫 المنت **a** نباید دارای صفت **target** باشد.

🚫 متن المنت **a** باید **Jump to Bottom** باشد.

🚫 المنت **footer** باید دارای صفت **id** با مقدار **footer** باشد.

✓ می‌توانیم لینک‌ها را درون عناصر متنی دیگر قرار بدیم، اصطلاحاً **nest link** داشته باشیم.

<p>

Here's a `` link to chandvajhi.com`` for you to follow.

</p>

یک متن عادی داریم که داخل المنت **p** قرار دارد.

<p> Here's a ... for you to follow. </p>

سپس در المنت **a** که باید تگ بسته داشته باشه (``)، صفت **target** رو داریم که تعیین می‌کنه مقصد در کجا باز بشه، که در این‌جا وقتی از مقدار **_blank** استفاده می‌کنیم یعنی اینکه لینک مورد نظر رو در یک تب جدید باز کن. لینک مورد نظرمون هم توسط صفت **href** تعیین شده و وبسایت `https://chandvajhi.com` هستش. متن `link to chandvajhi.com` هم anchor text هستش.

تمرین

📄 المنت **a** موجود رو توسط یک المنت جدید **p** تو در تو (nest) کنید. درست بعد از المنت **main**.

داخل پاراگراف جدید این متن را قرار دهید:

View more cat photos

cat photos لینک هست و بقیه متن، متن معمولی خواهد بود.

راهنما

🔥 باید یک المنت **a** داشته باشید که لینک باشد به `https://chandvajhi.com`.

🔥 متن المنت **a** باید `cat photos` باشد.

🔥 باید یک المنت **p** داشته باشید که المنت **a** را احاطه کرده باشد (المنت **a** فرزند المنت **p** باشد).

🔥 حداقل ۳ تگ **p** باید در صفحه موجود باشد.

🔥 المنت **a** باید داخل المنت **p** به صورت تودرتو (nest) شده باشد.

🔥 متن المنت **p** باید `"View more"` (با یک فاصله `space` بعد از آن) باشد.

🔥 المنت **a** نباید متن `"View more"` داشته باشد.

🔥 هرکدام از المنت‌های **p** باید دارای تگ بسته باشند (`</p>`).

🔥 هرکدام از المنت‌های **a** باید دارای تگ بسته باشند (``).

اهداف ما از ساخت لینک مرده:

- ✓ نمی‌دانیم مقصد لینک‌های ما به کدام صفحات است.
- ✓ تعیین رفتار پیش‌فرض یک لینک با Javascript .

تمرین

مقدار فعلی صفت **href** یک لینک است به `https://chandvajhi.com` .

این مقدار را با علامت `#` هاش جایگزین کنید تا یک `dead link` بسازیم.

برای مثال:

`href="#"`

راهنما

المنت `a` باید تبدیل به یک `dead link` شود و مقدار صفت **href** به `#` تغییر کند. 

✓ اگر المنتی رو با المنت **a** به صورت **تودرتو (nest)** در بیاریم، می‌تونیم تبدیل به لینکش کنیم.

✓ در مثال زیر یک تصویر رو با المنت **a** تودرتو (nest) کردیم و اون رو تبدیل به لینک کردیم.

```
<a href="#">
```

```

```

```
</a>
```

یادتون باشه فعلا در حال یادگیری هستیم و نمی‌خوایم تصویر به هیچ‌جا لینک باشه و باید یک dead link باشه، پس خاصیت href رو # قرار بدید.

📄 تصویر موجود رو داخل یک المنت **a** قرار بدید و اون رو تبدیل به یک لینک کنید.

وقتی این کار رو انجام دادید با موس روی تصویر برید، میبینید که نشانگر موس تبدیل به یک دست می‌شه، در این حالت تصویر به لینک تبدیل شده.

🧪 المنت **img** موجود باید با یک المنت **a** تودرتو (nest) شود.

🧪 المنت **a** باید یک dead link باشد و مقدار صفت **href** باید # باشد.

🧪 هر کدام از المنت‌های **a** باید دارای تگ بسته باشند (****).

✓ HTML المنتی داره که می‌تونیم توسط اون یک لیست نامرتب یا اصطلاحاً bullet point بسازیم.

✓ لیست هایی که توسط شماره بندی مرتب نشدن.

✓ لیست نامرتب یا unordered list ها با تگ باز المنت `` شروع می‌شوند.


✓ داخلشون یک یا چندتا المنت `` قرار می‌گیره.

✓ و در نهایت، زمانی که آیتم‌های لیستمون رو هر کدوم داخل یک `` قرار داریم، با تگ بسته `` پایان لیست رو اعلام می‌کنیم.

مثال:

```
<ul>
  <li>milk</li>
  <li>cheese</li>
</ul>
```

این کد یک لیست bullet point می‌سازه که دوتا آیتم milk و cheese داخلش قرار داره.

دوتا المنت آخری p رو حذف کنید و یک لیست نامرتب با این سه چیز بسازید: 

```
jump
eat the grass
rip the couch
```

🔥 یک المنت `ul` بسازید.

🔥 باید ۳ تا المنت `li` داخل این `ul` داشته باشید.

🔥 المنت `ul` باید تگ بسته داشته باشد (``).

🔥 المنت‌های `li` باید تگ‌های بسته داشته باشند (``).

🔥 هیچ‌کدام از المنت‌های `li` نباید خالی باشند و باید یک متن داخلشان وجود داشته باشد.

- ✓ این لیست توسط شماره‌گذاری و یا اعداد رومی و ... مرتب شده.
- ✓ لیست مرتب یا Ordered List با المنت `` شروع می‌شود.
- ✓ تعدادی `li` بعنوان آیتم‌های لیست، داخل آن قرار می‌گیرند.
- ✓ در نهایت با تگ بسته `` پایان لیست را اعلام می‌کنیم.

مثال:

```
<ol>
  <li>Garfield</li>
  <li>Sylvester</li>
</ol>
```

این مثال یک لیست مرتب می‌سازد که دارای ۲ آیتم شماره‌گذاری شده می‌باشد.

تمرین

با این سه چیز که یگ گربه از آن‌ها متنفر است، یک لیست مرتب بسازید.

Loud Noises

Car Rides

Strangers

راهنما

- 🔥 شما باید برای "Top 3 things cats hate": یک لیست مرتب بسازید.
- 🔥 شما باید یک لیست نامرتب داشته باشید از ۳ چیزی که گربه عاشق آن‌هاست " Things cats love".
- 🔥 باید فقط یک المنت `ul` داشته باشید.
- 🔥 باید فقط یک المنت `ol` داشته باشید.
- 🔥 باید ۳ المنت `li` در داخل المنت `ul` داشته باشید (۳ چیز که گربه عاشق آن‌هاست)
- 🔥 باید ۳ المنت `li` در داخل المنت `ol` داشته باشید (۳ چیز که گربه از آن‌ها متنفر است)
- 🔥 المنت `ul` باید دارای تگ بسته باشد (``).
- 🔥 المنت `ol` باید دارای تگ بسته باشد (``).
- 🔥 المنت‌های `li` باید دارای تگ بسته باشند (``).
- 🔥 المنت‌های `li` در `ul` و `ol` هیچ‌کدام نباید خالی و بدون متن باشند.

حالا بیایید یک فرم بسازیم!

✓ راحت‌ترین راه برای دریافت اطلاعات از کاربر استفاده از **input** هست.

✓ ما می‌توانیم یک **input** از نوع **text** را به شکل زیر بسازیم:

```
<input type="text">
```

✓ نکته : المنت **input** یک المنت **self-closing** می‌باشد.

به این معنا که نیازی به تگ بسته ندارد(مثل المنت **img**)

📄 زیر لیست‌ها، یک المنت **input** از نوع **text** بسازید.

🔥 سایت شما باید یک المنت **input** از نوع **text** داشته باشد.


- ✓ placeholder یک راهنماست تا کاربر متوجه بشه چه اطلاعاتی رو باید داخل این فیلد متنی وارد کنه.
- ✓ قبل از اینکه کاربر داخل این فیلد متنی چیزی بنویسه این متن نمایش داده می‌شه و به محض این‌که روی این فیلد کلیک کنه اون متن پنهان می‌شه.

مثال:

```
<input type="text" placeholder="this is placeholder text">
```

- ✓ نکته : یادتون باشه که المنت **input** از نوع self-closing هست و نیازی به تگ بسته نداره.

تمرین

برای المنت **input** خود که در درس قبل ساختید یک placeholder با این متن بنویسید: 
cat photo URL

راهنما

- 🔧 باید به المنت **input** موجود در اپلیکیشن خود، صفت placeholder را اضافه کنید.
- 🔧 مقدار صفت placeholder باید این متن باشد: "cat photo URL".
- 🔧 در نهایت المنت **input** نباید تگ بسته داشته باشد (self-closing می‌باشد).

- ✓ برای دریافت اطلاعات از کاربران راحت‌ترین راه استفاده از فیلد متنی یا همون **input** هستش.
- ✓ این **input**ها (که انواع مختلف دارند) باید فرزند المنتی به اسم **form** باشن، یعنی همه این‌ها باید در داخل المنت **form** قرار بگیرن تا مرورگر متوجه بشه این اطلاعات برای ارسال به سرور هستن.
- ✓ با استفاده از المنت **form** هست که ما به مرورگر می‌گیم این اطلاعات به کجا باید ارسال بشن، در واقع آدرس روی سرور خودمون رو بهش می‌دیم.

مثال :

```
<form action="/url-where-you-want-to-submit-form-data"></form>
```

تمرین

فیلد متنی خودتان را داخل یک فرم قرار بدید (اصطلاحاً اون رو nest کنید) و صفت **action** فرم را اینطور بنویسید :

```
action="/submit-cat-photo"
```

راهنما

- 🔥 فیلد متنی شما باید داخل المنت **form** قرار بگیرد.
- 🔥 فرم شما باید یک صفت **action** با مقدار `/submit-cat-photo` داشته باشد.
- 🔥 المنت فرم باید دارای تگ باز (**<form>**) و بسته (**</form>**) باشد.

✓ برای اینکه کاربر بتواند اطلاعات خودش یا هر اطلاعاتی که داخل فرم وارد کرده به سرور ما ارسال کند نیاز داریم که داخل المنت فرم دکمه ای قرار بدیم که بهش می‌گیم submit button یا دکمه submit .

✓ زمانی که روی این دکمه کلیک شود اطلاعات به همان URL ای ارسال می‌شود که در صفت action فرم خودمان تعیین کرده‌ایم.

مثال:

```
<button type="submit">this button submits the form</button>
```

تمرین

📄 بعنوان آخرین المنت داخل المنت **form** یک **button** (دکمه) قرار دهید.

صفت **type** دکمه را submit قرار دهید.

متن قابل نمایش روی دکمه را Submit قرار دهید.

راهنما

🔥 در داخل فرم شما باید یک دکمه وجود داشته باشد (بعنوان آخرین المنت داخل فرم).

🔥 دکمه submit باید دارای صفت **type** با مقدار submit باشد.

🔥 متن قابل نمایش بر روی دکمه باید "Submit" باشد.

🔥 المنت **button** باید دارای یک تگ بسته باشد (**</button>**).

✓ بعضی اطلاعات هستند که کاربر حتما باید آن‌ها را وارد نماید و نمی‌تواند آن فیلد را خالی رها کند،

اصطلاحا به این فیلدها می‌گیم Required (الزامی) و کاربر تا زمانی که مقدار مناسب رو داخل این input قرار نده نمی‌تونه فرم رو submit کند.

بعنوان مثال اگر بخواهیم یک input از نوع متن text داشته باشیم که کاربر حتما باید اطلاعاتی داخل آن وارد نماید،

اصطلاحا Required باشد، می‌توانیم به شکل زیر عمل کنیم:

```
<input type="text" required>
```

📄 فیلد متنی که در برنامه وجود دارد را Required کنید تا کاربر نتواند بدون پر کردن آن را ارسال کند.

سپس سعی کنید بدون این‌که چیزی داخل فیلد متنی بنویسید دکمه Submit را فشار دهید و مشاهده کنید که HTML5 چه پیغام خطایی به شما می‌دهد.

🔧 المنت input شما باید دارای صفت required باشد.

✓ گاهی کاربر باید از بین چند گزینه فقط یکی را انتخاب کند (مثلا در زمان تعیین جنسیت).

✓ سوالی داریم که می‌خواهیم کاربر به آن یک پاسخ از پاسخ‌های موجود را انتخاب کند.

✓ Radio Button ها یکی از انواع **input** ها هستند و هر Radio Button باید داخل یک المنت **label** باشد.

✓ برای گروه بندی دکمه‌های رادیویی که مربوط به یک سوال هستند باید صفت **name** آن ها را هم‌نام قرار بدیم.

```
<label>
```

```
<input type="radio" name="indoor-outdoor">Indoor
```

```
</label>
```

✓ نکته: بهتر است برای المنت **label** یک صفت **for** در نظر بگیریم و همینطور برای دکمه رادیویی خودمان هم یک

صفت **id** در نظر بگیریم و این دو (یعنی **for** و **id**) با هم برابر باشند. پس کد ما به این شکل تغییر خواهد کرد:

```
<label for="indoor">
```

```
<input id="indoor" type="radio" name="indoor-outdoor">Indoor
```

```
</label>
```

تمرین

یک جفت دکمه رادیویی به فرم اضافه کنید. 

هر کدام از دکمه ها باید در المنت **label** خودشان قرار داشته باشند.

یکی با متن Indoor و دیگری با متن Outdoor.

جهت گروه‌بندی دکمه‌ها صفت **name** هر دو باید indoor-outdoor باشد.

راهنما

🔥 برنامه شما باید دو دکمه رادیویی داشته باشد.

🔥 صفت **name** دکمه های رادیویی باید indoor-outdoor باشد.

🔥 هر کدام از دکمه‌های رادیویی باید داخل یک المنت **label** قرار بگیرند.

🔥 هر کدام از المنت‌های **label** باید دارای تگ بسته (**</label>**) باشند.

🔥 لیبل یکی از دکمه‌های رادیویی باید indoor باشد.

🔥 لیبل یکی از دکمه‌های رادیویی باید outdoor باشد.

🔥 دکمه‌های رادیویی باید داخل المنت **form** قرار بگیرند.

✓ اگر بخواهیم کاربر از بین پاسخ ها چندین گزینه را انتخاب کند باید از المنتی به نام **checkbox** استفاده کنیم.

✓ **checkbox** ها یکی از انواع **input** ها هستند.

✓ مانند دکمه های رادیویی، **checkbox** ها هم باید داخل یک المنت **label** قرار بگیرند.

✓ دلیل این کار این است که می خواهیم هر **checkbox** یک عنوان داشته باشد و کاربر متوجه شود چه پاسخی را انتخاب کند و با این کار دکمه و عنوان به هم می چسبند و اگر هر جای عنوان کلیک کنیم **checkbox** انتخاب می شود.

✓ **checkbox** های مرتبط به یکدیگر باید صفت **name** یکسانی داشته باشند.

نکته: بهتر است برای المنت **label** یک صفت **for** در نظر بگیریم و همینطور برای **checkbox** خودمان هم یک صفت **id** در نظر بگیریم و این دو (یعنی **for** و **id**) با هم برابر باشند.

مثال :

```
<label for="loving">
```

```
<input id="loving" type="checkbox" name="personality"> Loving
```

```
</label>
```

تمرین

داخل فرم ۳ تا چک باکس قرار دهید. هر کدام از چک باکس ها باید داخل یک المنت **label** قرار داشته باشند.

مطمئن شوید هر کدام از المنت های **label** دارای تگ بسته (**</label>**) باشند.

صفت **name** چک باکس ها باید **personality** باشد.

چک باکس ها باید داخل المنت فرم باشند.

برای این که دیگر المنت ها را به لاین بعد ببرید از المنت **
** استفاده کنید. (این المنت self-closing می باشد و نیازی به تگ بسته ندارد)

راهنما

داخل فرم ۳ تا چک باکس قرار دهید. 🚫

هر کدام از چک باکس ها باید داخل یک المنت **label** قرار داشته باشند. 🚫

مطمئن شوید هر کدام از المنت های **label** دارای تگ بسته (**</label>**) باشند. 🚫

صفت **name** چک باکس ها باید **personality** باشد. 🚫

چک باکس ها باید داخل المنت فرم باشند. 🚫

برای این که دیگر المنت ها را به لاین بعد ببرید از المنت **
** استفاده کنید. (این المنت self-closing می باشد و نیازی به تگ بسته ندارد) 🚫

چطور بفهمیم کاربر کدام دکمه رادیویی و کدام چک‌باکس‌ها را انتخاب کرده؟

وقتی یک فرم submit می‌شود و به سرور ارسال می‌شود، اطلاعاتی که به سرور ارسال می‌شود شامل گزینه‌های انتخاب شده در این **input** ها می‌باشد.

input ها با نوع **radio** و **checkbox** مقادیر خودشان را با عنوان **value** به سرور ارسال می‌کنند. مثال:

```
<label for="indoor">
  <input id="indoor" value="indoor" type="radio" name="indoor-outdoor">Indoor
</label>
<label for="outdoor">
  <input id="outdoor" value="outdoor" type="radio" name="indoor-outdoor">Outdoor
</label>
```

در اینجا، دو **input** از نوع دکمه رادیویی داریم. اگر کاربر دکمه indoor را انتخاب کند و فرم را submit کند، یکی از داده‌های که ارسال می‌شود به این شکل خواهد بود: indoor-outdoor=indoor

نام دکمه رادیویی و در مقابل آن مقدار دکمه. این مقدار را توسط صفت value تعیین می‌کنیم.

اگر این کار را نکنیم (یعنی value را ننویسیم) مقدار on به سرور ارسال می‌شود.

در این حالت چیزی که به سرور ارسال می‌شود اینطور خواهد بود: indoor-outdoor=on. این مقدار غیر قابل استفاده است و نمی‌توانیم تشخیص دهیم کدام گزینه انتخاب شده.

تمرین

داخل فرم ۳ تا چک‌باکس قرار دهید. هرکدام از چک‌باکس‌ها باید داخل یک المنت **label** قرار داشته باشند.

مطمئن شوید هر کدام از المنت‌های **label** دارای تگ بسته (**</label>**) باشند.

صفت **name** چک‌باکس‌ها باید personality باشد.

چک‌باکس‌ها باید داخل المنت فرم باشند.

برای این‌که دیگر المنت‌ها را به لاین بعد ببرید از المنت **
** استفاده کنید. (این المنت self-closing می‌باشد و نیازی به تگ بسته ندارد)

راهنما

داخل فرم ۳ تا چک‌باکس قرار دهید.

هرکدام از چک‌باکس‌ها باید داخل یک المنت **label** قرار داشته باشند.

مطمئن شوید هر کدام از المنت‌های **label** دارای تگ بسته (**</label>**) باشند.

صفت **name** چک‌باکس‌ها باید personality باشد.

چک‌باکس‌ها باید داخل المنت فرم باشند.

برای این‌که دیگر المنت‌ها را به لاین بعد ببرید از المنت **
** استفاده کنید. (این المنت self-closing می‌باشد و نیازی به تگ بسته ندارد)


✓ اگر بخواهیم یکی از چک باکس ها و یا دکمه های رادیویی را به صورت پیش فرض فعال کنیم می توانیم از صفت checked استفاده کنیم.

✓ برای این کار کافی هست تنها کلمه checked را در input خودمون بنویسیم. بعنوان مثال:

```
<input type="radio" name="test-name" checked>
```

✓ توصیه می کنیم این کلمه را بعنوان آخرین صفت یک input قرار دهید.

کارتی کنید اولین دکمه رادیویی و اولین چک باکس به صورت پیش فرض فعال شوند. 

اولین دکمه رادیویی در المنت فرم باید به صورت پیش فرض فعال باشد. 

اولین چک باکس در المنت فرم باید به صورت پیش فرض فعال باشد. 

چطور چندین المنت را داخل یک المنت **div** قرار دهیم؟

- ✓ المنت **div** که به اون division element هم می‌گن یک ظرف عمومی هست برای بقیه المنتها.
- ✓ المنت **div** احتمالاً پرکاربردترین و پرتکرارترین المنت در بین المنتهای HTML هستش.
- ✓ مثل بقیه المنتهای non-self-closing شما می‌توانید المنت **div** را با تگ **<div>** باز کنید و باید با تگ بسته **</div>** خاتمه آن را اعلام کنید.

“Things cats love” و “Things cats hate” را به طور کامل در یک المنت **div** قرار دهید. 

- 🧪 المنت **p** باید داخل المنت **div** باشد.
- 🧪 المنت **ul** باید داخل المنت **div** باشد.
- 🧪 المنت **ol** باید داخل المنت **div** باشد.
- 🧪 المنت **div** باید دارای تگ بسته باشد (**</div>**).

- ✓ برای اینکه به مرورگر بگیم از کدوم ورژن HTML داریم استفاده می‌کنیم باید از تگ DOCTYPE استفاده کنیم و بعد از این کلمه، ورژن HTML را قرار بدیم که برای HTML5 کافیه فقط بنویسیم html به این شکل: **<!DOCTYPE html>**
- ✓ این تگ باید اولین دستور در صفحه وب باشد و قبل از آن هیچ چیز نباید باشد.
- ✓ علامت ! و کلمه DOCTYPE با حروف بزرگ انگلیسی مهم هستند و دقیقاً باید به همین شکل نوشته شوند.
- ✓ کلمه html در اینجا اصطلاحاً case sensitive نیست (حساس به حروف بزرگ و کوچک نیست).
- ✓ بقیه کدهای HTML باید بین تگ‌های باز و بسته **html** قرار بگیرند.
- ✓ تگ باز **html** دقیقاً زیر تگ DOCTYPE قرار می‌گیرد و تگ بسته آن در انتهای صفحه بعنوان آخرین تگ قرار می‌گیرد. مثال :

```
<!DOCTYPE html>
```

```
<html>
```

```
<!-- کدهای ما در این جا قرار می‌گیرند -->
```

```
</html>
```

تمرین

- ✎ در داخل کد خود یک DOCTYPE در ابتدای صفحه اضافه کنید.
- ✎ زیر آن تگ‌های باز و بسته **html** را قرار دهید و داخل این تگ یک المنت **h1** قرار دهید.
- ✎ به المنت **h1** یک متن دلخواه بدهید.

راهنما

- 🔥 کد شما باید دارای تگ **<!DOCTYPE html>** باشد.
- 🔥 در کد شما باید یک المنت **html** وجود داشته باشد.
- 🔥 داخل المنت **html** باید یک المنت **h1** با متن دلخواه وجود داشته باشد.

- ✓ در داخل المنت **head** اطلاعاتی قرار می‌گیرد که صفحه وب ما را به موتورهای جستجو معرفی می‌کند.
 - ✓ اطلاعات Metadata مثل **title**، **meta**، **link** و **style** عمدتاً داخل **head** قرار می‌گیرند.
 - ✓ تقریباً هرچه در المنت **body** قرار بگیرد به کاربر نمایش داده خواهد شد.
- یک مثال از یک صفحه وب:

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata elements -->
  </head>
  <body>
    <!-- page contents -->
  </body>
</html>
```

- 📄 المنت‌های **head** و **body** را در این کد قرار دهید.
- 📄 المنت **title** داخل المنت **head** قرار می‌گیرد.
- 📄 المنت‌های **h1** و **p** داخل المنت **body** قرار می‌گیرند.

- 🔥 در کد شما فقط یک المنت **head** باید وجود داشته باشد.
- 🔥 در کد شما فقط یک المنت **body** باید وجود داشته باشد.
- 🔥 المنت **head** باید فرزند المنت **html** باشد (یعنی داخل المنت **html** باشد).
- 🔥 المنت **body** باید فرزند المنت **html** باشد (یعنی داخل المنت **html** باشد).
- 🔥 المنت **title** باید داخل المنت **head** باشد.
- 🔥 المنت‌های **h1** و **p** باید داخل المنت **body** باشند.

✓ برای تغییر رنگ متن، style المنت **h2** انجام دهیم.

✓ در CSS مرتب در مورد propertyها صحبت می‌کنیم، property یعنی ویژگی یا خاصیت.

✓ Property که مسئولیت رنگ متن یک المنت را به عهده دارد color است.

✓ به این شکل به رنگ متن داخل المنت **h2** استایل می‌دهیم و آن را آبی می‌کنیم:

```
<h2 style="color: blue;">CatPhotoApp</h2>
```

✓ بعد از هر property که در CSS تعریف می‌کنید به سمیکالن (;) قرار دهید.

استایل المنت **h2** را تغییر دهید طوری که رنگ متن آن قرمز باشد. 

🔥 المنت **h2** شما باید دارای style باشد.

🔥 رنگ متن المنت **h2** باید قرمز باشد.

🔥 پس از هر property داخل style حتما از سمیکالن (;) استفاده کنید.

✓ در این مثال به صورت inline CSS استایل دهی کردیم، که در درس مقدماتی گفتیم این روش اول است:

```
<h2 style="color: red;">CatPhotoApp</h2>
```

✓ روش دوم: در بالای کدتان یک بلاک style (همان المنت) درست کنید.

```
<style>
```

```
</style>
```

✓ اگر بخواهیم متن تمام المنت‌های h2 قرمز رنگ باشند به این شکل استایل‌دهی می‌کنیم:

```
<style>
h2 {
  color: red;
}
```

```
</style>
```

✓ قوانین استایل‌دهی (Rule) به هر المنت باید بین curly braces باز و بسته { } قرار بگیرند.

✓ تمام قوانین استایل‌دهی به المنت‌ها بین تگ‌های باز و بسته style قرار می‌گیرند.

✓ در انتهای هرکدام از rule ها سمیکالن قرار می‌گیرد.

تمرین

✎ استایل‌های inline المنت h2 را حذف کنید.

✎ بالای کدتان یک بلاک style درست کنید.

✎ در این بلاک style دستوراتی بنویسید که رنگ متن تمام المنت‌های h2 آبی شود.

راهنما

🔥 صفت style که به صورت inline نوشته شده باید از المنت h2 حذف شود.

🔥 یک المنت style در بالای کدتان بسازید.

🔥 المنت h2 باید آبی باشد.

🔥 استایل‌هایی که به h2 می‌دهید باید با سمیکالن و curly braces خاتمه یابند.

🔥 المنت استایل باید دارای تگ بسته (</style>) باشد.

✓ کلاس‌ها استایل‌های قابل استفاده مجدد هستند. مثال :

```
<style>
  .blue-text {
    color: blue;
  }
</style>
```

✓ چیزی که در بالا ساختیم، یک کلاس است با نام blue-text داخل تگ style .

✓ هر نامی می‌تونید بذارید، البته مهمه که معنادار باشه و قابل فهم و غیرتکراری.

✓ چطور می‌تونیم این کلاس رو به یک المنت HTML اعمال کنیم؟ به این شکل:

```
<h2 class="blue-text">CatPhotoApp</h2>
```

✓ نام کلاس‌ها همیشه با یک نقطه شروع می‌شه، اما در زمانی که می‌خوایم به یک المنت کلاس بدیم نیاز نیست

علامت نقطه رو قرار بدیم، فقط کافیه اسم کلاس رو بنویسیم.

تمرین

📄 در داخل المنت **style**، سلکتور **h2** را با **text-red**. جایگزین کنید و رنگ متن را از آبی به قرمز تغییر دهید.

📄 کلاس "red-text" را به المنت **h2** اعمال کنید.

راهنما

🔥 المنت **h2** باید قرمز باشد.

🔥 المنت **h2** باید دارای کلاس red-text باشد.

🔥 داخل استایل‌ها باید یک کلاس با نام red-text داشته باشید که رنگ متن را به قرمز تغییر دهد.

🔥 نباید از inline-style برای تغییر رنگ المنت **h2** استفاده کنید.

کلاس‌ها به ما اجازه می‌دن که یک استایل مشابه رو به هر تعداد المنتی که می‌خوایم بدیم. ✓

تمرین

کلاس text-red را به المنت **p** بدهید تا رنگ آن قرمز شود. 📄

راهنما

- المنت **h2** باید قرمز باشد. 🏠
- المنت **h2** باید دارای کلاس red-text باشد. 🏠
- اولین المنت **p** باید قرمز باشد. 🏠
- دومین و سومین المنت **p** نباید قرمز باشند. 🏠
- اولین المنت **p** باید دارای کلاس red-text باشد. 🏠

توضیحات

✓ برای تغییر اندازه فونت یک المنت باید از خاصیتی در CSS استفاده کنید با نام font-size .

✓ مثال تغییر اندازه فونت :

```
h1 {  
  font-size: 30px;  
}
```

✓ نکته : وقتی می‌گیم **تگ** منظورمون همون المنت هستش، منتها وقتی تکی هستش، مثل **p** می‌گیم تگ. زیاد سخت نگیرین.

تمرین

داخل تگ **style** (جایی که کلاس **red-text** وجود دارد)، یک سلکتور برای **p** (تگ‌های پاراگراف) ایجاد کنید و اندازه فونت آن را 16px قرار دهید.

راهنما

🔥 بین تگ‌های باز و بسته **style**، برای المنت‌های **p** اندازه فونت را 16px قرار دهید.

🔥 مطمئن باشید که مرورگر در حالت زوم نباشد.

✓ برای تغییر فونت المنت‌ها می‌توانیم از proerty به نام font-family یا همان خانواده فونت استفاده کنیم.

✓ یکسری فونت‌ها هستند که در بین اکثر سیستم عامل‌ها مثل ویندوز، مک اینتاش و لینوکس مشترک هستند، در حال حاضر، با دانشی که ما در این زمینه داریم با font-family فقط می‌تونیم از بین این فونت‌ها، یک فونت رو انتخاب کنیم.

✓ در آینده یاد می‌گیریم که چطور فونت‌های شخصی (و البته استاندارد) خودمون رو به المنت‌ها اعمال کنیم.

✓ برای مثال اگر بخوایم فونت یک المنت h2 را به خانواده sans تغییر دهیم اینطور عمل می‌کنیم:

```
h2 {  
  font-family: sans-serif;  
}
```

کارتی کنید تمام المنت‌های **p** دارای فونت monospace شوند. 

المنت‌های **p** باید دارای font-family با مقدار monospace باشند. 

توضیحات

✓ در اینجا از کتابخانه رایگان و استاندارد گوگل فونت استفاده می‌کنیم. یک URL یا آدرس اینترنتی وجود دارد که باید به صفحه ما پیوست (Import) بشه.

✓ می‌خواهیم خانواده فونت Lobster را از کتابخانه فونت گوگل به صفحه خودمون وارد کنیم. باید **قبل از شروع تگ style** این خط کد رو بنویسیم:

```
<link href="https://fonts.googleapis.com/css?family=Lobster" rel="stylesheet" type="text/css">
```

✓ می‌توانیم با استفاده از نام Lobster فونت المنت‌ها را تغییر دهیم و به جای FONT_FAMILY فقط کافیه Lobster رو بنویسیم.

```
font-family: FAMILY_NAME, GENERIC_NAME;
```

✓ GENERIC_NAME : برای زمانی که فونت دلخواه ما به هر دلیلی لود نشده، به صفحه پیوست نشده یا پاک شده از GENERIC_NAME استفاده می‌کنیم. اگر این موارد اتفاق بیافتد GENERIC_NAME به عنوان فونت جایگزین استفاده می‌شود. بهش می‌گن fallback font . در درس بعد به آن می‌پردازیم.

✓ فونت‌های دلخواه ما case-sensitive هستن و اگر در نامشان یک حرف بزرگ انگلیسی با یک فاصله وجود داشته باشد باید به طور کامل رعایت شود وگرنه فونت بارگذاری نمی‌شود و روی المنت ما اعمال نمی‌شود.

✓ در این حالت باید از دوتا دابل کوتیشن برای نام فونت استفاده کنیم چون دوکلمه‌است و دقیقاً باید به همین شکل باشد. "Open Sans".

✓ اگر یک کلمه‌ای باشد نیازی به استفاده از دابل کوتیشن نیست. Lobster

تمرین

یک class rule برای المنت **h2** بسازید و در آن font-family را به Loster تغییر دهید.

راهنما

خانواده فونت Lobster را باید import کنید. (قبل از تگ style)

المنت **h2** باید از فونت Lobster استفاده کند.

برای تغییر فونت المنت **h2** باید از سلکتور h2 استفاده کنید.

المنت‌های **p** همچنان باید از فونت monospace استفاده کنند.

چندین فونت پیش‌فرض در مرورگرها وجود دارد،

بعنوان مثال generic font family هایی از قبیل: serif, monospace, و sans-serif .

اگر به هر دلیلی فونت اولی که تعریف کردید لود نشد و پیدا نشد مرورگر به صورت خودکار به سراغ فونت بعدی

می‌رود. اصطلاحاً degrade می‌کند به فونت بعدی (اگر فونت تعریف کرده باشیم).

در مثال زیر، اگر فونت Helvetica وجود نداشته باشد مرورگر به سراغ فونت بعدی یعنی sans-serif می‌رود.

p {

font-family: Helvetica, sans-serif;

}

فونت های Generic حساس به حروف بزرگ و کوچک نیستند و نیازی نیست آن‌ها را بین دوتا دابل کوتیشن

بگذاریم چون آن‌ها در CSS به عنوان keyword شناخته می‌شوند.

Keywordها کلمات از پیش تعریف شده در CSS هستند.

تمرین

به المنت **h2** فونت monospace بدهید. این فونت به عنوان فونت دوم بعد از Lobster قرار می‌گیرد و برای زمان degrade شدن در نظر گرفته می‌شود.

در چالش قبلی شما با استفاده از تگ link فونت Lobster را import کردید. حالا آن خط از کد را کامنت کنید تا اجرا نشود، پس در نتیجه فونت Lobster به صفحه ما import نمی‌شود و در این حالت degrade انجام می‌شود و فونت دوم یعنی monospace به المنت **h2** داده می‌شود.

اگر فونت Lobster روی کامپیوتر شما از قبل نصب باشد این degrade انجام نمی‌شود، چون مرورگر توانایی این را دارد که اگر فونت مورد نظر را پیدا نکرد، روی کامپیوتر شما جستجو کند و آن‌را پیدا کند، در نتیجه شما تغییری نخواهید دید.

راهنما

المنت **h2** باید دارای فونت Lobster (بعنوان اولین فونت) باشد.



المنت **h2** باید در زمان عدم وجود فونت Lobster به فونت monospace اصطلاحاً degrade کند.

کدی که برای وارد کردن فونت Lobster از گوگل نوشتید را کامنت کنید.

✓ CSS برای کنترل عرض یک المنت یک property به نام width دارد. مانند فونت ها، از px برای تعیین عرض استفاده می‌کنیم (موارد دیگه‌ای هم هستن، که بعدا بررسی می‌کنیم).
مثال: می‌خواهیم برای تصاویر با عرض ۵۰۰ یک کلاس با نام larger-image بنویسیم:



```
<style>  
.larger-image {  
width: 500px;  
}  
</style>
```

تمرین

یک کلاس با نام smaller-image برای تصاویر با عرض ۱۰۰ پیکسل بسازید. 
این کلاس را به تصویر موجود در صفحه بدهید تا عرض آن ۱۰۰ پیکسل شود. 

✓ برای انجام تست هایی که طول و عرض و این چنین مقادیری داریم حتما مراقب باشید مرورگر شما در حالت زوم نباشد.

راهنما

المنت img باید دارای کلاس smaller-image باشد. 
عرض تصویر شما باید ۱۰۰ پیکسل باشد. 

- ✓ یکی از قابلیت‌های جالب در CSS امکان قرار دادن حاشیه یا border برای المنت‌ها هست.
 - ✓ Border دارای property های از قبیل: color، style و width می‌باشد.
- در مثال زیر یک border قرمز رنگ ۵ پیکسلی با رنگ ثابت در کلاسی تعریف کردیم:

```
<style>
.thin-red-border {
border-color: red;
border-width: 5px;
border-style: solid;
}
</style>
```

✓ یک کلاس با نام thick-green-border بسازید.

✓ این کلاس باید یک border با عرض ۱۰ پیکسل با رنگ ثابت سبز ایجاد کند.

✓ این کلاس را به تصویر گربه بدهید.

✓ می‌توانیم به یک المنت چندین کلاس را اختصاص بدهیم. کلاس‌ها را به ترتیب با قرار دادن یک فاصله بین‌شان می‌نویسیم. مثل این مورد:

```
<img class="class1 class2">
```

✓ دقت کنید که این کلاس را باید به تصویر گربه اضافه کنید.

🔥 المنت **img** باید دارای کلاس smaller-image باشد.

🔥 المنت **img** باید دارای کلاس thick-green-border باشد.

🔥 تصویر گربه باید یک border با عرض ۱۰ پیکسل داشته باشد.

🔥 رنگ border باید ثابت باشد.

🔥 border دور تصویر گربه باید سبز باشد.

تا حالا یاد گرفتیم که چطور به تصویر خودمون حاشیه بدیم زیباترش کنیم. اما این حاشیه با لبه های تیز مناسب نیست و ظاهر خیلی زیبایی نداره، در اکثر وبسایت‌ها border معمولا با لبه‌های گرد هستش .
می‌توانیم لبه‌ها را اصطلاحا round کنیم با اضافه کردن یک property دیگر با عنوان border-radius. ✓

تمرین

به تصویر گربه یک border-radius با اندازه ۱۰ پیکسل بدهید. 

در این تمرین شما می‌توانید از بیش از یک راه حل استفاده کنید.

مثلا می‌توانید این border-radius را به کلاس thick-green-border بدهید یا به کلاس smaller-green ، تصمیم با شماست.


پیشنهاد من به شما این است که این property را به کلاسی اضافه کنید که بقیه property های مخصوص border در آنجا هستند. منظورم thick-green-border هست.

راهنما

المنت **img** شما باید یک کلاس thick-green-border داشته باشد. 

تصویر گربه باید border radius با اندازه ۱۰ پیکسل داشته باشد. 

- ✓ برای این‌که لبه‌ها رو انقدر گرد کنید که به هم برسند و یک دایره تشکیل بدهند باید مقدار border-radius رو به اندازه نصف عرض المنت بدیم.
- ✓ برای مثال می‌تونیم برای مقدار border-radius تصویر گربه به جای ۵۰٪ استفاده کنیم.

به تصویر گربه یک border-radius با اندازه 50% بدهید. 

🔥 برای اینکه حاشیه تصویر گربه را به صورت کاملاً گرد در بیاورید باید خاصیت border-radius را 50% بدهید.

✓ برای تغییر رنگ پس‌زمینه یک المنت، باید از خاصیت background-color استفاده کنیم.

✓ در این مثال می‌خواهیم رنگ پس‌زمینه یک المنت سبز باشد:

```
.green-background {  
  background-color: green;  
}
```

📄 یک کلاس با عنوان silver-backgroud بسازید.

📄 رنگ پس‌زمینه را silver قرار دهید و این کلاس را به المنت **div** خودتان بدهید.

🔥 المنت **div** باید دارای کلاس silver-background باشد.

🔥 رنگ پس‌زمینه المنت **div** باید نقره‌ای باشد.

🔥 یک کلاس با نام silver-background باید در المنت **style** موجود باشد که مقدار خاصیت background-color آن silver است.

توضیحات

✓ هر المنت HTML علاوه بر کلاس می‌تونه id هم داشته باشه.

✓ برای استفاده از idها چندین مزیت وجود داره:

• ما از id برای استایل دادن به یک **تک المنت** استفاده می‌کنیم. در واقع خاص و یونیک هستن برای یک المنت.

✓ صفت id باید unique باشه. یعنی یکی باشه برای یک المنت.

✓ مرورگرها روی این قضیه حساس نیستن اما حتما شما این موضوع رو رعایت کنید.

✓ یک id رو به بیشتر از یک المنت ندید. در این مثال می‌خواهیم Cat-photo-app را به عنوان id المنت **h2** در نظر


بگیریم:

```
<h2 id="cat-photo-app">
```

تمرین

به المنت **form** یک id با عنوان cat-photo-form بدهید. 

راهنما

المنت فرم شما باید دارای id با عنوان cat-photo-form باشد. 

- ✓ می‌توانیم به صفت id درست مانند class استایل بدهیم.
- ✓ id قابل استفاده مجدد نیست و فقط باید به یک المنت اختصاص داده شود.
- ✓ به همین دلیل اولویت بیشتری دارد. فرض کنیم یک المنت هم دارای class هست هم id. و استایل‌هایی دارند که با هم تناقض دارند. یعنی بعنوان مثال در کلاس گفتیم رنگ متن المنت آبی باشد و در id گفتیم رنگ متن باید قرمز باشد. به این حالت conflict یا تناقض می‌گوییم. در این حالت هر استایلی که در id داده‌ایم، اعمال می‌شود.
- ✓ اولویت id بیشتر است. پس در نتیجه رنگ متن المنت قرمز خواهد بود. مثال:

```
#cat-photo-element {
  background-color: green;
}
```

- ✓ در المنت استایل، زمانی که کلاس تعریف می‌کنیم قبل از نام کلاس یک نقطه می‌گذاریم. برای id به جای نقطه باید از علامت # استفاده کنیم.

id فرم ما cat-photo-form می‌باشد. با استایل دادن به این id رنگ پس‌زمینه فرم را سبز کنید. 

- 🚫 id المنت فرم باید cat-photo-form باشد.
- 🚫 رنگ پس‌زمینه المنت فرم باید سبز (green) باشد.
- 🚫 نباید به فرم خود class یا استایل دیگری بدهید.

تمام المنت‌های HTML مستطیل شکل هستند. ✓

سه property مهم هستند که فضای داخلی و بیرونی المنت‌های HTML رو تنظیم می‌کنن: ✓
padding, margin و border.

padding، فاصله بین border و محتویات المنت را کنترل می‌کند. ✓

در مثال درس مشاهده می‌کنید که باکس آبی و قرمز داخل باکس زرد قرار دارند.

توجه کنید که باکس قرمز پدینگ بیشتری نسبت به باکس آبی دارد.

زمانی که مقدار پدینگ باکس آبی را افزایش دهید میزان فاصله بین border و متن بیشتری می‌شود.

اندازه padding باکس آبی را به حدی افزایش دهید تا به اندازه padding باکس قرمز برسد. 📄

مقدار padding کلاس blue-box باید 20px باشد. 🧪

✓ margin میزان فضا بین border یک المنت و المنت‌های اطراف آن را کنترل می‌کند.

در مثال درس مشاهده می‌کنید دو باکس آبی و قرمز داخل باکس زرد قرار دارند.

مشاهده می‌کنید که باکس قرمز نسبت به باکس آبی margin بیشتری دارد. که همین موضوع باعث شده کوچک‌تر دیده بشه.

وقتی margin باکس آبی را بیشتر کنیم فضای بین border این المنت و دیگر المنت‌ها بیشتر خواهد شد.

📄 margin باکس آبی را تغییر دهید تا مانند باکس قرمز شود.

🔥 مقدار margin در کلاس blue-box باید 20px باشد.

- ✓ margin فضای بین border یک المنت و المنتهای اطرافش رو کنترل می‌کنه.
- ✓ اگر یک مقدار منفی به margin بدهیم المنت ما بزرگ‌تر خواهد شد.

تمرین

سعی کنید مانند باکس قرمز که margin منفی دارد. به باکس آبی margin منفی 15px- بدهید تا تمام فضای افقی باکس زرد را اشغال کند.

راهنما

مقدار margin کلاس blue-box باید 15px- باشد. 🏠

بعضی مواقع ما نیاز داریم که مقادیر مختلف padding به یک المنت بدهیم.

✓ CSS این اجازه را به ما می‌دهد که برای کنترل فضای هر چهار طرف محتوای خودمان تا border مقادیری را تعیین کنیم.

✓ هر side یک property مخصوص به خودش رو داره:

padding-top •

padding-right •

padding-bottom •

padding-left •

به باکس آبی یک padding با مقدار 40px برای بالا و سمت چپ و 20px به سمت راست و پایین. 

🔥 مقدار padding بالا برای کلاس blue-box باید 40px باشد.

🔥 مقدار padding راست برای کلاس blue-box باید 20px باشد.

🔥 مقدار padding پایین برای کلاس blue-box باید 20px باشد.

🔥 مقدار padding چپ برای کلاس blue-box باید 40px باشد.

✓ دقیقا مانند کاری که در padding انجام دادیم اینجا هم می‌توانیم به هر چهار طرف مقادیر مختلف بدهیم. زمانی پیش می‌آید که می‌خواهیم المنت ما در هر طرف margin خاصی داشته باشد.


✓ CSS به ما این اجازه را می‌دهد که برای هر چهار طرف با استفاده از مقادیر زیر margin تعیین کنیم:

margin-top•


margin-right•

margin-bottom•


margin-left•

به باکس آبی، margin بالا و چپ 40px و برای margin پایین و راست 20px بدهید. 

مقدار margin بالا برای کلاس blue-box باید 40px باشد. 

مقدار margin راست برای کلاس blue-box باید 20px باشد. 

مقدار margin پایین برای کلاس blue-box باید 20px باشد. 

مقدار margin چپ برای کلاس blue-box باید 40px باشد. 

✓ در درس‌های قبل با استفاده از padding-top, padding-right, padding-bottom, padding-left یاد گرفتیم که می‌توانیم برای یک المنت چطور padding تعیین کنیم.

✓ اما یک راه ساده تر وجود دارد که کدتون رو حرفه‌ای تر نشون می‌ده و البته کوتاه‌تر و جمع و جور تر. اونم استفاده از مقداردهی Clockwise است. ساعتگرد.

✓ به این شکل:

```
padding: 10px 20px 10px 20px;
```

✓ این چهار مقدار دقیقا مثل ساعت عمل می‌کنن. بالا، راست، پایین، چپ.

✓ و دقیقا همون نتیجه‌ای رو بهمون می‌دن که روش قبل بهمون میداد. با این تفاوت که بسیار کوتاه‌تر و خواناتر شده.

با استفاده از مقداردهی Clockwise به کلاس blue-box مقدار padding بالا و چپ 40px و padding پایین و راست 20px بدهید.

🔥 مقدار padding بالا برای کلاس blue-box باید 40px باشد.

🔥 مقدار padding راست برای کلاس blue-box باید 20px باشد.

🔥 مقدار padding پایین برای کلاس blue-box باید 20px باشد.

🔥 مقدار padding چپ برای کلاس blue-box باید 40px باشد.

🔥 برای تعیین padding کلاس blue-box باید از روش Clockwise استفاده کنید.

✓ در درس‌های قبل با استفاده از margin-top, margin-right, margin-bottom, margin-left یاد گرفتیم که می‌توانیم برای اطراف یک المنت چطور margin تعیین کنیم.

✓ اما یک راه ساده تر وجود دارد که کدتون رو حرفه‌ای تر نشون می‌ده و البته کوتاه‌تر و جمع و جور تر.

✓ اونم استفاده از مقداردهی Clockwise هستش. ساعتگرد.

به این شکل:

```
margin: 10px 20px 10px 20px;
```

✓ این چهار مقدار دقیقا مثل ساعت عمل می‌کنن. بالا، راست، پایین، چپ.

✓ و دقیقا همون نتیجه‌ای رو بهمون می‌دن که روش قبل بهمون میداد. با این تفاوت که بسیار کوتاه‌تر و خواناتر شده.

تمرین

با استفاده از مقداردهی Clockwise به کلاس blue-box مقدار margin بالا و چپ 40px و margin پایین و راست 20px بدهید.

راهنما

🔥 مقدار margin بالا برای کلاس blue-box باید 40px باشد.

🔥 مقدار margin راست برای کلاس blue-box باید 20px باشد.

🔥 مقدار margin پایین برای کلاس blue-box باید 20px باشد.

🔥 مقدار margin چپ برای کلاس blue-box باید 40px باشد.

🔥 برای تعیین margin کلاس blue-box باید از روش Clockwise استفاده کنید.

روش‌های دیگری هم برای استایل دادن به المنت‌ها وجود دارد که یکی از آن‌ها استفاده از Attribute Selector می‌باشد. ✓

در این حالت می‌توانیم گروهی از المنت‌ها را انتخاب کنیم. ✓

برای استایل دادن به چک‌باکس‌ها می‌توانیم آن‌ها را با این حالت [attr=value] به صورت گروهی انتخاب کنیم. در این چالش شما باید از این حالت برای استایل دادن به چک‌باکس‌ها در CatPhotoApp استفاده کنید. ✓

این مثال تمام المنت‌هایی که دارای type از نوع radio هستند را انتخاب می‌کند و به آن‌ها استایل می‌دهد: ✓

```
[type='radio'] {
  margin: 20px 0px 20px 0px;
}
```


از سلکتور type attribute برای استایل دادن به چک‌باکس‌های CatPhotoApp استفاده کنید و به آن‌ها margin بالا 10px و margin پایین 15px بدهید. 📄

سلکتور type باید checkbox‌ها را انتخاب کند. 🏠

margin بالای چک‌باکس‌ها باید 10px باشد. 🏠

margin پایین چک‌باکس‌ها باید 15px باشد. 🏠

- ✓ تا الآن تمام مقادیری که به margin و padding دادیم همه‌گی px بودند. پیکسل‌ها یک نوع از واحد طولی می‌باشند که به مرورگر می‌گوید چه سایزی و چه فضایی را برای یک آیتم در نظر بگیرد.
- ✓ به علاوه px در CSS یونیت‌های دیگری برای تعیین این مقادیر داریم.
 - دو تا از اصلی‌ترین‌ها Absolute و Relative هستند.
- ✓ واحدهای Absolute با واحدهای فیزیکی طول گره خورده‌اند. برای مثال in و mm به ترتیب به معنای اینچ و میلیمتر می‌باشند.
- ✓ یونیت‌های Absolute اندازه‌ها رو بر روی صفحه نمایش به صورت تقریبی محاسبه می‌کنن اما بسته به وضوح صفحه نمایش یا همان resolution این اندازه‌ها متفاوت خواهند بود.
- ✓ یونیت‌های Relative مثل em یا rem وابسته به یک مقدار طول دیگر هستند.
- ✓ برای مثال: em بر اساس سایز فونت المنت می‌باشد. اگر از این واحد برای تعیین اندازه font-size یک المنت استفاده کنید چون وابسته است، در نتیجه به المنت والد خودش نگاه می‌کنه و بر اساس font-size اون، خودش رو تنظیم می‌کنه.
- ✓ چندین relative unit وجود دارد که مبتنی بر سایز viewport هستند. این موارد در مباحث آینده، اصول طراحی رسپانسیو بررسی می‌شوند.

مقدار padding کلاس red-box را 1.5em بدهید. 

- 🔥 کلاس red-box باید دارای padding باشد.
- 🔥 مقدار padding در کلاس red-box باید برابر 1.5em باشد.



توضیحات

- ✓ هر صفحه HTML یک المنت **body** دارد.
- ✓ المنت **body** به صورت ضمنی وجود دارد حتی اگر نوشته نشده باشد.

تمرین

یک المنت **style** بنویسید و داخل آن رنگ پس زمینه المنت **body** را سیاه در نظر بگیرید. 

راهنما

رنگ پس‌زمینه المنت **body** باید black باشد. 
هیچ کدی به جز المنت **style** و استایل‌دهی به المنت **body** نباید بنویسید. 

- ✓ هر سند HTML به صورت مخفی یا ضمنی دارای المنت **body** می‌باشد، پس می‌توانیم به این المنت استایل CSS بدهیم.
- ✓ مانند بقیه المنت‌های HTML شما به **body** می‌توانید استایل‌هایی بدهید که به المنت‌های داخلی این المنت به ارث داده می‌شود.
- ✓ هر عنصری که داخل المنت **body** وجود دارد تحت تاثیر این استایل‌ها قرار می‌گیرد و اصطلاحاً از این المنت ارث می‌برد.

- 📄 یک المنت **h1** با متن Hello World بسازید.
- 📄 سپس با تعیین `color: green` در استایل‌های `body`، برای تمام المنت‌های صفحه رنگ متن را سبز کنید.
- 📄 در نهایت، فونت کل المنت‌های داخلی `Body` را با تعیین `font-family: monospace` در استایل‌های المنت `body` تغییر دهید.

- 🔥 شما باید یک المنت **h1** بسازید.
- 🔥 متن داخل المنت **h1** باید Hello World باشد.
- 🔥 مطمئن شوید که المنت **h1** دارای تگ بسته (`</h1>`) می‌باشد.
- 🔥 خصوصیت `color` المنت **body** باید `green` باشد.
- 🔥 خصوصیت `font-family` المنت **body** باید `monospace` باشد.
- 🔥 فونت **h1** باید از المنت **body** ارث‌بری کرده باشد و `monospace` باشد.
- 🔥 رنگ متن **h1** باید از المنت **body** ارث‌بری کرده باشد و سبز باشد.

- ✓ بعضی مواقع استایل‌هایی به یک المنت HTML می‌دهیم که با هم اصطلاحاً **conflict** یا تناقض دارند. برای این‌که در این حالت استایل مورد نظر ما اعمال شود نیاز داریم تا اولویت استایل را مشخص کنیم.
- ✓ المنت **h1** نمی‌تواند همزمان هم سبز باشد هم صورتی.

- 📄 یک کلاس با نام pink-text بسازید که رنگ متن المنت را صورتی (pink) می‌کند.
- 📄 کلاس pink-text را به المنت **h1** بدهید.

- 🔧 المنت **h1** باید دارای کلاس pink-text باشد.
- 🔧 داخل المنت **style** باید کلاسی با نام pink-text داشته باشید که رنگ متن را صورتی می‌کند.
- 🔧 رنگ متن المنت **h1** باید صورتی باشد.

✓ اعمال اولویت یا override کردن استایل‌ها با دنباله کلاس‌های پشت سرهم:

✓ کلاس pink-text استایل CSS که در body داده بودیم را اصطلاحاً override کرد یعنی اون رو باطل کرد و استایل دیگری اعمال کرد.

✓ کلاسی که ما می‌نویسیم می‌تواند استایلی که در body داده بودیم را override کند.

سوالی که پیش می‌آید این است که آیا ما می‌توانیم استایلی که در کلاس pink-text خودمان دادیم را override کنیم؟

✓ زمانی که لیست کلاس‌ها را داخل المنت قرار می‌دهیم، ترتیبشان مهم نیست.

✓ اما، ترتیب قرار گرفتن استایل‌ها در المنت **style** مهم است و این جاست که ترتیب باید رعایت شود. بسیاری از برنامه نویسان تازه کار از این موضوع مطلع نیستند و هرچقدر به کلاس مورد نظرشان که قرار است override انجام دهد استایل می‌دهند تغییری نمی‌بینند، این به دلیل رعایت نکردن این موضوع می‌باشد.

✓ کلاسی که پایین‌تر باشد، در واقع دومین کلاس محسوب می‌شود و می‌تواند کار override را انجام دهد.

تمرین

یک کلاس با نام blue-text بسازید و در آن color را blue (آبی) قرار دهید.

مطمئن بشید که این کلاس، زیر کلاس pink-text قرار دارد.

کلاس blue-text را به صورت دنباله‌ای از کلاس‌ها، با یک فاصله به المنت h1 اضافه کنید.

یعنی در حال حاضر المنت **h1** کلاسی با نام pin-text دارد، یک فاصله (space) قرار دهید و نام کلاس blue-text را بنویسید.

برای استفاده از قابلیت multiple class یا چند کلاس با هم، باید بین کلاس‌ها یک فاصله خالی قرار داد، به این شکل:

class = "class1 class2"

در اینجا کلاس blue-text چون پایین‌تر از کلاس pink-text است پس override را انجام می‌دهد.

راهنما

المنت **h1** شما باید دارای کلاس pink-text باشد. 🧪

المنت **h1** شما باید دارای کلاس blue-text باشد. 🧪

هر دو کلاس blue-text و pink-text باید به المنت **h1** داده شده باشند. 🧪

المنت **h1** شما باید آبی باشد. 🧪

چطور توسط id کلاس‌ها را override کنیم؟

- ✔ ما ثابت کردیم که مرورگر استایل‌ها را از بالا به پایین می‌خواند، بر اساس ترتیب نوشتن استایل‌ها.
- ✔ این به این معناست که اگر تعارضی بین دو استایل پیش بیاید، یعنی دو استایل داشته باشیم که هر دو یک کار را انجام می‌دهند (مثلا یکی رنگ متن را آبی و دیگری رنگ متن را قرمز قرار می‌دهد) در این حالت اولویت اجرا و اعمال شدن با استایلی است که آخرین استایل باشد.
- ✔ در واقع مرورگر به declaration order نگاه می‌کند، نه order of use .

اما راه‌های دیگری برای override کردن CSS وجود دارد. صفت id رو یادتون می‌آد؟

- ✔ اهمیتی ندارد که id را کجای المنت **style** می‌نویسید، همیشه و در همه حالات id اولویت و تقدم دارد.

تمرین

به المنت **h1** یک صفت id با نام orange-text بدهید. 

المنت **h1** باید دارای کلاس‌های قبلی باشد، بذارید کلاس‌های blue-text و pink-text سر جاشون باشن. 


در المنت **style** یک id با نام orange-text بسازید. این یک مثال از نحوه نوشتن id هست: 


```
#brown-text {
  color: brown;
}
```

- ✔ نکته بسیار مهم: اهمیتی ندارد که این id را کجای المنت **style** می‌نویسید، قبل یا بعد از کلاس‌های pink-text و blue-text ، چون همیشه و در همه حالات id اولویت و تقدم دارد.


راهنما

المنت **h1** باید دارای کلاس pink-text باشد. 

المنت **h1** باید دارای کلاس blue-text باشد. 

المنت **h1** باید دارای id با مقدار orange-text باشد. 

فقط یک المنت **h1** باید داشته باشید. 


در المنت **style** باید یک id با نام orange-text داشته باشید. 

المنت **h1** نباید هیچ style به صورت inline داشته باشد. 

المنت **h1** باید نارنجی باشد. 

اعمال اولویت یا override کردن کلاس‌ها با استایل‌های inline :

- ✓ ما ثابت کردیم که id می‌تواند بدون در نظر گرفتن تقدم و تاخر و مکانش در استایل‌ها، کلاس را override کند، راه‌های دیگری برای override کردن وجود دارد:
- ✓ inline style ها رو یادتون هست؟


یک inline style بنویسید که المنت **h1** را سفید کند. 


استایل‌های inline این شکلی بودن: 

```
<h1 style="color: green;">
```

کلاس‌های **blue-text** و **pink-text** باید سر جای خودشون باشن. 

المنت **h1** باید دارای کلاس **pink-text** باشد. 

المنت **h1** باید دارای کلاس **blue-text** باشد. 

المنت **h1** باید دارای id با نام **orange-text** باشد. 

المنت **h1** باید سفید باشد. 

- ✔ ما ثابت کردیم که inline style می‌تواند تمام استایل‌های قبلی در المنت **style** را override کند.
- ✔ اما صبر کنید، یک راه آخری هم برای override کردن در CSS وجود دارد. این متد قوی‌ترین متد از بین همه این متدها هستش. اما قبل از این‌که بخوایم انجامش بدیم بیاین در مورد این حرف بزنینم که اصلا چرا ما باید override انجام بدیم.
- ✔ در بیشتر مواقع ما احتمالا از کتابخانه‌های CSS استفاده خواهیم کرد. این کتابخانه‌ها ممکن است به طور اتفاقی استایل‌هایی که خودتان نوشته‌اید را override کنند. در این حالت اگر می‌خواهید مطمئن شوید که استایلی که به یک المنت می‌دهید اعمال خواهد شد، می‌توانید از کلیدواژه **!important** استفاده کنید.
- ✔ بیایید به کلاس pink-text خودمون برگردیم، یادتان باشه که این کلاس با کلاسی دیگر، یک id و یک استایل inline قبلا override شده است.

تمرین

حالا بیایید در کلاس pink-text کلیدواژه important! را به color اضافه کنیم تا ۱۰۰٪ مطمئن بشیم که رنگ المنت **h1** صورتی خواهد بود.

اگر بخوام یک مثال بزوم که چطور این کار رو انجام بدید، این کد رو مشاهده کنید:

```
color: red !important;
```



راهنما

- 🔥 المنت **h1** باید دارای کلاس pink-text باشد.
- 🔥 المنت **h1** باید دارای کلاس blue-text باشد.
- 🔥 المنت **h1** باید دارای id با نام orange-text باشد.
- 🔥 المنت **h1** باید دارای استایل خطی color: white; باشد.
- 🔥 در کلاس pink-text خصوصیت color باید دارای کلیدواژه important! باشد تا تمام استایل‌های قبلی را override کند.
- 🔥 المنت **h1** باید صورتی باشد.

- ✔ تا اینجا هر وقت می‌خواستیم به المنت‌ها رنگ بدهیم از اسامی رنگ استفاده می‌کردیم، اما بدانید که راه‌های دیگری هم هست. یکی از این راه‌ها استفاده از کدهای هگزادسیمال هست که اصطلاحاً hex code هم بهش می‌گن.
- ✔ ما معمولا وقتی صحبت از دسیمال می‌کنیم منظورمان اعداد در مبنای ۱۰ هستند. که با اعداد ۰ تا ۹ شناخته می‌شوند.
- ✔ هگزادسیمال یا هگز بر مبنای ۱۶ هستند. به این معنا که از ۱۶ نماد مشخص استفاده می‌کنند.
- ✔ مثل اعداد دسیمال نمادهای ۰ تا ۹ بیانگر مقادیر ۰ تا ۹ هستند.
- ✔ و سپس A,B,C,D,E,F به ترتیب بیانگر مقادیر ۱۰ تا ۱۵.
- ✔ در نهایت، ۰ تا F بیانگر یک عدد هگزادسیمال است که ۱۶ مقدار ممکن را در اختیار ما قرار می‌دهد.
- ✔ در CSS، ما می‌توانیم از ۶ عدد هگزادسیمال برای نمایش دادن یک رنگ استفاده کنیم.
- ✔ هر جفت به ترتیب برای مقادیر یک رنگ، قرمز(R)، سبز(G)، آبی(B).
- ✔ برای مثال کمترین مقادیر ممکن را که بدهیم به رنگ سیاه می‌رسیم: #000000.

```
body {
  color: #000000;
}
```

رنگ black در استایل المنت **body** را با مقدار هگزادسیمال سیاه #000000 جایگزین کنید. 

المنت body باید دارای رنگ پس زمینه سیاه باشد. 
 به جای استفاده از کلمه black برای رنگ سیاه، باید از کد هگزادسیمال آن یعنی #000000 استفاده کنید. 

- ✓ کد هگز از ۶ رقم هگزادسیمال برای نمایش یک رنگ استفاده می‌کرد، که ۲ تا ۲ تا نمایانگر r,g,b بودن.
- ✓ از سه رنگ قرمز، سبز و آبی می‌توانیم بیش از ۱۶ میلیون رنگ تولید کنیم.
- ✓ برای مثال، رنگ نارنجی، قرمز خالص است که با کمی سبز ترکیب شده است و هیچ آبی در آن وجود ندارد.
- ✓ اگر بخواهیم کد هگز نارنجی را بنویسیم به این شکل می‌شود: FFA500.
- ✓ عدد صفر پایین‌ترین رقم در کد هگز می‌باشد و نمایانگر نبودن کامل یک رنگ می‌باشد.
- ✓ هروقت صفر است یعنی آن رنگ وجود ندارد.
- ✓ عدد F بالاترین رقم در کد هگز می‌باشد و نمایانگر نهایت شفافیت یک رنگ می‌باشد.

تمرین

رنگ‌های کلاس‌ها را به جای black با رنگ‌های متناسب خودشان از جدول زیر جایگزین کنید: 

Color	Hex Code
Dodger Blue	#1E90FF
Green	#00FF00
Orange	#FFA500
Red	#FF0000

راهنما

- 🔥 المنت **h1** با متن I am red! باید قرمز باشد.
- 🔥 به جای استفاده از اسم red باید از جدول کد هگزادسیمال را قرار دهید.
- 🔥 المنت **h1** با متن I am green! باید سبز باشد.
- 🔥 به جای استفاده از اسم green باید از جدول کد هگزادسیمال را قرار دهید.
- 🔥 المنت **h1** با متن I am dodger blue! باید سبز باشد.
- 🔥 به جای استفاده از اسم dodgerblue باید از جدول کد هگزادسیمال را قرار دهید.
- 🔥 المنت **h1** با متن I am orange! باید سبز باشد.
- 🔥 به جای استفاده از اسم orange باید از جدول کد هگزادسیمال را قرار دهید.

- ✓ هگز کد رنگ قرمز #FF0000 می‌تواند کوتاه شود و به این شکل در بیاید: #F00 .
- ✓ در این حالت برای هر کدام از رنگ‌های قرمز، سبز و آبی یک رقم در نظر می‌گیریم.
- ✓ در این حالت تمام رنگ‌هایی که در دسترس داریم حدود ۴۰۰۰ رنگ خواهد بود.
- ✓ مرورگر رنگ #F00 را تفسیر می‌کند به #FF0000 و هگز کد کامل را در نظر می‌گیرد.

تمرین

به جای رنگ‌های موجود از هگز کد کوتاه شده آن‌ها استفاده کنید:

Color	Short Hex Code
Cyan	#0FF
Green	#0F0
Red	#F00
Fuchsia	#F0F

راهنما

- 🔥 المنت **h1** با متن I am red! باید قرمز باشد.
- 🔥 به جای استفاده از هگز کد #FF0000 باید از جدول کد هگزادسیمال کوتاه شده را قرار دهید.
- 🔥
- 🔥 المنت **h1** با متن I am green! باید سبز باشد.
- 🔥 به جای استفاده از هگز کد #00FF00 باید از جدول کد هگزادسیمال کوتاه شده را قرار دهید.
- 🔥
- 🔥 المنت **h1** با متن I am cyan! باید سبز باشد.
- 🔥 به جای استفاده از هگز کد #00FFFF باید از جدول کد هگزادسیمال کوتاه شده را قرار دهید.
- 🔥
- 🔥 المنت **h1** با متن I am fuchsia! باید سبز باشد.
- 🔥 به جای استفاده از هگز کد #FF00FF باید از جدول کد هگزادسیمال کوتاه شده را قرار دهید.

اگر بخواهیم رنگ سیاه را با این روش نمایش بدهیم به این شکل عمل می‌کنیم: `rgb(0, 0, 0)` ✓

مقدار `rgb` برای رنگ سفید به این شکل خواهد بود: `rgb(255, 255, 255)` ✓

به جای استفاده از ۶ رقم مانند هگز کد، در `rgb`، برای میزان شفافیت هرکدام از رنگ‌ها، یک رقم از ۰ تا ۲۵۵ در نظر می‌گیریم. ✓

در این حالت بیشتر از ۱۶ میلیون رنگ را می‌توانیم تولید کنیم. ✓

می‌خواهیم رنگ پس زمینه `body` را نارنجی کنیم: ✓

```
body {  
background-color: rgb(255, 165, 0);  
}
```

رنگ پس‌زمینه `body` را با استفاده از `rgb value` سیاه کنید! 📄

رنگ پس‌زمینه المنت `body` باید سیاه باشد. 🏠

برای تعیین رنگ پس زمینه سیاه برای `body` باید از `rgb value` استفاده کنید. 🏠

✓ درست مانند کد هگز، می‌توانیم رنگ‌های rgb را با تغییر دادن ارقام آن با هم ترکیب کنیم.

تمرین

به جای هگز کدهای رنگ در المنت **style**، از مقدار rgb استفاده کنید.

Color	RGB
Blue	rgb(0, 0, 255)
Red	rgb(255, 0, 0)
Orchid	rgb(218, 112, 214)
Sienna	rgb(160, 82, 45)

راهنما

🔥 المنت **h1** با متن I am red! باید قرمز باشد.

🔥 برای این رنگ باید از rgb استفاده کنید.

🔥 المنت **h1** با متن I am orchid! باید سبز باشد.

🔥 برای این رنگ باید از rgb استفاده کنید.

🔥 المنت **h1** با متن I am blue! باید آبی باشد.

🔥 برای این رنگ باید از rgb استفاده کنید.

🔥 المنت **h1** با متن I am sienna! باید حنایی باشد.

🔥 برای این رنگ باید از rgb استفاده کنید.

✓ یکی از امکانات عالی در CSS ، متغیرها هستند. می‌توانیم برای مقادیری که شاید در بسیاری از جاها یکسان هستند و تکرار می‌شوند یک متغیر در نظر بگیریم و همه را با هم مقدار دهی کنیم.

تمرین

در کلاس penguin ، مقدار black را به gray ، مقدار gray را به white ، مقدار yellow را به orange تغییر دهید.

راهنما

- در کلاس penguin مقدار penguin-skin-- را به gray تغییر دهید. 🧴
- در کلاس penguin مقدار penguin-belly-- را به white تغییر دهید. 🧴
- در کلاس penguin مقدار penguin-beak-- را به orange تغییر دهید. 🧴

- ✓ برای ساخت متغیرهای دلخواه‌مان در CSS فقط کافی است قبل از اسم متغیر دو علامت خط تیره یا hyphen قرار دهیم. مثال: `--penguin-skin: gray;`
- ✓ دستور بالا یک متغیر با نام `--penguin-skin` می‌سازد و مقدار `gray` را به آن می‌دهد.
- ✓ شما می‌توانید از این متغیر هر جایی در CSS خودتان استفاده کنید تا به دیگر خصوصیت‌ها مقدار بدهید.

در کلاس `penguin` یک متغیر با نام `--penguin-skin` بسازید و مقدار `gray` را به آن بدهید. 

در کلاس `penguin` باید یک متغیر با نام `--penguin-skin` داشته باشید و مقدار آن `gray` باشد. 

✓ بعد از این که متغیر دلخواه خودتان را ساختید می‌توانید از آن برای مقدار دادن به دیگر خصوصیت‌ها استفاده کنید. فقط کافی است نام متغیر را به جای مقدار مورد نظر قرار دهید:

```
background: var(--penguin-skin);
```

✓ دستور بالا باعث می‌شود که رنگ پس زمینه المنتی که این خصوصیت را به آن داده‌اید خاکستری شود، چون ما قبلاً گفته بودیم که `--penguin-skin` دارای رنگ `gray` می‌باشد.

✓ یادتان باشد نام متغیر باید دقیق نوشته شود.

تمرین

رنگ پس‌زمینه (`background`) المنت‌هایی که دارای کلاس‌های `penguin-top`، `penguin-bottom`، `right-hand` و `left-hand` هستند را به `--penguin-skin` تغییر دهید.

راهنما

- ✎ خاصیت `background` کلاس `penguin-top` باید برابر با `--penguin-skin` باشد.
- ✎ خاصیت `background` کلاس `penguin-bottom` باید برابر با `--penguin-skin` باشد.
- ✎ خاصیت `background` کلاس `right-hand` باید برابر با `--penguin-skin` باشد.
- ✎ خاصیت `background` کلاس `left-hand` باید برابر با `--penguin-skin` باشد.

✓ اگر به هر دلیلی متغیری که به یک المنت نسبت می‌دهیم مقدار نداشته باشد یا غلط املایی داشته باشد یا در کل در دسترس نباشد می‌توانید کاری کنیم که مرورگر یک مقدار جایگزین را برای آن در نظر بگیرد، به این مقدار **Fallback value** می‌گویند.

✓ fallback برای بالا بردن سازگاری (compatibility) مرورگرها استفاده نمی‌شود و این قابلیت در مرورگرهای IE کار نمی‌کند. بلکه برای این استفاده می‌شود که در صورت عدم وجود متغیر مرورگر یک رنگ جایگزین برای نمایش دادن داشته باشد.
مثال:

```
background: var(--penguin-skin, black);
```

✓ اگر متغیر ما تعریف نشده باشد، پس‌زمینه سیاه خواهد بود. از این موضوع می‌توانید در دیباگ کردن (اشکال‌زدایی پروژه) استفاده کنید.

تمرین

📄 به نظر می‌رسد رنگ پس‌زمینه کلاس‌های `penguin-top` و `penguin-bottom`. مشکل دارند و نمایش داده نمی‌شوند.

📄 شما باید به جای اینکه غلط تایپی را رفع کنید یک Fallback value با مقدار `black` به رنگ پس‌زمینه این کلاس‌ها بدهید.

راهنما

🧪 در مقدار خصوصیت `background` کلاس `penguin-top` باید یک Fallback value با مقدار `black` وجود داشته باشد.

🧪 در مقدار خصوصیت `background` کلاس `penguin-bottom` باید یک Fallback value با مقدار `black` وجود داشته باشد.

- ✓ زمانی که سروکار ما با مرورگرها هست و برنامه نهایی ما در مرورگر اجرا خواهد شد باید آگاه باشیم که ممکن است بعضی مواقع مشکلات عدم سازگاری یا compatibility بوجود بیاید. به همین دلیل مهمه که برای جلوگیری از چنین مشکلاتی fallback در نظر بگیریم.
- ✓ وقتی مرورگر CSS هایی که ما نوشته ایم را اصطلاحاً parse (پارس) می کند یا برای اجرا و نمایش تجزیه و تحلیل میکند اگر به خصوصیتی برسد که برایش ناشناس است آن را نادیده می گیرد.
- ✓ اگر برای تعیین رنگ پس زمینه یک سایت از متغیرهای CSS استفاده کنید، مرورگر internet explorer پس زمینه را نادیده می گیرد چون از متغیرهای CSS پشتیبانی نمی کند.
- ✓ در این حالت اگر هیچ مقدار دیگری برای رنگ پس زمینه تعیین نکرده باشیم مرورگر برای آن رنگ از رنگ پس زمینه پیش فرضی که در مرورگر تعبیه شده استفاده خواهد کرد که این اصلاً قابل قبول نیست و ممکن است تمام طراحی ما را بهم بریزد.
- ✓ اگر می خواهید که یک browser fallback داشته باشید و با خیال راحت به طراحی پردازید باید قبل از استفاده از متغیرها، از حالت معمول که قبلاً توضیح داده ایم برای مقاردهی به خصوصیت ها استفاده کنید.
- ✓ در این حالت مرورگرهای قدیمی مقداری برای fallback می بینند و آن را نمایش می دهند و مرورگرهای جدید و بروزتر اول fallback را می بینند و وقتی از آن که عبور می کنند به مقداری که توسط متغیرها تعیین کرده ایم می رسند و آن را اجرا می کنند.

تمرین

📄 برای مقاردهی به رنگ پس زمینه کلاس red-box. از متغیر استفاده شده است.

برای این که استایل خودمان را با مرورگرهای قدیمی تر سازگار کنیم، قبل از جایی که از متغیرها استفاده کرده ایم یک بار دیگر خصوصیت background را بنویسید و مقدار آن را قرمز (red) قرار بدهید.

راهنما

🔧 برای داشتن یک fallback ، در کلاس red-box. باید قبل از background فعلی که موجود است یک background دیگر با مقدار red داشته باشیم.

توضیحات

- ✔ وقتی یک متغیر درست می‌کنید، این متغیر برای استفاده در دسترس شما هست اما فقط در آن سلکتوری که آن را آن‌جا تعریف کرده‌اید.
- ✔ این متغیر همچنین در تمام فرزندان آن سلکتور هم در دسترس است.
- ✔ در این حالت می‌گوییم متغیرهای CSS به ارث رسیده‌اند، مانند خصوصیت‌های معمولی که برای فرزندان به ارث می‌رسند.
- ✔ اگر بخواهیم از ارث بری طوری استفاده کنیم که متغیرها برای تمام سلکتورها به ارث برسند باید این متغیرها را در جایی به اسم المنت `root`: تعریف کنیم.
- ✔ `root`: اصطلاحاً یک سلکتور سودو کلاس (pseudo-class selector) است که با المنت ریشه سند ما یعنی `html` مطابقت دارد.
- ✔ اگر متغیرها را در `root`: تعریف کنیم آن‌ها به صورت **globally** برای تمام سلکتورها در دسترس خواهند بود و در همه سلکتورها می‌توانیم از آن‌ها استفاده کنیم.

تمرین

در سلکتور `root`: ، یک متغیر با نام `--penguin-belly` با مقدار `pink` تعریف کنید. خواهید دید که این متغیر در اختیار تمام فرزندان که از این متغیر استفاده کرده‌اند قرار می‌گیرد و رنگشان تغییر می‌کند.

راهنما

🔧 متغیر `--penguin-belly` باید در `root`: تعریف شود و مقدار آن `pink` باشد.

- وقتی که متغیر خودمان را در root: تعریف می‌کنیم، مقدار آن برای تمام صفحه در دسترس خواهد بود. ✓
- شما می‌توانید بعداً این متغیرها را در المنتی خاص مجدد تعریف کنید و مقدارشان را بازنویسی (overwrite) کنید. ✓

تمرین

مقدار متغیر --penguin-belly را در کلاس penguin به white تغییر دهید.

راهنما

در کلاس penguin باید متغیر --penguin-belly را مجدد بنویسید و مقدارش را برابر white قرار دهید. 🏠

- ✔ متغیرهای CSS می‌توانند استفاده از media query را ساده کنند.
- ✔ برای نمونه، متغیرها بر اساس اندازه صفحه نمایش می‌توانند تغییر کنند. می‌توانید تعریف کنیم که اگر صفحه نمایش کوچک یا بزرگ یا هر سایزی که ما تعریف می‌کنیم بود متغیرها چه مقداری داشته باشند.

تمرین

- داخل سلکتور root: مدیا کوئری که داریم، متغیر --penguin-size را مقداردهی مجدد کنید و آن را 200px قرار دهید.
- همچنین متغیر --penguin-skin را مقداردهی مجدد کنید و آن را black کنید.
- سپس صفحه خود را تغییر سایز دهید تا تغییرات را مشاهده کنید.

راهنما

- 🔥 در المنت root: داخل media query باید مقدار متغیر --penguin-size برابر 200px باشد.
- 🔥 در المنت root: داخل media query باید مقدار متغیر --penguin-skin برابر black باشد.