

در این پروژه ما از مفاهیم زیر استفاده کردیم که در ادامه توضیح خواهیم داد :

1. Semantic tags in HTML5
2. External CSS file
3. Font-awesome svg/icon library (cdn version)
4. CSS selectors such as :
  - a. All
  - b. Element
  - c. ID
  - d. Class
  - e. Direct child
  - f. first/last/nth child
  - g. Attribute
  - h. Pseudo
5. Flexbox CSS3 module
6. Grid CSS3 module
7. CSS variable

(1) خب اولین مفهوم تگ های مفهومی در HTML5 هستند که تفاوت این تگ ها با تگ های عادی از نظر موتور جستجو قابل تشخیص است و باعث فهمیده شدن بهتر وبسایت ما توسط خزنده جستجوگر میشود این تگها شامل تگ های section , nav, header و غیره میباشد .

(2) در این روش ما بجای نوشتن کدهای سی/اس/اس در همان فایل با پسوند .html و درون تگ style ، در واقع کد های سی/اس/اس را درون فایل با پسوند .css ذخیره میکنیم و سپس آن فایل را با استفاده از تگ link درون تگ head اضافه یا import میکنیم

(3) این کتابخانه یک کتابخانه تحت cdn است که به ما کمک میکند تا فقط با تغییر کلاس مربوط به تگ i و اضافه کردن تگ link مربوط به این کتابخانه درون تگ head ، به سرعت و سادگی هرچه تمام تر آیکون های متنوعی را درون سند خود داشته باشیم.

(4) از انتخابگر یا سلکتور های سی/اس/اس زیر استفاده کردیم :

(a) انتخابگر همه المنت ها که با \* در فایل سی/اس/اس قابل مشاهده است و وقتی به کار ما می آید که بخواهیم یک سری استایل را به همه المنت ها اعمال کنیم مثلا در کد ما صفر کردن مارجین و پدینگ و اعمال box-sizing و تغییر فونت برای کل سند html .

(b) سلکتور المنت که ساده ترین نوع انتخاب کردن است و کافی است که نام آن عنصر را نوشته و درون آکولاد مربوطه استایل مد نظر خودمان را وارد کنیم ، باید دقت شود که این استایل به همه المنت های داخل سند اعمال میشود .

(c) سلکتور یا انتخابگر آیدی، در این روش ما المنت را درون فایل سی/اس/اس انتخاب میکنیم با توجه به مقدار مشخص شده اتریبیوت (صفت) آیدی درون فایل html و با علامت # یعنی

HTML:

```
<a id="example"> TEXT </a>
```

CSS:

```
#example { STYLE }
```

(d) سلکتور کلاس دقیقاً مشابه سلکتور آیدی است با این تفاوت که مقدار صفت آیدی را از فایل html میخواند و بجای علامت # از نقطه برای انتخاب آن استفاده میکنیم.

(e) این انتخابگر همه فرزندان های مستقیم را انتخاب میکند که در مثال زیر با رنگ قرمز مشخص شده اند :

HTML:

```
<div>
  <p></p>
  <p></p>
  <p></p>
  <p></p>
  <div>
    <p></p>
  </div>
</div>
```

CSS:

```
div > p { STYLE }
```

(f) با این سه شبه کلاس یا شبه سلکتور ما میتوانیم اولین فرزند یا آخرین فرزند یا n امین فرزند را انتخاب کنیم.

(g) در این انتخابگر ما یک المنت را طبق صفتی که دارد انتخاب میکنیم مثلاً داریم که :

HTML:

```
<div class="search-container">
  <input type="submit" class="btn" value="جستجو"/>
  <input type="search" placeholder="آموزش خود را جستجو کنید"
name="search" id="search" />
</div>
```

CSS:

```
.search-container input[type="submit"] {
  background: var(--primary-color);
  color: white;
  margin-right: 4px;
}
.search-container input[type="search"] {
  width: 100%;
  direction: rtl;}
```

در اینجا تگ های input را برحسب صفت type انتخاب کردیم

(h) شبه کلاس ها که در بالا یکی از آنها را دیدیم (مورد f) به صورت زیر هستند

CSS :

```
.option-box a:first-child {  
  background: var(--primary-color);  
  color: white;}
```

(5) ماژول فلکس باکس تحت نسخه سوم سی اس اس ۳ ، در این ماژول ما به المنت پدر مقدار عرض و ارتفاع را میدهم و با پراپرتی های مخصوص این ماژول در سی اس اس به عناصر فرزند استایل میدهم و موقعیت آنها را در صفحه مشخص میکنیم ، استفاده از این ماژول و ماژولی که در بخش بعدی بررسی میکنیم ما را از استفاده از float و clear کردن بعد از استفاده از آن راحت میکند :

CSS:

```
.filter-product {  
  width: 100%; عرض  
  height: 35vh; ارتفاع  
  display: flex; استفاده از ماژول  
  justify-content: space-evenly; محل قرار گیری افقی  
  align-items: center; محل قرار گیری عمودی  
}
```

(6) در این ماژول (ماژول گرید) همانند ماژول قبلی ما یک عرض و ارتفاع برای تگ پدر تعریف کردیم و با استفاده از پراپرتی های مخصوص این ماژول عناصر را در شبکه تولید شده جایگذاری کردیم :

CSS:

```
.resources {  
  width: 95%; عرض  
  height: 100%; ارتفاع  
  display: grid; استفاده از ماژول  
  grid-template-columns: repeat(4, 1fr); چهار ستون  
  grid-template-rows: repeat(3, 1fr); سه سطر  
  grid-gap: 25px; فاصله بین ستون ها  
  margin: 0 auto;  
}
```

7) متغیر ها در CSS به ما این کمک را میکنند که مثلا در پروژه ما که در اکثر جاهای پروژه ما از رنگ بنفش استفاده میکنیم ، در واقع به جای تکرار خط کدی مانند زیر

CSS :

```
Background یا color: purple;
```

ابتدا در ROOT فایلمان به صورت زیر عمل کنیم و یک متغیر بسازیم

```
:root {  
  --primary-color: purple;  
  --gray-bg: #f2f2f2;  
}
```

و حالا در هر جایی از سند CSS نیاز به رنگ بنفش یا `purple` داشتیم از متغیر به صورت زیر استفاده کنیم :

```
i {color: var(--primary-color);}
```

و یا

```
input:focus {  
  border: var(--primary-color) 1.5px solid;  
  outline: none;  
}
```

یا

```
.create-shop {  
  background: var(--primary-color);  
  color: white;  
}
```