

تمرین A02-06: مهمترین units در CSS

تقسیم بندی کلی واحدهای اندازه گیری یا CSS Units

واحدهای اندازه گیری در **CSS** به دو بخش کلی تقسیم می شوند.

۱- واحدهای اندازه گیری مطلق یا **Absolute Lengths** :

واحدهای اندازه گیری مطلق یک مقدار ثلثت دارند. یعنی عددی که همراه واحد مربوطه می آید، دقیقاً همان اندازه ای است که در صفحه وب ظاهر می شود. واحدهای اندازه گیری مطلق برای استفاده در صفحات وب خیلی توصیه نمی شوند و در صورت نیاز بایستی به مقدار کم استفاده شوند، زیرا اینگونه واحدها برای هماهنگ شدن از نظر اندازه در دستگاه ها با رزولوشن های مختلف چندان مناسب نیستند. اما می توانند جهت لایه بندی صفحه پرینت همچون چاپ بفرمهای تبلیغاتی و... مناسب واقع شوند.

cm	mm	in
px	pt	pc

واحد های مطلق عبارتند از:

- پیکسل (px)
- نقطه یا پوینت (pt)
- پیکا (pc)
- سانتی متر (cm)
- میلی متر (mm)
- اینچ (in)

حال به تعریف هر کدام از این واحدها می پردازیم :

واحد px یا پیکسل : در بین واحدهای اندازه گیری مطلق، واحد اندازه گیری **px** کاربرد متداول تری دارد. به صورت پیشفرض، اندازه پیکسل ها در دستگاه های مختلف می تواند متفاوت باشد، اما **CSS** این مشکل را با در نظر گرفتن اندازه یک پیکسل در دستگاهی با تراکم پیکسل **۹۶ DPI (نقطه در اینچ)** در نظر گرفته است. بدین ترتیب، با توجه به نامتغیر بودن واحد اینچ، ابعاد یک پیکسل در تمامی دستگاه ها یکسان خواهد بود.

واحد pt یا نقطه : واحد نقطه، شباهت بسیاری به واحد پیکسل دارد. این واحد، ثابت و نامتغیر است و در تمامی دستگاه ها به یک اندازه ظاهر می شود. تفاوت واحد نقطه و پیکسل، تنها در اندازه آن هاست.

واحد pc یا پیکا : واحد پیکا، واحدی است که به اندازه **px** و **pt** استفاده نمی شود. اندازه آن هم یک مقدار نامتغیر است.

واحد های اینچ، میلی متر و سانتی متر : این واحدها، واحدهایی هستند که همگی آن ها را می شناسیم و در واقع با این واحدها بزرگ شدیم. از این واحدها، به اندازه واحد های قبلی استفاده نمی شود، چون کاربردشان در مسائل دیگه ای است.

واحد اندازه گیری	توضیح
cm یا سانتی متر	تنظیم اندازه بصورت سانتیمتر
mm یا میلی متر	تنظیم اندازه بصورت میلیمتر
in یا اینچ	۱ اینچ برابر است با ۹۶ پیکسل
px یا پیکسل	تنظیم اندازه بصورت پیکسلی
pt یا پوینت	۱ پوینت برابر است با ۱۰۷۲ از ۱ اینچ
pc یا پیکا	۱ پیکا برابر است با ۱۲ پوینت

۲- واحدهای اندازه گیری نسبی یا Relative Lengths :

واحدهای اندازه گیری نسبی را برای هر تگ **HTML** نسبت به یک تگ **HTML** دیگر و یا نسبت به واحد اندازه گیری دیگر تعیین می کنند. به عبارتی دیگر واحدهای نسبی، واحدهایی هستند که اندازه ی ثابتی ندارند و بر اساس اندازه های دیگر در صفحه وب سایت ما تعریف می شوند.

em	ex	ch
rem	vw	vh
vmin	vmax	%

مهم ترین این واحد ها، عبارتند از:

- percent (%)
- vw - vh
- em
- rem

واحد اندازه گیری	توضیح
em	به نسبت اندازه فونت تگ والد تغییر میکند
ex	به نسبت اندازه ارتفاع حرف X (ایکس انگلیسی) کوچک
ch	به نسبت اندازه عرض کاراکتر
rem	به نسبت اندازه فونت ریشه صفحه تگ (html)
vw	به نسبت ۱ درصد از عرض کل صفحه نمایش
vh	به نسبت ۱ درصد از ارتفاع کل صفحه نمایش
vmin	به نسبت ۱ درصد از عرض کل صفحه نمایش (در ابعاد کوچکتر)
vmax	به نسبت ۱ درصد از ارتفاع کل صفحه نمایش (در ابعاد بزرگتر)
% درصد	به نسبت اندازه کل صفحه (بصورت درصدی)

تعریف هر کدام :

واحد % یا درصد : این واحد، عمومی ترین واحد مستقلی هست که در **CSS** مورد استفاده قرار می گیرد.

یک عنصر **div** با کد زیر رو فرض کنید :

```
div {  
    width: 90%;  
}
```

این عنصر، همواره دارای عرضی برابر با نود درصد عرض عنصر والد خواهد بود. چه در صفحه دسکتاپ، چه در صفحه تبلت و چه در صفحه تلفن همراه. در صورت تغییر کردن عرض عنصر والد، عرض عنصر **div** هم به تناسب تغییر خواهد کرد، اما همیشه نود درصد عنصر والد خودش خواهد ماند.

واحد های vw و vh : این واحد ها، مخفف و کوتاه شده عبارات **viewport width** و **viewport height** محسوب می شوند. **viewport**، به قسمتی از صفحه وب سایت که جلوی ما قرار دارد و ما آن را می بینیم اطلاق می شود. این واحد ها، مشابه واحد درصد عمل می کنند با این تفاوت که اندازه آن ها به اندازه عنصر والد بستگی ندارد بلکه مستقیماً به طول و عرض **viewport** ما بستگی دارد. به کد زیر توجه کنید:

```
div {  
    width: 40vw;  
}
```

کد بالا، بیانگر این است که عرض عنصر **div**، در هر دستگاه و صفحه نمایشی، همواره باید چهل درصد عرض **viewport** باشد. مشابه **vw** که عرض عنصر را بر حسب درصدی از عرض **viewport** تعیین می کند، **vh** هم ارتفاع عنصر را بر حسب درصدی از ارتفاع **viewport** تعریف می نماید.

واحد em : این واحد، تغییرات خود را بر حسب سایز فونت عنصر والد انجام می دهد. برای درک بهتر این واحد، به کد زیر توجه کنید:

کد **CSS :**

```
div .parent {  
    font-size: 16px;  
}
```

```
div .child{
    width: 3em;
}
```

واحد **em**، همواره به عنوان شاخصی از سایز فونت عنصر والد عمل می‌کند. بر اساس کد بالا، عرض عنصر **child** سه برابر سایز فونت عنصر والد یا ۴۸ پیکسل خواهد بود.

واحد rem : این واحد، مشابه واحد **em** عمل می‌کند. با این تفاوت که از سایز فونت عنصر والد تاثیر نمی‌گیرد، بلکه از سایز فونت خود عنصر **HTML** تاثیر می‌پذیرد. به همین دلیل، به این واحد **rem** یا **root em** (مربوط به عنصر ریشه وب سایت یا **HTML**) گفته می‌شود. به کد زیر توجه کنید:

کد **CSS** :

```
* {
    font-size: 20px;
}
div .parent{
    font-size: 16px;
}
div .child{
    width: 3rem;
}
```

بر خلاف واحد **em**، توسط واحد **rem**، عرض عنصر **child** به اندازه ۶۰ پیکسل تنظیم خواهد شد (سه برابر سایز فونت عنصر **root**).

۱- بهترین واحدهای اندازه گیری نسبی جهت طراحی وب سایت های ریسپانسیو ، **em** و **rem** هستند.

۲- واحدهای اندازه گیری **vmax** ، **vh** ، **vw** و **vmin** به عنوان واحدهای نسبی وابسته به **viewport** هستند.

۳- واحدهای **vmax** ، **vh** ، **vw** و **vmin** در طراحی سایت های مقیاس پذیر یا واکنشگرا موثر خواهد بود.

نکته : واحد های اندازه گیری **em** ، **px** و **rem** به عنوان بهترین واحدهای اندازه گیری در طراحی سایت واکنشگرا هستند.

نکات مهم : ۱- هرگز بین عدد و واحد اندازه گیری **CSS** فاصله نگذارید، چرا که یک **خطای سینتکسی** خواهد بود. به عنوان مثال مقادیر **۵ em** و یا **۲۰٪** که بین عدد و واحد اندازه گیری فاصله افتاده اشتباه است و حالت صحیح آن **۵em** و **۲۰٪** است.

۲- در صورتی که مقدار ۰ را برای ویژگی **CSS** مربوطه در نظر گرفتید، نیاز به استفاده از **CSS Units** یا واحدهای اندازه گیری ندارید.

۳- در برخی ویژگی های **CSS** شما اجازه استفاده از اندازه های **منفی** یا **negative** را دارید. ولی در برخی ویژگی های دیگر **CSS** اجازه استفاده از اندازه های **منفی** یا **negative** را ندارید.