**بسم الله الرحمن الرحیم**

**درس اول :**

برای ایجاد لینک در css از دستور زیر استفاده می‌کنیم

 <link rel="stylesheet" href="../funcss/1.css">

برای نوشتن تگ های css در ابتدا باید تگ های Html را بنویسیم

برای تغییر background image باید از تگ body استفاده نمود

برای اینکه به کل یک تگ Css را اعمال کنیم در فایل css نام تگ Html را نوشته و یگ brace در آن باز می کنیم

به عنوان مثال

body

{

}

**Background-color :**

این دستور در css باعث تغییر رنگ در پس زمینه می‌گردد

body

{

    background-color:aquamarine;

}

رنگ را می توان به صورت کدهای RGB وارد کرد

   background-color:rgb(234,208,234)

روش دیگر بر اساس کد رنگ Hex است

 background-color: #ffe2c2;

روش دیگر بر اساس نام رنگ است

  background-color:bisque;

 }

روش دیگر بر اساس کد Hsl است

    background-color: hsl(33, 100%, 88%);

روش دیگر بر اساس نام hwb است

 background-color:hwb(33 77% 0%);

**دستور background-image :**

می توان از تصاویر نیز برای background استفاده کرد

برای اینکار از دستور background-image استفاده کرد

   background-image:url(../image/1.jpg);

می توان داخل “” یا داخل ‘’ نیز آدرس تصویر را قرار داد

**دستور background--repeat :**

برای جلوگیری از تکرار عکس در background می باشد

 background-repeat: no-repeat;

**دستور background-size :**

برای تنظیم ابعاد عکس به کار می رود که دارای چندین حالت است در حالتی که مقدار cover بگیرد کل صفحه را پوشش می دهد

   background-size: cover;

**نحوه استفاده از کلاس ها در Css :**

برای این کار در کنار نام کلاس از علامت . استفاده می‌کنیم و در کنار تگ html از class استفاده می کنیم

<p class="a">Hi</p>

**نحوه استفاده از id در Css :**

برای این کار در کنار نام id از علامت # استفاده می‌کنیم و در کنار تگ html از عبارت id استفاده می‌کنیم

<p id="a">Hi</p>

#a

{

   color: brown;

}

**تغییر رنگ Font :**

دستور color در تگ <p> باعث تغییر رنگ در رنگ فونت می شود

  color: brown;

**دستور Font-size :**

این دستور باعث افزایش سایز Font می شود و برا اساس px است می توان با درصد هم کار کرد

    color: blue;

    font-size: 55;

**دستور font – weigth :**

باعث می شود که بتوانیم font را bold کنیم

    font-weight:bold;

**دستور Font-family:**

باعث می‌شود که بتوانیم نوع Font را عوض کنیم

 font-family:'Tahoma';

در این حالت می‌توان چند فونت را با هم قرار داد که اگر یک فونت به هر دلیلی عمل نکرد فونت بعدی عمل کند هر فونت در داخل ‘ ‘ قرار می‌گیرد و در صورت وجود چند فونت با , از هم جدا می‌شوند

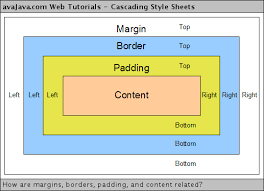
**دستور Text – align :**

با این دستور می توان چینش متن را مرتب کرد در حالت پیش فرض به صورت Left است ولی می توان آن را Rigth و Center نمود

**Comment :**

برای Comment گذاری در Css از /\* \*/ استفاده می‌کنیم

**Margin , Padding :**

****

دستور Margin دارای چهار جهت است

1. Margin-top که فاصله را از بالا می‌دهد بر اساس دو مقدار % و px است
2. margin-top: 50px;
3. Margin-rigth که از سمت راست به مقدار 50px فاصله می‌دهد
4. margin-right: 50px;
5. }

توجه به این نکته ضروری است که در margin باید عنصر content از سمت مورد نظر بتواند فاصله بگیرد و اگر نتواند از سمت مقابل کم می شود

و اگر میزان فاصله زیاد باشد و از حد مجاز بیشتر شود متن را می شکند و در دو خط نمایش می دهد

**درس آخر css مقدماتی :**

**انواع Button :**

درhtml چند تگ برای Button وجود دارد

   <button>button</button>

   <button disabled="disabled">Disable</button>

   <button type="reset">Reset</button>

<button type="submit">Submit</button>

**دستور Input :**

با استفاده از این دستور نیز می توان دکمه های مختلف ساخت

**Input button :**

<input type="button" value="">

برای ساخت button به کار می رود مقدار داخل value مقدار داخل دکمه را نشان می دهد

**Input checkbox :**

برای ساخت checkbox به کار می رود

<input type="checkbox" name="" id="">

Name و id برای backend به کار می‌رود

**Input color :**

این تگ برای ساخت یک Check color به کار می رود و یک لیست از رنگ ها ایجاد می‌کند

<input type="color" name="" id="">

**Input date :**

این تگ برای ساخت یک check date به کار می رود در این دستور نیز Name و id برای backend به کار می رود

 <input type="date" name="" id="">

**Input date-time local :**

این تگ برای ورود یک input که تاریخ و ساعت را نشان می دهد به کار می رود

<input type="datetime-local" name="" id="">

در این تگ نیز name و id برای backend می‌رود

**Input Email :**

این تگ برای ایجاد یک ورود Email یه کار می‌رود

 <input type="email" name="" id="">

در این تگ نیز name و id برای backend می‌رود

**Input file :َ**

این تگ برای انتخاب یک فایل به کار می رود

 <input type="file" name="" id="">

در این تگ نیز name و id برای backend می‌رود

**Input password :**

این تگ برای قرار دادن پسورد به صورت ستاره به کار می‌رود

 <input type="password" name="" id="">

در این تگ نیز name و id برای backend می‌رود

**Input radio :**

این تگ برای قرار دادن یک چک باکس به صورت دایره به کار می رود با این تفاوت که نمی توان تیک آن را برداشت

 <input type="radio" name="" id="">

در این تگ نیز name و id برای backend می‌رود

# راهنمای کامل EM و REM

[[](https://css-tricks.ir/author/seyedi/)مجتبی سیدی](https://css-tricks.ir/author/seyedi/)

یکی از روش هایی که واقعا باعث بهبود پیاده سازی در طراحی وب می شود استفاده از **واحدهای نسبی** مثل em و rem می باشد.

برای ساختن کامپوننت های مستقل که دوباره قابل استفاده و یا اینکه به راحتی قابل ویرایش باشند, و اینکه بتوانیم با تغییرات کمی به طرح جدیدی از کامپوننت دست پیدا کنیم, به عنوان فقط یکی از نیازها باید از **واحدهای نسبی** (relative units) مناسب استفاده کنیم.

سوال اینجا است که **از کدام یک باید استفاده کرد؟** این سوال چندین سال است که مورد بحث قرار می گیرد و طرفداران هریک از این واحدها به دنبال این هستند که یکی را بر دیگری غالب بدانند.

در این مطلب و مطالب بعدی با هر دو واحد em و rem کاملا آشنا می شویم و همینطور نقاط مثبت و منفی هر کدام را بررسی کرده, و یاد می گیریم چگونه از این واحدها برای ساخت کامپوننت های ماژولار استفاده کنیم.

(اولین بار کلمه ماژول و ماژولار را در یک سفر در قطار از یک استاد دانشگاه در رشته مکانیک شنیدم. مطلب ویکی پدیا در مورد [طراحی ماژولار](https://en.wikipedia.org/wiki/Modular_design" \t "_blank) نیاز به ترجمه نیز دارد. در کل ایده طراحی ماژولار بیانگر ساختن یک سیستم از زیر سیستم های مستقل است که فواید بسیاری را در بر دارد.)

## EM چیست؟

یک **em** یک واحد [تایپوگرافی](https://fa.wikipedia.org/wiki/%D8%AA%D8%A7%DB%8C%D9%BE%D9%88%DA%AF%D8%B1%D8%A7%D9%81%DB%8C" \t "_blank) است که برابر با font-size کنونی تعیین شده می باشد.

یعنی چه؟ یعنی اگر عنصری دارای اندازه فونت 20px باشد پس 1em=20px است.

h1 { font-size: 20px } /\* 1em = 20px \*/

p { font-size: 16px } /\* 1em = 16px \*/

به عنوان نمونه برای عنصر h1 چون اندازه فونت 20 پیسکل تعیین شده است پس اگر برای این عنصر از واحد em برای تعیین برخی از ویژگی های آن استفاده کنیم هر em برابر با 20 پیکسل می باشد, مثلا:

h1 {

font-size: 20px; /\* 1em = 20px \*/

padding-right: 2em; /\* 2em = 2\*20 = 40px \*/

}

همانطور که دیدیم برای دانستن اندازه هر em در انتخابگر, به ویژگی font-size رجوع کردیم. حال اگر این ویژگی برای انتخابگر تعیین نشده بود چه؟

کافی است به ویژگی font-size پدر آن عنصر رجوع کنیم و اگر آن عنصر هم این ویژگی برایش تعیین نشده بود همینطور به بالا حرکت می کنیم تا در اجدادش عنصری پیدا کنیم که ویژگی font-size دارد و اگر در نهایت عنصری پیدا نشد, به عنصر ریشه که همان html است می رسیم و بصورت پیشفرض font-size برای html برابر با 16px می باشد مگر اینکه کاربر از طریق تنظیمات مرورگر این عدد را تغییر دهد. و یا ما در CSS اندازه فونت html را تغییر دهیم.

معمولا کار جالبی نیست که اندازه فونت html را تغییر دهیم, چون با این کار یک حالت پیشفرض را برای مرورگر کاربر تغییر می دهیم. حالتی که شاید کاربر به آن خو گرفته باشد. بهتر است بدانیم اگر font-size را برای html تعیین نکنیم مقدار آن 100% می باشد که همان 16 پیکسل مربوط به تنظیمات مرورگر است.

از em برای تعیین اندازه ویژگی font-size هم می توان استفاده کرد.

h1 {

font-size: 2em; /\* یعنی چی؟ \*/

}

اکنون اندازه حقیقی فونت برای انتخابگر h1 چند است؟ با توجه به بحثی که شد دوباره به اندازه فونت عناصر پدر رجوع می کنیم. این بار فرض کنید h1 فرزند header می باشد, داریم:

<header>

<h1>HAVE FUN!</h1>

</header>

header {

font-size: 18px;

}

header h1 {

font-size: 2em; /\* 2\*18 = 36px \*/

}

اکنون فرض کنید داخل عنصر h1 عنصری دیگر داریم که باید یک مقدار padding داشته باشد, محاسبه مقدار آن بصورت زیر است:

header {

font-size: 18px;

}

header h1 {

font-size: 2em; /\* 2\*18 = 36px \*/

}

h1 span {

padding-right: 0.5em; /\* 0.5\*36 = 18px \*/

}

همانطور که قبلا گفته شد چون عنصر span در حال حاضر font-size ندارد ما به font-size پدرش رجوع می کنیم.

(توجه داشته باشین که ***استفاده از نام تگ ها به عنوان انتخابگر روش مناسبی نمی باشد*** و در اینجا به منظور ساده تر بیان کردن موضوع به این صورت عمل می کنیم. دلیل این مورد هم واضح است چراکه ما اطمینان نداریم که در آینده تگ ها به همین صورت در پروژه بمانند شاید از تگ دیگری استفاده شد و غیره…, نباید استایل ما به این صورت شکننده باشد. ولی ما مطمئن هستیم که نام کلاس ها توسط هیچ تیم دیگری تغییر نخواهد کرد و اگر هم تغییر کند می دانند که باید در استایل هم آن نام را بروز کنند.)

اکثر افرادی که تازه با em آشنا می شوند, کمی برایشان اینکه 1em در مکان های مختلف عددی متفاوت است و باید دنبال حساب کردن آن باشند, گیج کننده و کمی خسته کننده است اما این فقط یک حس اولیه است. چرا که در خیلی مواقع شما نیاز به دانستن اینکه اندازه یک عنصر چند پیکسل است ندارید و مستقیم با خود em به ظاهری که می خواهید می رسید و نکته دیگر اینکه به دلیل فواید زیاد em در زمان پیاده سازی و نگه داری, همیشه با اشتیاق از آن استفاده خواهید کرد.

بعد از em به rem می رسیم…

## REM چیست؟

rem یا Root EM متولد شده است تا آن محساباتی که در em وجود دارند را نداشته باشد. این واحد تایپوگرافی برابر با font-size ریشه است یعنی همیشه برابر با اندازه فونتی که برای html تعیین می شود می باشد.

پیچیده ترین مورد بالا را این بار با rem می نویسیم:

header {

font-size: 18px;

}

header h1 {

font-size: 2rem; /\* 2\*16 = 32px \*/

}

h1 span {

padding-right: 0.5rem; /\* 0.5\*16 = 8px \*/

}

همانطور که مشاهده می کنید 1rem همیشه 16px است, مهم نیست که کجا استفاده می شود و مهم نیست که در اینجا پدر این عناصر اندازه فونت 18px دارد, بلکه همیشه مقدار font-size عنصر html تعیین کننده خواهد بود, که بصورت پیشفرض این مقدار 16 پیکسل است مگر اینکه تغییرش بدهیم.

**قابل اطمینان است, فهم آن ساده است.**

زمانی که بدانیم em چیست فهمیدن rem بسیار ساده است. قبول دارید؟