



طراحی وب سایت

CSS

فصل پنجم

مدرس : محمد امین زاده

HTML



CSS



Bootstrap

B

ویژگی display در CSS

با استفاده از ویژگی display در CSS می‌توان حالت یا نحوه نمایش یک تگ رو مشخص کرد، مثلا می‌توانیم تعریف کنیم که یک تگ بصورت بلاکی (Block) نمایش داده شود یا خیر مثلا بصورت اینلاینی (Inline)

برای مثال تگ P حالت نمایش به صورت بلاکی یا Block می‌باشد، این یعنی تمام عرض سطر را به خودشان اختصاص می‌دهند.

و در مقابل تگ‌هایی هم هستن که حالت نمایششان به صورت درون خطی می‌باشد، مثل تگ a و b و span و .. (این‌ها در یک سطر کنار همدیگه قرار می‌گیرند و فقط به اندازه خودشان فضا را اشغال می‌کنند.)

حال ما به راحتی با استفاده از ویژگی display در CSS می‌توانیم حالت نمایش یک تگ را به دلخواه خودمان در صورت نیاز تغییر بدهیم، مثلا می‌توانیم حالت نمایش تگ P که بلاکی هست را به اینلاینی یا Inline تغییر بدهیم.



ویژگی display در CSS



inline	تگ را به صورت درون خطی نمایش می دهد. (استفاده از Width و Height و Padding و.. تاثیری ندارد.)
inline-block	تگ را به صورت درون خطی نمایش می دهد. (استفاده از Width و Height و Padding و.. تاثیری دارد.)
block	تگ را به صورت تمام عرض (۱۰۰ درصدی) نمایش می دهد.
list-item	تگ را همانند آیتم های یک لیست نمایش می دهد. همانند تگ Li
none	برای مخفی / پنهان کردن تگ در صفحه
flex	تگ رو در حالت فلکس (تمام عرض) نمایش میدهد



CSS hover

ویژگی hover زمانی قابل مشاهده است که با ماوس بر روی آن عنصر بروید و تغییر را مشاهده خواهید کرد. بعد نوشتن نام کلاس، با اضافه کردن hover: بعد از آن، این ویژگی به آن عنصر اضافه می شود.

```
.text-hover:hover {  
    color: red;  
}
```

```
.img-hover:hover {  
    transform : scale(1.5);  
    -webkit-transform : scale(1.5); /* Safari 3-8 */;  
}
```



Transition in css

ویژگی transition یا در لغت انتقال، یکی از ویژگی‌های است که با استفاده از آن می‌توان به تگ‌های داخل صفحه افکت انتقال (Transition) داد.

منظور از افکت انتقال، مثلاً می‌توانیم کاری کنیم که وقتی ماوس بر روی یک تگ قرار گرفت آن تگ در مدت زمان مشخصی برای مثال رنگ زمینه آن عوض شود یا اندازه آن تگ به آرامی تغییر کند و...

Transition-Delay : تاخیر در اجرای انتقال

Transition-Duration : مدت زمان اجرای انتقال

Transition-Property : آن ویژگی که باید تحت تاثیر انتقال قرار بگیرد

Transition-Timing-Function : یک تابع برای اجرای عمل انتقال



Transition-Delay in css

با استفاده از این ویژگی می توان مدت زمانی را برای تاخیر در اجرای عمل انتقال مشخص کرد، برای مثال ۲ ثانیه طول بکشد تا عمل انتقال شروع شود.

مقداری که این ویژگی می پذیرد از نوع تایم یا Time می باشد، یعنی یک عدد به همراه پسوند s یعنی ثانیه و ms یعنی میلی ثانیه، برای مثال ۵ ثانیه را باید به این شکل بنویسیم. « 5s

```
.MyBox {  
  transition-delay: 1s;  
}
```



Transition-Duration in css

با استفاده از این ویژگی می‌توانیم مشخص کنیم که چند ثانیه طول بکشد تا انتقال انجام شود. برای این کار باید از واحد زمان (ثانیه یا میلی‌ثانیه) استفاده کنیم.

```
.MyBox {  
  transition-duration: 3s;  
}
```



Transition-Property in css

با استفاده از این ویژگی می‌توانیم مشخص کنیم که کدام ویژگی تحت تاثیر عمل انتقال قرار بگیرد، برای مثال وقتی ماوس بر روی تگ رفت، ویژگی width آن تحت تاثیر عمل انتقال قرار بگیرد.

مثال شماره ۱: فقط ویژگی width تحت تاثیر عمل انتقال قرار می‌گیرد.

```
.MyBox {  
  transition-property: width;  
}
```

مثال شماره ۲: ویژگی‌های width و heighth تحت تاثیر عمل انتقال قرار می‌گیرند.

```
.MyBox {  
  transition-property: width,height;  
}
```



example



```
.main-div {  
  background-color : #f0005a;  
  width : 100px;  
  height : 80px;  
}
```

در فایل **style.css**

```
.mybox {  
  transition-duration : 1s;  
  transition-property : width,height;  
}
```

```
.mybox :hover {  
  width : 150px;  
  height : 150px;  
}
```

```
<body>      در فایل index.html  
<p>  
  <b> لطفا ماوس را بروی باکس زیر ببرید </b>  
</p>  
<div class="main-div mybox"></div>  
</body>
```



Transition-Timing-Function in css

با استفاده از این ویژگی می توان نحوه (روش) انتقال را با استفاده از توابع مشخص کرد، یا به طور کلی با استفاده از این ویژگی می توان سرعت عمل انتقال را تغییر داد.
این ویژگی مقادیر زیر را می تواند بپذیرد :

linear : با یه سرعت یکسان شروع و تا پایان ادامه دارد

ease-in : در شروع آرام می شود

ease-out : در پایان آرام می شود

ease-in-out : در شروع و پایان آرام می شود

steps(int,start|end) : تکه تکه کردن انتقال از شروع یا پایان



example

در فایل **style.css**

```
.main-div-1 {  
  background-color : #f0005a;  
  width : 100px;  
  height : 50px;  
  transition-duration : 1s;  
}
```

در فایل **index.html**

```
<body>  
  <div class="main-div-1 mybox1"></div>  
  <div class="main-div-1 mybox2"></div>  
  <div class="main-div-1 mybox3"></div>  
  <div class="main-div-1 mybox4"></div>  
  <div class="main-div-1 mybox5"></div>  
</body>
```

```
.mybox1 {  
  transition-timing-function:linear;  
}  
.mybox2 {  
  transition-timing-function:ease-in;  
}  
.mybox3 {  
  transition-timing-function:ease-out;  
}  
.mybox4 {  
  transition-timing-function:ease-in-out;  
}  
.mybox5 {  
  transition-timing-function:steps(5,start);  
}  
.mybox1:hover, .mybox2:hover, .mybox3:hover, .mybox4:hover, .mybox5:hover {  
  width: 100%;  
}
```



animation-iteration-count

با استفاده از ویژگی **animation-iteration-count** در CSS می توان تعداد دفعات اجرا شدن انیمیشن را مشخص کرد، یعنی مثلا می توانیم مشخص کنیم که انیمیشن فقط ۲ بار اجرا شود یا مثلا تا بی نهایت اجرا شود.

این ویژگی ۲ مقدار زیر را می تواند بپذیرد :

number : به صورت عددی

infinite : نامحدود و تا همیشه اجرا می شود



example

مثال شماره ۱ : انیمیشن ۳ بار اجرا می شود.

```
.MyBox{  
  animation-iteration-count: 3;  
}
```

مثال شماره ۲ : استفاده از مقدار `infinite` یعنی اجرای بی نهایت

```
.MyBox{  
  animation-iteration-count: infinite;  
}
```



دستور keyframes در CSS

با استفاده از دستور `@keyframes` در CSS می‌توان نحوه اجرای انیمیشن‌ها را تعریف کرد، مثلا می‌توانیم مشخص کنیم یک باکس در مدت زمان ۵ ثانیه از رنگی به رنگ دیگر تغییر پیدا کند و...

ما می‌توانیم به راحتی از دیگر ویژگی‌های CSS در این دستور استفاده نماییم و نحوه اجرای انیمیشن را کنترل کنیم. انیمیشن‌ها برای اجرا شدن نیاز به ۲ نقطه اصلی دارن، یکی نقطه شروع و دیگری نقطه پایان، که این نقاط با استفاده از کلمات کلیدی زیر مشخص می‌شوند:

{ نام دلخواه @keyframes

{ 0% بخش استایل در این جا تعریف می‌شود. **}**

{ 25% بخش استایل در این جا تعریف می‌شود. **}**

{ 50% بخش استایل در این جا تعریف می‌شود. **}**

{ 75% بخش استایل در این جا تعریف می‌شود. **}**

{ 100% بخش استایل در این جا تعریف می‌شود. **}**

}



example

در این مثال و در طی بخش‌هایی که تعیین کرده‌اید، رنگ پس‌زمینه body در ۱۰ ثانیه تغییر پیدا می‌کند.

```
body{  
  animation: soora-learn 10s infinite;  
  color: #fff;  
  text-align: center;  
}  
  
@keyframes soora-learn {  
  0% { background: #2196F3; }  
  25% { background: #8AC007; }  
  50% { background: #9C27B0; }  
  75% { background: #607D8B; }  
  100% { background: #2196F3; }  
}
```



تمرین اول

در فایل `index.html` خود یک تگ `div` باز کرده و در آن یک تگ `img` تعریف کنید. به وسیله‌ی استایل‌دهی کاری کنید که با مواس بر روی عکس می‌رویم، کمی زوم شود و این زوم شدن در ۱ ثانیه اتفاق بیافتد و وقتی از روی عکس ماوس کنار رفت، در زمان ۱ ثانیه به حالت اولیه خود برگردد.





طراحی وب سایت

CSS

پایان فصل پنجم

مدرس : محمد امین زاده

HTML



CSS



Bootstrap

