

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

مقالات

CSS

منبع:

[www.Dev.ir](http://www.Dev.ir)

## ده نکته برای افزایش سرعت بارگزاری صفحات وب

افزایش سرعت بارگزاری صفحات یکی از مسائل مهم در طراحی صفحات وب می‌باشد. در صورتی که زمان بارگزاری یک صفحه وب طولانی باشد، معمولاً کاربران از مشاهده آن صفحه صرف نظر می‌کنند و دیگر به آن باز نمی‌گردند این به معنی از دست دادن کاربران سایت می‌باشد که برای مدیران یک سایت اصلاً خوشایند نیست. در ادامه ده روش موثر برای افزایش سرعت بارگزاری صفحات وب بیان می‌شود.

### ۱. استفاده از CSS برای ایجاد قالب و طرح صفحه به جای Table

CSS یا Cascading Style Sheets ها به دلایل زیر از Table ها سریعتر بارگزاری می‌شوند:

مرورگرها قبل از نمایش محتویات یک جدول دو بار آن را مرور می‌کنند. یک بار برای مشخص شدن ساختار آن و یک بار هم برای تعیین محتویات آن. جداول (Tables) در یک مرحله بر روی صفحه به نمایش در می‌آیند. هیچ قسمتی از یک جدول بر روی صفحه ظاهر نمی‌شوند تا اینکه تمام جدول همراه با محتویات آن به طور کامل بارگزاری و پردازش شود.

عمل موقعیت دهی و یا ایجاد یک فضای خاص در جداول معمولاً با استفاده از تصاویر کوچک خاصی (Spacer Images) صورت می‌پذیرد.

بطور کلی CSS ها کد کمتری نسبت به جداول نیاز دارند.

تمام کدهای مورد نیاز برای ایجاد طرح و نمای صفحه قابلیت جایگزین شدن با یک فایل CSS خارجی را دارند. که این فایل پس از یک بار فراخوانی در کامپیوتر کاربر ذخیره (cache) می‌شود. اما قالب‌های ایجاد شده با استفاده از جداول در هر صفحه HTML تکرار می‌شوند و با درخواست هر صفحه جدید دوباره باید بارگزاری (Download) شوند.

با استفاده از CSS ها امکان تعیین ترتیب بارگزاری صفحه وجود دارد. یعنی می‌توان محتویات اصلی صفحه را قبل از بارگزاری تصاویر با حجم بالا به نمایش درآورد. این کار قطعاً کاربران سایت شما را خوشحال خواهد کرد.

برای آشنایی بیشتر با CSS و کارهای جالبی که می‌توان با آن در یک سایت انجام داد می‌توانید از آموزشهای خوب و مفید سایت <http://www.htmldog.com> استفاده کنید.

### ۲. از تصاویر برای نمایش متن استفاده نکنید

در اینجا نیز CSS به ما کمک خواهد کرد. در مواردی که کار را می‌توان با CSS بطور کامل انجام داد از تصاویر استفاده نکنید. به کد زیر دقت کنید:

```
a:link, a:visited, a:active {
width: 7em;
font: bold 0.8em Georgia;
text-decoration: none;
display: block;
margin-left: 0;
margin-bottom: 0.5em;
margin-top: 0.5em;
color: white;
background: #39c;
border-left: 1px solid #6cf;
border-bottom: 1px solid #068;
border-right: 1px solid #068;
```

```
padding: 0.25em 0.5em 0.4em 0.75em;
border-top: 1px solid #6cf;
}
```

```
a:hover {
background: #28b;
padding: 0.35em 0.35em 0.25em 0.9em;
border-top: #069;
border-right: #6cf;
border-bottom: #6cf;
border-left: #069;
}
```

با استفاده از این کد می‌توانید یک کلید جذاب ایجاد کنید که با رفتن موس بر روی آن پایین می‌رود.

### ۳. فراخوانی تصاویر تزئینی بوسیله CSS

با CSS می‌توان تصاویر را بصورت قسمتی از یک زمینه (background) نمایش داد. بطور مثال یک تصویر ۲۰۰x۲۰۰ را می‌توان بصورت زیر نمایش داد:

```
<div class="pretty-image"></div>
```

کد HTML

```
.pretty-image {
background: url(filename.gif);
width: 200px;
height: 200px
}
```

کد CSS

در ابتدا شاید این کار بی معنی به نظر برسد اما این کار سرعت بارگذاری و نمایش صفحات را افزایش می‌دهد. بطور کلی مرورگرها تصاویر زمینه را بعد از بقیه اجزا بارگذاری می‌کنند. با استفاده از این تکنیک متن درون صفحه فوراً به نمایش درآمده و کاربر می‌تواند آن را مشاهده و در بین آن گردش کند و در همین هنگام تصاویر با حجم زیاد بارگذاری می‌شوند.

در این روش نمی‌توان از خصوصیت ALT استفاده نمود اگر واقعاً می‌خواهید که از این خصوصیت استفاده کنید، کد HTML بالا را بصورت زیر تغییر دهید.

```
</ "image src="spacer.gif" class="pretty-image" alt="description">
```

در اینجا spacer.gif یک تصویر 1x1 پیکسل شفاف (transparent) است که ۵۰ بایت حجم دارد. در این حالت ابتدا محتوای اصلی بارگذاری می‌شود و بعد از آن تصویر بزرگ و حجیم همراه با خصوصیت ALT بطور کامل بارگذاری می‌شود. دقت داشته باشید که این روش برای نمایش تصاویر تزئینی مناسب می‌باشد نه برای تصاویر حاوی اطلاعات. همچنین کاربرانی که CSS آنها غیر فعال است قادر به مشاهده تصویر (یا متن مربوط به ALT) نمی‌باشند.

## ۴. استفاده از انتخاب کننده‌های (selectors) مناسب

به کد نامناسب زیر توجه کنید:

```
<p class="text">This is a sentence</p>
<p class="text">This is another sentence</p>
<p class="text">This is yet another sentence</p>
<p class="text">This is one more sentence</p>
```

```
.text {
  color: #03c;
  font-size: 2em
}
```

به جای اختصاص دادن مقدار به هر پاراگراف، می‌توان همه را در یک تگ <div> قرار داده و مقدار را به آن اختصاص دهیم:

```
<div class="text">
<p>This is a sentence</p>

<p>This is another sentence</p>
<p>This is yet another sentence</p>
<p>This is one more sentence</p>
</div>
```

```
.text p {
  color: #03c;
  font-size: 2em
}
```

در ابتدا شاید این موضوع زیاد با اهمیت به نظر نرسد اما اگر شما بتوانید از آن بطور صحیح در تمام صفحه استفاده کنید به راحتی ۲۰٪ از حجم فایل شما کم خواهد شد. همچنین شاید توجه کرده باشید که مقادیر مربوط به رنگها کوتاه‌تر از حالت عادی است. #۰۳C کوتاه شده مقدار #۰۰۳۳CC است شما می‌توانید از این روش خلاصه کردن در هر جایی که مقادیر با این فرم قرار دارند استفاده کنید.

## ۵. استفاده از خلاصه نویسی خصوصیات در CSS

در زیر روش خلاصه نویسی بعضی المانهای CSS را مشاهده می‌کنید:

```
font: 1em/1.5em bold italic serif
```

```
font-size: 1em;
line-height: 1.5em;
font-weight: bold;
font-style: italic;
font-family: serif
```

Font

border: 1px black solid

border-width: 1px;  
border-color: black;  
border-style: solid

Border

background: #fff url(image.gif) no-repeat top left

background-color: #fff;  
background-image: url(image.gif);  
background-repeat: no-repeat;  
background-position: top left;

Background

margin: 2px 1px 3px 4px (top, right, bottom, left)

margin-top: 2px;  
margin-right: 1px;  
margin-bottom: 3px;  
margin-left: 4px

Margin

margin: 5em 1em 3em (top, left and right, bottom)

margin-top: 5em;  
margin-bottom: 1em;  
margin-right: 1em;  
margin-left: 4em

Margin

margin: 5% 1% (top and bottom, left and right)

margin-top: 5%;  
margin-bottom: 5%;  
margin-right: 1%;  
margin-left: 1%

Margin

این قوانین بطور عملی برای border و padding نیز قابل استفاده است.

## ۶. استفاده از فراخوانی نسبی

سعی کنید از آدرس دهی مطلق پرهیز کنید زیرا فضای بیشتری را اشغال می‌کند و تغییر دادن آنها در آینده مشکل است. یک مثال از فراخوانی مطلق بصورت زیر است:

```
<a href="http://www.URL.com/filename.htm">
```

که بهتر است به این صورت باشد:

```
<a href="filename.htm">
```

اما اگر فایل‌ها در فهرست‌های مختلف باشند از خلاصه نویسی‌های زیر استفاده کنید:

```
<a href="/filename.html">
فراخوانی http://www.URL.com/filename.html
```

```
<a href="/directory/filename.html">
فراخوانی http://www.URL.com/directory/filename.html
```

```
</."=a href">
فراخوانی index.html در فهرست جاری
```

```
</.."=a href">
فراخوانی index.html از یک فهرست بالاتر
```

```
<a href="../filename.html">
فراخوانی filename.html از یک فهرست بالاتر
```

```
</../.."=a href">
فراخوانی index.html از دو فهرست بالاتر
```

## ۸. حذف تگ‌های غیر ضروری META و مقادیر درون آن

بسیاری از تگ‌های META غیر ضروری هستند و کار زیادی انجام نمی‌دهند. در صورت علاقه می‌توانید لیست تگ‌های META را در این آدرس (<http://www.submitcorner.com/Guide/Meta>) مشاهده کنید. تگ‌های META با اهمیت برای موتورهای جستجو تگ‌های keywords و description هستند. البته استفاده نادرست و بیش از اندازه از آنها به تازگی باعث کاهش اهمیت آنها شده است. در هنگام استفاده از هرکدام از این تگ‌ها سعی کنید حجم محتوای آنها برای هر یک کمتر از ۲۰۰ کاراکتر (حرف) باشد. هر مقداری بیش از این باعث افزایش حجم صفحه شما خواهد شد. متا تگ‌های طولانی برای موتورهای جستجو مناسب نیستند زیرا کلمات کلیدی شما را کم رنگ می‌کنند.

## ۹. انتقال JavaScript و CSS درون صفحه به فایل‌های مستقل

برای استفاده از CSS که در فایل خارجی قرار دارد از کد زیر استفاده کنید:

```
<link type="text/css" rel="stylesheet" href="filename.css" />
```

و برای استفاده از JavaScript که در فایل خارجی قرار دارد از کد زیر استفاده کنید:

```
<script language="JavaScript" src="filename.js" type="text/javascript"> </script>
```

هر فایل خارجی یک بار فراخوانی شده و بعد از آن در کامپیوتر کاربر (برای استفاده‌های بعدی) ذخیره می‌شود. بجای قرار دادن JavaScript و CSS در تک تک صفحات HTML آنها را برای یک بار در یک فایل خارجی قرار دهید و از آن درون صفحات استفاده کنید. فراموش نکنید که هیچ محدودیتی برای استفاده از این فایل‌های خارجی وجود ندارد. برای مثال بجای ساختن یک فایل CSS بزرگ، یک فایل برای قسمت‌های یکسان در تمام صفحات و چند فایل هم برای قسمت‌هایی که در صفحات خاص استفاده می‌شوند بسازید.

## ۱۰. استفاده از / در انتهای آدرس فهرست‌ها:

بجای استفاده از این کد:

```
<a href="http://www.URL.com/directoryname" >
```

از کد زیر استفاده کنید:

```
<a href="http://www.URL.com/directoryname/" >
```

زیرا اگر از / در انتهای آدرس استفاده نکنید سرور متوجه نخواهد شد که اشاره به فایل شده یا فهرست. اما با اضافه کردن / سرور فوراً متوجه می‌شود که به یک فهرست اشاره شده و آنرا نمایش می‌دهد.

## استایل (CSS) چیست؟

- CSS مخفف کلمه Cascading Style Sheets میباشد.
- اولین ورژن استایلها در سال ۱۹۹۶ ساخته شد. که اولین ورژن آن CSS1 بود.
- دیگر ورژن آن CSS2 می باشد که در سال ۱۹۹۸ ساخته شد که بیشتر برای ویرایش صفحات، نحوه نمایشها تگها و ... کاربرد دارد.
- فایل استایلها با فرمت CSS ذخیره میشوند.

### انواع استایلها:

استایلها بر سه نوع میباشد:

۱. Inline Style: این نوع استایل در داخل تگهای HTML به کار گرفته میشوند، و این استایل فقط بروی همان تگ به خصوص تاثیر خواهد گذاشت و به صورت زیر نوشته خواهد شد.

```
<tag style="Style Declarations">
```

Example:

```
<p style="font-family: Tahoma">This is a paragraph</p>
```

در مثال بالا با استفاده از حالتی مختلف در استایلها نوع فونت آنرا تغییر دادیم و همانطوری که در تعریف این نوع استایل نوشتیم این استایل فقط و فقط تغییرات را بروی همین پاراگراف اعمال خواهد کرد.

۲. Embedded or Global Style: در این نوع، استایل نوشته شده در تمامی صفحه تاثیر خواهد گذاشت، اینرا بین دو تگ (<head></head>) به صورت زیر نوشته خواهد شد.

```
<style>
<!--
Style Declaration
-->
</style>
```

Example:

```
<head>
<style type="text/css">
<!--
p { color: red; text-align: left; font-size: 8pt }
-->
</style>
</head>
```

این نوع حالت از تغییرات تعریف شده بروی تمامی تگها پاراگراف اعمال خواهد شد که رنگ متن: قرمز، جهت متن: چپ، و اندازه فونت ۸ پوینت خواهد بود. توجه داشته باشید در این مثال سه حالت مختلف یک استایل را برای تگ پاراگراف تعریف کردیم که با استفاده از (;) حالتی مختلف آن از هم جدا شدند.

اگر بخواهید این حالت نوشته شده را برای چند تگ دیگر اعمال کنید می توانید تگها را با استفاده از کاما (,) از هم جدا کنید برای مثال:

h1, h2, h3, h4 { color: gold; font-family: sans-serif }

۳. Linked or External style sheet: نوع آخر هم معروف به استایل‌های خارجی هستند که به صورت لینک فایل استایل را به صفحات خود لینک می‌دهید. فایل‌های استایل با فرمت CSS ذخیره میشوند و به صورت زیر به صفحات لینک داده میشود.

```
<link href="url" rel="relation type" type="link type">
```

**Example:**

```
<head>  
<link rel="stylesheet" type="text/css" href="mystyle.css">  
</head>
```

در این نوع از استایل‌ها بعد از اینکه فایل استایل خود را آماده کردید آنرا با فرمت CSS ذخیره کرده و آنرا همانند بالا فایل استایل را به قالب لینک می‌کنیم. برای راحت بود در کار می‌توانید ابتدا استایل‌ها را به صورت Internal استفاده کرده و بعد از اینکه کارتان تمام شد کدهای نوشته شده بین دو تگ (<style></style>) را در نت پد کپی کرده و آنرا با فرمت CSS ذخیره کرده و سپس با استفاده از فرمول بالا آن فایل را به قالب لینک دهید. یکی از مزیت‌های این نوع استایل این می‌باشد که می‌توانید با داشتن یک فایل استایل برای چندین صفحه استفاده کنید و از دیگر مزیت‌های آن با این کار کدهای قالب را به چند فایل تقسیم کرده و قالب را منظم تر میشود.

## صفحات دو ستونی در CSS

در هنگام طراحی صفحات با استفاده از CSS بعضی مرتکب این اشتباه میشوند که بخش کناری صفحه را در اول صفحه قرار میدهند و بخش اصلی را در زیر آن؛ به همین علت بازدیدکنندگان تا پایان کار لود شدن بخش کناری باید صبر کنند.

در قسمت کناری صفحه اغلب کدهای javascript برای استفاده از سرویسهای سایتهای مختلف (مثل سایتهای اندازه گیری بازدیدکنندگان یا سایت blogrolling برای لینکهای سایت) وجود دارد به همین خاطر کار لود شدن آن قسمت مدت به نسبت زیادی را به خود اختصاص میدهد. به همین دلیل در صورت قرار گیری این بخش در قسمت بالاتر صفحه کار لود شدن نوشته های اصلی صفحه به تعویق خواهد افتاد.

این کار بر خلاف فلسفه طراحی با استفاده از CSS است. ما با استفاده از CSS سعی میکنیم کاری کنیم که مطالب مهمتر زودتر از بقیه موارد ظاهر شوند.

برای قرار دادن بخش اصلی در ابتدای صفحه تگ DIV مربوط به بخش اصلی را در ابتدا و تگ DIV مربوط به بخش کناری را بعد از آن قرار دهید. سپس خصوصیت float مربوط به style بخش اصلی را تعیین کنید. مثال:

```
<html>
<head>
<style>
.Page_body { background-color: #FF0000;float:left }
.Page_side { background-color: #0000FF; }
</style>
</head>
<body>
<div class="Page_body">Page contents here</div>
<div class="Page_side">Page Side Contents</div>
</body>
</html>
```

## رنگ ScrollBar صفحه سایت خود را عوض کنید

برای اینکه رنگ ScrollBar صفحه سایت خود را عوض کنید کافی است کد زیر را در ابتدای برنامه خود اضافه کنید. (تقریباً بعد از تگ )

```
<STYLE type=text/css>
BODY {
SCROLLBAR-FACE-COLOR: #739cbd;
SCROLLBAR-HIGHLIGHT-COLOR: #ffffff;
SCROLLBAR-SHADOW-COLOR: #1f1f1f;
SCROLLBAR-3DLIGHT-COLOR: #739cbd;
SCROLLBAR-ARROW-COLOR: #ffffff;
SCROLLBAR-TRACK-COLOR: #cee6f2;
SCROLLBAR-DARKSHADOW-COLOR: #ffffff
}
</STYLE>
```

خب يك توضیحي در مورد این سورس بدم...  
اول اینکه این چند خط کد از دستورات CSS پیرو می‌کنه که در تگ Style قرار گرفته است.  
**Scrollbar-face-color**: در این قسمت رنگ اصلی اسکرول را تعیین می‌کنید.

**Scrollbar-shadow-color**: در این قسمت رنگ سایه اسکرول را مشخص می‌کنید (خط موجود در سمت راست و پایین اسکرول)

**Scrollbar-highlight-color**: در این قسمت رنگ قسمت روشن اسکرول را مشخص می‌کنید (خط موجود در سمت چپ و بالای اسکرول)

**Scrollbar-3dhigh-color**: این پارامتر رنگ بوردر (خط) دور اسکرول را مشخص می‌کند. (خط کلی در دور تا دور اسکرول)

**Scrollbar-darkshadow-color**: خط دوم سایه اسکرول را مشخص می‌کند. (این خط بر روی خط سایه که در سمت راست و بالای اسکرول است قرار می‌گیرد)

**Scrollbar-track-color**: این قسمت رنگ زمینه اسکرول را مشخص می‌کند.

**Scrollbar-arrow-color**: رنگ فلش‌های موجود در کلیدهای بالا و پایین اسکرول را تغییر می‌دهد.

## CSS و کاربرد آن در طرح های امروزی

CSS جایگزینی برای نمایش مبتنی بر HTML پیشنهاد می کند و اکنون به وسیله ی انجمن طراحی وب پذیرفته شده است . سازگاری CSS نیازمند برخی تغییرات از سوی تولید کنندگان وب است . CSS بسیار حساس به نحو دستوری است ، و اشتباهات کوچک می تواند باعث شکست کامل يك قانون CSS شود . بنابراین ، طراحان باید از مشخصات آن آگاه باشند .

متأسفانه ، حتی اگر از قوانین پیروی کنید ، خطاهای آشکاری در پیاده سازی مرورگرهای پیشرفته وجود دارد . چیزهای زیادی وجود دارد که باید درباره ی CSS بدانید ، و امکانات جدید هر لحظه اضافه می شود . CSS1 تعیین اولین شیوه نامه ، بیش از ۵۰ خاصیت را تعریف می کند ، و CSS2 بیش از ۵۰ خاصیت دیگر را . CSS2 تسهیلات تغییر مکان که CSS-P نامیده می شود را به وجود آورده است . نگارش جدید که CSS3 نام دارد قصد دارد امکانات ارائه بیشتری را به وجود آورد ، و با فناوری های دیگر ، نظیر اسکریپت و گرافیک های برداری ، بیشتر یکپارچه شود .

مهم است به خاطر داشته باشید که HTML اساس و پایه ی يك صفحه وب است . CSS در واقع ، به طور مستقیم به استفاده صحیح از عناصر HTML یا XHTML استناد می کند . ولی ، اگر به کد ناهنجاری محدود شده باشند لزوماً به طور قابل پیش بینی ای کار نمی کنند .

هر چند واضح است که می خواهیم به برجسب های منطقی معنی بدهیم ، ولی مواظب باشید که این به شما مجوز نمی دهد از CSS به جای کد HTML استفاده کنید . قوانینی که ارائه محدود برخی برجسب ها را تغییر می دهند نباید به کار روند . برای مثال در اینجا باعث می شویم که برجسب های **<b>** به طور غیر معمولی عمل کنند و تمام اشکال تزئینی پیوندها را حذف نمایند :

```
b {font-style:italic; font-weight:normal;}
a {text-decoration:none; color: black; }
```

( اگر به دلیل خاصی این قوانین را بازنویسی کرده اید ، حتماً باید آن ها را توضیح گذاری کنید ) به خاطر مشکل معنای پیش فرض در HTML ، شاید برجسب های **<div>** و **<span>** فوق العاده مفید باشند . هم چنین برجسب **<div>** يك برجسب بلوکی است که هیچ نمایش پیش فرضی ندارد ، بنابراین در حالت دهی به بخش های بزرگ يك سند مفید است . برای مثال :

```
<div style="background: lightblue; font-weight: bold; color: black;">
<p>This Paragraph is highlighted in blue.</p>
</div>
```

به طور مشابه برجسب **<span>** هیچ معنی یا نمایش از پیش تعریف شده ای ندارد ، ولی وقتی يك برجسب هیچ کاراکتر بازگشت به ابتدای خط ( carriage return ) را نداشته باشد ، هنگام الصاق اطلاعات سبک به چند کلمه یا حرف ، مفید به نظر می آید ، مانند مثال زیر :

```
<p>Calling out<span style="background-color:#99CC00; font-weight:bold;
color:#000000"></span>isn't hard with SPAN</p>
```

روش های متفاوتی برای اضافه کردن CSS به سند HTML وجود دارد ، و همه آن ها داری مزایا و معایبی است که قبل از اینکه روشی در نظر گرفته شود باید به دقت سبک و سنگین شود . ابتدا می توانید در بخش سر صفحه ی يك سند HTML با مشخص ساختن رابطه پیوندی به يك شیوه نامه ی خارجی ، به آن متصل شوید . برای مثال :

```
<link rel="stylesheet" href="styles/global.css" media="screen" />
```

روش دوم برای اضافه کردن CSS ، نهفته کردن آن است . وقتی يك شیوه نامه نهفته می شود ، قوانین شیوه نامه را به طور مستقیم درون سند HTML بنویسید . شیوه نامه ی گسترده شدن درون سند ( document-wide style ) يك راه بسیار ساده برای شروع استفاده از CSS است . يك برچسب <style> درون سر صفحه ی سند HTML قرار دهید . به دلیل اینکه چند شکل از شیوه نامه ها ممکن است اضافه شود ( فراتر از شکل استاندارد CSS ) باید هنوز هم صفت type را اضافه کنید تا نشان دهید که کدام شکل از شیوه نامه را به کار می برید ، البته صرفنظر از پشتیبانی پیش فرض مرورگر از فناوری های شیوه نامه ها . به مثال های زیر توجه کنید :

شیوه نامه گسترده درون برچسب استایل

```
<style type="text/css">
body {background-color:#666666; font-size:9px; }
h1 {color:#990000;}
</style>
```

شیوه نامه پیوندی درون برچسب استایل

```
<style type="text/css">
@import url( wp-layout.css );
</style>
```

يك نگرانی در مورد شیوه نامه ها ی گنجانده شده این است که همه ی مرورگرها اطلاعات شیوه نامه را به درستی درک نمی کنند . برای اجتناب از مشکلات ، اطلاعات شیوه نامه ها را با استفاده از روش توضیح گذاری HTML ، تبدیل به توضیحات کنید . به طوری که این توضیحات روی صفحه نمایش نشان داده نمی شوند یا توسط مرورگرهای قدیمی تر ، به طور نادرستی تفسیر نمی گردند . ( نمی دانم نویسنده این مطلب را برای چه عصری از مرورگرها نوشته ! اما احتیاط شرط عقل است ! مهم تر از اون این يك استاندارد است ! پس طراحان موظف به انجام آن هستند )

CSS گسترده شده درون سند در مقایسه با CSS پیوندشده دارای معایب قابل ملاحظه ای است زیرا باید در هر صفحه که استفاده می شود يك نسخه از آن کپی شود . این کار باعث سخت تر شدن این گونه سایت ها می شود و از حافظه نهائی محلی مرورگر نیز به شکل بهینه استفاده نمی شود .

آخرین روش به کار بردن CSS اعمال آن درون مولفه های ویژه با استفاده از صفت رایج style است . نیازی نیست که این گونه اطلاعات مربوط به شیوه نامه ها از مرورگرهایی که نسبت به شیوه نامه ها اطلاعی ندارند مخفی شود ، زیرا مرورگرها صفاتی که نمی فهمند را در نظر نمی گیرند . به مثال زیر توجه کنید :

```
<p style="color:#3366FF; font-size:12px;">This is red blue text!</p>
```

ولی بسیاری از مزایای اصلی استفاده از CSS به این روش به دست نمی آید . زیرا CSS خیلی به برچسب وابسته هستند . در واقع ، همه توافق دارند که این شیوه خیلی بهتر از به کار بردن برچسب های جدید جهت تشریح ظاهر صفحه نمی باشد . ولی برخی از طراحان از CSS به این روش استفاده می کنند تا از کد HTML قدیمی به CSS پلی بزنند .