

طراحی صفحات وب

Web Programming

مهندس حمید رضا نیرومند

در دانشگاه جامع علمی-کاربردی

<http://niroomand.ir>

منابع:

- **Beginning Web Programming with HTML, XHTML, and CSS, 2nd ed.**
(Author: Jon Duckett) (<http://yourl.ir/xhtmll>)
- <http://w3schools.com>

توجه:

- استفاده از جزوه بدون حضور در کلاس‌ها پیشنهاد نمی‌شود.
- این یک انتشار غیررسمی از جزوه و ویژه دانشجویان مهندس نیرومند است.
- ویدئوهای آموزشی طراحی وب مقدماتی و پیشرفته مهندس نیرومند از طریق لینک زیر قابل تهیه است:
 - http://yourl.ir/niroomand_products
- این، نسخه ۱.۰.۰ از جزوه است و به مرور غنی‌تر خواهد شد. (لطفاً اشتباهات سهوی که مشاهده می‌کنید را به ایمیل info@niroomand.ir ارسال نمایید).

فهرست مطالب

۶	HTML :
۸	توضیح تگ‌های مهم در HTML:
۸	تگ DOCTYPE:
۹	تگ head:
۹	تگ title:
۹	تعریف SEO:
۱۰	تگ body:
۱۱	آدرس‌دهی:
۱۲	تگ‌های مربوط به شکل دهی متن:
۱۲	تیتراهای صفحه:
۱۳	تگ‌های sub و sup:
۱۳	تگ pre:
۱۴	تگ abbr:
۱۴	لیست‌ها:
۱۶	تگ font:
۱۶	تگ meta:
۱۸	لینک‌ها (links and Navigations) :
۱۸	انواع لینک:
۱۸	تگ a:
۲۱	تشریح اجزای یک URL:
۲۲	تگ img:
۲۴	تگ center:
۲۴	فرمت عکس در طراحی وب:
۲۶	جدول (Table) :
۲۶	صفات تگ table:

۲۷.....	نکاتی درباره قالب اصلی سایت:
۲۷.....	تگ td:
۲۹	قالب (Frames):
۲۹.....	معایب فریم:
۳۰	ایده ها:
۳۱	انواع سایت از نظر تبادل داده با کاربر:
۳۲	فرم (Form):
۳۳	تگ input:
۳۵	فیلد textarea:
۳۶.....	صفات مشترک فیلدها:
۳۷	دکمه ها:
۳۸	تگ video:
۳۸	تگ audio:
۳۹	CSS
۳۹.....	روش های اعمال کدهای CSS به صفحه:
۴۱.....	شکل دهی به متن با CSS:
۴۱.....	صفات background:
۴۲.....	روش دوم استفاده از کدهای CSS:
۴۲.....	انتخاب گرهای CSS یا Selectors:
۴۳.....	تگ div و span:
۴۳.....	رخدادها در (Events in CSS):
۴۴.....	ترفند CSS sprite:
۴۴.....	background-repeat:
۴۵.....	background-position:
۴۶.....	ساخت قالب سایت با div:
۴۷.....	border:
۴۷.....	صفات border:

۴۸.....	:margin
۴۹.....	:padding
۴۹.....	:float
۴۹.....	صفاتی برای تعیین ارتفاع:
۵۰.....	:Position
۵۱.....	صفت z-index
۵۱.....	صفت :overflow
۵۳.....	ضمیمه فونت به سایت:
۵۴.....	Frameworks
۶۱.....	JavaScript
۶۱.....	انواع زبان‌های برنامه نویسی وب از نظر محل اجرای کد:
۶۱.....	انواع اعمال کدهای جاوا اسکریپت به صفحه وب:
۶۳.....	JQuery
۶۴.....	PHP
۶۴.....	تفاوت‌های بین دو زبان PHP و ASP :
۶۸.....	CMS
۶۹.....	مراحل طراحی سایت:

:HTML

- ¹ HTML به معنای زبان نشانه‌گذاری ابرمتن است.
- «ابرمتن» یعنی هر چند ما در این زبان یک سری متن را تایپ می‌کنیم اما این متون، معنایی فراتر از متن دارند.
- در این جزوه، HTML5 تدریس می‌شود.
- هر چند html به حروف کوچک و بزرگ حساس نیست اما بهتر است نام تگ‌ها و نام صفات با حروف کوچک انگلیسی نوشته شوند.

ساده‌ترین برنامه به زبان HTML:

```
<!DOCTYPE html>
<html>
  <head>
    <title>First Page</title>
  </head>
  <body>
    <h1 align="center">Hello World!</h1>
  </body>
</html>
```

اولین گام در طراحی هر سایتی:

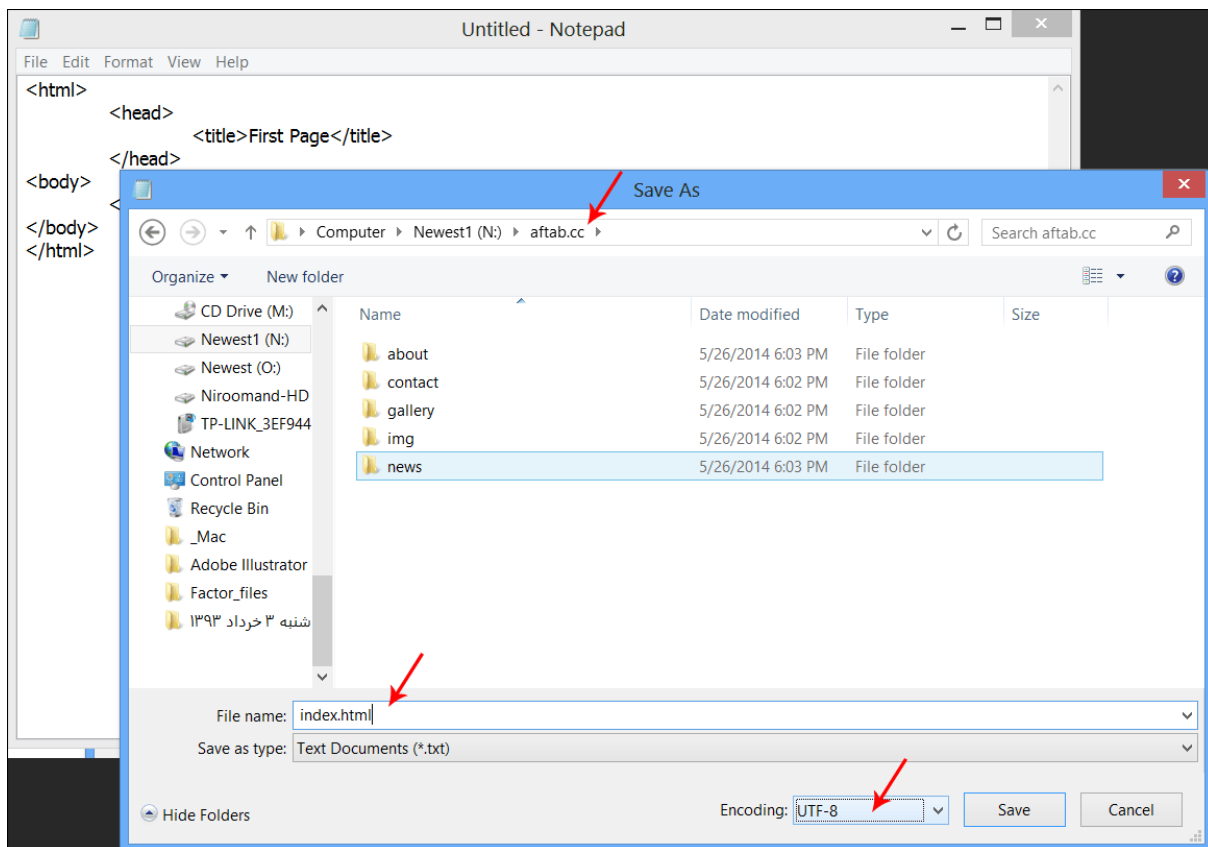
اولین گام در شروع به طراحی، ایجاد یک پوشه برای آن سایت است. این پوشه می‌تواند به هر نامی انتخاب شود اما چه بهتر که هم‌نام دامنه‌ی سایت باشد؛ مثلاً `aftab.cc`. اما نکته مهم این است که با خودتان قرارداد کنید که هر چیزی که قرار است در سایت شما استفاده شود، باید به این پوشه منتقل شود. عکس‌ها، ویدئوها، اصوات و ... همه باید ابتدا به این پوشه منتقل شوند سپس در کدنویسی به آن‌ها اشاره شود. یعنی نمی‌شود یکی در درایو C شما باشد، یکی در درایو D شما، یکی روی دسکتاپ شما!!

در این پوشه، زیرپوشه‌های مرتبط با محتوای خود بسازید. مثلاً پوشه‌ای به نام `img` برای عکس‌های سایت. پوشه‌ای به نام `video` برای ویدئوهای سایت...

نحوه‌ی اجرای کدهای HTML:

برنامه بالا را در NotePad تایپ کرده و در پوشه سایت که قبلاً ساخته‌ایم، با نام `index.html` و با Encoding، UTF-8 ذخیره می‌کنیم. (توجه داشته باشید که Encoding پیش‌فرض یعنی ANSI زبان فارسی را پشتیبانی نمی‌کند.)

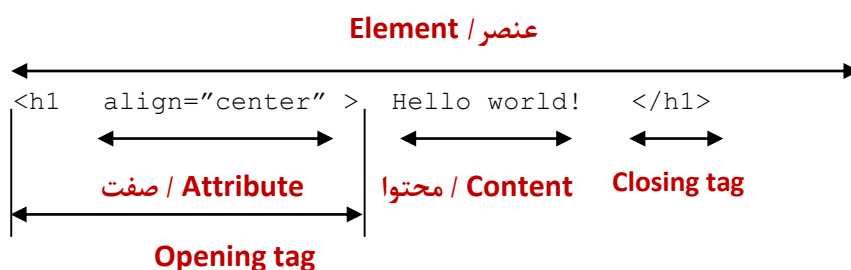
¹ Hyper Text Markup Language



مفهوم tag:

- تگ یعنی برچسب
- ما عناصری داریم؛ مانند عکس، متن، ویدئو و ... قرار است آن‌ها را با برچسب‌ها به دیوار سایت خود بچسبانیم.
- در HTML محدوده‌ی عناصر با تگ‌ها مشخص می‌شوند. (مانند {} و () که برای مشخص کردن یک محدوده در زبان هستند)

تگ‌ها و اجزای تشکیل دهنده آن:



- صفات‌ها به تگ دستور می‌دهند که به چه روشی عمل کنند. در html درج صفات‌ها به این صورت است:

name="value"

"مقدار صفت"=نام صفت

- مقدار هر صفت باید در دابل کوتیشن (") قرار بگیرد.
- قبل از نام هر صفت، یک فاصله بزنید.

فضای سفید در HTML:

در HTML هر تعداد فاصله (Space) که بیش از یک فاصله باشد، یک فاصله به حساب می‌آید. یعنی به طور مثال اگر در بخش کد نویسی بین دو کلمه، ۱۰۰ فاصله وارد کنیم در مرورگر یک فاصله بیشتر نمایش داده نمی‌شود. برای درج فاصله‌های بیش‌تر از یکی، از عبارت ` ` استفاده کنید.

توضیح تگ‌های مهم در HTML:

تگ DOCTYPE:

این تگ در بر دارنده‌ی کل محتویات صفحه است که همه‌ی تگ‌ها را احاطه می‌کند. تنها عبارتی که قبل از تگ `<html>` باز باید نوشته شود، قطعه کد `<DOCTYPE html!>` است که استثنائاً با حروف بزرگ نوشته می‌شود.

اما این تگ چیست؟

تگ `<!DOCTYPE html>` به مرورگر می‌فهماند که شما قصد استفاده از HTML نسخه ۵ را برای طراحی صفحه‌ی وب خود دارید. در صورتی که بخواهید از XHTML^۲ جهت طراحی صفحه وب خود استفاده کنید، باید کد فوق را کمی تغییر دهید.

شاید به این فکر کرده باشید، پس چرا حذف آن تأثیری در صفحه وب ما ندارد؟ این کد در ایجاد صفحه HTML اختیاری و در صفحات XHTML اجباری است. اگر این تگ را به همراه یک URL^۳ در بالای صفحه وب خود اعلان کنید سبب می‌شود که مرورگر فعلی، هنگام باز کردن صفحه شما در مد standard کار کنند.

بدین صورت می‌توانید کدهای خود را مطابق استاندارد نوشته و کنترل بیشتری بر چگونگی نمایش این صفحه در مرورگرهای مختلف داشته باشید. در صورت نگذاشتن این کد، مرورگر از روش‌های قدیمی‌تر برای نمایش صفحه شما استفاده می‌کند.

^۲ EXtensible HyperText Markup Language

^۳ Uniform Resource Locator

یک مثال عملی: در هنگام تعیین رنگ با شماره های هگزاد سیگمال، یک علامت # (شارپ) در ابتدای شماره رنگ قرار می گیرد. که حذف آن باعث نادیده گرفته شدن رنگ در زمان نمایش صفحه می شود، اما اگر کد DOCTYPE را از ابتدای صفحه پاک کنید. مرورگرتان از این مورد ایراد نمی گیرد و آن رنگ را نشان می دهد.

نکته: برای تست این مثال از مرورگر فایرفاکس استفاده کنید. چون مرورگر Internet Explorer به طور معمولی هم از حذف علامت # ایراد نمی گیرد و ممکن است شما فکر کنید گذاشتن علامت # اختیاری است.

تگ head:

تگ head یا معادل فارسی آن تگ سراندا؛ اولین تگ در بدنه تگ html می باشد که حاوی یک سری اطلاعات برای مرورگر و نیز جستجوگرهای وب می باشد. اطلاعات این بخش در قسمت بدنه سند html ما نمایش داده نمی شود. ولی اطلاعاتی همچون عنوان صفحه، کلمات کلیدی، استایل های خارجی و داخلی، جاوا اسکریپت های خارجی و داخلی، آیکون صفحه، توضیحاتی راجع به سازنده و... را تعریف می کند. که برای مرورگر و نیز ایندکس کردن توسط جستجوگرهای وب مانند گوگل و یاهو بسیار مهم است.

در کل تگ head حاوی کدهایی است که مستقیماً روی صفحه دیده نمی شوند ولی تأثیر آنها دیده می شود.

تگ head هیچ صفتی ندارد.

تگ هایی که بین تگ head قرار می گیرند عبارتند از:

```
<title>
<link>
<script>
<style>
<meta>
```

که در آینده کاربرد و نحوه ی استفاده از آنها را توضیح خواهیم داد.

تگ title:

- تگ title عنوان صفحه را تعیین می کند و هیچ صفتی ندارد.
- محتوای تگ تایتل نباید بیش از ده کلمه باشد.

تأثیر title در سئو (SEO⁴):

تعریف SEO:

⁴ Search Engine Optimization

به معنای «بهبود در موتور جستجو» است. که در اصل قواعدی است که تعیین می‌کند سایت خود را به چه صورت طراحی کنید تا در نتایج جستجو، در رتبه‌های بالاتری قرار بگیرد.

از روی تگ **title** است که مشخص می‌شود یک صفحه درباره چیست، پس محتوای تگ تایتل را مرتبط با موضوع آن صفحه انتخاب کنید. همه سئو کارهای معروف و معتبر روی این نکته تأکید خاصی دارند.

وقتی ربات‌های جستجو به تگ تایتل می‌رسند، این تگ را از جهات مختلف مورد بازبینی قرار می‌دهند و کلمات کلیدی صفحه را استخراج می‌کنند و در مرحله بعد تمامی تگ‌ها را از روی این تگ می‌سنجند. که چقدر بقیه تگ‌های به‌کار رفته در صفحه، مانند **<h1>** با تگ تایتل شما همخوانی دارد؟

اولویت‌ها در موتور جستجو:

معیار اول، آدرس URL (یعنی کلمات به کار رفته در نام پوشه و فایل منتهی به صفحه) است که کلمه جستجو شده در آن‌ها موجود باشد. معیار دوم تگ **<title>** و معیار سوم متن صفحه است.

تگ **body**:

تگ **body**، یا همان بدنه سایت، شامل تمام تگ‌هایی می‌شود که برای ساختن قالب اصلی سایت نیاز داریم. متن بین تگ **body** باز و بسته تنها اطلاعاتی است که توسط مرورگر نمایش داده خواهد شد.

صفات تگ **body**:

bgcolor: صفت **bgcolor** رنگ بک‌گراند را تعیین می‌کند. که داخل تگ **body** و با یک فاصله بعد از آن به صورت زیر قرار می‌گیرد:

```
<body bgcolor="نام یا کد رنگ">
```

نکته: اگر از کد رنگ استفاده می‌کنید باید قبل از آن، از علامت # استفاده کنید.

dir^۵: راست به چپ بودن یا چپ به راست بودن صفحه را تعیین می‌کند **ltr**^۶ و **rtl**^۷ مقادیری هستند که می‌توان به این صفت نسبت داد.

```
<body dir="rtl">
```

background: این صفت هنگامی که می‌خواهید یک عکس را به عنوان بک‌گراند معرفی کنید استفاده

می‌شود. قبل از انتخاب عکس و درج آن باید به نکات زیر توجه کنید:

خصوصیات نام عکس:

^۵ direction

^۶ Left To Right

^۷ Right To Left

در نام‌گذاری فایل‌ها و پوشه‌های سایت خود به نکات زیر دقت کنید:

- نام فایل انگلیسی باشد.
- با حروف کوچک نوشته شود.
- بین حروف نباید فاصله درج شود. به جای آن می‌توان از (_) آندرلاین استفاده کرد.
- نام فایل را مرتبط با محتوا انتخاب کنید. (مثلاً اگر یک عکس برای بک‌گراند استفاده می‌شود آن را `bg` بنامید و نه مثلاً `pic1`)
- فراموش نکنید که پسوند فایل، جزئی از نام فایل است. پس هر کجا خواستید آدرس یک فایل را درج کنید، پسوند را نیز باید بیان کنید.

آدرس‌دهی:

دو نوع آدرس‌دهی برای بیان مسیر فایل‌ها وجود دارد:

۱- **مطلق (Absolute):** یعنی آدرس یک فایل را نسبت به ریشه سایت یا درایو بیان کنیم.

- مثال اول)

```
<body background="file:///x.com/img/y.jpg">
```

- مثال دوم)

```
<body background="http://persiangig.com/img/x.jpg">
```

۲- **نسبی (Relative):** یعنی آدرس یک فایل را نسبت به فایلی که در حال نوشتن کدهای آن هستیم

بیان کنیم.

(مثال)

```
<body background="img/sky.jpg">
```

آدرس بالا می‌گوید فایلی به نام `sky.jpg` در پوشه `img` در کنار فایل فعلی (که `index.html` باشد) قرار دارد.

آدرس‌دهی نسبی مناسب‌ترین نوع آدرس‌دهی در وب می‌باشد. البته گاهی مجبورید به صورت مطلق آدرس‌دهی کنید.

تگ‌های مربوط به شکل دهی متن:

<code>Hello</code>	Hello	متن را Bold می‌کند.
<code>Hello</code>	Hello	متن را Bold می‌کند، نماد تأکید است.
<code><i>Hello</i></code>	<i>Hello</i>	حروف را خمیده می‌کند.
<code>Hello</code>	<i>Hello</i>	حروف را خمیده می‌کند، نماد تأکید است.
<code><u>Hello</u></code>	<u>Hello</u>	به متن زیرخط می‌دهد.
<code><s>Hello</s></code>	Hello	متن را خط می‌زند.

نکته: می‌توان از تگ‌های بالا به طور همزمان استفاده کرد.

پاراگراف‌بندی در متون وب:

در word برای ایجاد یک پاراگراف جدید از کلید Enter استفاده می‌کنیم که cursor به خط بعدی انتقال می‌یابد. یعنی به ابتدای پاراگراف جدید می‌رود. اما برای رفتن به خط بعدی بدون ایجاد پاراگراف از کلیدهای Shift+Enter استفاده می‌کنیم. در کد نویسی وب نیز دو تگ برای پاراگراف‌بندی و رفتن به خط بعدی وجود دارد:

<code><p></code> برای ایجاد پاراگراف	<code>
</code> برای رفتن به خط بعدی بدون ایجاد پاراگراف
--	--

نکته: تگ p می‌تواند صفت dir یا align بگیرد.

align محل متن را مشخص می‌کند و مقادیر right و left و center به خود می‌گیرد.

تیتراهای صفحه:

به تیترا در انگلیسی Heading گفته می‌شود. در وب برای صفحه، شش سطح تیترا در نظر گرفته شده است که به صورت زیر تعیین نمی‌شوند:

<code><h1></code> تیترا اول
<code><h2></code> تیترا دوم
<code><h3></code> تیترا سوم
<code><h4></code> تیترا چهارم
<code><h5></code> تیترا پنجم
<code><h6></code> تیترا ششم

توجه: در هر صفحه فقط یک بار می‌توان از تگ h1 استفاده کرد. در غیر اینصورت صفحه شما از نظر سئو امتیاز منفی می‌گیرد.

برای مثال شما می‌خواهید یک مقاله درباره «آموزش طراحی وب» بنویسید. برای داشتن تیترا با این نام؛ کد شما باید به این صورت باشد:

<code><h1>آموزش طراحی وب</h1></code>
--

تگ‌های sub و sup :

متن داخل تگ sub و تگ sup به صورت توان (sup^8) و یا زیرنویس (sub^9) در می‌آید. از این تگ‌ها برای قرار دادن حروف یا اعداد کمی بالاتر یا پایین‌تر از بقیه متن در صفحه استفاده می‌شود.

برای نوشتن متنی به شکل زیر :

Designed by MicrosoftTM Office₂₀₁₀

از این دو کد استفاده می‌کنیم:

^{حروف یا شماره ای که بالاتر از سطر باشد}

_{حروف یا شماره ای که پایین تر از سطر باشد}

تگ pre :

تگ `pre`¹⁰ زمانی به کار می‌رود که شما قصد دارید یک قطعه کد برنامه‌نویسی (مثلاً به زبان C) به همان صورت که تایپ کرده‌اید به کاربر نمایش داده شوند. یعنی قصد دارید قسمتی از کد را در قسمت آموزش به کاربر نشان دهید.

برای مثال شما قصد دارید این برنامه به زبان C را در سایت خود قرار دهید:

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int x=2 ,y=5 ,sum=0;
    clrscr();
    sum=x+y;
    printf("sum=%d",sum);
    getch();
    return(0);
}
```

برای این کار آن را با تگ <pre> باز و بسته احاطه می‌کنید و همچنین برای این که مرورگر شما قسمت <stdio.h> را، تگ‌های html در نظر نگیرد از معادل آن‌ها استفاده کنید:

به این علامت‌ها در اصطلاح «کاراکترهای خاص» یا Special Characters گفته می‌شود:

< —> <

> —> >

⁸ superscripted

⁹ subscripted

¹⁰ preformatted

تگ abbr:

این تگ برای نمایش متن کامل کلمات اختصار یافته به کار می‌رود. بدین صورت که شما عبارت کامل را بین (") دابل کوتیشن صفت title می‌نویسید و کلمه اختصار یافته را بین تگ باز و بسته <abbr>^{۱۱} قرار می‌دهید. به این صورت:

```
<abbr title="عبارت کامل">کلمه مختصر</abbr>  
<abbr title="Personal Computer">PC</abbr>
```

لیست‌ها:

برای نمایش یک سری اقلام به صورت لیستی منظم، به عنوان مثال منوها و زیرمنوها و... می‌توانید از دو تگ و استفاده کنید. به طور کلی ایجاد لیست در وب به دو صورت امکان‌پذیر است:

۱- لیست‌های مرتب (Ordered List):

لیستی است که ترتیب در آن‌ها با شماره یا علامت‌های ترتیبی مشخص شده است. از این نوع لیست‌ها هنگامی استفاده می‌شود که ترتیب اقلام لیست برای ما مهم است.

انواع لیست مرتب:

1,2,3,4,...
i,ii,iii,iiii,...
I,II,III,IIII,...
a,b,c,d,...
A, B, C, D,...

- یک لیست مرتب با تگ آغاز می‌شود.
- هرکدام از اقلام لیست با تگ (مخفف List Item) آغاز می‌شوند. مانند:
- صفت type نوع لیست مرتبط را مشخص می‌کند. اگر کاراکتر اول هر یک از انواع لیست مرتب را به این صفت بدهید، به آن روش شماره‌گذاری خواهد کرد.

(مثال)

```
<ol type="1">  
  <li>CPU</li>  
  <li>RAM</li>  
</ol>
```

¹¹abbreviation

۲- لیست‌های نامرتب (Unordered List) :

از این نوع لیست گذاری زمانی استفاده می‌شود که ترتیب اقلام برای ما اهمیتی نداشته باشد. اقلام لیست به وسیله اشکال مختلف مانند گلوله نشانه گذاری می‌شوند. یک لیست نامرتب با یک تگ `<ul>` آغاز می‌شود. هر کدام از اقلام لیست بین تگ `<li>` قرار می‌گیرند.

انواع نشانه‌های لیست نامرتب همراه با کلمه مربوط به type آن‌ها:

- —→ circle
- —→ disc
- —→ square

مثال)

```
<ul type="disc">  
  <li>منوی اصلی</li>  
  <li>گالری عکس</li>  
</ul>
```

تگ font:

تگ font برای تعیین خصوصیات متن است. مثل نوع فونت، سایز آن و رنگ نوشته، که با صفات color, size و face قابل اجراست.

(مثال)

```
<font face="Tahoma" size="2" color="blue">طراحی صفحات وب</font>
```

این کد در مرورگر نوع فونت را Tahoma سایز آن را ۲ و رنگ نوشته شما را آبی نمایش می‌دهد.

تگ meta:

متا تگ‌ها تگ‌هایی هستند که یک سری اطلاعات اضافه در مورد صفحه به مرورگر و موتورهای جستجو ارائه می‌کنند.

اولین و مهم‌ترین نوع تگ متا مربوط به تعیین charset (گروه کاراکترهای) صفحه است.

```
<meta charset="utf-8" >
```

به وسیله این تگ شما می‌توانید زبان فارسی را در سایت خود بدون نگرانی استفاده کنید. بدون این تگ کاراکترهای فارسی در خروجی نمایش داده نمی‌شود.

مابقی متا تگ‌ها یک مقدار name و یک مقدار content می‌پذیرند.

زمانی که یک موتور جستجو صفحه شما را پیدا می‌کند، باید آن را در اصطلاح index کند. (یعنی آن را به بانک اطلاعاتی قابل جستجوی خود اضافه کند) البته همراه با اطلاعاتی از سایت شما. هم اکنون موتورهای جستجوی زیادی از برچسب‌های <meta> استفاده می‌کنند که به شما اجازه می‌دهد کلمات کلیدی و توصیف صفحه خود را در آن قرار دهید. این به شما کنترل بیشتر می‌دهد تا صفحه شما در طول یک جستجو چگونه نمایش داده شود و اغلب باعث ترافیک بیشتر به سایت شما می‌شود. برچسب‌های meta برای چند هدف مختلف می‌توانند مورد استفاده قرار گیرند معمولاً شما باید تگ‌های meta را در تگ <head> در شروع صفحه خود قرار دهید. برای بهبود بخشیدن به نتایج موتور جستجو از دو نشان مخصوص استفاده می‌شود:

```
<meta name="description" content="توضیحات مربوط به صفحه">
<meta name="keywords" content="کلمات کلیدی">
```

زمانی که یک کاربر در حال جستجو با موتور جستجویی که از تگ‌های متا و کلمات کلیدی پشتیبانی می‌کند باشد. اگر از کلمات کلیدی یا متای موجود در سایت شما به عنوان عبارات مورد جستجو استفاده کند ممکن است سایت شما با آن تیترا در لیست ظاهر شود و در زیر این تیترا مقداری یا تمام کلماتی که شما در توصیف صفحه خود در تگ متا قرار داده‌اید قرار گیرد.

نکته: بهتر است که توصیف صفحه دارای بیش از ۲۰۰ کاراکتر (حروف و...) نباشد. همچنین کلمات کلیدی در نتایج جستجو دیده نمی‌شود و بهتر است آن‌ها را زیر ۱۰۰۰ کاراکتر نگه دارید. زیرا اگر بیش از این باشد موتور جستجو اضافه آن را نادیده می‌گیرد و یا شما را از فهرست حذف می‌کند (برای جدا کردن کلمات کلیدی احتیاجی به ویرگول نیست).

(مثال)

```
<head>
  <title>first page</title>
  <meta name="description" content="آموزش متا تگ">
  <meta name="keywords" content="meta web computer">
</head>
```

انواع متا تگ‌ها:

<code><meta http-equiv="expires" content="0"></code>	برای تعیین تاریخ انقضای صفحه استفاده می‌شود مثال: اطلاعات
<code><meta name="resource-type" content="document"></code>	نوع صفحه شما را مشخص می‌کند مثال: این صفحه image است یا document
<code><meta name="distribution" content="global"></code>	محدودیت نمایش سایت شما را برای یک کشور تعیین می‌کند. مثال: جهانی باشد یا فقط در ایران
<code><meta name="author" content="hr.niroomand"></code>	نویسنده یا مدیر را مشخص می‌کند مثال: اگر مقاله بود نام نویسنده را نوشته می‌شود
<code><meta name="copyright" content="copyright © by Aftabgardan"></code>	حق کپی رایت را مشخص می‌کند. اینکه کاربر اجازه استفاده از مطالب سایت را در سایت یا وبلاگ خود دارد یا خیر؟
<code><meta name="robots" content="INDEX, FOLLOW"></code>	برای ایندکس کردن صفحه شما توسط ربات‌های موتور جستجو به کار می‌رود.
<code><meta name="revisit-after" content="1 DAYS"></code>	به ربات‌های جستجو می‌فهماند که چند روز یک بار از صفحه بازدید کنند
<code><meta name="rating" content="general"></code>	به ربات‌ها می‌فهماند که سایت شما برای چه رده سنی مناسب است.
<code><meta name="generator" content="wordpress"></code>	نوع سیستمی که سایت با آن نوشته شده را معرفی می‌کند مثال: wordpress

برچسب‌های متا در صفحه وب قابل مشاهده نیستند.

لینک‌ها (links and Navigations):

لینک‌ها بنیادی‌ترین قسمت از صفحات وب و شبکه ارتباط جهانی (^{۱۲}WWW) هستند. بنابراین توصیه می‌شود زمان بیشتری برای این قسمت صرف کنید.

انواع لینک:

- لینک سراسری (Global Links): لینک‌هایی که به صفحات یک سایت دیگر اشاره دارند.
- لینک محلی یا موضعی (Local Link): لینک‌هایی که به صفحات دیگر همان سایت اشاره می‌کند.
- لینک داخلی (Internal Links): لینک‌هایی هستند که به قسمتی از صفحه‌ای که کاربر در آن قرار دارد اشاره می‌کند.

چگونه لینک بسازیم؟

تگ a:

برای ساختن لینک از تگ `<a>`^{۱۳} استفاده کنید. هر آنچه که بین تگ `<a>` باز بسته قرار بگیرد به عنوان لینک شناخته می‌شود. برای اینکه مقصد لینک نیز مشخص شود از صفت `href`^{۱۴} استفاده کنید.

توجه: هر المانی از جمله متن، عکس و... می‌تواند بین تگ `a` قرار بگیرد. شکل کلی این دستور در زبان `html` به صورت زیر است:

```
<a href="آدرس مورد نظر">متنی که به عنوان لینک شناخته می‌شود</a>
```

برای لینک‌دار کردن یک عکس باید آدرس آن نیز ست شود مانند:

```
<a href="آدرس مورد نظر"></a>
```

لینک‌های سراسری و محلی با دو مثال بالا قابل انجام هستند. اما برای این که به قسمت خاصی از صفحه وبی که در آن قرار دارید، پیوند برقرار کنید یا به عبارت دیگر به قسمتی از همان صفحه پرش کنید باید از صفت `name` از تگ `a` برای نام گذاری آن بخش استفاده کنید. این نام از حروف تشکیل می‌شود که باید منحصر به فرد نیز باشد.

مثلاً قسمتی از صفحه را انتخاب کنید و در آن این دستور را بنویسید.

```
<a name="mesal"></a>
```

¹² World Wide Web

¹³ anchor

¹⁴ hyper text reference

حال برای مراجعه به آن بخش کافیسیت علامت # را قبل از نامی که انتخاب کرده بودید در مکانی که می خواهید پرش کنید درج شود. به این صورت:

```
<a name="#mesal"></a>
```

لینک های رنگی بسازید!

برای این که لینک ها را در صفحه خود به صورت متمایز نمایش دهید می توانید برای آن ها رنگ مشخص کنید.

دو حالت برای تعریف کردن رنگ لینک ها وجود دارد:

۱- رنگ تمامی لینک ها در صفحه به یک صورت تعریف شود.

۲- برای بعضی از لینک ها رنگ متفاوتی تعریف شود.

صفت های link , vlink , alink :

به وسیله این صفات می توانید برای حالات مختلف لینک رنگ های دلخواه خود را انتخاب کنید. برای مثال بعد از کلیک روی یک لینک رنگ آن قرمز شود.

link: بیان کننده رنگ پیوندهایی است که تاکنون ملاقات نشده اند. استاندارد و حالت پیش فرض آن آبی یا blue است.

vlink : vlink که مخفف visited link است، رنگ لینک را پس از ملاقات شدن نمایش می دهد. استاندارد آن ارغوانی یا #800080 یا purple است.

alink :alink که مخفف Active link است، رنگ لینک را وقتی که با ماوس در حال کلیک روی آن هستید مشخص می کند. پیش فرض آن قرمز (#FF0000) یا red است.

اگر می خواهید یک رنگ برای تمام لینک های موجود سِت کنید، کافیسیت صفت های موجود را به تگ <body> نسبت دهید.

```
<body link="green" vlink="red" alink="pink">
```

در این صورت تمام لینک ها با رنگ های مورد نظر نمایش داده می شوند. اما اگر می خواهید یک یا چند لینک مجزا با رنگ های متفاوت داشته باشید می توانید از تگ بین تگ <a> استفاده کنید و به فونت رنگ مورد نظر را نسبت دهید.

نکته: به جای نوشتن اسامی رنگ‌ها می‌توان از فرمت **rgb** استفاده کرد.

rgb به ترتیب مخفف **red, green, blue** است که هر یک می‌توانند مقداری بین ۰ تا ۲۵۵ بگیرند رنگ مورد نظر از ترکیب این رنگ‌ها به دست خواهد آمد مثلاً ۱۷, ۱۸۷, ۱۴۲ رنگی مایل به بنفش خواهد بود البته اکثر طراحان ترجیح می‌دهند که این رنگ‌ها را در مبنای شانزده (hex) بیان کنند. مثلاً رنگ معادل مثال قبلی در مبنای ۱۶ به صورت **#E11BB8** نوشته می‌شود.

از صفات دیگری که برای تگ **<a>** می‌توان نام برد صفت **title** است. با استفاده از این صفت وقتی ماوس بر روی پیوند یا لینک مورد نظر قرار می‌گیرد متنی که به این صفت نسبت داده شده نمایش داده می‌شود.

مثال:

```
<a href="http://www.google.com" title="سایت شرکت گوگل">google</a>
```

وقتی ماوس روی کلمه **google** قرار بگیرد، پیغام «سایت شرکت گوگل» ظاهر می‌شود.

صفت **target**:

به صورت پیش فرض، لینک‌ها در همان پنجره با فریمی که در آن هستید باز می‌شوند. با استفاده از صفت **target** می‌توانید تعیین کنید که لینک مورد نظر در چه پنجره‌ای باز شود. با این صفت در بخش فریم‌ها بیشتر آشنا خواهید شد.

مثال:

هدف
↔

```
<a href="" target="" _blank
```

در پنجره جدید

_self در پنجره خودش

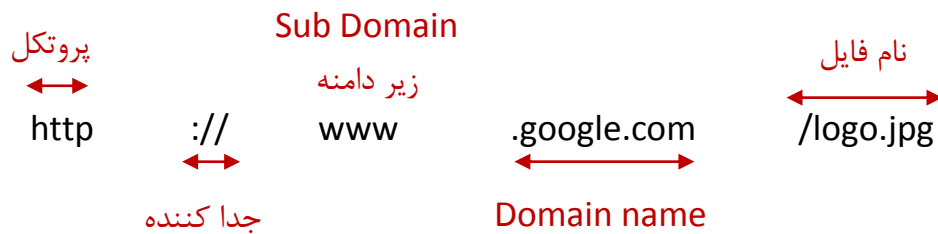
_top در بالاترین فریم

_parent در فریم پدرش (فریم بالایی)

نام فریم در فریم نامبرده باز شود

تشریح اجزای یک URL

Uniform resource locator به معنای «مشخص کننده منبع یک شکل» است.



پروتکل‌های مشهور عبارتند از:

http: hypertext transfer protocol

ftp: file transfer protocol

https: s → SSL → secure socket layer

تحقیق: در مورد پروتکل https بیشتر تحقیق کنید...

تگ

یکی از مهم‌ترین عناصر در صفحات وب عنصر عکس است. برای قرار دادن عکس در فایل‌های html از تگ استفاده می‌شود. این تگ یک تگ تهی بوده و نیازی به تگ پایانی ندارد. برای مشخص کردن تصویری که باید در صفحه قرار بگیرد باید از ویژگی ^{۱۵}src استفاده کرد که توسط این ویژگی آدرس و نام فایل مشخص می‌شود.

صفت alt:

این ویژگی، متنی را به عنوان مقدار می‌پذیرد و برای ایجاد یک متن توصیفی بر روی تصویر به کار می‌رود. اگر عکس مورد نظر پیدا نشود به جای عکس، آن متن را نمایش می‌دهد. این مشخصه بیشتر زمانی کاربرد دارد که وقتی تصویر به دلیل هر مشکلی در صفحه دیده نشد کاربر بداند که چه تصویری را ندیده است که در این حالت متن توصیفی ویژگی alt به جای تصویر دیده خواهد شد.

موتورهای جستجو برای ایندکس کردن یک صفحه فقط نام تصویر و توضیحی که برای آن تصویر داده شده است را می‌خوانند، پس برای تصاویری که به موضوع صفحه شما مرتبط هستند و یا می‌خواهید بازدید کنندگان بتوانند در موتورهای جستجو آنها را پیدا کنند حتماً از این صفت استفاده کنید.

مثال:

```

```

نکته: برای استفاده از تصویر در وب قبل از هر کاری یک پوشه با نام img بسازید و هر تصویری که قصد استفاده از آن را دارید در این پوشه قرار دهید.

توجه داشته باشید تصویر در مکانی قرار خواهد گرفت که تگ img در آنجا نوشته شده باشد.

صفات width و height :

امکان دیگری که شما می‌توانید در تگ داشته باشید صفات width و height هست. با استفاده این دو صفت می‌توانید ابعاد یک تصویر را مشخص کنید.

¹⁵ source

مثال:

```

```

لغت **width** به معنای عرض و لغت **height** به معنای ارتفاع است. به وسیله این صفات می‌توانید ابعاد تصویر مورد نظر را تعیین کنید. به جای 16px می‌توانید از $\%$ نیز استفاده کنید. برای فهمیدن ابعاد تصویر شما می‌توانید از برنامه‌های گرافیکی سودمند برای اینکه این اعداد را تخمین بزنید استفاده کنید. روش دیگر برای پیدا کردن ابعاد عکس این است که شما می‌توانید آیکون مورد نظر را با **drag and drop** به پنجره مرورگر بکشید و طول و عرض تصویر در **titlebar** مرورگر نشان داده می‌شود.

برای ایجاد گالری و چینش تصاویر کنار هم هیچگاه تصاویر بزرگ را کوچک نمایش ندهید. چون حجم عکس تغییری نمی‌کند و در این صورت صفحه شما دیر **load** می‌شود و کاربر همان عکس بزرگ را در نمای کوچک می‌بیند. بنابراین بهتر است تصویر را **resize** کرده و تصویر کوچک را به کاربر نشان دهید و تصاویر بزرگتر را به آن لینک دهید. (برای لینک دادن آن را با تگ **<a>** احاطه کنید).

نکته: برای ساخت گالری عکس، تصاویر کوچک را در پوشه **small** و تصاویر بزرگ را در پوشه **large** و در پوشه **img** ذخیره کنید.

مثال:

```
<a href="../../../img/large/نام عکس" target="_blank">  
</a>
```

صفت title: این صفت برای عکس نیز سِت می‌شود که مقداری که می‌گیرد با قرار گرفتن موس روی عکس دیده می‌شود..

صفت align:

به وسیله صفت **align** می‌توان نحوه‌ی قرار گرفتن تصویر را نسبت به عناصری که در اطراف آن است تعیین کرد.

این صفت مقادیر زیر را می‌پذیرد:

۱. **right:** اگر ویژگی **align="right"** باشد در این صورت تصویر در سمت راست عناصر موجود قرار می‌گیرد.

۲. **left:** اگر ویژگی **align="left"** باشد در این صورت تصویر در سمت چپ عناصر موجود قرار می‌گیرد.

۳. **top:** اگر ویژگی **align="top"** باشد در این صورت تصویر در بالای خطی که درج شده قرار می‌گیرد.

¹⁶ Pixel

۴. bottom: اگر ویژگی align="bottom" باشد در این صورت تصویر در پایین خطی که درج شده قرار می‌گیرد.

تگ center:

همانطور که از نام آن پیداست این تگ هر عنصری را که احاطه کند آن را به وسط صفحه وب انتقال می‌دهد. این تگ از عملکرد بالایی برخوردار است و در جاهایی که align کارساز نیست جایگزین خوبی است.

فرمت عکس در طراحی وب:

فرمت‌های مختلفی برای عکس‌ها وجود دارد اما فرمت‌هایی که در وب کاربرد دارد عبارتند از:

فرمت jpeg یا jpg^{۱۷}:

- در تصاویر، رنگ‌های مشابه و کنار هم را خلاصه نویسی می‌کند.
 - حجم این فرمت قابل محاسبه نیست.
 - ۲۴ بیتی است، یعنی هر پیکسل ۱۶ میلیون رنگ را پشتیبانی می‌کند.
- نکته مهم: jpg برای **عکس‌های طبیعی** مناسب است.

فرمت gif^{۱۸}:

این فرمت به دو صورت عکس ثابت و متحرک (انیمیشن) ممکن است دیده شود. نهایت رنگ در این فرمت ۲۵۶ رنگ است. بنابراین حجم عکس نسبت به تعداد رنگ استفاده شده افزایش می‌یابد. در واقع gif عکس‌هایی هستند که در آن تعدادی عکس ثابت قرار دارد که فریم به فریم پخش می‌شوند تا یک انیمیشن را خلق کنند.

از این فرمت برای لوگوهای تبلیغاتی استفاده می‌شود که در کل تنوع بسیار کمی دارند و حاوی نوشته هستند.

خصوصیات gif:

- ابتدا عکس را بررسی می‌کند تا نهایتاً ۲۵۶ رنگ که بیشترین کاربرد را داشته‌اند استخراج کند سپس به هر رنگ یک شماره اختصاص می‌دهد.
 - به جای ذخیره سازی هر پیکسل شماره اختصاصی آن را ذخیره می‌کند.
 - تنها فرمتی است که متحرک بودن را پشتیبانی می‌کند.
 - بک گراند transparent را پشتیبانی می‌کند.
- نکته مهم: gif برای **عکس‌های متحرک** مناسب است.

¹⁷ Joint Photographic Experts Group

¹⁸ graphic interchange format

فرمت ^{۱۹}png:

این فرمت قابلیت‌های خوب gif و jpg را با هم دارد. این فرمت در دو نوع است یکی ۸ بیتی که شبیه فرمت gif است و تقریباً تمام ویژگی‌های آن غیر از قابلیت پویا نمایی (متحرک) را داراست و نوع دیگر آن ۲۴ بیتی است که قادر به نمایش میلیون‌ها رنگ مانند jpg است.

نکته مهم: png برای تصاویر ترسیمی مناسب است. (یعنی تصاویری که توسط کامپیوتر ترسیم شده‌اند)

¹⁹ Portable Network Graphics

جدول (Table):

جدول یکی از سنگین‌ترین عناصر وب است و سرعت لود شدن سایت را پایین می‌آورد. بنابراین تا می‌توانید کمتر استفاده کنید. برای ترسیم جدول به دو تگ `<tr۲۰>` و `<td۲۱>` نیاز داریم که `tr`ها بیان‌گر تعداد سطر و `td`ها بیان‌گر تعداد ستون‌های جدول هستند. ابتدای جدول با `<table>` و انتهای آن با `</table>` مشخص می‌شود.

برای رسم جدولی به شکل زیر، از این کدها استفاده می‌شود:

A diagram of a 2x3 table. The first row contains 'Cell1', 'Cell2', and 'Cell3'. The second row contains 'Cell4', 'Cell5', and 'Cell6'. A red arrow labeled 'tr' points to the first row. A red arrow labeled 'td' points to the first cell of the first row.

```
<table>
  <tr>
    <td>Cell1</td>
    <td>Cell2</td>
    <td>Cell3</td>
  </tr>
  <tr>
    <td>Cell4</td>
    <td>Cell5</td>
    <td>Cell6</td>
  </tr>
</table>
```

صفات تگ table:

<code>valign="top"</code> <code>middle</code> <code>bottom</code>	چینش ارتفاعی نوشته را در سلول‌ها مشخص می‌کند
<code>border="1"</code>	ضخامت خط‌های جدول را مشخص می‌کند
<code>width="500px" height="400px"</code>	طول و ارتفاع جدول را مشخص می‌کند
<code>bgcolor="green"</code>	رنگ پس زمینه را مشخص می‌کند

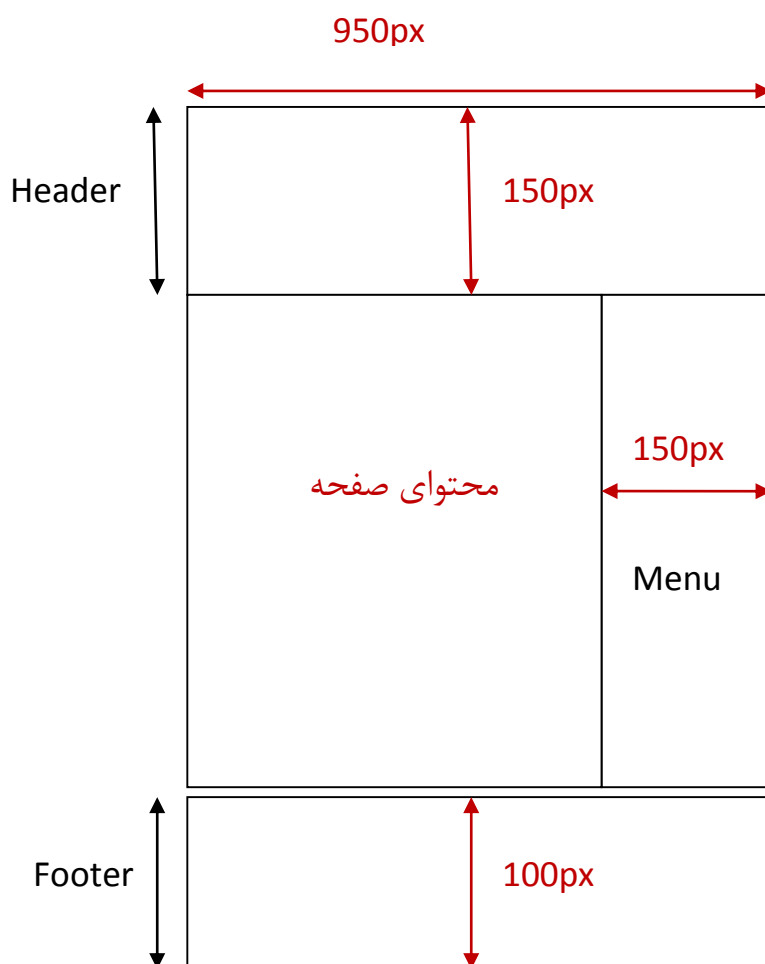
²⁰ table row

²¹ table division

<code>bordercolor="red"</code>	رنگ خط دور جدول را مشخص می‌کند
<code>cellspacing="10px"</code>	فاصله متن را از خط دور سلول‌ها تعیین می‌کند
<code>cellpadding="5px"</code>	فاصله سلول‌های جدول از یکدیگر را تعیین می‌کند
<code>align="center"</code> <code>right</code> <code>left</code>	چینش جدول در صفحه را مشخص می‌کند

نکاتی درباره قالب اصلی سایت:

طول سایت نباید طولانی شود؛ یعنی نباید scroll افقی داشته باشد. طبق آمار، رزولوشن برای اکثر کاربران بیشتر از ۱۰۲۴ در ۷۶۸ است. پس ساختار کلی سایت باید چیزی شبیه این باشد:



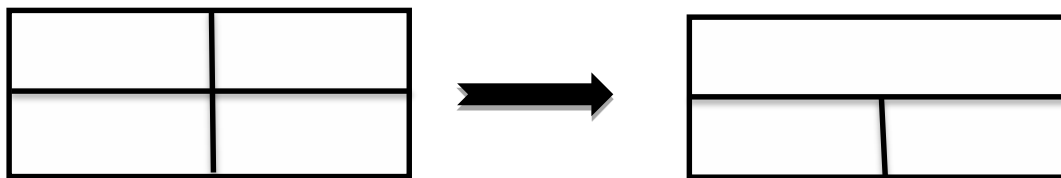
تگ `td`:

- در تگ `td` نمی‌توان `height` داد. باید از `width` استفاده کنید.
- بهتر است بین `td`های خالی ` ` تایپ شود.

برای ادغام ستونی، از صفت `colspan` استفاده می‌شود. مثلاً برای ادغام دو سلول از کد `colspan="2"` استفاده کنید و به تعداد `colspan`، 1- عدد از `td`ها را حذف کنید. یعنی برای همین مثال، یک `<td>` باید حذف کنید و برای ادغام ۳ ستون دو عدد از `td`ها را پاک می‌شود.

(مثال)

```
<table border="1" width="200" height="200">
  <tr>
    <td colspan="2">
    </td>
  </tr>
  <tr>
    <td>
    </td>
    <td>
    </td>
  </tr>
</table>
```



نکته: در ادغام سطری به جای `colspan` از `rowspan` استفاده کنید.

قاب (Frames):

فریم‌ها برای نمایش یک صفحه وب در صفحه دیگر به کار می‌روند. برای استفاده از فریم در مکان مورد نظر از تگ `<iframe>` استفاده کنید.

برای مثال قطعه کد زیر، صفحه‌ای که حاوی اوقات شرعی ساوه است را در سایت شما نشان خواهد داد:

```
<iframe src=http://savehsara.aftab.cc/modules/owghat/" frameborder="0"
width="150px" height="200px" scrolling="no">
```

معایب فریم:

- موتور جستجو به محتوای `iframe` دسترسی ندارد.
- سرعت سایت را در صورتی که وابسته به سایت دیگر باشد پایین می‌آورد.
- از صفات `iframe` می‌توان به `scrolling, height, width, allowtransparency` اشاره کرد.
- صفت `scrolling` برای زمانی است که ما نمی‌خواهیم دور فریم، اسکرول بار دیده شود.
- صفت `allowtransparency="true"` به معنای شفافیت است.

چگونه یک لینک را در یک فریم باز کنیم؟

گام اول) یک فریم به نام `f` بسازید، مثلاً در `td` مربوط به محتوا، کدی به صورت زیر وارد کنید:

```
<td>
  <iframe src="" width="100%" height="500px" name="f"></iframe>
</td>
```

گام دوم) در لینکی که می‌خواهید در این فریم باز شود، `target` را مساوی `f` قرار دهید.

```
<a href="r.html" target="f"></a>
```

IDEها:

یک IDE^{۲۲} یا «محیط توسعه یکپارچه» نرم‌افزاری است که برای کمک به برنامه‌نویسان و توسعه‌دهندگان جهت ساخت نرم‌افزار طراحی شده است.

اکثر IDEها شامل موارد زیر هستند:

- یک ویرایشگر کد:

یک ویرایشگر کد شبیه Html Text Editor مکانی است که برنامه‌نویسان کد منبع خود را برای این برنامه‌ها می‌نویسند.

- یک کامپایلر یا مفسر:

کامپایلری که کد را به یک برنامه اجرایی کامپایل می‌کند و یک مفسر برنامه‌ها و اسکریپت‌هایی که نیازی به کامپایل شدن ندارند را تفسیر می‌کند.

- یک اصلاح کننده خطا:

اصلاح کننده خطاها یا debugger مشکلات به وجود آمده در کد منبع را حل می‌کنند.

IDEهای مطرح دنیا در زمینه طراحی وب:

Microsoft FrontPage

Microsoft Expression Studio

Adobe Dreamweaver

Sublime Text

²²Integrated Development Environment

انواع سایت از نظر تبادل داده با کاربر:

۱- ایستا (Static)

ایستا دریافت داده چندانی از کاربر ندارد و برای همه کاربران به یک صورت نمایش داده می‌شود.
مانند: سایت معرفی شرکت‌ها، سایت معرفی شخصیت‌های معروف مانند بازیگران و ...

۲- پویا (Dynamic)

سایت پویا دائماً از کاربر داده دریافت می‌کند و نسبت به نوع داده‌ای که از کاربر دریافت کرده نمایش متفاوتی دارد.
مانند: انجمن‌ها، ایمیل‌ها و ...

فرم (Form):

فرم‌ها از عناصری تشکیل شده‌اند که به کاربر اجازه ورود اطلاعات را می‌دهند. در واقع اطلاعات فرم به طرف برنامه‌ای که بر روی سرور قرار دارد، فرستاده می‌شود و بعد از تجزیه و تحلیل بر روی سرور، نتیجه به مخاطب نمایش داده می‌شود.

تگ form در واقع نقطه آغاز و پایان فرم را تعیین می‌کند.

صفات تگ form:

۱- method: به وسیله این صفت تعیین می‌کنید که اطلاعات فرم به چه نحوی به مقصد مورد نظر منتقل شوند. دو نوع ارسال اطلاعات برای فرم وجود دارد:

○ get: در این حالت داده‌ها و اطلاعات فرم از طریق url مرورگر منتقل خواهند شد و محدودیت‌هایی نیز خواهید داشت. همچنین از آن جا که داده‌ها در url مرورگر نمایش داده می‌شوند برای انتقال داده‌هایی مانند پسورد و ... مناسب نمی‌باشند. و فقط مناسب فرم‌های جستجو است.

○ post: در این روش، داده‌ها در پشت صحنه مرورگر و به صورت internal منتقل می‌شوند و در url نمایش داده نمی‌شوند. در این روش هیچ محدودیتی برای حجم داده‌های ارسالی و نوع کاراکتر نخواهید داشت.

۲- action: مقدار این صفت آدرس فایلی است که پردازش فرم را انجام می‌دهد برای آدرس دهی باید طبق قوانین آدرس دهی که قبلاً آموخته‌اید عمل کنید.

```
<form action="آدرس صفحه پردازشگر" method="نوع ارسال">
```

بر خلاف بسیاری از دستورات html، فرم‌ها در مرورگر به تنهایی نمایش داده نمی‌شوند. به عنوان مثال دستور بالا هیچ چیزی را در مرورگر نمایش نخواهد داد. برای اینکه اطلاعات را از کاربر دریافت کنید. علاوه بر دستور فرم، نیازمند دستورات دیگری نیز هستید که فیلدهای ورودی را در فرم مشخص کنند.

اکثر فیلدهای ورودی در زبان html به وسیله تگ input مشخص می‌شوند. شکل کلی دستور input به صورت زیر است. که در این حالت type، نوع فیلد و name، نام فیلد ورودی را مشخص می‌کند.

```
<input type="نوع فیلد" name="نام فیلد">
```

تگ input:

از تگ input برای قرار دادن انواع کنترل‌های دریافت اطلاعات از کاربر در فرم‌های html استفاده می‌شود. به وسیله خاصیت type در تگ input می‌توانید نوع کنترل آن را مشخص نمایید.

انواع فیلد در تگ <input> در html5:		
<code><input type="color" name="favcolor" /></code>	مقدار ورودی از نوع کد هگزادسیمال یک رنگ مثل FF8800	color
<code><input type="date" name="bday"/></code>	مقدار ورودی از نوع تاریخ	date
<code><input type="datetime" name="bdaytime" /></code>	مقدار ورودی از نوع زمان و یا تاریخ	datetime
<code><input type="email" name="usremail" /></code>	مقدار ورودی از نوع یک یا چند آدرس ایمیل این کنترل فقط روی مرورگرهای کروم، فایرفاکس و اپرا جواب می‌دهد.	email
<code><input type="month" name="bdaymonth" /></code>	مقدار ورودی از نوع ماه این کنترل فقط در مرورگرهای کروم و اپرا جواب می‌دهد.	month
<code><input type="number" name="quantity" min="1" max="5"/></code>	مقدار ورودی از نوع عدد در صورت وارد کردن مقادیر غیر عددی و همچنین مقادیر بیشتر و یا کمتر از محدوده تعیین شده، برنامه این مقادیر را نادیده گرفته و ارسال نمی‌کند.	number
<code><input type="range" name="points" min="1" max="10"/></code>	مقدار ورودی از نوع عددی در رنج داده شده در مرورگرهای کروم و اپرا جواب می‌دهد.	range
<code><input type="search" name="googlesearch" /></code>	مقدار ورودی از نوع فیلد جستجو این کنترل از موتور جستجو گوگل استفاده می‌کند. در مرورگرهای کروم و اپرا جواب می‌دهد.	search

<code><input type="time" name="usr_time" /></code>	مقدار ورودی از نوع زمان در مرورگرهای کروم و اپرا جواب می‌دهد.	time
<code><input type="url" name="homepage" /></code>	مقدار ورودی از نوع یک آدرس وب اگر مقدار دارای فرمت صحیح یک آدرس اینترنتی نباشد، کنترل error داده و مانع ارسال اطلاعات می‌شود. در مرورگرهای کروم، فایرفاکس و اپرا جواب می‌دهد.	url
<code><input type="week" name="week_year" /></code>	مقدار ورودی از نوع هفته	week
<code><input type="tel" name="usrtel"/></code>	مقدار ورودی از نوع شماره تلفن	tel
<code><input type="datetime-local" name="bdaytime" /></code>	مقدار ورودی از نوع زمان/تاریخ محلی	datetime- local
<code><input type="hidden" /></code>	استفاده از این فیلد برای زمانی است که شما مقادیری دارید که می‌خواهید از یک فرم به فرم های دیگر ارسال کنید بدون اینکه کاربر مجبور باشد اطلاعات را دوباره و دوباره تکرار کند! این نوع تایپ در فرم می‌تواند شامل هر مقداری باشد و از دید کاربر مخفی می‌باشد.	Hidden

نکته قابل ذکر در رابطه با نام فیلدها این است که وقتی فرمی به برنامه‌ای که بر روی سرور قرار دارد، فرستاده می‌شود هنگامی که می‌خواهید به اطلاعات هر فیلد از قبیل نام و نام خانوادگی دسترسی پیدا کنید `name` آن‌ها را بررسی کنید.

مثال)

```
First Name:<input name="FirstName" type="text" value="enter your  
firstname" maxlength="25">
```

- `maxlength` حداکثر تعداد کاراکترهای ورودی را کنترل می‌کند.

- value یک متن اولیه در فیلد چاپ می‌کند.
- بقیه فیلدهای متنی نیز مانند فیلد text هستند با این تفاوت که type آن‌ها متفاوت است.

(مثال)

```
password → type="password"
email → type="email"
tel → type="tel"
```

فیلد textarea:

- در این فیلد شما می‌توانید به میزان نامحدود کاراکتر وارد کنید. این فیلد برای دریافت پیغام متنی استفاده می‌شود که می‌توانید برای کادر آن طول و ارتفاع مشخص کنید.

(مثال)

```
<textarea name="content" rows="10" cols="30"></textarea>
```

- فیلد radio: این فیلد معمولاً برای تعیین جنسیت یا وضعیت تأهل استفاده می‌شود یا سوالات چند گزینه‌ای.
- در این نوع فیلد name را برای همه حالات یکی در نظر بگیرید اما value‌ها باید متفاوت باشند.

(مثال)

```
<input type="radio" name="gender" value="0">خانم
<input type="radio" name="gender" value="1" checked>آقا
```

برای فعال بودن یکی از گزینه‌ها به صورت پیش‌فرض از صفت checked استفاده کنید.

- فیلد checkbox: فیلد چک باکس مثلاً می‌تواند برای قسمت علاقه‌مندی‌ها به کار رود.

(مثال)

```
<input type="checkbox" name="fav[0]">ورزش
<input type="checkbox" name="fav[1]">هنر
<input type="checkbox" name="fav[2]">موسیقی
fav[...]
```

- فیلد عکس: برای دریافت عکس به کار می‌رود.

(مثال)

```
<input type="file" name="photo">
```

فیلد select:

- برای اینکه بتوانید لیستی از موارد انتخابی تهیه کنید از تگ select استفاده کنید، که هر مورد از موارد لیست در تگ option نوشته می‌شود.

(مثال)

```
<select name="grade">
  <option value="1">دیپلم</option>
  <option value="2">کار دانی</option>
  <option value="3" selected>کارشناسی</option>
</select>
```

selected: یک گزینه را به صورت پیشفرض انتخاب می‌کند.

صفات مشترک فیلدها:

readonly: از این صفت برای فقط خواندنی کردن عناصر فرم استفاده می‌شود. این صفت که جزء صفت‌های boolean است در صورتی که برای عناصر فرم استفاده شود، کاربر امکان تغییر محتویات آن عنصر را نخواهد داشت.

disabled: این صفت بولین عناصر را به حالت غیر فعال در می‌آورد. مقدار این فیلدها انتخاب و حتی به صفحه پردازشگر ارسال نمی‌شود.

required: از این صفت بولین برای آن دسته از عناصری که نمی‌توان آن‌ها را خالی رها کرد و در واقع وارد کردن اطلاعات در آن الزامی است استفاده می‌شود. از این صفت می‌توان در تگ‌های `textarea` و اغلب تگ‌های `input` استفاده کرد. مرورگرهایی که از این صفت پشتیبانی می‌کنند. در صورت خالی رها کردن فیلدهایی که این صفت را اختیار کرده‌اند پیغام خطا نمایش می‌دهند.

autocomplete: خاصیت `autocomplete` مشخص می‌کند که آیا `autocomplete` یک فرم یا فیلد ورودی باید روشن باشد یا خاموش. هنگامی که `autocomplete` روشن است، مرورگر به صورت خودکار شروع به تکمیل مقادیر مبتنی بر مقادیری که کاربر قبلاً وارد کرده است، می‌کند.

دکمه‌ها:

- دکمه **submit**: این عنصر برای تأیید فرم استفاده می‌شود. زمانی که بر روی دکمه submit کلیک می‌شود فرم به آدرسی که به action نسبت داده شده، فرستاده می‌شود.

```
<input type="submit" name="submit-reg-form" value="ارسال">
```

value مقدار نیست که روی دکمه نوشته می‌شود.

- دکمه **reset**: این دکمه داده‌های فرم را به حالت اولیه باز می‌گرداند.

```
<input type="reset" name="reset" value="پاک کردن">
```

- دکمه **button**: برای ساختن دکمه به کار برده می‌شود. این دکمه به تنهایی کاری انجام نمی‌دهد. با دانستن زبان script می‌توان دستوراتی به این دکمه‌ها داد تا کاری را انجام دهند.

```
<input type="button" name="button" value="Click Me!">
```

تگ video:

این تگ برای گذاشتن ویدئو در سایت استفاده می‌شود. که فقط فرمت‌های ^{۲۳}webM، ^{۲۴}mp4 و ^{۲۵}ogg را پشتیبانی می‌کند. طریقه استفاده از این کد به صورت زیر است:

```
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  Your browser does not support the video tag.
</video>
```

تگ audio:

این تگ برای گذاشتن صدا در وب استفاده می‌شود. که فقط فرمت‌های ^{۲۶}mp3، ^{۲۷}wav و ^{۲۸}ogg را پشتیبانی می‌کند. طریقه استفاده از این کد به گونه زیر است:

```
<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
  Your browser does not support the audio element.
</audio>
```

^{۲۳} Web Matroska

«یک قالب چند رسانه‌ای است که برای ارایه یک فرمت ویدیویی آزاد و با کیفیت بالا برای استفاده در اچ تی ام ال ۵ ارایه شده»

^{۲۴} MPEG-4 = Moving Picture Expert group

^{۲۵} ogg

^{۲۶} MPEG audio layer3

^{۲۷} Windows Media Audio

CSS

^{۲۸} CSS به معنای «صفحات استایلی آبشاری» است و روشی است برای قالب بندی و طراحی اجزای صفحه از قبیل متن، تصویر، زمینه صفحه، کادرها و تقریباً هر چیزی که در طراحی صفحه استفاده می‌شود. از طریق CSS می‌توانیم از دوباره نویسی کدهای HTML که باعث ایجاد مشکل در کدنویسی و همچنین کند شدن سرعت بارگذاری صفحه در اینترنت می‌شود جلوگیری کنیم. برای مثال می‌توانید یک بار نوع فونت مورد استفاده در یک صفحه را تعیین کنید و دیگر مجبور نباشید کد مربوط به فونت را در صفحه ده‌ها بار به کار ببرید. همچنین می‌توانید از این ابزار قدرتمند در مواردی به جای جاوا اسکریپت نیز استفاده کنید و چون مشکلات جاوا اسکریپت مثلاً غیر فعال بودن جاوا اسکریپت در مرورگر را ندارد با خیالی آسوده‌تر می‌توانید از آن استفاده کنید.

CSS هم مانند جاوا اسکریپت این امکان را به شما می‌دهد که از یک فایل استایل خارجی برای تعیین خواص بخش‌های مختلف صفحه‌های یک سایت استفاده کنید. در این صورت می‌توانید ظاهر همه صفحه‌ها را با ایجاد یک تغییر کوچک در فایل استایل خارجی تغییر دهید. مثلاً فرض کنید که رنگ متن همچنین نوع فونت متن صفحه‌هایتان را در یک فایل استایل تعیین کرده‌اید و از آن در طراحی چندین صفحه استفاده کرده‌اید، حالا اگر بخواهید رنگ متن همه این صفحات را تغییر دهید تنها کافی است تا رنگ متن را در همان فایل استایل تغییر دهید و نیازی به تغییر دادن همه صفحات ندارید.

روش‌های اعمال کدهای CSS به صفحه:

inline (در خط): در این روش، کدهای CSS در **صفت style** در بین صفات دیگرِ تگ مورد نظر نوشته می‌شود.

```
<h1 style="CSS کدهای">Hello!</h1>
```

Embedded (توکار): در این روش، کد معمولاً بین تگ head و در میان **تگ <style>** نوشته می‌شود.

```
<head>
  <style>
    h1 { CSS کدهای }
  </style>
</head>
```

²⁸ Cascading Style Sheet

External (خارجی): در این روش کدهای CSS در فایلی با پسوند CSS نوشته می‌شوند و توسط link که بین تگ head قرار می‌گیرد، به صورت زیر به صفحه پیوند زده می‌شود.

```
<head>  
<link rel="stylesheet" href="آدرس فایل سی.اس.اس">  
</head>
```

یک نمونه کد CSS:

جدا کننده صفات

نام مقدار

```
<h1 style="color:red; font-family:Tahoma; font-size:12px;">Hello CSS!</h1>
```

شکل دهی به متن با CSS:

فونت:

```
font-family:Tahoma;  
font-size:13px;  
font-weight:bold;
```

صفات text :

```
color:rgba (255,0,0,0.5);
```

alpha حالت نامرئی و شیشه‌ای به قالب و نوشته می‌دهد. برای متن استفاده نکنید بهتر است.

direction: rtl;	متن صفحه را راست چین می‌کند.
letter-spacing: 5px;	۵ پیکسل بین حروف فاصله می‌دهد.
text-align: center;	متن را وسط صفحه نمایش می‌دهد.
word-spacing: 10px;	فاصله بین کلمات را زیاد می‌کند تا هماهنگ شوند.
text-decoration: blink; text-decoration: line-through; text-decoration: none;	چشمک زدن حروف خط زدن ساده
text-shadow: 3px 3px 3px black; 	به متن سایه می‌دهد. اگر X و Y سایه صفر باشد حرفه‌ای‌تر و زیباتر به نظر می‌رسد.

صفات background :

```
background-color:rgb (شماره رنگ);  
background-image:url('آدرس عکس');  
background-attachment:fixed;
```

وقتی background-attachment را مساوی fixed قرار دهید. بک‌گراند ثابت می‌ماند و اسکرول نمی‌شود.

روش دوم استفاده از کدهای CSS

انتخاب‌گرهای CSS یا Selectors:

وقتی قصد استفاده از روش دوم اعمال کدهای CSS را دارید، باید در بالای صفحه بین تگ style کدها را بنویسید. حالا سؤالی که مطرح می‌شود این است که کدهایی که آن بالا نوشته می‌شود، به کدام تگ یا تگ‌ها اعمال می‌شود؟

این شما هستید که باید انتخاب کنید که این کدها مربوط به کدام تگ یا تگ‌ها هستند. برای این کار باید آن تگ‌ها را در اصطلاح select کنید.

روش‌های مختلفی برای انتخاب وجود دارد که ما به معرفی چند مورد که پرکاربردتر هستند می‌پردازیم:

۱. **بیان نام تگ:** در این حالت کدهای CSS به تمام تگ‌های نامبرده اعمال می‌شوند.

```
<head>
  <style>
    a {color:red;}
  </style>
</head>
```

۲. **تعریف کلاس:** در این روش کدهای CSS فقط به تگ‌هایی اعمال می‌شوند که نام کلاس مورد نظر برای آن‌ها تعریف شده باشد.
(مثال)

```
.right_links
{
    color:darkred;
}
.....
<a class="right_links"></a>
```

۳. **بیان id تگ:** در این حالت کدهای CSS فقط به تگی که id آن برابر با id نامبرده باشد اعمال می‌شود.

^{۲۹}id به معنای شناسه است. id در یک صفحه یکتاست؛ یعنی دو تگ نمی‌توانند یک id یکسان داشته باشند.

²⁹ identifier

```

<head>
  <style>
    #header
    {
      صفات مورد نظر
    }
  </style>
  .....
<body>
  <div id="header">

```

۴. بیان مسیر تگ: در این حالت، کدهای CSS فقط به تگ‌هایی اعمال می‌شود که در مسیر ذکر شده قرار دارند.

```

#header a
{
  صفات مورد نظر
}

```

عبارت بالا یعنی تمام تگ‌های a که در عنصری با آی.دی برابر با header قرار دارند تحت تأثیر این صفات باشند.

توضیحات در CSS:

برای نوشتن توضیحات در CSS به این صورت عمل می‌کنیم.

```

/*
توضیحات
*/

```

تگ div و span:

دو تگ نامرئی هستند برای اینکه به محتوای داخل آن‌ها کد CSS اعمال کنید.

```

<div>...</div>
<span>...</span>

```

div خاصیت block دارد ولی span خاصیت inline دارد یعنی span فقط دور محوطه تعیین شده را به خود اختصاص می‌دهد، اما div کل خط را به خود اختصاص می‌دهد.

رخدادها در (Events in CSS):

در CSS می‌توان به وسیله صفات مختلف رخدادهایی را ایجاد کرد که جلوه خاصی به سایت شما می‌بخشند. که پر کاربردترین آن‌ها hover است.

```

a
{
    color:blue;
    background-color:yellow;
}
a:hover
{
    color:black;
    background-color:orange;
}

```

در مثال بالا تگ‌های a صفاتی را دارا هستند که با قرار گرفتن ماوس بر روی آن‌ها این صفات تغییر می‌کنند. مثلاً رنگ متن این تگ‌ها آبی بوده که با قرار گرفتن ماوس بر روی آن سیاه می‌شوند. یا رنگ پس زمینه از زرد به نارنجی تغییر می‌یابد.

آشنایی با مابقی رخدادهای active و visited و ... را به پشتکار شما می‌سپاریم. بحث طراحی وب بسیار گسترده و متنوع است و نمی‌توان تمام کدها و نحوه استفاده از آن‌ها را در یک جزوه گنجاند. ما سعی داریم تا پرکاربردترین کدها را به شما آموزش دهیم.

ترفند CSS sprite :

استفاده از CSS Sprites برای عکس‌های مربوط به iconهای سایت و عکس‌های مربوط به منوی سایت و کلاً برای عکس‌های کم‌حجم، پیشنهاد می‌شود. به این ترتیب که عکس‌های مورد نظر را در یک تصویر بزرگتر قرار می‌دهید و با استفاده از CSS قسمتی از تصویر بزرگتر را که تصویر مورد نظر شما قرار دارد، در پشت یک div یا span قرار می‌دهید.

با این روش، فقط یک بار «تقاضا» یا Request به سرور سایت شما ارسال می‌شود و عکس بزرگتر یک بار لود شده و با تکنیک CSS تمام iconها و backgroundهای کم‌حجم روی صفحه قرار داده می‌شوند. علاوه بر این شما می‌توانید با تکنیک hover با قرار گرفتن ماوس بر روی تصویر، تصویر دیگری را به کاربر نشان دهید.

یعنی اگر آیکون شما ۵۰ پیکسلی هستند وقتی ماوس بر روی آیکون قرار گرفت ۵۰ پیکسل بعدی و در حقیقت تصویر بعدی نمایش داده شود.

:background-repeat

این ویژگی تعیین کننده نوع تکرار شدن تصویر زمینه است. یعنی تعیین می‌کند تصویر بک‌گراند در چه محوری تکرار شود. مقادیری را که می‌توان به این ویژگی اختصاص داد عبارتند از:

• repeat :

این مقدار تعیین می‌کند که تصویر زمینه در صورتی که از اندازه پنجره کوچکتر بود هم به صورت افقی و هم عمودی تکرار شود تا همه زمینه پنجره را بپوشاند.

• **:no-repeat**

این گزینه مشخص می‌کند که تصویر زمینه نباید تکرار شود حتی اگر از اندازه پنجره کوچکتر بود.

• **:repeat-x**

این گزینه مشخص می‌کند که تصویر باید به صورت افقی تکرار شود نه عمودی.

• **:repeat-y**

این گزینه مشخص می‌کند که تصویر زمینه باید به صورت عمودی تکرار شود.

:background-position

این ویژگی برای تعیین موقعیت تصویر زمینه در پشت عنصر استفاده می‌شود. این ویژگی زمانی مورد استفاده قرار می‌گیرد که از **background-repeat: no-repeat** استفاده شده باشد. این ویژگی می‌تواند دو مقدار را همزمان به خود اختصاص دهد. با این دو مقدار می‌توانید موقعیت عمودی و افقی تصویر در صفحه را تعیین کنید. توجه داشته باشید که این دو مقدار باید با یک فاصله خالی از هم جدا شوند نه کاما یا نقطه ویرگول.

دو مقداری که می‌توان به این ویژگی اختصاص داد از مقادیر زیر انتخاب می‌شوند:

• **top, center or bottom**

• **left, center or right**

برای این ویژگی باید از هر دسته از مقادیر بالا فقط یکی انتخاب شود. در اینجا می‌توانید برخی از موارد قابل استفاده را مشاهده کنید:

- `background-position: top right;`
این ویژگی تصویر زمینه را در بالا و سمت راست عنصر نمایش می‌دهد.
- `background-position: center right;`
این ویژگی تصویر را در سمت راست وسط عنصر نمایش می‌دهد.
- `background-position: bottom center;`
این ویژگی تصویر زمینه را در پایین مرکز عنصر نمایش می‌دهد.

می‌توانید به جای گزینه‌های بالا از مقداری در مقیاس پیکسل یا درصد نیز استفاده کنید تا تصویر زمینه را در هر فاصله‌ای از بالا و سمت چپ عنصر که بخواهید قرار دهید. در این روش فقط کافیست مقدار فاصله از

سمت چپ عنصر را نوشته و مقیاس آن را مشخص کنید و با یک فاصله مقدار فاصله از بالای عنصر همراه با مقیاس آن را بنویسید.

ساخت قالب سایت با div:

همانطور که گفتیم جدول یکی از سنگین‌ترین عناصر وب است شما می‌توانید برای کادر بندی و حتی قالب کلی از این دو تکنیک استفاده کنید. برای مثال برای ایجاد قالب کلی به وسیله div، که قبلاً با جدول طراحی کرده بودیم به کدهای زیر نیاز دارید.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Fisrt Page</title>
    <style type="text/CSS">
      body
      {
        direction: rtl;
      }
      #container
      {
        width: 950px;
        min-height: 1000px;
        padding: 10px;
        border: 1px dashed gray;
        margin: 0 auto;
      }
      .brd
      {
        border: 1px solid black;
      }
      #header
      {
        height: 150px;
        margin-bottom: 10px;
      }
      #rightcol
      {
        width: 200px;
        float: right;
        min-height: 700px;
      }
      #main_container
      {
        margin-right: 10px;
        width: 735px;
        min-height: 700px;
        float: right;
      }
      #footer
      {
        height: 100px;
      }
    </style>
  </head>
  <body>
  </body>
</html>
```

```

margin-top: 10px;
}

</style>
</head>

<body>
<div id="container">
<div id="header" class="brd"></div>
<div id="rightcol" class="brd"></div>
<div id="main_container" class="brd"></div>
<br clear="both">
<div id="footer" class="brd"></div>
</div>
</body>
</html>

```

:border

border را می‌توان به تمام عناصر وب نسبت داد همچنین به وسیله border می‌توانید محدوده divها را مشخص کنید؛ استفاده از نوع و رنگ بُردر امری کاملاً سلیقه‌ای است. بُردرها همچنین صفاتی را در بر می‌گیرند.

صفات border

border-style: با استفاده از پارامتر border-style می‌توانید مشخص کنید که آیا می‌خواهید کادر داشته باشیم یا نه، و همچنین نوع کادری را که می‌خواهید ایجاد کنید مشخص کنید. مقداری را که می‌توانید برای border-style استفاده کنید. عبارتند از:

- **none**: با استفاده از این مقدار تعیین می‌شود که نمی‌خواهید کادر داشته باشید. در صورتی که مقداری غیر از none تعیین شود کادر خواهید داشت.

• **solid**:

این یک div با بُردر solid است.

• **dashed**:

این یک div با بُردر dashed است.

• **dotted**:

این یک div با بُردر dotted است.

• **double**:

این یک div با بُردر double است.

همانطور که می‌بینید عرض این کادرها بیش از آن چیزی است که نیاز دارید چون برای این کادرها عرضی مشخص نشده اگر بخواهید عرض این کادرها به اندازه مشخصی باشد باید از پارامتر width که قبلاً با آن آشنا شدید استفاده کنید. به مثال زیر توجه کنید:

```
<div style="width:300px ; border-style:solid"> ۳۰۰ پیکسل  
</div> تعیین شده است
```

اکنون **div**، با ابعاد مورد نظر شما ساخته می‌شود:

برای این کادر عرض ۳۰۰ پیکسل تعیین شده است

border-width: این پارامتر به اجازه می‌دهد تا میزان ضخامت کادرها را تعیین کنید.

```
border-width:5px;
```

border-color: این صفت رنگ برادر را تعیین می‌کند.

```
border-color:red;
```

یک روش خلاصه‌تر این که تمام خصوصیات را در یک دستور قرار دهید:

```
border:red 1px solid;
```

اگر خواستید هر برادر دارای یک صفت خاص باشد به روش زیر عمل کنید:

```
border-top:2px yellow dashed;  
border-right:5px red solid;  
border-bottom:10px green dotted;  
border-left:1px blue double;
```

border-radius: به وسیله این صفت می‌توان گوشه‌های برادر را قوس داد و حتی **div**های دایره‌ای شکل

ایجاد کرد. به مثال زیر توجه کنید:

```
<div style="width:200px; height:200px; border:1px solid black; border-  
radius:100px 100px 100px 100px;">یک دایره</div>
```



:margin

margin: پارامتر margin فاصله عنصر با عناصر اطراف را تعیین می‌کند. این گزینه اجازه می‌دهد تا برای

هر یک از اجزای صفحه حاشیه مشخصی را تعیین کنید. این صفت اگر به تنهایی ست شود به همه جهات اعمال می‌شود.

```
margin:10px;
```

در مثال بالا عنصر مورد نظر از چهار جهت 10px فاصله می‌گیرد. البته می‌توان فقط از یک سمت فاصله تعیین کرد فقط کافیسست جهت مورد نظر را ذکر کرد. برای مثال:

```
margin-top:10px;
```

این کد تنها از بالا 10px حاشیه قرار می‌دهد. البته این صفت مقادیر left, right, bottom را نیز به خود می‌گیرد.

برای قرار دادن یک div در وسط یک div دیگر (پدرش)، از صفت margin به صورت زیر استفاده کنید.

```
margin:0 auto;
```

padding:

پارامتر padding فاصله محتوای عنصر تا خط دور را تعیین می‌کند و مشابه شناسه cellpadding در جدول‌هاست با این تفاوت که می‌توان آن را برای بالا، پایین، چپ و راست به صورت جداگانه تعریف کرد به جای اینکه به صورت یکجا تعیین شود. و می‌توان مقادیر top, left, right, bottom را به آن نسبت داد. توجه داشته باشید padding نیز اگر به تنهایی قید شود روی همه جهات اعمال می‌شود.

نکته: مقادیر padding و border به width، div اضافه می‌شود. پس باید مقدار padding و border را از طول div کم کنید.

float:

float به معنای شناور است. این خاصیت وقتی مورد استفاده قرار می‌گیرد که قصد داشته باشید بدون استفاده از جدول چهارچوب صفحه را طراحی کنید.

این خاصیت باعث می‌شود تا یک بلوک بسته به مقداری که به این ویژگی داده می‌شود تا حد امکان در سمت چپ یا راست عنصر والدش قرار گیرد و سایر عناصر و متن موجود در زیر عنصر فعلی در اطراف این بلوک قرار گیرند.

برای پایان دادن به حالت float از قطعه کد زیر استفاده کنید:

```
<br clear="both">
```

صفات برای تعیین ارتفاع:

صفت min-height: این پارامتر حداقل ارتفاع مجاز را برای عنصر تعیین می‌کند یعنی اگر محتوای آن بیشتر از مقدار داده شده باشد، به ارتفاع آن اضافه می‌شود.

line-height: این ویژگی این امکان را می‌دهد تا ارتفاع سطرها را از هم مشخص کنید. برای مثال اگر بخواهید سطرها از هم ۵۰ پیکسل فاصله داشته باشند می‌توانید به صورت زیر عمل کنید:

```
<div style="line-height: 50px">سطر اول<br>سطر دوم</div>
```

:Position

ویژگی position اجازه می‌دهد تا تعیین کنید که موقعیت یک عنصر نسبت به پنجره مرورگر و یا نسبت به محل درج در صفحه، مشخص شود. چندین مقدار را می‌توان برای این ویژگی در نظر گرفت:

:static

این مقدار پیش فرضی است که در صورت تعریف نشدن position در صفحه استفاده می‌شود. با این مقدار اجزا در همان جایی که کد آنها را می‌نویسید دیده می‌شوند.

:absolute

عنصر در صفحه معلق می‌شود و نسبت به مرورگر کاربر تعیین موقعیت می‌شود اما نسبت به اسکرول کردن مقاومت نشان نمی‌دهد؛ یعنی با اسکرول، جابجا می‌شود.

:fixed

همانند absolute نسبت به مرورگر تعیین موقعیت می‌شود اما با اسکرول کردن، جای عنصر تغییری نمی‌کند.

:relative

عنصر نسبت به پدرش تعیین موقعیت می‌شود.

مثال: گاهی اوقات در بعضی سایت‌ها می‌بینید که یک تصویر روی همه عناصر وب قرار گرفته، مثل اینکه در هوا معلق باشد و با اسکرول جابجا نمی‌شود. (برای دیدن نمونه به سایت aftab.cc مراجعه کنید) برای داشتن یک تصویر ثابت (نوار ابزار یا عکس) از قطعه کد زیر استفاده می‌شود:

```
#ads
{
    position: fixed;
    bottom: 0px;
    right: 0px;
}
```

```
<div id="ads"></div>
```

ابتدا تصویر را در یک div قرار داده؛ سپس به آن یک id نسبت داده دهید و برای آن یک position در نظر بگیرید. همچنین می‌توانید تعیین موقعیت کنید یعنی از مقادیر top, right, bottom, left استفاده کنید.

ترتیب اعمال کدهای CSS به صفحه: خارجی، inline، Embedded. اما **اولویت**، برعکس این ترتیب است. یعنی همیشه اولویت با آخرین کد و نزدیکترین کد نوشته شده است. برای اجرا شدن یک کد بدون در نظر گرفتن اولویت به شکل زیر عمل کنید:

```
color:red !important;
```

important! به مرورگر می‌فهماند که این کد در اولویت قرار دارد و تنها این کد باید اجرا شود پس این کد را روی کدهای قبلی overwrite می‌کند یعنی دوباره نویسی می‌کند.

صفت z-index

به کمک این صفت بالا و پایین بودن یک عنصر نسبت به عناصر دیگر را مشخص می‌شود. هر چه عددی که به این صفت اختصاص داده شده بیشتر باشد اولویت بالاتر و در نتیجه بالاتر از همه (یعنی روی همه) عناصر قرار خواهد گرفت. اگر بخواهید زیر عناصر دیگر قرار گیرد می‌توانید حتی از اعداد منفی استفاده کنید.

صفت overflow

وقتی محتویات قسمتی از صفحه طول یا عرضی بزرگتر از آن در برگیرنده داشته باشد (بیرون بزند)، با پارامتر overflow می‌توان مشخص کرد که آن قسمت زیادی نمایش داده شود یا مخفی شود. به صورت پیش فرض قسمت‌های اضافی نمایش داده می‌شوند ولی با این پارامتر می‌توانیم آن‌ها را مخفی کرد. به طور کلی چند مقدار را می‌توان برای پارامتر overflow استفاده کرد:

۱. **visible**:

با قرار دادن این مقدار برای overflow همه محتویاتی که طول یا عرض بیش از اندازه داشته‌اند نمایش داده خواهند شد.

۲. **hidden**:

این مقدار باعث خواهد شد تا محتویات اضافه، نمایش داده نشوند.

۳. **scroll**:

این مقدار باعث خواهد شد تا در صورتی که محتویات اندازه‌ای بیش از طول و عرض لازم دارند با استفاده از اسکرول بار قابل مشاهده باشند.

صفت display: به کمک این صفت و نسبت دادن مقدار none به آن، می‌توانید عناصری را از صفحه حذف کنید در صورتی که در کدها وجود دارند. همچنین می‌توانید با خاصیت inline-block آن منوهای افقی بسازید. البته خاصیت‌های دیگری دارد که خیلی پرکاربرد نیستند.

صفت list-style-type: بر اساس نوع لیست، عناصر لیست با شماره یا شکل‌های خاصی مثل دایره تو پر، تو خالی و غیره مشخص می‌شوند که مقدار none آن بیشترین استفاده را دارد که موجب حذف شکلک کنار آیتم‌های لیست می‌شود.

صفت box-shadow: برای سایه دادن به تمامی عناصر وب به جز متن استفاده می‌شود. مقادیری که می‌گیرد همانند text-shadow است.

صفت border-image: به وسیله این خاصیت می‌توان یک عکس را به عنوان خطوط حاشیه برای یک عنصر به کار برد.

صفت background-size: اندازه عکس پس زمینه عنصر را تعیین می‌کند.

صفت opacity: برای ایجاد شفافیت و حالت نامرئی در عناصر وب استفاده می‌شود.

صفت transition: خاصیت transition به همراه افکت‌هایی که با آن به کار می‌برید، این امکان را به یک عنصر می‌دهد تا به صورت تدریجی از یک اندازه و شکل به یک اندازه و شکل دیگر تغییر وضعیت دهد؛ برای مثال فرض کنید، شما یک عکس دارید که نمونه کوچک آن را بر روی صفحه قرار داده‌اید و قصد دارید تا زمانی که کاربر موس خود را بر روی آن می‌برد، به صورت تدریجی به اندازه بزرگ و اصلی خود در بیاید و کاربر آن را مشاهده نماید، سپس هنگامی که کاربر موس خود را از روی آن می‌برد، به اندازه قبلی و کوچک خود برگردد.

صفت animation: استفاده از انیمیشن بدون نیاز به استفاده از کتابخانه‌های jQuery باعث می‌شود سایت طراحی شده سبک‌تر و در عین حال زیبا باشد. انیمیشن استفاده‌های زیادی دارد. برای مثال برای حرکت لوگوی سایت از کد زیر استفاده کنید:

```

@keyframes logoPlay
{
    from{opacity:0; margin-right:100px;}
    to{opacity:1; margin-right:0px;}
}
#logo
{
    animation:logoPlay 2s;
}
.....


```

ضمیمه فونت به سایت:

برای پیوست فونت از @font-face استفاده می‌شود. این دستور راهکاری است کاملاً بر پایه‌ی CSS که امکان پیوست فونت را به صفحه می‌دهد.

گام اول) یک پوشه به نام فونت بسازید و فونت خود را در آن کپی کنید.
گام دوم) کد زیر را در فایل CSS خود وارد کنید.

```

@font-face
{
    font-family:نام دلخواه;
    src:url('آدرس فونت مورد نظر');
}

```

گام سوم) فونت خود را به body نسبت دهید.

```

body
{
    font-family:نام انتخابی;
}

```

Frameworks

فرض کنید شما قرار است یک سایت طراحی کنید که در آن، افراد ثبت نام می کنند و مثلاً شارژ تلفن همراه می خردند و بر اساس خرید، تخفیف می گیرند و در قرعه کشی شرکت می کنند و امثالهم. یا مثلاً ثبت نام می کنند و در مسابقه پیش بینی فوتبال جام جهانی شرکت می کنند و امتیاز و جایزه می گیرند. سؤال این است: آیا فرضاً اعتبارسنجی ایمیل، جزء اصلی این پروژه به حساب می آید؟ (اعتبار سنجی ایمیل یعنی مثلاً اگر کاربری gsddf را به عنوان ایمیل وارد کرد، ما هشدار بدهیم که این چیزی که شما وارد کردید، ایمیل نیست! ایمیل ساختاری شبیه به z@y.z دارد.) یا اعتبار سنجی نام کاربری (که فقط حاوی حروف الفبای انگلیسی و عدد و چند کاراکتر دیگر مثل - و _ باشد)...

طبیعتاً کار اصلی شما متمرکز شدن روی طراحی سیستم فروش شارژ یا پیش بینی است و این عاقلانه نیست که ساعت ها وقت خود را صرف نوشتن کد اعتبارسنجی ایمیل و نام کاربری کنید.

یا مثلاً در برنامه نویسی ویندوز، فرض کنید می خواهید یک فرم طراحی کنید و چند دکمه روی آن بگذارید که هر دکمه کار خاصی انجام می دهند.

در حالت عادی (بدون استفاده از فریم ورک ها) می دانید چند ساعت زمان لازم دارید که فقط مفهوم یک دکمه یا فرم را به ویندوز بفهمانید؟ (ویندوز چه می فهمد دکمه چیست؟ فرم چیست؟ فقط می تواند کدها را اجرا کند. حالا شما ممکن است با کدها به او بفهمانید که دکمه یعنی یک باکس که یک هاله رنگی درون آن است و اگر موس روی آن قرار گرفت کمی نارنجی شود و خلاصه کلی کدنویسی دیگر که شاید روزها وقت شما را بگیرد)

راه حل صرفه جویی در زمان: فریم ورک!

مثلاً میکروسافت می گوید: ای برنامه نویس! شما نمی خواهید درگیر نوشتن کد دکمه و فرم شوی، ما یک بار نوشته ایم و آن را در یک بسته قرار داده ایم (که در اصطلاح به آن Framework یا چارچوب می گوییم) شما فقط کمی وقت بگذار و یاد بگیر که چطور با آن فریم ورک کار کنی، چطور در چارچوب ما حرکت کنی...

در این صورت، هر وقت یک فرم خواستی، فقط کافی ست کتابخانه ای که در آن فریم ورک هست را فراخونی کنی و بنویسی:

```
Form frm = new Form();
```

همین! ما یک فرم با تمام جزئیاتش برای تو می سازیم!

مثلاً میکروسافت در مجموعه NET Framework. خود، هزاران امکان را تعبیه کرده و فقط با چند خط کد می‌شود از آن‌ها استفاده کرد. مثلاً در پارس‌خوان، خواندن متون انگلیسی در دات نت فریم‌ورک تعبیه شده بود، بنابراین ما دیگر ننشستیم آن‌را دوباره بنویسیم، بلکه خواندن متن انگلیسی را با چند خط کد به میکروسافت محول کردیم و وقتمان را صرف خواندن متن فارسی کردیم که در آن پکیج نبود...

فریم‌ورک در برنامه‌نویسی تحت وب به چه معنی است؟

همانطور که در برنامه‌نویسی ویندوز، فریم‌ورک داریم، در برنامه‌نویسی تحت وب هم ده‌ها فریم‌ورک داریم. مثلاً برای زبان جاوااسکریپت (JavaScript) فریم‌ورکی داریم به نام jQuery که هزاران کار را به راحتی برای شما انجام می‌دهد. با چند خط کد، بهترین اسلایدها روی عکس‌های شما اعمال می‌شود. انواع و اقسام پنجره‌های pop-up و prototype به راحتی به کاربر نمایش داده می‌شود و بسیاری کارهای دیگر که شاید باور نکنید، تماماً با چند خط کد در پروژه شما تعبیه می‌شود.

یا مثلاً یک فریم‌ورک برای زبان CSS می‌تواند bootstrap باشد که ما در پروژه **نمرا** از آن استفاده کردیم. اگر بدانید چقدر راحت و زیبا می‌توان انواع جلوه‌ها را روی عناصر صفحه ایجاد کرد! یا مثلاً فقط با یک خط کد، سایت شما برای تبلت و موبایل آماده می‌شود!!!



getbootstrap.com

برای PHP^{۳۰} چه فریم‌ورک‌هایی داریم؟

³⁰ PHP: Hypertext Preprocessor

برای زبان PHP هم فریم‌ورک‌های مختلفی توسط خوره‌های این زبان طراحی شده است. لیستی از این فریم‌ورک‌ها را می‌توانید در سایت زیر مشاهده کنید:

<http://phpframeworks.com>

یکی از آن‌ها که طرفداران بیشتری دارد و انصافاً مستندات بهتری هم دارد، CakePHP است.

مثلاً فرض کنید می‌خواهید یک فرم را به طور کامل Validate کنید (اعتبارسنجی). نام کاربری و ایمیل و تاریخ تولد و کد پستی و غیره... لازم نیست درگیر قواعد و قوانین و پیغام‌ها شوید! می‌توانید در <http://book.cakephp.org/2.0/en/models/data-validation.html> ببینید که چقدر ساده می‌شود یک فرم را اعتبارسنجی کرد. (توجه: ممکن است در نگاه اول کمی پیچیده به نظر برسد اما اگر کدنویسی کرده باشید در پنج دقیقه می‌توانید یاد بگیرید که چطور باید از آن استفاده کنید)

آیا WordPress و Joomla و امثالهم هم یک فریم‌ورک به حساب می‌آیند؟ فرق فریم‌ورک و پلتفرم چیست؟

در کل، کلمه فریم‌ورک را می‌توان به هر قطعه کد نسبتاً بزرگی که از انجام کارهای تکراری جلوگیری کند نسبت داد. پس با این حساب، بله، وردپرس و جوملا هم می‌تواند یک فریم‌ورک به حساب آید. اما دقت کنید که ما یک کلمه دیگر در دنیای برنامه‌نویسی داریم و آن کلمه (Platform بستر) است.

فرق فریم‌ورک و پلتفرم را شاید بتوان اینطور گفت که فریم‌ورک است که به شما کمک می‌کند تا کارتان را توسعه دهید اما معمولاً این شما هستید که به پلتفرم کمک می‌کنید تا توسعه پیدا کند!

یعنی مثلاً وردپرس به تنهایی بخشی از کارهای شما را انجام می‌دهد و بخشی از کارهایتان باقی می‌ماند که شما با توسعه وردپرس، آن امکانات را به آن اضافه می‌کنید و نهایتاً پروژه شما آماده است، این یعنی شما بر روی بستر وردپرس کارتان را انجام می‌دهید. اما در فریم‌ورک، چیزی آماده نیست. شما از صفر کار را شروع می‌کنید و با استفاده از فریم‌ورک کارتان را انجام و توسعه می‌دهید.

توضیح این دو کلمه کمی سخت است و باید اعتراف کرد که بسیار شبیه به هم هستند و خیلی از اوقات، به جای هم به کار می‌روند و هیچ مشکلی هم نیست.

برخی اصطلاحات دیگر در این زمینه:

اسکرپیت به چه معناست؟

یک کلمه دیگر هم در این زمینه مطرح است: اسکریپت (Script)

این کلمه، دامنه کوچک‌تری دارد. یعنی به وسعت فریم‌ورک نیست. ما به قطعه کد کوچکی که یک کار تکراری را از روی دوش شما بردارد، اسکریپت می‌گوییم. مثلاً ممکن است در یک پروژه، برای اعتبارسنجی ایمیل، کار شما با همین قطعه کد ساده در این آموزش راه بیفتد: [آموزش اعتبارسنجی ایمیل و URL در PHP با تابع filter_var](#)

یا برای داشتن یک گالری عکس، نیازی به استفاده از کتابخانه حجیم jQuery نداشته باشید و با این اسکریپت کارتان راه بیفتد: [قطعه کد جاوااسکریپت برای نمایش رندوم چند عکس در صفحه](#)

کامپوننت چیست؟

کلمه دیگری هم در این زمینه وجود دارد: Component یا مؤلفه. کامپوننت را چیزی تصور کنید بین اسکریپت و فریم‌ورک. یعنی نه به کوچکی اسکریپت است و نه به وسعت فریم‌ورک. کامپوننت معمولاً روی یک بستر نصب می‌شود و یک سری کارهای تکراری مرتبط با هم را انجام می‌دهد. مثلاً یک کامپوننت گیر می‌آورد برای وردپرس که مدیریت گالری عکس را به آن اضافه می‌کند.

پلاگاین چیست؟

کلمه دیگری که ممکن است در این زمینه بشنوید، Plug-in یا افزونه یا (Extension) است. پلاگاین هم همان مفهوم کامپوننت را دارد. یعنی روی یک بستر نصب می‌شود و یک کار تکراری را انجام می‌دهد. فقط یک فرق کوچک می‌توان بین کامپوننت و پلاگاین بیان کرد: شما می‌توانید از کامپوننت استفاده کنید تا یک کار را انجام دهید اما پلاگاین آن کاری که شما می‌خواستید انجام دهید را انجام داده! یعنی حتی نیازی به صرف زمان برای هماهنگ کردن و استفاده از پلاگاین نیست! (یعنی آن استفاده‌ای که شما می‌خواستید از کامپوننت کنید و چیزی را تولید کنید، پلاگاین آن استفاده را کرده و آن چیز را تولید کرده!) یک پلاگاین ممکن است خودش از چند کامپوننت استفاده کرده باشد...

توجه: هیچ وقت درگیر کلمات نشوید. فراموش نکنید که این‌ها همه از زبان انسان‌ها بیان شده و انسان‌ها ممکن است برای یک چیز مشابه کلمات مختلفی به کار ببرند و نهایتاً همه منظورشان یکی باشد! پس در این مورد بحث نکنید که تعریف کامپوننت را درست نگفتید یا فلان کلمه فلان تعریف را دارد! این‌ها وحی منزل نیست. ممکن است یکی bootstrap را فریم‌ورک بداند و همان لحظه یکی دیگر، آن را script و یا حتی هیچ کدام از این کلمات!! مهم این است که همه این‌ها قرار است باری را از روی دوش شما بردارند، همین!

استفاده از فریم‌ورک، آری یا خیر؟

نظرات مختلفی در این زمینه وجود دارد. خیلی‌ها می‌گویند: استفاده نکردن از فریم‌ورک یعنی اختراع مجدد چرخ! پس عقل حکم می‌کند که شما از فریم‌ورک استفاده کنید و چرخ را از اول اختراع نکنید. اما برخی افراد مثل من می‌گویند: شما اگر چرخ را از اول اختراع کنید دستتان بازتر است تا بخواهید از چرخ‌های

آماده استفاده کنید! (من معتقدم دلیل عقب ماندگی ما در رشته کامپیوتر این است که همیشه از چرخ‌های آماده استفاده کرده‌ایم! هیچ وقت نرفته‌ایم مثلاً CPU³¹ را از صفر دوباره اختراع کنیم تا فردا بتوانیم یک CPU با ویژگی‌های دلخواه‌مان تولید کنیم) مثلاً در پروژه‌های بزرگی مثل کارهای Adobe، آن‌ها هیچ وقت نمی‌آیند از فریم‌ورک استفاده کنند! معمولاً تمام کدهای خود را از صفر و خودشان می‌نویسند به همین دلیل هم هست که می‌توانند به راحتی کدها را هر تغییری که خواستند بدهند. (مثلاً برای تمام سیستم عامل‌ها و بسترهای سخت‌افزاری مختلف آماده کنند) در برنامه‌نویسی، معتاد شدن به یک فریم‌ورک، زمانی که برای انجام کار مورد نظر شما چیزی تعبیه نکرده است، کار را سخت می‌کند! مثلاً فریم‌ورک bootstrap برای سایت‌های RTL (راست به چپ) تقریباً تدابیر درستی نیاندیشیده و این کار شما را گاهی اوقات آنقدر سخت می‌کند که نهایتاً به این نتیجه می‌رسید که اگر خودم از اول می‌نشستم کد می‌نوشتم سریع‌تر می‌شد تا بخواهم مشکلات این فریم‌ورک را رفع کنم تا با شرایطم جور در بیاید.

استفاده از فریم‌ورک معایب دیگری نیز دارد:

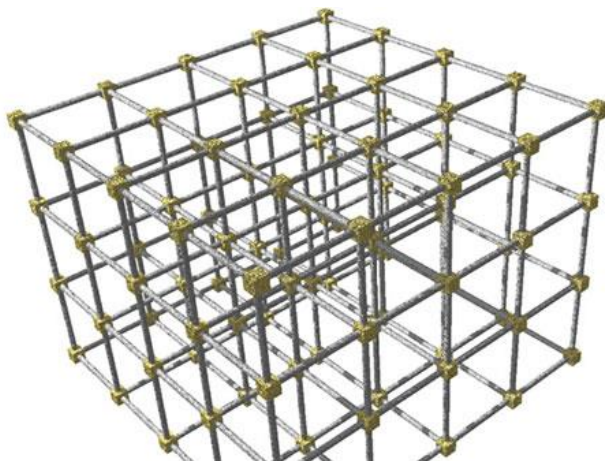
- اگر یک باگ ساده در فریم‌ورک مورد استفاده شما کشف شود، چون آن فریم‌ورک در اختیار همه است، سایت شما به راحتی در خطر می‌افتد. مثلاً سایت‌هایی که با وردپرس بالا می‌آیند، اگر یک باگ ساده کشف شود، می‌بینید یک دفعه صدها سایت وردپرسی هک شدند!
- شما گاهی مجبورید دانلود حجم زیادی داده به کاربر تحویل کنید در حالی که می‌خواهید یک کار ساده انجام دهید که بدون فریم‌ورک در چند خط انجام می‌شد! مثلاً bootstrap حدود ۱۰۰ کیلوبایت است! در حالی که شاید ما فقط از ۱۰ کیلوبایت آن استفاده کرده‌ایم. (شاید بگویید خوب در بین کدها بگرد و اضافات را پاک کن! خوب، اگر می‌خواستیم وقت‌م را صرف پیدا کردن و پاک کردن کدهای اضافه کنم که از اول کد می‌نوشتیم که سنگین‌تر بودم!)
- خطر تکراری شدن پروژه شما وجود دارد. شاید یکی از بزرگ‌ترین عیب‌ها همین باشد! مثلاً تمام سایت‌هایی که از فریم‌ورک bootstrap یا jQuery استفاده می‌کنند، ظاهر تقریباً یکسانی دارند و این ممکن است کلاس کاری پروژه شما را به خطر بیندازد. (مشتري شما فکر می‌کند شما زمان زیادی صرف نکرده‌اید یا دزدی کرده‌اید!)
- دست شما باز نیست! معمولاً محدودیت‌هایی در کدهای فریم‌ورک‌ها وجود دارد که باعث می‌شود شما نتوانید آن‌طور که می‌خواهید روی پروژه‌تان مانور بدهید.
- استفاده از فریم‌ورک، عیب‌یابی و رفع عیب پروژه را بسیار سخت می‌کند. اگر فرضاً یک خطا در یک قطعه از کدهای فریم‌ورک ببینید، تا بخواهید بفهمید دلیل آن چه بوده، کلی زمان می‌برد! اما اگر خودتان آن کد را نوشته باشید، دقیقاً می‌روید بالای سر همان فایل و همان خطی که باعث شده آن خطا رخ دهد!

³¹ Central Processing Unit

با همه این معایب، اما شکی نیست که در برخی موارد نمی‌توان بدون Framework زندگی کرد! به خصوص زمانی که کلمه **Rapid** در پروژه مطرح شود. (یعنی بخواهید کار را سریع و کم هزینه تحویل دهید)

در کل، من خودم به این نتیجه رسیده‌ام که انسان خودش برای پروژه‌های خودش یک فریم‌ورک بسازد موفق‌تر است تا بخواهد از فریم‌ورک‌های دیگران استفاده کند (مگر اینکه هزینه‌ها اجازه ندهد). (تنوع فریم‌ورک‌ها هم به خاطر همین است! همه به مرور به این نتیجه رسیده‌اند که خودمان برای خودمان فریم‌ورک داشته باشیم دستمان بازتر است و وقتی آن‌را در اختیار عموم قرار داده‌اند، تعداد آن‌ها بالا رفته است. وگرنه کسی شک ندارد که جوملا و وردپرس می‌توانستند یک پروژه باشند! چرا یک پروژه نشدند؟ چون هر کدام احساس می‌کنند فریم‌ورک خودشان را بسازند راحت‌ترند!!!) در کل در استفاده از فریم‌ورک‌ها این آیه و حدیث یادتان باشد:

كُلُوا و اشْرَبُوا و لَا تُسْرِفُوا (بخورید و بیاشامید ولی اسراف نکنید) (آیه)
خَيْرُ الْأُمُورِ أَوْسَطُهَا (بهترین کار در امور، میانه‌روی در آن‌هاست) (حدیث)



کدام فریم‌ورک را انتخاب کنم؟

- مهم‌ترین دلیل انتخاب فریم‌ورک، مستندات آن فریم‌ورک است. در سایت رسمی هر کدام از آن‌ها که وارد شوید بخشی می‌بینید به نام Docs یا Documentations یا Help یا Support و امثالهم... آن‌جا را بررسی کنید و ببینید آیا با مستندات کنار می‌آیید یا خیر؟ به راحتی و با یک نگاه می‌شود فهمید که این فریم‌ورک آینده‌دار است یا خیر! روی آن سرمایه‌گذاری شده است یا خیر؟
- در سایت رسمی، برخی نمونه سایت‌ها هم که با آن فریم‌ورک کار شده باشد قرار داده شده‌اند، آن‌را هم بررسی کنید و ببینید آیا قوی کار شده یا خیر؟

- در کل بهترین‌ها در جستجوهای گوگلی در رتبه بالاتری قرار می‌گیرند. مثلاً به نتیجه [این جستجو \(PHP Framework\)](#) دقت کنید.

سخن پایانی:

وقتی برای مصاحبه برای استخدام در یک شرکت مرتبط با برنامه‌نویسی می‌روید، یکی از سؤالاتی که از شما می‌پرسند این است که:

با چه فریم‌ورک‌ها و پلتفرم‌هایی کار کرده‌اید؟

سعی کنید تجربه کار با چند تا از برترین‌ها را در کارنامه خود داشته باشید. این روزها WordPress آنقدر قوی شده است که می‌توان به عنوان فریم‌ورک و بستر بسیاری از کارها از آن استفاده کرد. شاید این پیشنهاد خوبی باشد که آن‌را بستر کارهای خود قرار دهید و در کنارش از فریم‌ورک‌های دیگر برای توسعه استفاده کنید.

طریقه ایجاد **alert** ها در **bootstrap**: یعنی چطور یک پیغام موفقیت (success) و یا خطا (error) ایجاد کنیم:

گام اول) دانلود فریم ورک bootstrap، یک پوشه دانلود می‌شود که حاوی چند فایل CSS، یک فایل javascript، و چند تصویر است.

گام دوم) قرار دادن فایل‌ها در پوشه‌های سایت. محتویات دانلود شده را در پوشه theme کپی می‌کنید. در CSS از فایل min استفاده نکنید چون می‌خواهید در آن تغییرات ایجاد کنید؛ در غیر اینصورت یعنی در javascript، مین‌ها را پاک کنید. اگر می‌خواهید سایت خود را به گونه‌ای طراحی کنید که در گوشی و تبلت قابل مشاهده باشد فایل اصلی responsive را پاک نکنید در غیر این صورت پاک کنید.

گام سوم) فایل‌ها را به صفحه لینک کنید. سپس به سایتی که بوت‌استرپ را دانلود کرده‌اید بروید روی گزینه Base CSS کلیک کرده و در قسمت Emphasis classes رنگ alert‌ها را انتخاب کنید؛ برای پیام خطا قرمز و برای صحیح سبز برای پیغام خطا کافیس `class="text-warninig"`، و برای پیغام موفقیت‌آمیز، `class="text- success"` قرار دهید. همین!

و یا برای قرار دادن یک آیکون پرینت در مستندات باز هم در Base CSS، به قسمت Icon glyphs مراجعه کنید. فقط کافیس `<i>` تگ `<i>` تایپ کرده و کلاس آن را برابر آیکون قرار دهید.

```
<i class="icon-print"></i>
```

JavaScript

چند نکته در مورد این زبان:

- جاوا اسکریپت با جاوا متفاوت است.
- ساختاری شبیه زبان C دارد.
- کار در جاوا اسکریپت راحت تر شده است.
- در انتهای خط اختیاری است.
- یک زبان سمت client است.
- یک زبان تابعی است.
- گاهی اوقات یک خطا در یک خط، کل ادامه کدها را از کار می اندازد.
- زبان JS به حروف کوچک و بزرگ حساس است.

انواع زبان های برنامه نویسی وب از نظر محل اجرای کد:

- **client-side** (سمت کلاینت): کدها ابتدا روی سیستم کاربر دانلود می شوند و سپس اجرا می شوند. یعنی بعد از دانلود کدها، اجرای آن ها وابسته به سرور نیست. مانند HTML, CSS, Javascript
- **server-side** (سمت سرور): اصل کدها روی سرور است و بعد از اجرا روی سرور، خروجی آن ها روی سیستم کاربر منتقل می شود. مانند PHP, ASP^{۳۲}, JSP^{۳۳}, Python, Ruby On Rails

انواع اعمال کدهای جاوا اسکریپت به صفحه وب:

- **inline**: کدهای JS در یکی از Event های هر عنصر دلخواه نوشته می شوند.

```
<input type="button" onclick="alert(' Hello js! ');">
```

- **Embedded**: کدهای JS معمولاً در بالای صفحه وب (در بین تگ <head>) و داخل تگ <script> نوشته می شوند.

³² Application Service Provider

³³ java server page

```

<head>
  <script>
    function hello()
    {
      alert('Hello js!');
    }
  </script>
</head>
<body>
  <a href="javascript:hello();" >Click me!</a>

```

- **External:** کدهای JS در یک فایل با پسوند js ذخیره شده و به صفحه مورد نظر لینک می‌شوند.

```

<script src="js آدرس فایل"></script>

```

تعریف متغیر در جاوا اسکریپت بسیار ساده است var انواع عدد و کاراکتر را می‌پذیرد.

نحوه دسترسی به محتویات یک فیلد در یک فرم در جاوا اسکریپت:

برای این که کاربر تمام فیلدهای مورد نظر را پر کند استفاده می‌شود. می‌توان به وسیله ساختارهای کنترلی یک یا چند شرط گذاشت.

```

if(شرط)
{
  دستورات در صورت برقراری شرط
}
else
{
  دستورات در صورت عدم برقراری شرط
}

```

مثال، اگر می‌خواهید فیلد خاصی خالی نباشد. برای مثال فیلد نام:

```

<script type="text/javascript">
function checkForm()
{
  if(document.getElementById('fname').value=="")
  {
    alert('نام خود را وارد نمایید');
    document.getElementById('fname').focus();
    return false;
  }

  return true;
}

```

JQuery

جی کوئری یکی از قدرتمندترین فریم‌ورک‌های جاوا اسکریپتی است که شما را قادر به انجام فعالیت‌های گوناگونی می‌سازد. طبیعت توسعه‌پذیر بودن جی کوئری به شما کمک می‌کند که در صورت نبود قابلیت خاص در هسته jQuery خودتان آن‌را به جی کوئری اضافه کنید.

توجه: در مجموع جی کوئری حافظه رم کاربر را بیش از حد مشغول خواهد کرد. پس سعی کنید تا جایی که می‌شود از معادل‌های CSS و HTML5 برای آنچه نیاز دارید استفاده کنید.

یک مثال: درج یک اسلاید شو توسط jQuery:

گام اول) به سایت www.jquery.com رفته نسخه ۱ آن را دانلود کرده و در پوشه JS، ذخیره کرده و به صفحه لینک کنید.

گام دوم) کلمه jquery slideshow را در اینترنت جستجو کرده و یکی از اسلایدشوها که بیشتر دوست دارید را دانلود کنید. (<http://slidesjs.com> یکی از بهترین‌هاست)

گام سوم) فایل‌های جاوا اسکریپتی آن را دانلود کرده و در پوشه JS قرار دهید.

گام چهارم) فایل‌ها را به صفحه لینک کنید. بعد به مستندات اسلایدشو که روی سایت آن قرار دارد مراجعه کرده و کدهای مورد نظر را کپی کرده و در مکان ذکر شده مابین کدهای خود قرار دهید. احتمالاً لازم است کلاس div خود را برابر با کلاس ذکر شده در مستندات قرار دهید.

گام پنجم) عکس‌های خود را به اسلایدشو معرفی کنید.

PHP

تفاوت‌های بین دو زبان ASP و PHP:

:PHP

- PHP یک زبان رایگان و منبع باز (open source) است.
- در زبان PHP به خاطر رایگان بودن، کدهای آماده بسیار فراوان است.
- در PHP، IDE خوب و معتبر نداریم چون یک زبان Run-time است.
- PHP بر روی windows, linux و MacOS قابل اجراست.
- Web Server مربوط به این زبان، Apache است.
- Data Base مربوط به این زبان، MySQL است.
- برای اجرای برنامه‌ها به زبان PHP مجموعاً به سه نرم افزار نیاز دارید: Apache-MySQL-PHP که باید از سایت‌های رسمی آن‌ها دانلود و روی سیستم یا سرور خود نصب کنید.
- به جای نصب کردن چند نرم‌افزار می‌توانید یک پکیج دانلود و نصب کنید فقط باید بستر خود را انتخاب کنید یعنی برای ویندوز می‌خواهید یا لینوکس یا مک؟
- به پکیجی که روی ویندوز نصب می‌شود در اصطلاح ^{۳۴}WAMP گفته می‌شود. به پکیج لینوکس، LAMP و به پکیج مک، MAMP گفته می‌شود.

:ASP

- ASP یک زبان انحصاری است. (در انحصار مایکروسافت)
- ASP کدهای آماده کمتری دارد.
- ASP دارای محیط توسعه بسیار قوی به نام Visual Studio است.
- زبان ASP تنها بر روی Windows قابل اجراست.
- Web Server مربوط به این زبان ^{۳۵}IIS است.
- Data Base مربوط به این زبان SQL Server است.

³⁴ WAMP = Windows Apache MySQL PHP

³⁵ Internet Information Services

کار با PHP:

پکیج WAMP خود را Extract و نصب کنید. به پوشه **www** رفته تمام فایل‌ها به جز پوشه **phpmyadmin** را حذف کنید. سپس تمام پوشه‌های سایت خود را در این پوشه کپی کنید. از این پس باید برای دیدن سایت خود در مرورگر وارد کنید: **http://localhost**.

نکته: از این پس از دابل کلیک کردن روی فایل‌های خود بپرهیزید و فایل‌ها را در **localhost** اجرا کنید.

اولویت‌بندی در **web server** برای فراخوانی فایل‌های **index** به این ترتیب است:

۱. **index.html**

۲. **index.htm**

۳. **index.php**

– **PHP** مخفف **PHP Hypertext Preprocessor** است.

– ساختاری شبیه زبان **C** دارد.

– کدهای **PHP** را در هر قسمتی می‌توان تایپ کرد. یعنی مابین کدهای **html** قرار می‌گیرد و کمبدهای این زبان را رفع می‌کند.

– اگر یک خط هم **PHP** در فایل خود دارید باید پسوند فایل را به **(.php)** تغییر دهید.

ساده ترین برنامه به زبان **PHP**:

```
<?php
echo 'hello PHP!';
?>
```

متغیرها در زبان PHP: در **PHP** نیازی به تعریف کردن نوع عدد یا کاراکتر برای متغیر نیست؛ یعنی می‌توانید هر نوع کاراکتر یا عددی را به متغیر که با علامت دلار در ابتدایش مشخص می‌شود (مانند **\$X**) نسبت دهید.

مثال:

```
$X=5;
$X= Hamid;
```

if و else در زبان PHP: ساختار **if** و **else** در این زبان کاملاً شبیه زبان **C** هستند. به این صورت:

```
<?php
if (شرط)
{
    دستورات در صورت برقراری شرط
}
else
{
    دستورات در صورت عدم برقراری شرط
}
```

```
}  
?>
```

یک مثال ساده:

```
<?php  
$x=4;  
if($x>=5)  
{  
    echo 'عدد بزرگتر یا مساوی پنج است';  
}  
else  
{  
    echo 'عدد کوچکتر از پنج است';  
}  
?>
```

حلقه‌ها در زبان PHP:

حلقه برای عدم تکرار در کدنویسی بهترین گزینه است. برای مثال شما می‌خواهید یک منوی کرکره‌ای برای تاریخ تولد قرار دهید تا کاربر بین ۳۱ روز یکی را انتخاب کند. اگر شما بخواهید بدون زبان PHP این کار را انجام دهید یعنی باید ۳۱ بار کد option را تکرار کنید اما به وسیله یک حلقه، با یک خط کد همان کار را انجام می‌دهید. مثال:

```
<?php  
for($i=1; $i<=31; $i++)  
{  
    echo "<option value=\"\$i\">$i</option>";  
}  
?>
```

در مثال بالا دلیل استفاده از دابل کوتیشن(“) و بک اسلش (\)، این است که هنگامی که به echo سینگل کوتیشن نسبت دهید. به جای تعداد روزهای ماه؛ \$i است که 31 بار تکرار می‌شود. پس بهتر است از دابل کوتیشن استفاده کنید. ولی یک خطا رخ می‌دهد چون دابل کوتیشنِ value، دابل کوتیشن پایانی فرض می‌شود و مابقی کدها نادیده گرفته می‌شود. برای جلوگیری از این کار، قبل از دابل کوتیشنِ value، یک بک اسلش قرار دهید. که به این کار در اصطلاح فرار از خطا یا error scaping گفته می‌شود.

طراحی صفحه پردازشگر فرم تماس با ما با استفاده از زبان PHP:

در این بخش می‌توانید با استفاده از زبان PHP از طریق فرم تماس پیام کاربر را در ایمیل خود دریافت کنید. یعنی به عبارتی پیام کاربر به آدرس ایمیل شما ارسال شود.

به خاطر دارید که در تگ form، صفت action را مساوی handle.php قرار دادید. action یعنی آدرس فایل پردازشگر؛ حال با استفاده از زبان PHP این فایل را ایجاد خواهید کرد.

برای اینکه مقادیر یک فیلد را از کاربر بگیرید از نام (name) آن فیلد استفاده کنید یعنی برای هر فیلد نام را این گونه در نظر بگیرید:

```
$نام فیلد=$_REQUEST['نام فیلد']
```

برای ارسال داده‌ها از طریق ایمیل از تابع mail استفاده کنید. در کل برای دریافت مقادیر نام و نام خانوادگی، ایمیل، موضوع و پیغام از کد زیر استفاده می‌شود:

```
<?php
$fname=$_REQUEST['fname'];
$lname=$_REQUEST['lname'];
$email=$_REQUEST['email'];
$subject=$_REQUEST['subject'];
$message=$_REQUEST['message'];
if(mail('yourmail@gmail.com', $subject, $message, "From: $fname $lname
<$email>"))
{
    echo 'پیغام شما با موفقیت دریافت شد';
}
else
{
    echo 'مشکلی در ارسال ایمیل به وجود آمد. لطفاً دوباره
    !!امتحان کنید';
}
?>
```

فراموش نکنید که name فیلدها در فرم سایت شما باید دقیقاً همان fname و lname و ... که در این کد آمده باشد.

ضمناً به جای yourmail@gmail.com آدرس ایمیل خودتان را بنویسید.

CMS

CMS^{۳۶} به معنی «سیستم مدیریت محتوا» است و در حقیقت سیستم‌های آماده‌ای است که زحمت کدنویسی را از دوش مدیر سایت برداشته و او را مشغول «تولید محتوای سایت» می‌کند. با استفاده از این سیستم‌ها دیگر نیاز به کدنویسی از ابتدا را ندارید.

برای هر موضوعی که مد نظر شما باشد، یک یا چند سیستم مدیریت محتوا وجود دارد:

Word Press Drupal	وبلاگ آموزشی
Joomla Mambo	سایت سازمان
4images	گالری عکس
PHPBB MyBB MySMF VBulletinBoard	انجمن گفتگو
PHP-Nuke	همه چیز با هم

هر کدام از این سیستم‌ها سایت‌های رسمی و سایت‌های پشتیبان به زبان فارسی دارند که با یک جستجوی ساده می‌توانید آن‌ها را پیدا کنید. راهنمای نصب آن‌ها در سایت‌هایشان قابل مشاهده است.

³⁶ System Content Management

مراحل طراحی سایت:

۱. انتخاب و ثبت دامنه

– نکاتی در مورد ثبت دامنه:

- اولاً دامنه باید توسط دیگران رزرو نشده باشد.

فراموش نکنید که هر دامنه فقط توسط یک نفر می‌تواند ثبت شود و تا انتهای تاریخ انقضایش در مالکیت اوست. مثلاً شما نمی‌توانید دامنه aftab.cc را ثبت کنید.

دامنه‌ای که به ذهنتان می‌رسد را در سایت domaintyper.com و یا ud.com چک کنید تا از آزاد (Available) بودن آن مطمئن شوید.

- دامنه باید تا حد ممکن کوتاه باشد:

تا حد ممکن دنبال دامنه‌ای بگردید که بیش از ۸ کاراکتر نباشد. کاربران معمولاً دامنه‌های طولانی را دیرتر به خاطر می‌سپارند یا ارزش کمتری برایش قائلند. ضمناً موتورهای جستجو نیز دامنه‌های کوتاه را در رتبه‌های بالاتری قرار می‌دهند.

ممکن است در نگاه اول به نظر برسد کلمات کوتاه و روند رزرو شده باشد اما من مطمئنم با کمی فکر و بررسی کردن می‌توانید کلمات جذابی که به ذهن هیچ کس نرسیده باشد پیدا کنید.

- دامنه ترجیحاً متشکل از حروف مخفف نباشد:

تحقیقات نشان می‌دهد کاربران کلمات مخفف را خیلی زود فراموش می‌کنند مگر اینکه حروف مخفف در کنار هم یک کلمه نسبتاً با معنی بسازند. مثلاً دامنه avd.ir هر چند کوتاه است اما به مرور از ذهن کاربران پاک خواهد شد اما مخفف‌هایی مثل ناجا و امثالهم بهتر به خاطر می‌مانند.

- اگر به حروف مخفف اصرار دارید، از حروف **c** و حروف دیگر که با فارسی مشکل ایجاد می‌کند استفاده نکنید:

برخی حروف مانند **z** و **c** و **a** و **i** و **o** و **q** هستند که با چیزهای دیگر اشتباه گرفته می‌شوند. مثلاً **c** با عدد ۳۰ و **z** با **g** و **q** با **g** و **o** با صفر و **i** بزرگ با **l** کوچک اشتباه گرفته می‌شوند. بعدها در بیان آدرس به دیگران با مشکلات زیادی مواجه خواهید شد.

- از کلمات انگلیسی با املای سخت و خاص استفاده نکنید:

برخی کلمات انگلیسی هستند که نوشتن آن‌ها واقعاً مشکل است. مثلاً یکی از دانشجوها برای سایتش با عنوان «کودکانه» می‌خواست کلمه **puerile** را انتخاب کند! اولاً این کلمه به معنای «بچگانه» (به معنی احمقانه) بود و ثانیاً من نتوانستم این کلمه را حتی بخوانم چه برسد به اینکه حفظ کنم!

- در نام دامنه فقط استفاده از - مجاز است که آن را هم استفاده نکنید:

سعی کنید هرگز از - (dash یا همان خط تیره) در نام دامنه استفاده نکنید چون فهماندن آن به کاربران فارسی‌زبان واقعاً مشکل است. اگر مثلاً بگویید دامنه سایت من **aftab-host.com** است، باید کلی جواب پس بدهید: دش یعنی چه؟ منظورت همان آندرلاین است؟ یعنی شیف‌ت را بگیرم بزنم یا نه؟...

- دامنه مرتبط با موضوع سایتتان باشد:

خیلی مهم است که کاربران وقتی نام سایت شما را می‌آورند، کاری که در سایتتان انجام می‌دهید به ذهنشان بیاید. برای موتورهای جستجو، این خیلی مهم است که کلمه کلیدی مرتبط با شغل شما در دامنه بیان شده باشد.

- اگر کلمات روند (round) و با معنا، آزاد نبود، از ترکیب دو یا چند کلمه استفاده کنید:

گاهی با ترکیب دو کلمه، کلمات بسیار جالبی ایجاد می‌شود که بهتر از کلمات تنها به خاطر می‌ماند! می‌توانید از ترکیب کلمات فارسی و انگلیسی استفاده کنید. مثلاً ما در **testa.cc** کلمه **test** را با «الف فاعل‌ساز» ترکیب کرده‌ایم تا به معنی «تست گیرنده» شود.

- پسوند مرتبط با موضوع کار خود را انتخاب کنید:

این روزها بیش از ده پسوند برای دامنه‌ها قابل تصور است؛ مانند: **ir, com, net, org, biz, ws, us, tv** و **cc** و ... و پسوندهای جدید مانند **game, .company, .land** و ...

سعی کنید برای سایت خود پسوندی را انتخاب کنید که مرتبط با موضوع سایت باشد؛ مثلاً برای کارهای تجاری، **com** را انتخاب کنید. برای کارهای شبکه، **net** را، اگر سازمان هستید، **org** را و اگر سایتی فقط ویژه کاربران ایرانی هستید، **ir** را انتخاب کنید... اگر پسوند و دامنه مورد نظرتان آزاد نبود، پسوندهای ناشناخته مانند **ws** و **us** و ... را در نظر بگیرید.

- دامنه ترجیحاً حاوی عدد نباشد:

این روزها وجود عدد در دامنه یک نوع بی‌اعتمادی را القا می‌کند. از بس سایت‌هایی بوده‌اند که فیلتر شده‌اند و انتهای دامنه‌شان یک ۲ یا ۳ قرار داده‌اند و دوباره فعالیت غیرقانونی خود را ادامه داده‌اند، کاربران به محض دیدن عدد احساس غیرقانونی بودن فعالیت شما را پیدا می‌کنند.

شما هیچ شرکت یا سایت مشهوری را نخواهید یافت که عدد در دامنه‌اش به کار ببرد.

۲. انتخاب هاست

- انواع هاست:

- ویندوزی

- ASP, PHP, ...

- گران‌تر

- ناامن‌تر

- کنترل پنل برتر: Plesk

- لینوکسی

- PHP

- ارزان‌تر

- امن‌تر

- کنترل پنل برتر: CPanel

۳. طراحی سایت

۴. آپلود