



دانشگاه پیام نور

فراگیری و استفاده کاربردی از زبان های

برنامه نویسی تحت وب

HTML, CSS, JavaScript

کارآموزی دوره کارشناسی

مهندسی کامپیوتر

ارائه شده به:

گروه مهندسی کامپیوتر و فناوری اطلاعات

دانشکده فنی و مهندسی

دانشگاه پیام نور مرکز تهران شمال

استاد راهنما:

جناب دکتر سید علی رضوی ابراهیمی

توسط:

محسن گلی

تیر ۱۴۰۰

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

فراگیری و استفاده کاربردی از زبان‌های

برنامه‌نویسی تحت وب

HTML, CSS, JavaScript

محل کارآموزی: تحت نظر استاد درس

استاد کارآموزی: دکتر سید علی رضوی ابراهیمی

نام و نام خانوادگی: محسن گلی

شماره دانشجویی: ۹۷۰۱۶۱۴۶۷

ترم تابستان سال تحصیل ۱۴۰۰ - ۹۹

آدرس محل کارآموزی: تحت نظر استاد درس

تاریخ شروع: ۱۴۰۰/۰۹/۱۳

تاریخ پایان: ۱۴۰۰/۱۱/۰۷

چکیده

فهم HTML^۱ و CSS^۲ می‌تواند به هر شخصی که با وب سر و کار دارد، کمک کند. طراحان می‌توانند سایت‌های جذاب و قابل استفاده‌تری بسازند، ویرایش‌گران وب‌سایت می‌توانند محتوای بهتری تولید کنند، بازاریابان می‌توانند با مخاطبان‌شان به طور موثرتری ارتباط برقرار کنند و مدیران می‌توانند سایت‌های بهتری را سفارش دهند و از تیم‌هایشان بهترین بهره را ببرند. در این آموزش فرا می‌گیریم، چگونه HTML برای ساخت صفحات وب به کار می‌رود. ابتدا با نوشتن کلماتی که می‌خواهیم در صفحه دیده شوند، شروع می‌کنیم و سپس **تگ‌ها**^۳ و **عناصری**^۴ را به کلمات اضافه می‌کنیم که به وسیله آن مرورگر متوجه می‌شود که عنوان چیست، یک **بند**^۵ یا پاراگراف از کجا آغاز و در کجا پایان می‌یابد، و سایر موارد. سپس به معرفی تگ‌هایی همانند text, link, image, table, form, video و عناصر متفرقه‌ای می‌پردازد که در ساخت صفحات وب کاربرد دارند.

در ادامه می‌آموزیم چگونه CSS با استفاده از دستورات، ما را قادر می‌سازد تا شکل و چیدمان صفحات وب را کنترل کنیم که جالب‌تر به نظر برسند. این ویژگی‌ها برای مثال در کنترل رنگ متن، اندازه و نوع Font^۶، رنگ پس‌زمینه^۷، و نحوه اضافه نمودن عکس به پس زمینه صفحات وب به کار رفته و این که هر یک از عناصر مختلف در کجای صفحه قرار گیرند را کنترل می‌کنند.

بعد از یادگیری HTML و CSS نوبت به جاوااسکریپت^۸ می‌رسد. این زبان برنامه نویسی برای ساخت وب‌سایت‌های **واکنش‌گرا**^۹ و پویا به کار می‌رود. به عبارت دیگر به وسیله جاوااسکریپت شما می‌توانید، هر چیزی را در صفحه وب حرکت یا تغییر دهید یا تصاویر متحرک بسازید و کارهای بسیار دیگری انجام دهید. هر چند این قبیل کارها را با CSS نیز می‌توان انجام داد ولی با جاوااسکریپت، انجام این کارها سهل‌تر و قدرتمندتر امکان‌پذیر می‌باشد. امروزه این زبان یکی از محبوب‌ترین زبان‌های برنامه‌نویسی می‌باشد که به صورت گسترده‌ای استفاده می‌گردد. جاوااسکریپت برای استفاده **سمت کاربر**^{۱۰} ساخته شده است منظور این که موتور جاوااسکریپت در درون **مرورگرها**^{۱۱} وجود دارد که به ما اجازه می‌دهد تا کدهای آن را بدون نیاز به ابزارها یا نرم‌افزارهای اضافی به کار ببریم و به لطف فناوری Node.js این امکان برای به کارگیری جاوااسکریپت در **سمت سرویس‌دهنده**^{۱۲} نیز فراهم گردیده است. این موارد باعث گردیده تا امروزه این زبان بسیار پرطرفدار و قدرتمند باشد.

^۱Hyper Text Markup Language

^۲ Cascading Style Sheets

^۳ Tags

^۴ Elements

^۵ Paragraph

^۶ مجموعه حروف

^۷ Background

^۸ JavaScript

^۹ Responsive

^{۱۰} Client-Side

^{۱۱} Browsers

^{۱۲} Server-Side

فهرست مطالب

عنوان	صفحه
چکیده	یک
فهرست مطالب	دو
فصل اول	۱
مقدمه	۱
مقدمه	۲
۱-۱ معرفی کارآموزی	۳
۲-۱ ساختار گزارش	۳
فصل دوم	۴
معرفی محل کارآموزی	۴
مقدمه	۵
۱-۲ مبانی و نیازمندی‌های آموزش	۵
۲-۲ هدف این دوره آموزشی	۵
۳-۲ جمع‌بندی	۶
فصل سوم	۷
بستره کارآموزی	۷
مقدمه	۸
۱-۳ معرفی موضوع کارآموزی	۹
۲-۳ وظایف دوره کارآموزی	۹
۳-۳ جمع‌بندی	۹
فصل چهارم	۱۰
فعالیت‌های انجام‌شده توسط کارآموز	۱۰
مقدمه	۱۱
۱-۴ ساختار کلی فایل‌ها و ویژگی‌های دیگر	۱۱
۴-۲ نمونه‌سازی	۱۱
۳-۴ ساخت پوشه فایل‌ها	۱۴

۴-۴	مبانی تگ HTML.....	۱۵
۴-۵	ساختار صفحه وب Head و Body.....	۱۷
۴-۶	معرفی عناصر بلوکی و عناصر درون خطی.....	۲۰
۴-۷	ساخت بدنه و محتوای صفحه.....	۲۲
۴-۸	CSS چیست؟.....	۲۷
۴-۹	چگونه مرورگر CSS را از HTML تشخیص می‌دهد؟.....	۲۹
۴-۱۰	انتخاب‌گرهای Tag، Class و ID: سه بلوک اصلی ساختمان CSS.....	۳۰
۴-۱۱	اهمیت RESETS.....	۳۱
۴-۱۲	چگونه از رنگ در CSS استفاده کنیم؟.....	۳۱
۴-۱۳	فونت.....	۳۲
۴-۱۴	با فایل HTML شروع کنیم.....	۳۳
۴-۱۵	طراحی واکنش‌گرا با استفاده از CSS شرطی.....	۳۸
۴-۱۶	پیوندهای رسانه اجتماعی.....	۳۹
۴-۱۷	دوره توسعه وب سمت کاربر و شبیه‌سازی Twitter.....	۴۰
۴-۱۸	جمع‌بندی.....	۴۰
	فصل پنجم.....	۴۱
	جمع‌بندی و پیشنهادها.....	۴۱
	مقدمه.....	۴۲
۵-۱	چگونه به یک توسعه‌دهنده وب سمت‌کاربر تبدیل شویم؟.....	۴۲
۵-۲	از کجا باید شروع کرد؟.....	۴۲
۵-۳	کسب مهارت‌های مرتبط.....	۴۲
	مراجع.....	۴۵
	واژه‌نامه.....	۴۶
	واژه‌نامه فارسی به انگلیسی.....	۴۷
	واژه‌نامه انگلیسی به فارسی.....	۴۸

فهرست اشکال

شکل ۱-۴	نمای کلی یک صفحه وب	۱۲
شکل ۲-۴	نمونه‌سازی رسم شده شکل ۱-۴	۱۳
شکل ۳-۴	استفاده از Wireframing به عنوان نقشه هنگام پیاده‌سازی کد	۱۳
شکل ۴-۴	ساخت فایل index.html	۱۴
شکل ۵-۴	تگ	۱۵
شکل ۶-۴	تگ جفت و تگ تکی	۱۶
شکل ۷-۴	هدر تگ	۱۶
شکل ۸-۴	ساختار کلی فایل html	۱۷
شکل ۹-۴	تگ‌های بیرونی	۱۸
شکل ۱۰-۴	تگ‌های درونی	۱۸
شکل ۱۱-۴	بخش عنوان وب‌سایت	۱۹
شکل ۱۲-۴	اختصاص عنوان در زبانه مرورگر	۱۹
شکل ۱۳-۴	اضافه نمودن عنوان و رعایت دندانه‌گذاری	۲۰
شکل ۱۴-۴	اضافه شدن عنوان	۲۰
شکل ۱۵-۴	تگ‌های تودرتو	۲۱
شکل ۱۶-۴	عناصر بلوکی	۲۱
شکل ۱۷-۴	عناصر خطی	۲۱
شکل ۱۸-۴	استفاده از تگ div حامل تمام محتوای صفحه	۲۲
شکل ۱۹-۴	Wireframe	۲۳
شکل ۲۰-۴	تگ img	۲۳
شکل ۲۱-۴	استفاده از تگ <code><p></p></code>	۲۴
شکل ۲۲-۴	تگ <code></code> و <code></code>	۲۴
شکل ۲۳-۴	ایجاد لیست نامرتب و Bullet Point	۲۵
شکل ۲۴-۴	استفاده از تگ <code><sup></code> جهت بالانویس کردن متن	۲۵
شکل ۲۵-۴	استفاده از تگ <code><sup></code> جهت بالانویس کردن متن	۲۶
شکل ۲۶-۴	مراحل دریافت قطعه کد رسانه‌های اجتماعی از سایت	۲۷
شکل ۲۷-۴	فایل آماده شده HTML	۲۷
شکل ۲۸-۴	فهرست مجموعه قواعد	۲۸
شکل ۲۹-۴	اختصاص نام به مجموعه قواعد CSS	۲۸
شکل ۳۰-۴	Selector	۲۹
شکل ۳۱-۴	استفاده از ویژگی class	۲۹
شکل ۳۲-۴	Selector's Syntax	۳۰
شکل ۳۳-۴	CSS RESET	۳۱
شکل ۳۴-۴	ویژگی فونت	۳۲
شکل ۳۵-۴	استفاده از تگ لینک	۳۳
شکل ۳۶-۴	استفاده از تگ <code><div></code> در تگ <code><body></code>	۳۴
شکل ۳۷-۴	margin با یک مقدار، margin با دو مقدار و margin با سه مقدار	۳۴
شکل ۳۸-۴	استفاده از <code>margin: ۰ auto</code>	۳۵
شکل ۳۹-۴	به وجود آمدن Collapse بعد از استفاده از Float	Error! Bookmark not defined.
شکل ۴۰-۴	ساخت باکس برای عنوان باکس‌های راست و چپ	Error! Bookmark not defined.
شکل ۴۱-۴	استفاده از Negative Margin	۳۷

- شکل ۴-۴۲ استفاده از Negative Margin ۳۷
- شکل ۴-۴۳ پس از استفاده از Negative Margin ۳۷
- شکل ۴-۴۴ ساخت انتخاب‌گر خاص برای ۳۷
- شکل ۴-۴۵ ساخت انتخاب‌گر خاص برای لیست ۳۸
- شکل ۴-۴۶ انتخاب دو id در یک انتخاب‌گر ۳۸
- شکل ۴-۴۷ منو با استفاده از تگ ۳۹
- شکل ۴-۴۸ تغییر وضعیت تگ از عمودی به افقی ۳۹

فهرست جدول‌ها

فهرست علائم اختصاری

API: Application Programming Interface	رابط برنامه‌نویسی نرم‌افزار
CSS: Cascading Style Sheets	صفحات آبخاری
DOM: Document Object Model	مدل شیء‌گرای سند
HTML: Hyper Text Markup Language	زبان نشانه‌گذاری ابرمتن
PHP: Hypertext Preprocessor	پیش‌پردازنده ابرمتن
RESTful: Representational State Transfer	انتقال بازنمودی حالت
SASS: Syntactically Awesome Stylesheet	
SEO: Search Engine Optimization	بهینه‌سازی موتور جستجو
SQL: Structured Query Language	زبان پرس‌وجوی ساخت‌یافته

فصل اول

مقدمه

در این آموزش می‌خواهیم فرا بگیریم که محصول ما در نهایت چه شکلی خواهد داشت. از انتها شروع کرده و به عقب باز می‌گردیم تا مفهوم را به خوبی درک کنیم و با تکرار و تداوم آموزش بتوانیم پروژه‌های بعدی را از صفر شروع کرده و به پایان برسانیم.

این گزارش بر اساس مجموعه ویدئوهایی است که به آموزش HTML، CSS، و JavaScript به عنوان بنیان و شالوده توسعه وب سمت کاربر^۱ جهت طراحی و پیاده‌سازی وبسایت‌های واکنش‌گرا با استفاده از آخرین منابع تدریس برنامه‌نویسی و بهترین شیوه‌ها در این خصوص می‌پردازد تا در سریعترین زمان ممکن مهارت لازم را کسب نموده و بتوان آن را در پروژه‌ها به کار برد.

HTML یا زبان نشانه‌گذاری ابر متن در ساخت صفحات استاتیک وب مورد استفاده قرار می‌گیرد. مبتنی بر متن^۲ بوده و ساختار آن از برجسبها تشکیل شده است. هر یک از تگ‌ها دارای معنی و مفهوم خاصی بوده و بر محتوای صفحه تاثیر می‌گذارند. این زبان برای سازمان‌دهی اطلاعات و تفکیک اجزای یک صفحه به کار می‌رود و برای صفحه آرایی نظیر رنگ‌نوشته‌ها و قالب بندی متون، فناوری دیگری به نام CSS مورد استفاده قرار می‌گیرد.

از مفاهیم مرتبط با آموزش CSS، مفهوم FlexBox^۳ می‌باشد. یک مدل لایه‌بندی صفحات وب که در طراحی سایت، به ما کمک می‌کند تا بدون استفاده از Float^۴ و Positioning^۵ به طراحی صفحات انعطاف‌پذیر و واکنش‌گرا بپردازیم. این مفهوم ایرادات و محدودیت‌های موجود در ویژگی Float را حل کرده و با تغییر شیوه واکنش‌گرایی انعطاف‌پذیر، نیاز ما به استفاده از این خاصیت را برطرف می‌سازد.

یکی دیگر از مباحث مطرح در این آموزش زبان برنامه‌نویسی جاوااسکریپت می‌باشد. زبانی محبوب و قدرتمند، که با یک جستجوی در اینترنت می‌توان دلیل آن را متوجه شد. شاید بد نباشد بدانید که فریم ورک‌های قدرتمندی مثل jQuery، React، React Native، Node.js، و بسیاری دیگر، همه بر پایه جاوااسکریپت می‌باشند. پس با یادگیری این زبان برنامه‌نویسی می‌توان کارهای بسیاری انجام داد. نکته مهم درباره یادگیری زبان جاوااسکریپت، این است که برخلاف زبان‌های HTML و CSS که زبان‌های نشانه‌گذاری محسوب می‌شوند، جاوااسکریپت یک زبان برنامه‌نویسی است که نباید انتظار یادگیری سریع و بدون دردسر آن را داشته باشیم و بایستی خود را برای چالش‌ها و درگیری‌های ذهنی بیشتر آماده کنیم.

به صورت پیش‌فرض مرورگرها در درون خودشان موتور پردازش دستورات زبان جاوااسکریپت را دارند. ویژگی‌های بسیار این زبان نظیر کنترل مدل-شی-گرا-سند (DOM)^۶، قابلیت برنامه‌نویسی سمت سرور با استفاده از کتابخانه‌ها، پشتیبانی از بسترهای مختلف ویندوزی و غیر ویندوزی، معتبرسازی فرم‌ها، ساخت توابع در اسکریپت‌ها، باعث گردیده است تا شکل کاربردی‌تری به خود گرفته و فراتر از یک زبان اسکریپتی باشد و به دلیل مجهز بودن به بهترین فریم ورک‌ها و کتابخانه‌ها به عنوان بی‌نظیرترین زبان برنامه‌نویسی حال دنیا شناخته شود.

^۱ Client-side Web Development

^۲ Text Base

^۳ Flexible Box

^۴ شناور بودن

^۵ موقعیت دهی

^۶ Document-Object-Model

۱-۱ معرفی کارآموزی

در این دوره ما سه هسته اصلی فناوری توسعه وب سمت کاربر HTML، CSS و JavaScript را از صفر می‌آموزیم. فناوری‌هایی که بسیار محبوب هستند و به شکل گسترده‌ای مورد استفاده قرار می‌گیرند. با دروه‌های فشرده این آموزش ابتدا با مفاهیم اصلی HTML آشنا می‌شویم، بعد از آن مباحث پایه‌ای CSS را فرا می‌گیریم و کارمان که با آن به پایان رسید به مفهوم CSS Flexbox می‌پردازیم که به ما اجازه می‌دهد بدون استفاده از Float عناصر صفحه را در جای مناسب قرار دهیم. بحث دیگر CSS Grid می‌باشد که همانند Flexbox برای شکل دادن به صفحه استفاده می‌گردد، با این تفاوت که برخلاف Flexbox یک‌بعدی است، به صورت دوبعدی می‌باشد. سپس می‌توانیم ساخت پروژه را شروع کنیم و در ادامه به یادگیری JavaScript بپردازیم که با دروه کوتاه DOM همراه می‌گردد. به وسیله DOM می‌توان با سند HTML تعامل کرده و توسط کدهای جاوااسکریپت آن را با مهارت دستکاری نمود. افزودن عملکرد به پروژه با جاوااسکریپت و DOM ممکن می‌گردد. در آخر نیز به واکنش‌گرایی صفحاتی که ساخته‌ایم برای هر اندازه صفحه نمایشی خواهیم پرداخت.

۲-۱ ساختار گزارش

در فصل دوم: به معرفی محل کارآموزی و انتخاب کارآموزی می‌پردازیم.

در فصل سوم: به معرفی بستره، مقدمات، پیش نیازها و هم چنین نحوه آماده‌سازی محیط لازم جهت شروع برنامه‌نویسی و طراحی سایت خواهیم پرداخت.

در فصل چهارم: به تفصیل به شرح هر یک از مطالب پرداخته و مراحل آموزش با ذکر جزئیات، بیان گردیده است، سعی گردیده از تکرار مطالب خودداری شود تا بحث طولانی و کسل کننده نگردد.

فصل آخر نیز به جمع‌بندی کلی از آموخته‌ها و نتایجی که می‌توان از این آموزش گرفت، اختصاص دارد.

فصل دوم

معرفی محل کار آموزشی

مقدمه

برای هر دانشجویی، آموزش مستمر و منظم در طول دوره تحصیل و بهره‌مندی از وجود اساتید محترم، می‌تواند بخش مهمی از نیل به هدف نهایی که پرورش خلاقیت و بروز استعدادهای آنان می‌باشد را ممکن سازد. انجام فعالیت کارآموزی توسط دانشجوی به منظور آشنایی و نحوه به کارگیری مطالب علمی به صورت عملی و کاربردی در تکمیل این روند بسیار مهم و موثر می‌باشد.

برای بهره‌مندی از تجربیات اساتید در این دوره با استفاده از مجموعه ویدئوهایی که همانند کلاس‌های درس مجازی و آنلاین انجام این فعالیت را ممکن ساخته است و با رعایت ملزومات و ضوابط با مشاهده این آموزش‌ها و تمرین مستمر مطالب به صورت عملی و درگیر شدن در پروژه مطرح در آموزش، گذراندن دوره کارآموزی و کسب مهارت، مطابق با رشته تحصیلی میسر گردیده است.

در طول ساعات کارآموزی به سرعت مهارت‌های مقدماتی که برای طراحی صفحات وب واکنش‌گرا، Wireframing^۱، HTML5 و CSS3 بدان نیاز دارید را کسب خواهید کرد. اگر قبلاً با هیچ یک از این موارد برخورد نکرده‌اید، ساخت یک صفحه وب صد در صد شخصی که موتور جستجو پسند هم می‌باشد و محتوای آن نیز کاملاً تحت کنترل شما است را به صورت عملی فرا خواهید گرفت. به وسیله این آموزش، طراحی یک صفحه وب را به طور کامل از اول یاد می‌گیرید. هم چنین خواهید دید با استفاده از **تکه‌کدهای**^۲ صفحات وب، نظیر Google Map^۳ به سادگی و به سرعت می‌توان عناصر جذابی را بدون کدزدن^۴ اضافی به طرح‌مان بیفزاییم.

در پایان این کارآموزی پایه‌ای مستحکم از دو بنیان برنامه‌نویسی سمت‌کاربر یعنی HTML5 و CSS3 را فرا خواهید گرفت. هم چنین با یادگیری JavaScript که تکمیل‌کننده این آموزش می‌باشد برای انجام پروژه‌های آتی آماده خواهید گردید.

۲-۱ مبانی و نیازمندی‌های آموزش

- داشتن یک کامپیوتر به همراه دسترسی به اینترنت
- میل به یادگیری درباره HTML5 و CSS3 و چگونگی خواندن و نوشتن کدها به هر دلیل و هدفی
- یک قلم و کاغذ برای برداشتن یادداشت، به طور علمی ثابت شده است که برداشتن یادداشت در به خاطر سپردن اطلاعات به ما کمک می‌کند.

۲-۲ هدف این دوره آموزشی

با گذراندن این دوره آموزشی در نهایت می‌توانیم صفحه وب کاملاً شخصی‌سازی شده‌ای را از صفر و با استفاده از کدهای HTML، CSS و جاوااسکریپت خودمان بسازیم و قادر خواهیم بود به سرعت محتوای آن را تغییر داده و ظاهر آن را شکل دهیم.

^۱ نمونه‌سازی

^۲ Snippets

^۳ نقشه گوگل

^۴ Coding

با ساعت‌ها آموزش به وسیله این خودآموز و بهره بردن از منابع فنی رایگان آنلاین، در نهایت به درک درستی از مفاهیم HTML، CSS و JavaScript خواهید رسید و قادر به خواندن، نوشتن و فهم آنان خواهید بود، حتی شخصی که تا به حال کدنویسی نکرده و به دنبال کسب مهارت و دانش لازم در این خصوص می‌باشد.

۲-۳ جمع‌بندی

در این فصل به معرفی محل کارآموزی و نحوه انجام این دوره با توجه به شرایط حال حاضر پرداخته شده و نیازهای اولیه برای انجام این آموزش نام برده شده است و آنچه که در نهایت از این دوره کارآموزی انتظار داریم، مختصری توضیح داده شد تا در فصل‌های بعدی به طور مفصل به آن‌ها بپردازیم.

در فصل بعد نیز به پیش نیازها و نحوه آماده‌سازی محیط جهت شروع این دوره آموزشی و معرفی موضوعاتی که قرار است، فرا بگیریم خواهیم پرداخت .

فصل سوم

بستره کار آموزشی

برای شروع دوره آموزش HTML و CSS چند پیش نیاز را می بایست فراهم کنیم. برای این منظور به دو ابزار اولیه نیازمندیم:

- ویرایش گر متن^۱
- مرورگر وب^۲

ممکن است این دو به صورت نصب شده بر روی سیستم نباشند اما انتخاب های بسیاری موجود است که به طور رایگان قابل دسترسند. برای نمونه ویرایش گر متن رایگان Sublime Text که کاملاً خودکار، کدها و عناصر HTML و CSS را تشخیص داده و آن ها را به صورت رنگی نمایش می دهد که این قابلیت، کدنویسی را برای چشم بسیار راحت تر می کند. این ویرایش گر خصوصیت های کاربر پسند دیگری را نیز ارائه می کند که توسعه وب را روان و دلچسب می گرداند. لازم به ذکر است که استفاده از Sublime Text اجباری نیست و هر ویرایش گری برای این کار مناسب است اما باید دقت شود که اگر از نرم افزار دیگری می خواهیم استفاده کنیم **ویرایش گر متن غنی شده**^۳ نباشد، مثل WordPad یا Microsoft Word. این ویرایش گر ها با افزودن متون اضافی یا **کاراکترهای مخفی**^۴ به سند باعث بروز خطا در صفحه وب شما می گردند.

به بیان دیگر، یک **متن ساده**^۵ ما را قادر می سازد تا از ابزارهای خاصی که برای این منظور ساخته شده اند به بهترین وجه استفاده کنیم. آن چه که برای کدنویسی نیاز داریم ویرایش گر متن ساده خوانده می شود مثل Sublime که می توان آن را با توجه به سیستم عامل از www.sublimetext.com/3 دانلود و نصب نمود.

همان طور که گفته شد ابزار بعدی که به آن نیاز داریم یک مرورگر است که پیشنهاد ما Firefox می باشد. می توان از هر مرورگری نظیر Google Chrome، Microsoft Edge، Safari نیز استفاده نمود اما Firefox دارای **افزونه های**^۶ بسیار خوبی برای این منظور می باشد که ساخت و توسعه راحت تر و سریعتر صفحات وب را دلیزیرتر می سازد.

در خصوص جاوااسکریپت نیز باید گفت که ابزار خاصی برای اجرای آن نیاز نیست چرا که به صورت پیش فرض در هر مرورگری کد آن قابل اجراست و کافی است تا به همان ترتیب که برای کدنویسی فایل های HTML و CSS گفته شد با استفاده از یک ویرایش گر متن نظیر Visual Studio Code کد جاوااسکریپت را نوشته و در مرورگر نتیجه ی اجرای آن را مشاهده کنیم. در فصل چهار در خصوص نحوه نوشتن و پیاده سازی هر کدام از موارد بالا توضیح داده خواهد شد.

^۱ Text Editor

^۲ Web Browser

^۳ Rich Text Editor

^۴ Invisible Character

^۵ Plain Text

^۶ Extension

۳-۱ معرفی موضوع کارآموزی

با استفاده از چندین مجموعه آموزشی شامل:

- آموزش ساخت وبسایت واکنش گرا با HTML5 و CSS3
- دوره طراحی وب با HTML، CSS و JavaScript برای مبتدیان
- آموزش توسعه کامل وب - کپی تویتر
- JavaScript – The Complete Guide 2020 Beginner Advanced

طراحی صفحات وب را تجربه کرده و بر دانش خود در زمینه طراحی و توسعه وبسایت به صورت عملی می‌افزاییم.

در آموزش اول یاد می‌گیریم که چگونه صفحات وب واکنش‌گرای شخصی‌سازی شده را از پایه و بدون هیچ تجربه‌ای بسازیم. در ابتدا متوجه می‌شویم که یادگیری، چگونگی استفاده از تکنولوژی‌های جدید در زمان بسیار کوتاه می‌تواند ناامیدکننده باشد و چون مسلط شدن در یک مهارت جدید معمولاً ساده نیست به وسیله دوره‌های کوتاه شده در این آموزش نه تنها درک مفاهیم، بسیار واضح شده بلکه به خاطر سپردن آن‌ها نیز بسیار سریع، ممکن شده است.

در این آموزش این موضوع که چرا کدها این گونه کار می‌کنند را فرا می‌گیریم، علاوه بر آن خواهیم توانست این دانش را در هر جای دیگری به کار ببریم. این بدان معنی است که قادر خواهیم بود کارهای جالبی انجام دهیم. این که کسب و کارمان را راه‌اندازی کرده و با هزینه کم به بازار عرضه کنیم یا فقط یک مهارت بازارپسند برای ارتقای رزومه‌مان فرا بگیریم و در پایان این دوره به درک عمیقی درباره چگونگی به کار بردن هر آن چه که فرا گرفته‌ایم در پروژه‌های آتی خواهیم رسید.

در آموزش دوم علاوه بر مرور مطالبی که درباره HTML و CSS فرا گرفته‌ایم به یادگیری JavaScript می‌پردازیم که در نهایت بتوانیم پروژه این دوره را پیاده‌سازی نماییم.

۳-۲ وظایف دوره کارآموزی

برای توسعه دانش و پیشرفت در زمینه برنامه‌نویسی وب و طراحی صفحات تحت وب، انجام این دوره کارآموزی به صورت مستمر و مداوم نقش بسیار موثری ایفا می‌کند. یادگیری هر یک از مراحل این آموزش و پیاده‌سازی و انجام آن به صورت عملی با استفاده از ابزارهای لازم، مهارت برنامه‌نویسی و درک مفاهیم این آموزش را بهتر و کامل‌تر می‌نماید. در کنار این مجموعه‌های آموزشی، مراجعه به سایر منابع برای افزایش بهره‌وری و آشنایی با نحوه استفاده از این مطالب به صورت کاربردی می‌تواند مفید واقع شود.

۳-۳ جمع‌بندی

در این فصل به معرفی ابزارها و پیش‌نیازهای برنامه‌نویسی و طراحی صفحات وب پرداختیم و در ادامه هر یک از موضوعاتی را که در این دوره کارآموزی قرار است فرا بگیریم، ذکر کردیم. در فصل بعد با جزئیات بیشتر این موضوعات را بررسی کرده و به هریک می‌پردازیم.

فصل چهارم

فعالیت‌های انجام‌شده

توسط کارآموز

فعالیت‌هایی این‌جانب در دوران کارآموزی به‌طور عمده استفاده از ویدئوها جهت فراگیری مطالب و به کارگیری عملی این آموزش با تمرین در محیط ویرایش‌گر متن و پیاده‌سازی مثال‌ها بوده است.

در این فصل سعی شده تا بخشی از ساعت‌ها فایل و ویدئوی آموزشی با جزئیات آورده شود هر چند که بیان تمامی کارهای انجام‌شده همراه با جزئیات خارج از حیطه این گزارش است اما تا آنجا که امکان داشته، مباحث اصلی با توضیح و شکل مستند گردیده است.

۴-۱ ساختار کلی فایل‌ها و ویژگی‌های دیگر

قبل از کدنویسی بایستی از ساختار و محل دقیق فایل‌ها مطمئن شویم که به راحتی قابل دسترس و منظم باشند. برای این منظور روش‌های مختلفی جهت نام‌گذاری و آدرس‌دهی فایل‌ها و فولدرهای صفحات وب وجود دارد ولی در این آموزش سعی می‌شود تا از بهترین شیوه‌ها و روش‌ها جهت این کار بهره جسته تا در آینده اگر پروژه‌هایی مثل ایجاد یک وب‌سایت یک صفحه‌ای در یک روز محقق شد بتوان آن را برای بازدیدکنندگان غنی‌تر و پربارتر نمود. هرگونه سرمایه‌گذاری اندکی که برای بهتر بودن کار اکنون انجام شود حتما در آینده از فواید آن بسیار بیشتر بهره‌مند خواهیم شد.

۴-۲ نمونه‌سازی^۱

این که نمونه‌سازی چیست و چرا دیگر هرگز بدون نمونه‌سازی طراحی سایت را انجام نخواهیم داد، بحثی است که در این قسمت به آن می‌پردازیم.

HTML یا زبان نشانه‌گذاری ابر متن در ساخت صفحات استاتیک وب مورد استفاده قرار می‌گیرد. مبتنی بر متن بوده و ساختار آن از برچسب‌ها تشکیل شده است. هر یک از تگ‌ها دارای معنی و مفهوم خاصی بوده و بر محتوی صفحه تاثیر می‌گذارند. این زبان برای سازمان‌دهی اطلاعات و تفکیک اجزای یک صفحه استفاده می‌گردد و برای صفحه آرایی نظیر رنگ‌نوشته‌ها و قالب بندی متون فن آوری دیگری به نام CSS به کار می‌رود.

یکی دیگر از مباحث مطرح شده در این آموزش نمونه‌سازی می‌باشد. شبیه پروژه‌های ساختمانی اولین قدم همیشه داشتن یک طرح می‌باشد. هرگاه می‌خواهیم ساختمان بسازیم، هرگز بدون در دست داشتن یک نقشه^۲ اقدام به ساخت نمی‌کنیم. توسعه وب نیز مشابه همین امر است که به جای نقشه در این جا آن را Wireframing می‌نامیم.

نمونه‌سازی برای هر وب‌سایتی ضروری است و در دو مورد موثر می‌باشد، اول این که بدانیم محصول نهایی ما چگونه خواهد بود و دوم به دیگران در نحوه استفاده از آن وب‌سایت کمک خواهد کرد. هر صفحه شامل بلوک‌های اصلی سازنده صفحه وب است و نمونه‌سازی به وضوح، محل و اندازه آنان را در تعامل با یک دیگر نشان می‌دهد.

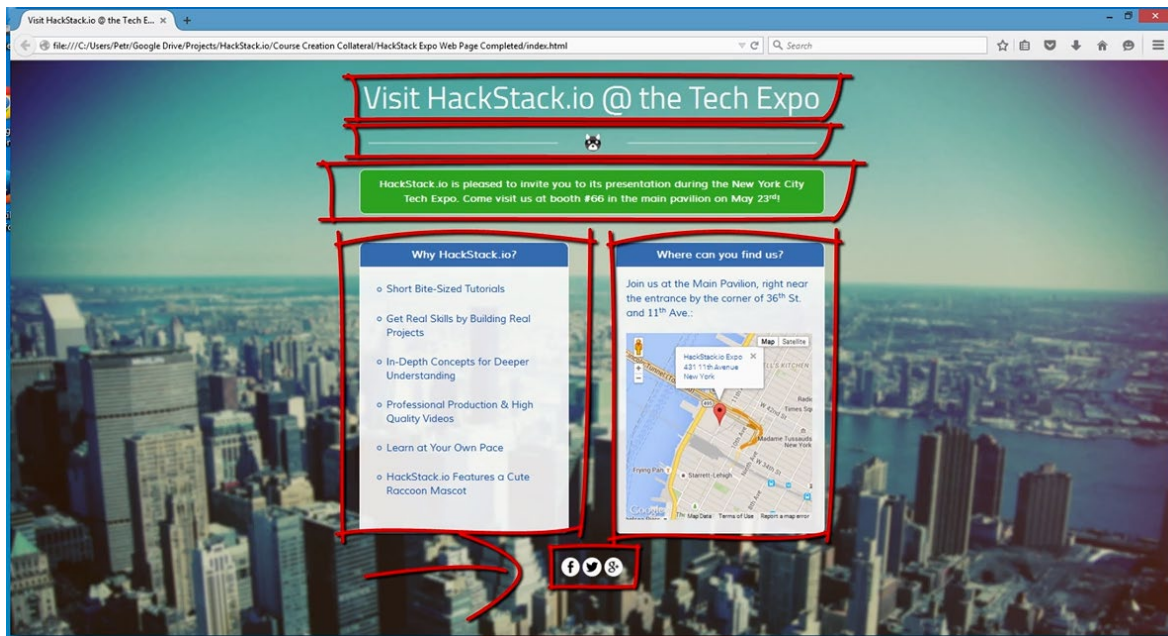
به عنوان بهترین قسمت تنها چیزی که نیاز دارید یک کاغذ و قلم است. نرم افزارهای بسیاری نیز برای این کار وجود دارد از PowerPoint گرفته تا نرم افزار balsamiq، اما باز هم یادآور می‌شود که بهترین ابزار برای این مورد همان قلم و کاغذ است

^۱ Wireframing

^۲ Blueprint

و در صورت امکان، داشتن وایت برد کفایت می‌کند. هم چنین می‌توانید نرم افزار نمونه‌سازی را از لینک <http://pencil.evolus.vn/Default.html> دانلود بنمایید.

اگر به یک صفحه وب نگاه کنیم (شکل ۴-۱)، متوجه تعدادی بلاک اصلی می‌شویم که آن را تشکیل داده اند، برای مثال بخش عنوان، تصویری که به عنوان جداکننده به کار رفته، کادر توضیحات، بخش‌هایی در سمت راست و چپ که محتوایی را در برگرفته و در انتها نماد^۱ صفحات اجتماعی قرار دارد.



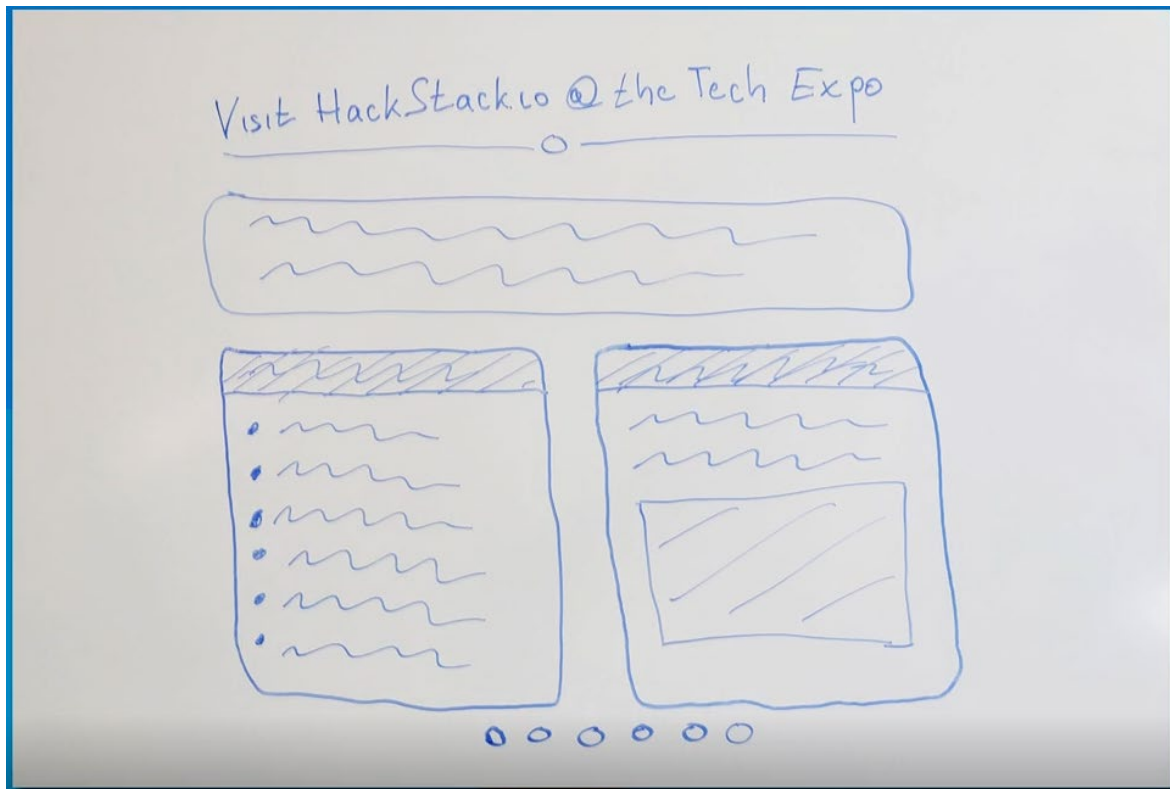
شکل ۴-۱ نمای کلی یک صفحه وب (HackStack.io, 2015)

حال می‌توان بخش‌های بالا را بر روی یک وایت برد کشید (شکل ۴-۲)، و از این نمونه‌سازی شروع کرده و در طول کار تا مرحله تکمیل به آن مراجعه کنیم. با استفاده از این طرح به پیاده‌سازی بخش‌های مختلف صفحه پرداخته و محتوای لازم را به آن می‌افزاییم و در نهایت به وسیله CSS سایر خصوصیات، نظیر اندازه و نوع فونت، موقعیت و نحوه قرارگیری آن‌ها و اضافه کردن رنگ به عناصر صفحه شکل می‌بخشیم. می‌توان CSS را شبیه تم^۲ برای یک صفحه وب تصور کرد برای مثال وقتی که یک گوشی تلفن همراه^۳ خریداری می‌کنید، رنگ صفحه، ترتیب آیکن‌های صفحه اصلی و تصویر پس زمینه را می‌توان تغییر داد یا نحوه کار کردن و نوع ذخیره داده را تنظیم نمود.

^۱ Icon

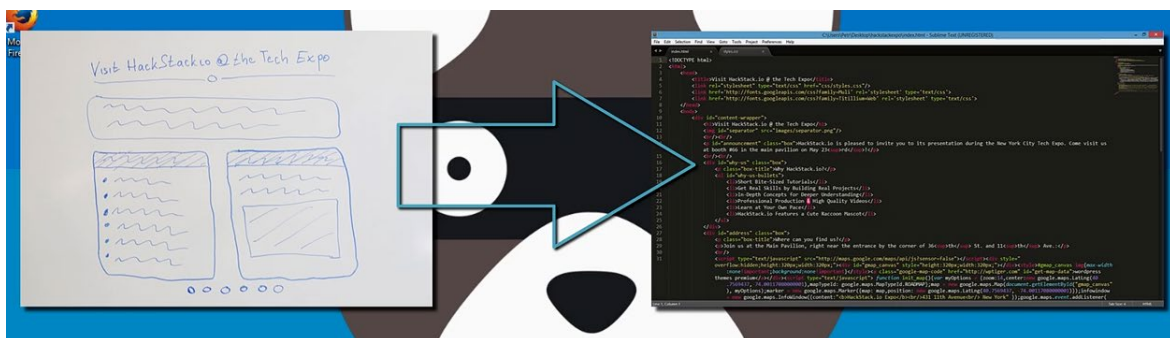
^۲ Theme

^۳ Cell Phone



شکل ۴-۲ نمونه‌سازی رسم شده شکل ۴-۱ (HackStack.io, 2015)

آنچه که اکنون مهم است به کارگیری این شیوه در زمان طراحی یک صفحه وب جدید می‌باشد. شما می‌بایست در ابتدا بلوک‌های اصلی را شناسایی کرده و سپس با نمونه‌سازی و استفاده از آن به عنوان نقشه برای ساخت کد آن اقدام کنید (شکل ۴-۳). کار به این شیوه نه تنها روشی مطمئن برای شروع طراحی صفحه و توجه به نکات فنی پیاده‌سازی کد آن می‌باشد بلکه باعث می‌گردد شما از ساختار پروژه، تصویر مشخصی داشته باشید و جزئیاتی که در نسخه نهایی می‌خواهید به آن برسید را مشاهده کنید.



شکل ۴-۳ استفاده از Wireframing به عنوان نقشه هنگام پیاده‌سازی کد (HackStack.io, 2015)

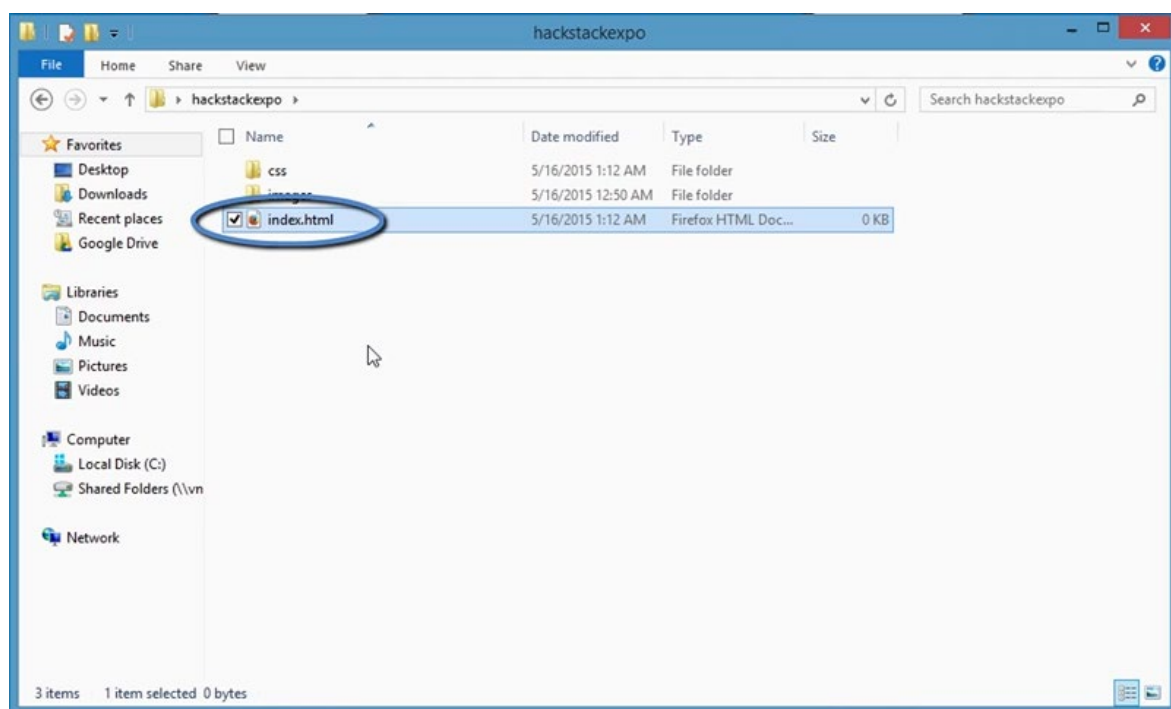
نمونه‌سازی به شما این امکان را می‌دهد تا محصول را قبل از آن که به طور واقعی ساخته شود، مشاهده کرده و همیشه پیش از اولین خط کد، نتیجه تغییرات در محصول نهایی را بدانید.

۴-۳ ساخت پوشه^۱ فایل‌ها

با ساخت یک پوشه روی دسکتاپ^۲ شروع کرده و آن را نام‌گذاری می‌کنیم، البته هر جای دیگری می‌توانیم این فولدر را ایجاد کنیم ولی برای سهولت در دسترسی، آن را در دسکتاپ می‌سازیم. نامی که انتخاب می‌کنیم می‌بایست نشان دهنده نام و دلیل صفحه‌ای که ایجاد می‌کنیم باشد. سپس درون این فولدر دو فولدر دیگر به نام‌های css و images می‌سازیم.

دقت داشته باشید که بسیاری از خدمات میزبانی وب سایت‌ها^۳، حساس به متن^۴ می‌باشند. اگر شما فولدری با نام /images بسازید و سپس مسیر فایل‌تان را به فولدری به نام IMAGES / آدرس‌دهی کنید، بازدیدکنندگان با خطا مواجه شده یا سایت را به صورت ناقص مشاهده خواهند کرد. پس به عنوان توصیه، همیشه فولدرها و فایل‌ها را با حروف کوچک نام‌گذاری کنید. فولدر CSS، برای نگهداری فایل‌های css و فولدر /images مکان مناسبی برای نگهداری عکس‌های صفحه وب می‌باشد.

مورد دیگری که باید به آن توجه نمود، دیدن **فرمت فایل**^۵ نظیر html یا css. می‌باشد که پس از فعال کردن این گزینه در ویندوز، یک فایل متنی خالی در فولدر CSS ساخته و نام آن را به style.css تغییر می‌دهیم تا کدهای css مان را در آن ذخیره کنیم. حال بایستی در فولدر اصلی یک فایل به نام index.html نیز بسازیم (شکل ۴-۴) که در واقع کدهای html صفحه وب ما در آن قرار خواهد گرفت.



^۱ Folder

^۲ Desktop

^۳ Web Site Hosting Services

^۴ Case Sensitive

^۵ File Extension

این فایل در واقع موارد زیر را شامل می‌شود:

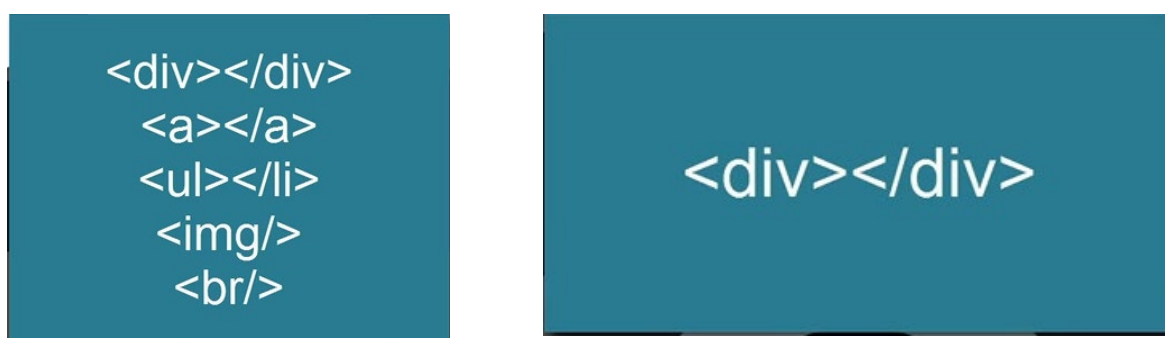
- محتوای کلی صفحه وب را تعیین می‌کند.
- بلوک‌های سازنده صفحه وب را که قبلاً در Wireframe تعیین کرده ایم، ارائه می‌دهد.
- بلوک‌ها را با محتوای ما پر می‌کند.
- فایل css را با فولدر عکس‌ها در یک مکان پیوند می‌دهد.

چرا این فایل index.html نامیده می‌شود؟ چون این فایلی است که به صورت پیش فرض تمام وب سرورها هنگام بازدید دامین^۱ به دنبال آن هستند. هر زمان که شما به‌سایتی مثل www.anythingyouwant.com می‌روید، سرور مقدار وارد شده را به صورت www.anythingyouwant.com/index.html تفسیر می‌کند.

۴-۴ مبانی تگ HTML

صفحه وب ما از کدهای HTML و CSS ساخته شده است. در ابتدا روی کدهای HTML تمرکز کرده و سپس در بخش‌های بعدی به کدهای CSS می‌پردازیم. این کار باعث می‌شود تا مشخص گردد کدام بخش از صفحه نهایی وب ما توسط HTML کنترل می‌شود و کدام بخش توسط CSS.

همان طور که قبلاً ذکر شد، HTML را زبان نشانه‌گذاری ابر متن خوانده و توسط عناصری به نام برچسب یا همان تگ‌ها ساخته می‌شود که در بین دو علامت کوچکتر و بزرگتر محصور شده‌اند (شکل ۴-۵).



شکل ۴-۵ تگ (HackStack.io, 2015)

تگ‌ها به مرورگر وب ما می‌گویند که چه چیز و کجا آن را نمایش دهد، حال می‌خواهد متن باشد یا تصویر. هم چنین می‌توانند برای دسترسی به منابع خارجی جهت ساخت صفحه وب آن طور که می‌خواهیم به کار روند. چیزی شبیه Webfont از google.com/fonts که در جایی از وب به صورت آنلاین وجود دارد و لزوماً به صورت پیش فرض بر روی سیستم شما نصب نیستند، مثالی از این نوع می‌باشد.

به طور خلاصه، تگ‌ها همانند بلوک‌های ساختمانی می‌باشند که از آن‌ها برای ساخت هر صفحه وبی استفاده می‌گردد. تگ‌ها دو نوع هستند، **تگ‌های جفت**^۲ مثل `<div></div>` که متن یا تگ‌های دیگر و یا به صورت ترکیبی از هر دو بین تگ آغاز

^۱ Domain

^۲ Paired Tags

و تگ پایان آن قرار می‌گیرد و تگ‌های تکی^۱ مثل `` که نوع مشخصی از دستورات را به مرورگر اعمال کرده و نحوه نمایش آن را مشخص می‌کند (شکل ۴-۶).

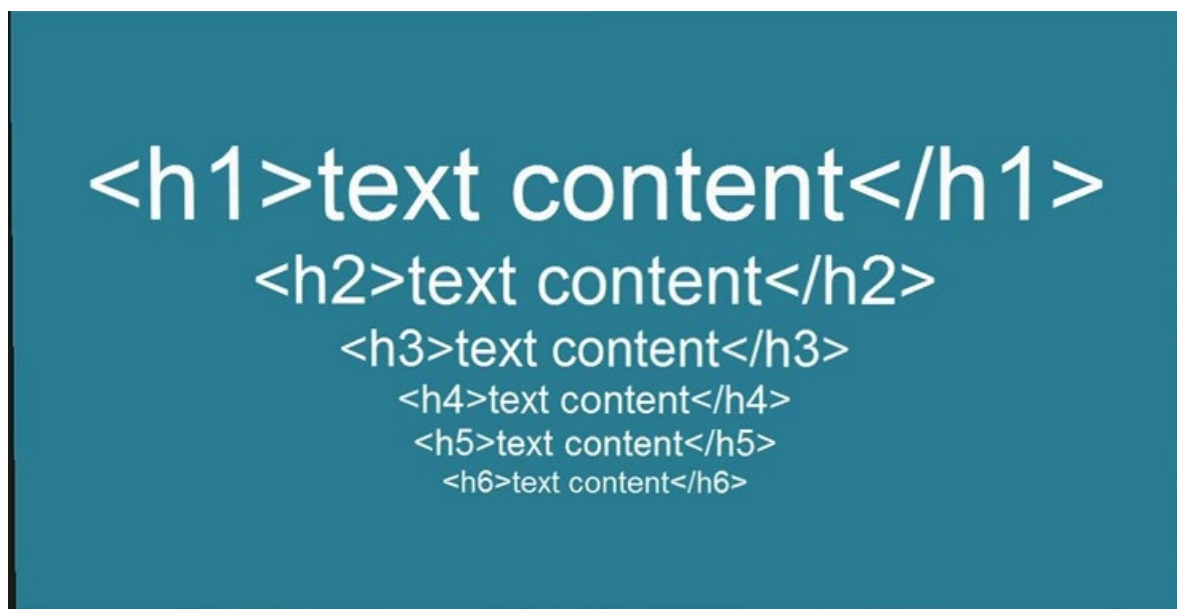


شکل ۴-۶ تگ جفت و تگ تکی (HackStack.io, 2015)

برای مثال تگ `
` به مرورگر دستور می‌دهد متن را به سطر بعدی ببرد همانند مایکروسافت ورد یا سایر پردازش‌ها که با فشردن کلید Enter این کار انجام می‌گردد.

ساختار کلی تگ‌ها جفت به صورت `<Start Tag>Content goes here</End Tag>` می‌باشد برای مثال:

`<h1>Hello World</h1>` تگ `h1` به عنوان **هدر تگ**^۲ یا برچسب سرآیند شناخته شده و مشخصاً به نام هدر یک نامیده می‌شود. چندین نوع متفاوت هدر تگ داریم که محتوای بین تگ آغاز و پایان را به صورت بزرگتر از سایر قسمت‌های متن اصلی صفحه وب نمایش می‌دهند (شکل ۴-۷).



شکل ۴-۷ هدر تگ (HackStack.io, 2015)

^۱ Unpaired Tags

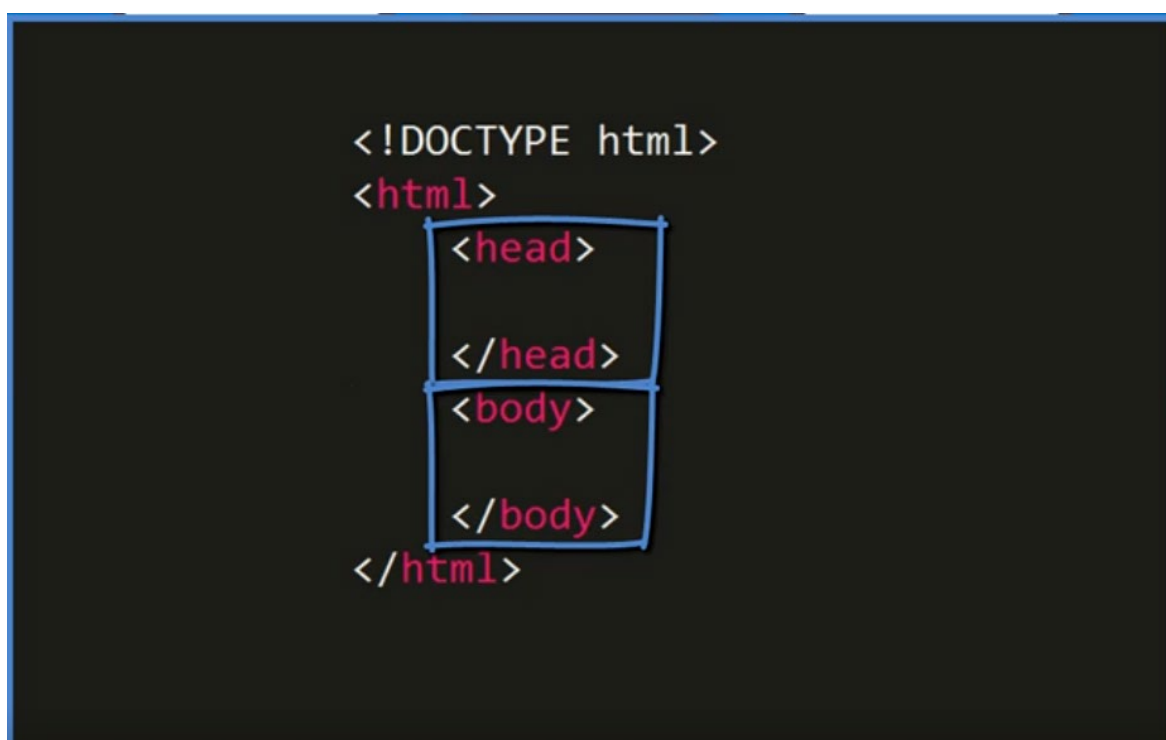
^۲ Header Tag

تمامی کدهای HTML به همین شکل کار می‌کنند و با توجه به نوع تگی که به کار می‌بریم، بسته به این که چه کاری باید انجام دهند و نحوه تنظیم آن‌ها به چه شکل باشد، مورد استفاده قرار می‌گیرند.

۴-۵ ساختار صفحه وب Head^۱ و Body^۲

تمامی فایل‌های HTML به دو بخش اصلی، سرآیند یا هد و بدنه تقسیم می‌گردند (شکل ۴-۸). بخش هد محتوی عناصر زیر است:

<script>, <link>, <meta>, <style>, <title> و <base> هم چنین محلی برای metadata است که شامل داده‌هایی درباره سند HTML می‌باشد. در این بخش تمامی جزئیات موثر در سئو^۳ سایت قرار می‌گیرد. به بیان دیگر، هر چه که در کد ما به google و دیگر موتورهای جستجو برای پیدا کردن آسان صفحه و نشان دادن آن به کاربر کمک کند، سئو گفته می‌شود. در بخش بدنه HTML محتوایی را قرار می‌دهیم که هنگام بازدید از صفحه وب قابل رویت است، از محتوا گرفته تا طرح اولیه صفحه وب ما به طور مشخص در این بخش قرار دارد.



شکل ۴-۸ ساختار کلی فایل html (HackStack.io, 2015)

^۱ سرآیند

^۲ بدنه

^۳ SEO (Search Engine Optimization)

حال از خارج به داخل شروع می‌کنیم، منظور ما از این جمله این است که ابتدا تگ جفت بیرونی^۱ را می‌نویسیم (شکل ۴-۹) و سپس به نوشتن دیگر تگ‌های درونی^۲ آن ادامه می‌دهیم (شکل ۴-۱۰).

```
<!DOCTYPE html>
<html>
<head>
<title>Visit HackStack.io @ the Tech Expo</title>
<link rel="stylesheet" type="text/css" href="css/styles.css"/>
<link href="http://fonts.googleapis.com/css?family=Muli:400,300italic" rel="stylesheet" type="text/css">
<link href="http://fonts.googleapis.com/css?family=Titillium+Web" rel="stylesheet" type="text/css">
</head>
<body>
<div id="content-wrapper">
<div>
<h1>Visit HackStack.io @ the Tech Expo</h1>

<br/><br/>
<p id="announcement" class="box">HackStack.io is pleased to invite you to its presentation during the New York City Tech Expo. Come visit us at booth #66 in the main pavilion on May 23<sup>rd</sup></p>
</div>
<div id="why-us" class="box">
<p class="box-title">Why HackStack.io?</p>
<ul id="why-us-bullets">
<li>Short Bite-Sized Tutorials</li>
<li>Get Real Skills by Building Real Projects</li>
<li>In-Depth Concepts for Deeper Understanding</li>
<li>Professional Production & High Quality Videos</li>
<li>Learn At Your Own Pace</li>
<li>HackStack.io Features a Cute Raccoon Mascot</li>
</ul>
</div>
<div id="address" class="box">
<p class="box-title">Where can you find us?</p>
<p>Join us in the Main Pavilion, right near the entrance by the corner of 36<sup>th</sup> St. and 11<sup>th</sup> Ave.</p>
<br/>
<script type="text/javascript" src="http://maps.google.com/maps/api/js?sensor=false"></script><div style="overflow:hidden;height:320px;width:320px;"><div id="gmap_canvas" style="height:320px;width:320px;"></div><style>#gmap_canvas img(max-width:none!important;background:none!important)</style><a class="google-map-code" href="http://premium-wordpress-themes.org" id="get-map-data">http://premium-wordpress-themes.org/</a></div><script type="text/javascript"> function init_map(){var myOptions = {zoom:15,center:new google.
```

شکل ۴-۹ تگ‌های بیرونی (HackStack.io, 2015)

```
<!DOCTYPE html>
<html>
<head>
<title>Visit HackStack.io @ the Tech Expo</title>
<link rel="stylesheet" type="text/css" href="css/styles.css"/>
<link href="http://fonts.googleapis.com/css?family=Muli:400,300italic" rel="stylesheet" type="text/css">
<link href="http://fonts.googleapis.com/css?family=Titillium+Web" rel="stylesheet" type="text/css">
</head>
<body>
<div id="content-wrapper">
<div>
<h1>Visit HackStack.io @ the Tech Expo</h1>

<br/><br/>
<p id="announcement" class="box">HackStack.io is pleased to invite you to its presentation during the New York City Tech Expo. Come visit us at booth #66 in the main pavilion on May 23<sup>rd</sup></p>
</div>
<br/><br/>
<div id="why-us" class="box">
<p class="box-title">Why HackStack.io?</p>
<ul id="why-us-bullets">
<li>Short Bite-Sized Tutorials</li>
<li>Get Real Skills by Building Real Projects</li>
<li>In-Depth Concepts for Deeper Understanding</li>
<li>Professional Production & High Quality Videos</li>
<li>Learn At Your Own Pace</li>
<li>HackStack.io Features a Cute Raccoon Mascot</li>
</ul>
</div>
<div id="address" class="box">
<p class="box-title">Where can you find us?</p>
<p>Join us in the Main Pavilion, right near the entrance by the corner of 36<sup>th</sup> St. and 11<sup>th</sup> Ave.</p>
<br/>
<script type="text/javascript" src="http://maps.google.com/maps/api/js?sensor=false"></script><div style="overflow:hidden;height:320px;width:320px;"><div id="gmap_canvas" style="height:320px;width:320px;"></div><style>#gmap_canvas img(max-width:none!important;background:none!important)</style><a class="google-map-code" href="http://premium-wordpress-themes.org" id="get-map-data">http://premium-wordpress-themes.org/</a></div><script type="text/javascript"> function init_map(){var myOptions = {zoom:15,center:new google.
```

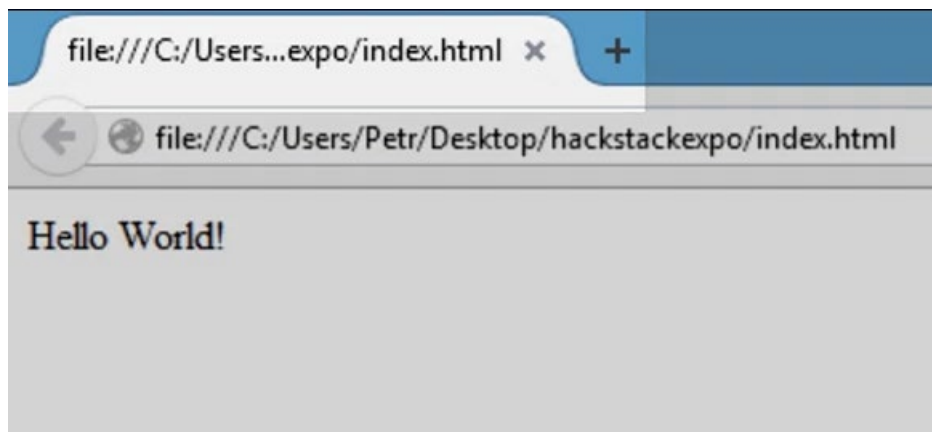
شکل ۴-۱۰ تگ‌های درونی (HackStack.io, 2015)

^۱ Outer

^۲ Inner

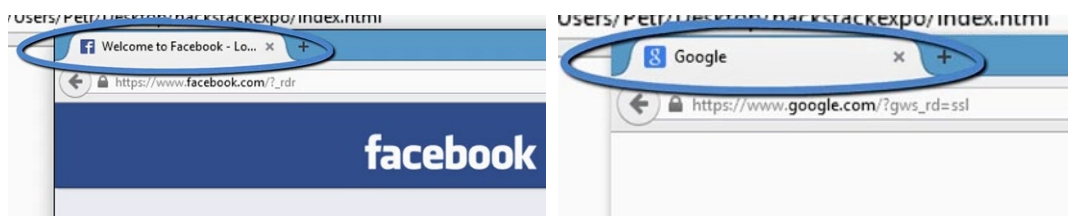
می‌دانیم که هر فایل HTML با تگ `<!DOCTYPE html>` شروع می‌گردد که شبیه نشانه پستی برای مرورگر وب عمل می‌کند تا متوجه گردد که این یک صفحه وب می‌باشد و می‌بایست محتوای آن را نشان دهد. در زیر این تگ اولین تگ جفت `<html></html>` را داریم و درون این دو تگ آغاز و پایان، تگ‌های `<head></head>` و تگ `<body></body>` را می‌نویسیم. هر چه که در درون تگ `<body>` نوشته شود، در صفحه وب ما نشان داده خواهد شد.

هنوز در بخش head چیزی اضافه نکرده ایم. اگر دقت کنیم هیچ **عنوانی**^۱ در Tab^۲ بالای صفحه وجود ندارد (شکل ۴-۱۱).



شکل ۴-۱۱ بخش عنوان وبسایت (HackStack.io, 2015)

اگر به وبسایت‌هایی نظیر فیسبوک، گوگل یا اپل مراجعه کنیم، همه دارای عنوان در بخش **زبانه مرورگر**^۳ می‌باشند. Title برای SEO بسیار مهم می‌باشد هم چنین برای کاربران که بدانند کدام زبانه باز مربوط به کدام صفحه وب می‌باشد (شکل ۴-۱۲).



شکل ۴-۱۲ اختصاص عنوان در زبانه مرورگر (HackStack.io, 2015)

حال می‌خواهیم برای صفحه وب، عنوان قرار دهیم، کافی است در ویرایش‌گر متن در قسمت head کافی است تگ جفت `<title></title>` را بیفزاییم. در این جا لازم است به این نکته اشاره کنیم که هنگام کدنویسی، **فضای خالی**^۴ بین خطوط و

^۱ Title

^۲ زبانه

^۳ Browser Title

^۴ Space

دندانه‌گذاری‌ها^۱ هنگام ترجمه و تفسیر^۲ توسط ویرایش‌گر متن نادیده گرفته شده و در مرورگر دیده نمی‌شوند. اما برای ما و هر کسی که کدهای ما را می‌خواند رعایت این کارها باعث می‌گردد تا کدها خوانا باشند و تگ‌ها را جدا کرده تا بعداً پیدا کردن و کار بر روی آن‌ها راحت باشد (شکل ۴-۱۳).

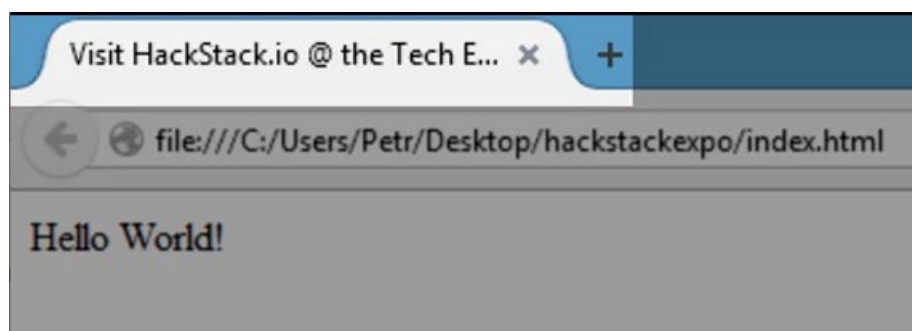
```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Visit HackStack.io @ the Tech Expo</title>
5   </head>
6   <body>
7     Hello World!
8   </body>
9 </html>

```

شکل ۴-۱۳ اضافه نمودن عنوان و رعایت دندانه‌گذاری (HackStack.io, 2015)

بعد از ذخیره کردن^۳، در مرورگر خواهیم دید که عنوان به صفحه ما اضافه گردیده است (شکل ۴-۱۴) که به بازدیدکنندگان صفحه و موتورهای جستجو می‌گوید که صفحه وب درباره چیست.



شکل ۴-۱۴ اضافه شدن عنوان (HackStack.io, 2015)

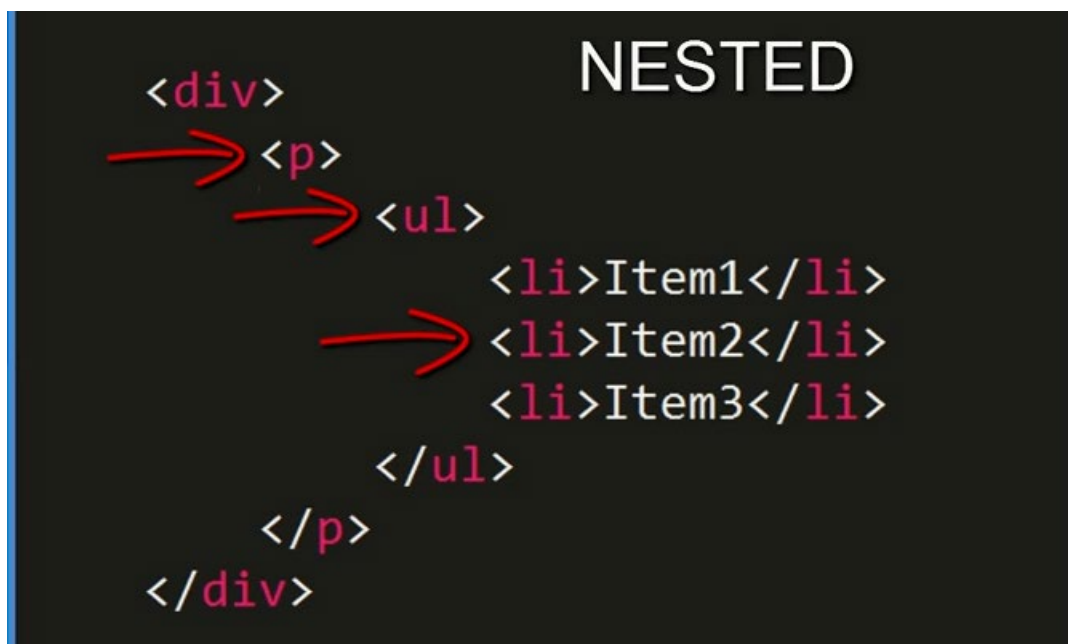
^۱ Indentation

^۲ Rendering

^۳ Save

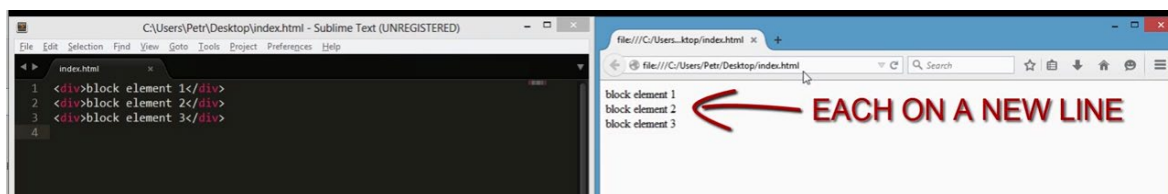
۴-۶ معرفی عناصر بلوکی^۱ و عناصر درون خطی^۲

عناصر و تگ‌ها می‌توانند شامل عناصر و تگ‌های دیگری باشند که تودرتو^۳ نامیده می‌شود (شکل ۴-۱۵). قبلاً دیدیم که تگ‌های head و body بین دو تگ html قرار دارند که در این جا باعث می‌شود تگ html^۴ این دو باشد که تگ head و body به صورت تودرتو بین آن آمده اند به عبارت دیگر تگ‌های head و body فرزندهای^۵ تگ html می‌باشند. این نوع رابطه تودرتوی، والد و فرزندی برای تمام عناصری که ساخته خواهند شد، برقرار می‌باشد.



شکل ۴-۱۵ تگ‌های تودرتو (HackStack.io, 2015)

در یک فایل html دو نوع عنصر به کار می‌رود، عناصر بلوکی و عناصر خطی. عناصر بلوکی همیشه عناصری هستند که به صورت عمودی روی هم قرار می‌گیرند به عبارت دیگر هر خط به صورت خودکار به خط بعد رفته و از یک خط جدید آغاز می‌شود (شکل ۴-۱۶) مثل این که بعد از وارد کردن کلمات کلید Enter را بزنیم. به همین دلیل عناصر بلوکی خیلی کاربردی هستند زمانی که می‌خواهیم محتواها به صورت عمودی و زیر هم مرتب شوند.



شکل ۴-۱۶ عناصر بلوکی (HackStack.io, 2015)

^۱ Block Elements

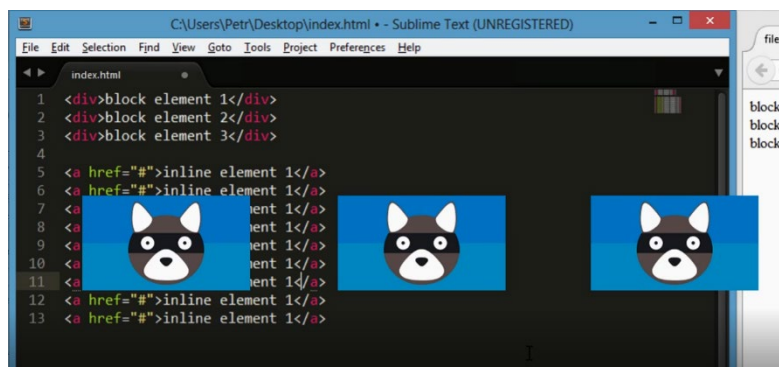
^۲ Inline Elements

^۳ Nested

^۴ Parent

^۵ Child

در مقابل عناصر خطی، عناصری هستند که گزینه رفتن به خط بعد را نداشته و در یک خط پشت سرهم مرتب می‌شوند تا زمانی که به صورت دستی به خط بعد برویم (شکل ۴-۱۷).



شکل ۴-۱۷ عناصر خطی (HackStack.io, 2015)

به همین دلیل زمانی که بخواهیم عناصر ما به صورت پشت سرهم در یک خط قرار گیرند، بسیار سودمند خواهد بود. البته امکان تبدیل عناصر بلوکی به عناصر خطی و بالعکس وجود دارد. یک مثال از این نمونه تگ `<a>` می‌باشد که باعث می‌گردد قابلیت کلیک بر روی لینک قرار گرفته در صفحه وب، فعال شود.

۴-۷ ساخت بدنه^۱ و محتوای^۲ صفحه

حال که با مفاهیم عناصر خطی و بلوکی آشنا شدیم، نوبت ساخت محتوای صفحه می‌باشد. در قسمت قبل با ساخت بخش‌های `head` و `body` آشنا شدیم حال می‌خواهیم عناصر اصلی صفحه را بسازیم و سپس محتوای لازم را درون آن‌ها قرار دهیم. این روش باعث می‌گردد تا بهتر متوجه تاثیر متقابل عناصر ساخته شده بر یکدیگر شویم و در این آموزش، **صفحه اول سایت**^۳ عالی و فوق العاده ای داشته باشیم.

تمرین بسیار خوبی است که با تگ `<div></div>` شروع کنیم که شامل هر چیزی است که در درون تگ `body` می‌باشد. می‌خواهیم مطمئن شویم که تمام محتوای ما درون یک حامل^۴ یا بسته بندی محتوا^۵ قرار دارد تا بعدا امکان جابجا کردن کلی آن فراهم گردد و برنامه‌نویسی عناصر داخل آن چه حال و چه در آینده به سادگی ممکن شود (شکل ۴-۱۸). این روش در زمان استفاده از CSS به خوبی مزیت خود را نشان خواهد داد.

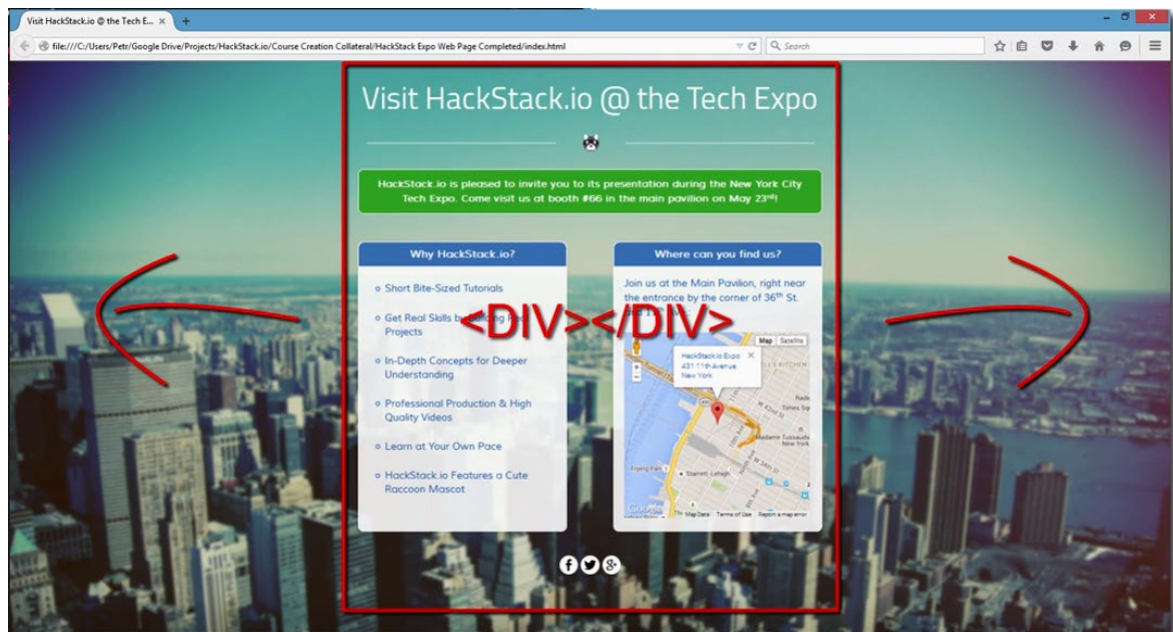
^۱ Body

^۲ Content

^۳ Landing Page

^۴ Container

^۵ Content Wrapper



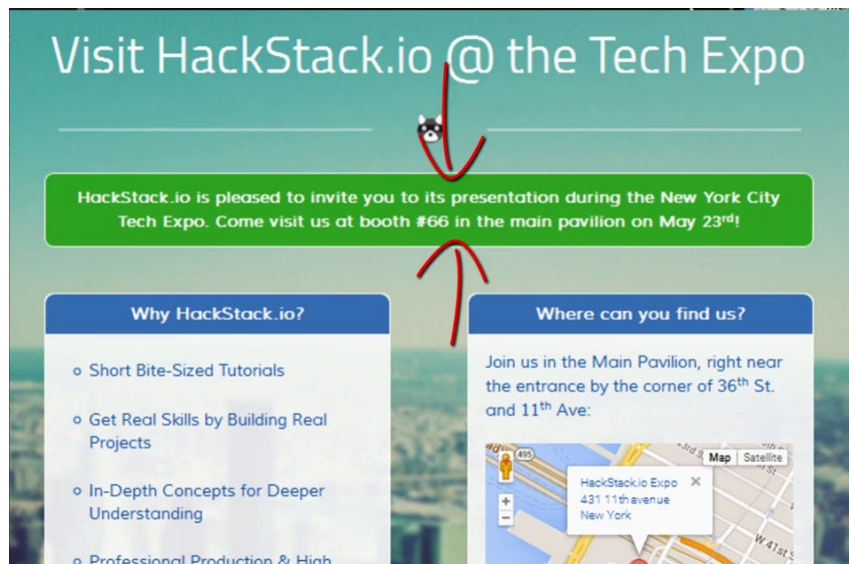
شکل ۴-۱۸ / استفاده از تگ div حامل تمام محتوای صفحه (HackStack.io, 2015)

تمامی عناصر دیگر صفحه وب ما بعد از این می‌بایست درون این تگ div خاص به عنوان Wrapper قرار گیرند که حاشیه بیرونی آنچه را که کد می‌زنیم، شکل می‌دهد.

از بالای Wireframe شروع کرده و به پایین می‌آییم. در بالا، اولین عنصری که دیده می‌شود عنوان صفحه است که می‌خواهیم به صورت بصری آن را در صفحه وب قرار دهیم. این کار را با یک تگ جفت `<h1></h1>` انجام می‌دهیم و متن مورد نظر را درون آن می‌نویسیم.

در پایین title یک تصویر جداکننده داریم (شکل ۴-۱۹) که به وسیله تگ `` آن را اضافه می‌کنیم. تگ `img` یک Unpaired Tag می‌باشد. در HTML یک ویژگی بسیار مهم داریم که Attribute^۱ نامیده می‌شود و هنگام استفاده از CSS به ما کمک می‌کند تا فایل `html` ما با فایل `css` مرتبط گردد. تعیین مقدار برای آن سبب می‌گردد که تگ مورد نظر چگونه کار کند، به نظر برسد یا عمل کند. همان طور که در شکل می‌بینید (شکل ۴-۲۰) تگ `img` دارای یک Attribute به نام `src` دارد که مخفف Source است و آدرس تصویر می‌باشد.

^۱ خصوصیت



شکل ۴-۲۱ استفاده از تگ `<p>` (HackStack.io, 2015)

بخش مهم بعدی صفحه وب ما، اضافه نمودن دو چارچوب^۱ در چپ و راست صفحه می‌باشد. برای این منظور یک تگ `div` برای باکس سمت چپ و یک تگ `div` جهت باکس سمت راست به کدمان اضافه می‌کنیم. این کار ساختار اولیه را ایجاد خواهد کرد. حال محتویات باکس سمت چپ را درون آن قرار می‌دهیم. یک تگ `<p>`، جهت عنوان درون باکس و سپس همان طور که در Wireframe یک فهرست نقطه‌دار^۲ داریم، در ویرایش‌گر متن با استفاده از تگ `` که مخفف Unordered List^۳ می‌باشد و تگ ``، مخفف List Item^۴ و تودرتو درون تگ `` (شکل ۴-۲۲)، فهرست را ایجاد می‌کنیم (شکل ۴-۲۳).

```
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
  ...
</ul>
```

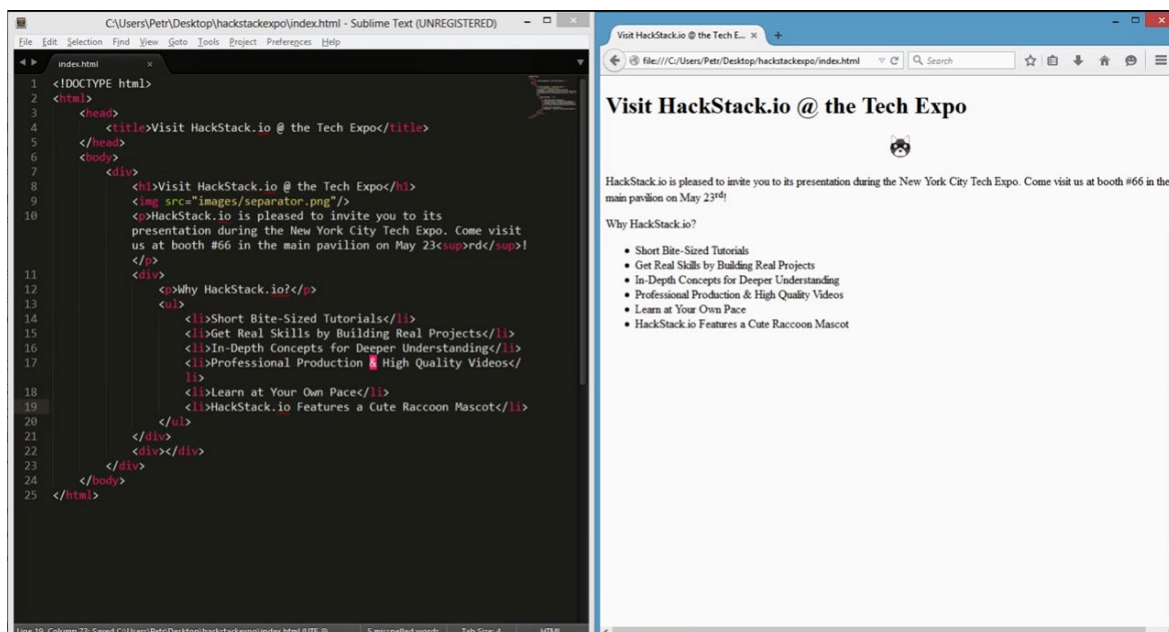
شکل ۴-۲۲ تگ `` و `` (HackStack.io, 2015)

^۱ Box

^۲ Bullet Point

^۳ فهرست نامرتب

^۴ فهرست مرتب



شکل ۴-۲۳ / ایجاد لیست نامرتب و Bullet Point (HackStack.io, 2015)

حال به باکس سمت راست می‌پردازیم. به روش قبل با استفاده از یک تگ `<p>`، عنوان را اضافه می‌کنیم و می‌خواهیم آدرس غرفه^۱ نمایشگاه^۲ را درون آن قرار دهیم. یک تگ `<p>` دیگر برای این منظور به کار می‌بریم. در متن آدرس برای نگارش صحیح^۳ 36th، می‌بایست از یک تگ `` استفاده نمود که دو حرف `th` درون آن قرار می‌گیرند. تگ `<sup>`، مخفف SuperScript^۳ می‌باشد و برای بالانویس کردن متن به کار می‌رود (شکل ۴-۲۴).



شکل ۴-۲۴ / استفاده از تگ `<sup>` جهت بالانویس کردن متن (HackStack.io, 2015)

برای وارد نمودن نقشه گوگل^۴، می‌توانیم مستقیماً از آدرس `map-embed.com` به صورت رایگان کد مورد نظر را دریافت کنیم. در مرورگر آدرس بالا را تایپ نموده و به این صفحه می‌رویم. در محل‌های مربوطه نام مکان مورد نظر، آدرس و اندازه

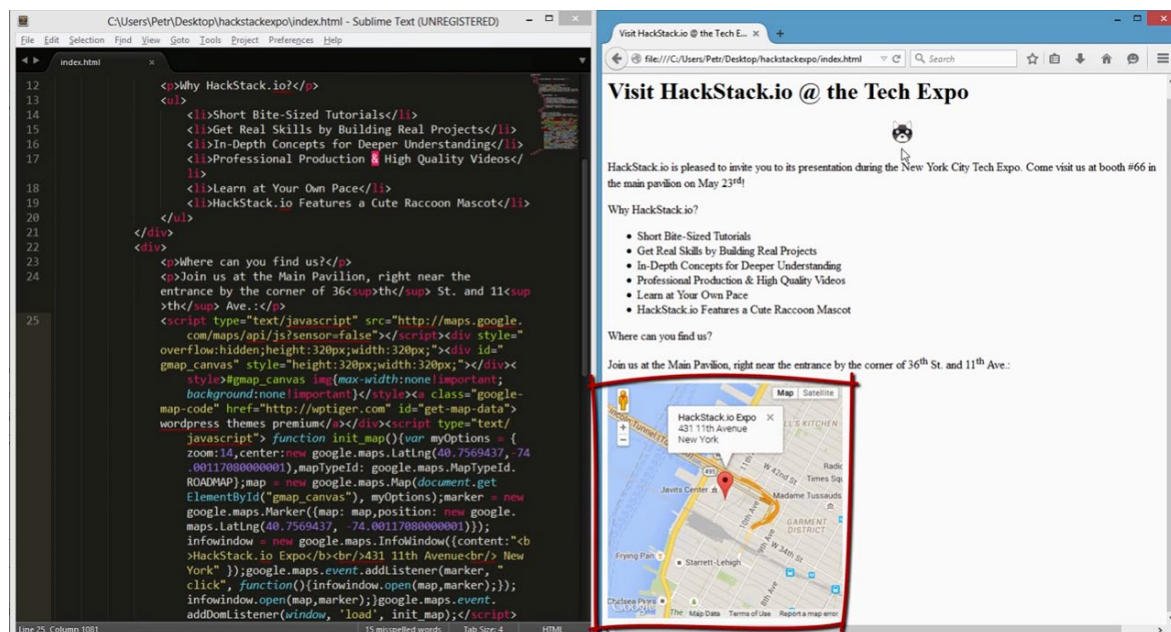
^۱ Pavilion

^۲ Exposition

^۴ Google Map

^۳ بالانویس

ارتفاع و عرض نقشه جهت نمایش در صفحه وب را وارد کرده و در آخر بر روی گزینه Generate Code کلیک کرده تا قطعه کد آن جهت وارد کردن در ویرایشگر متن آماده گردد. کافی است بعد از تگ `<p></p>` این قطعه وارد کنید (شکل ۴-۲۵).



شکل ۴-۲۵ / استفاده از تگ `<sup>` جهت بالانویس کردن متن (HackStack.io, 2015)

حال که نقشه ما در صفحه تعبیه شده است، با یک تگ `
` یک سطر فاصله بین آن و پاراگراف بالا ایجاد می‌کنیم. تنها بخش باقیمانده از صفحه بر طبق نمونه‌سازی ما، اضافه نمودن نماد تصویری^۱ رسانه اجتماعی^۲ می‌باشد. همانند نقشه که از سایت Map-Embed.com قابل دانلود بود، می‌توان با استفاده از خدمات شخص ثالث^۳ این کار را نیز ساده نمود. این دکمه‌های رسانه‌های اجتماعی، شبیه یک لیست عمل می‌کنند با این تفاوت که به جای متن در این جا تصویر داریم. به سایت simplesharingbuttons.com مراجعه نموده و به صورت رایگان، کد را تولید^۴ و دریافت می‌کنیم.

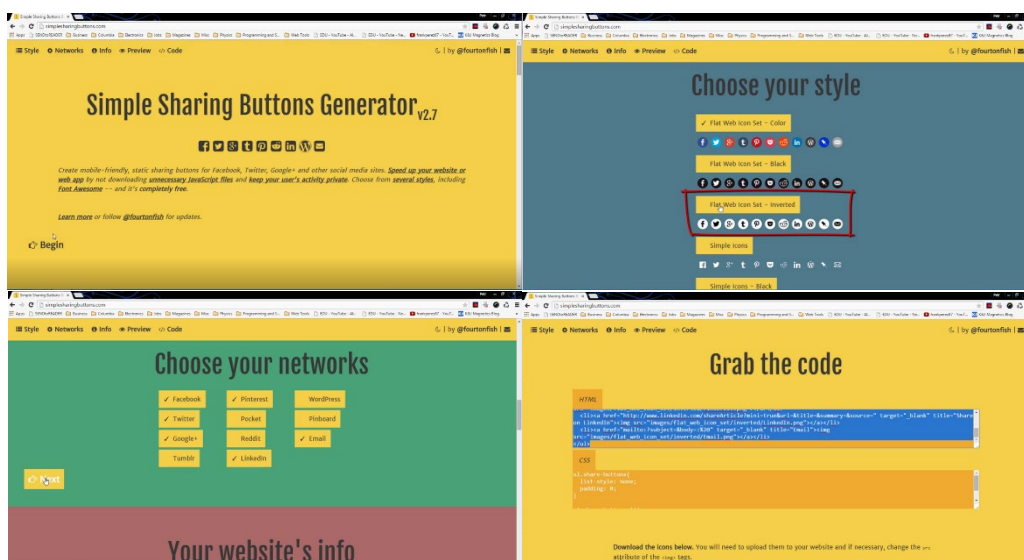
با کلیک بر روی دکمه Begin و انتخاب مجموعه آیکون‌های مورد نظرمان که در این جا Flat Web Icon Set - Inverted می‌باشد، شروع می‌کنیم. سپس از قسمت Choose your networks، آیکون‌هایی را که مد نظرمان است را انتخاب می‌کنیم و دکمه Next را می‌زنیم. می‌توانیم از بخش بعد صرف نظر کنیم و با زدن دکمه Next به بخش Grab the code رفته و قطعه کد را کپی کرده و در ویرایشگر متن مان Paste بنماییم (شکل ۴-۲۶).

^۱ Icon

^۲ Social Media

^۳ Third Party Service

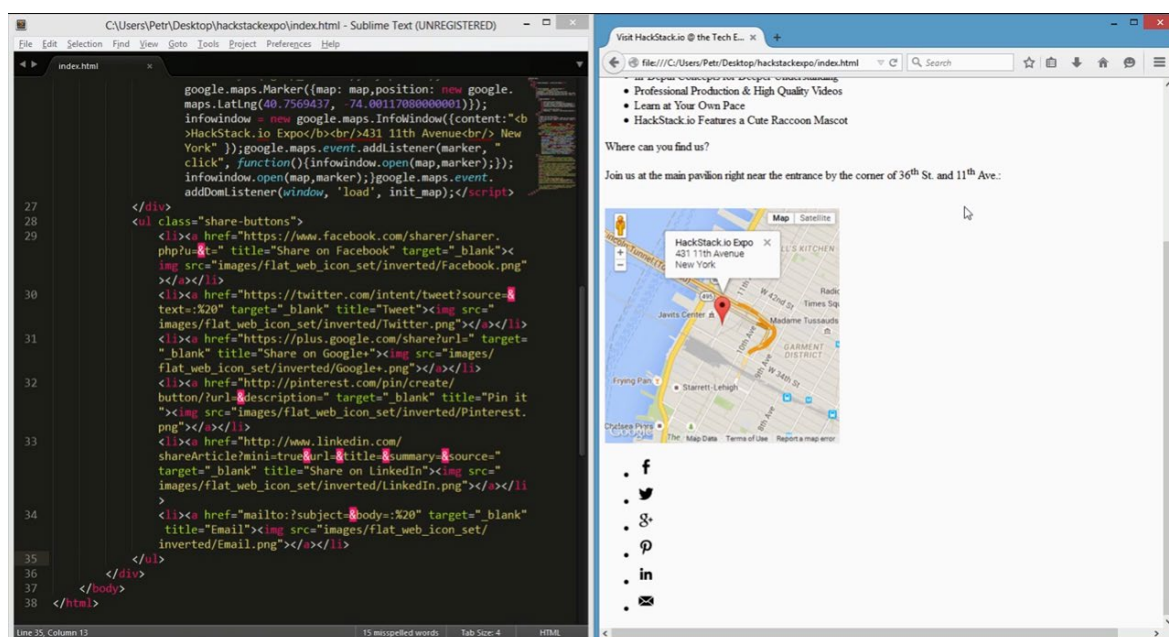
^۴ Generate



شکل ۴-۲۶ مراحل دریافت قطعه کد رسانه‌های اجتماعی از سایت simplesharingbuttons.com

در این سایت قطعه کد CSS، نیز قرار داده شده است که از آن صرف نظر کرده چون می‌خواهیم فایل CSS خودمان را ایجاد کنیم.

به این ترتیب فایل HTML ما آماده گردیده (شکل ۴-۲۷) و حال می‌توانیم به بخش CSS بپردازیم.



شکل ۴-۲۷ فایل آماده شده HTML (HackStack.io, 2015)

۴-۸ CSS چیست؟

CSS و HTML، هر دو با یکدیگر می‌آیند و هر کدام به وسیله دیگری بهتر و کامل‌تر می‌شوند. قبلاً به HTML پرداختیم اما CSS چیست؟

CSS، مخفف Cascading Style Sheets و به معنی برگه‌های آبشاری است. نوعی از کد می‌باشد که جهت طراحی و قالب بندی صفحات وب به کار می‌رود. به چیزهایی نظیر رنگ‌ها، سایزبندی، فاصله‌گذاری، ^۱وضوح، ^۲صفحه آرایشی، تصاویر پس‌زمینه و غیره فکر کنید، اگر HTML اسکلت خانه باشد، CSS عبارت است از رنگ آمیزی، کاغذ دیواری، مفروش کردن و موارد دیگری که باعث می‌شوند به شما حس خانه بودن القا گردد. درست مثل خانه که اندازه اتاق‌ها یا محل قرارگیری پنجره‌ها بر این که چه مبلمانی بخریم یا چه رنگی را انتخاب کنیم، تاثیر می‌گذارد، HTML و CSS به همین شکل بر یک دیگر اثر می‌گذارند. هر کدام مستقل از دیگریست ولی به خودی خود کاملاً بلااستفاده و ناقص می‌باشند. این دو با هم دیگر است که می‌توانند مقصود و شکلی مطلوب را ارائه دهند.

ما می‌دانیم که این دو بایستی با یک دیگر کار کنند اما چگونه این کار را انجام می‌دهند و با هم ارتباط برقرار می‌کنند؟ در ابتدا باید بدانیم که CSS چیزی نیست جز گروهی از مجموعه قواعد قالب‌های بصری، تحت نامی که ما برای آن‌ها انتخاب کرده ایم. برای مثال ما مجموعه ای از قواعد را تعیین می‌کنیم و سپس بر عناصر فایل html نظیر تگ `<div></div>` یا `<p></p>` اعمال می‌کنیم. ابتدا این قواعد را فهرست کرده (شکل ۴-۲۸) و سپس به این فهرست یک نام اختصاص می‌دهیم (شکل ۴-۲۹) که بعداً بتوانیم به آن ارجاع داشته باشیم. این گروه از قواعد را Selector^۳ می‌نامیم (شکل ۴-۳۰).

```
.text-in-box {
  font-weight: bold;
  background-color: blue;
  color: white;
  text-align: center;
  border: 1px solid white;
}
```

شکل ۴-۲۸ فهرست مجموعه قواعد CSS (HackStack.io, 2015)

```
→ .text-in-box {
  font-weight: bold;
  background-color: blue;
  color: white;
  text-align: center;
  border: 1px solid white;
}
```

شکل ۴-۲۹ اختصاص نام به مجموعه قواعد CSS (HackStack.io, 2015)

^۱ Transparency

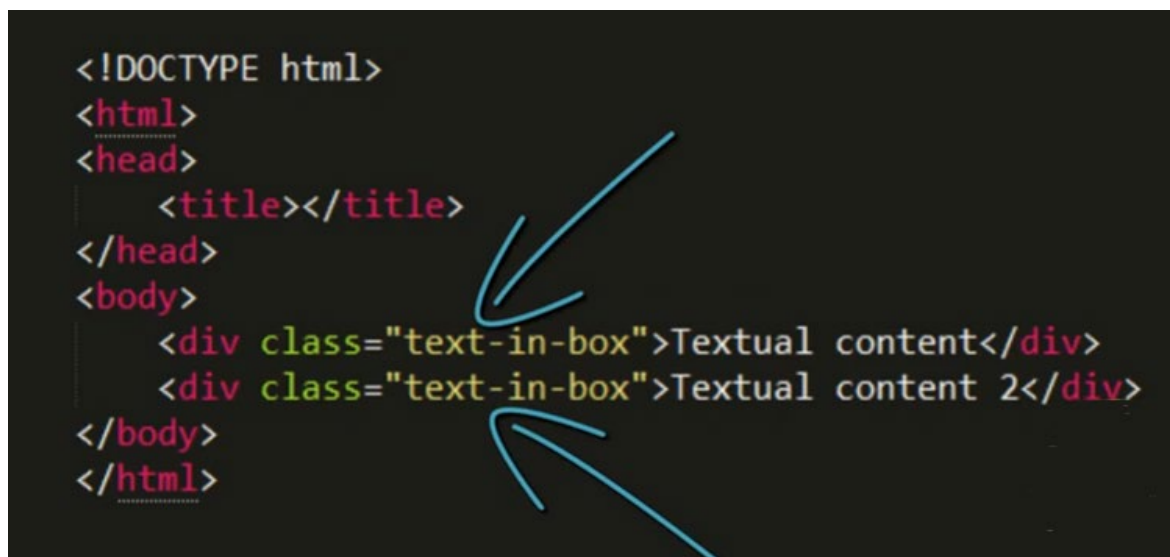
^۲ Layout

^۳ انتخاب‌گر



شکل ۴-۳۰ Selector (HackStack.io, 2015)

چندین روش برای استفاده از CSS در فایل html وجود دارد که بهترین آن اختصاص یک فایل CSS جداگانه می‌باشد که با استفاده از ویژگی‌های class و id^۱ می‌توان به این قابلیت دسترسی داشت. با فهرست کردن یک گروه از ویژگی‌ها و سپس دادن یک نام مشترک، می‌توان در فایل html به آن‌ها رجوع کرده و تمام این ویژگی‌ها را به صورت یک جا به یک یا چندین تگ اعمال نمود (شکل ۴-۳۱).



شکل ۴-۳۱ استفاده از ویژگی class (HackStack.io, 2015)

۴-۹ چگونه مرورگر CSS را از HTML تشخیص می‌دهد؟

مرورگرها به علت syntax^۲ یا قالب^۳ نوشتاری فایل CSS می‌توانند آن را تشخیص دهند. ما از یک فایل جداگانه برای CSS استفاده می‌کنیم. البته روش‌های دیگری نیز برای این منظور وجود دارد.

^۱ Identification

^۲ نحو

^۳ Format

- می‌تواند در قسمت head فایل HTML قرار گیرد اما باعث خطاهای انسانی و کاهش سادگی و تسهیل کار شود.
 - می‌تواند درون تگ نوشته شود که به شدت حجم کد ما را افزایش می‌دهد و دردسرهای بسیاری ایجاد می‌کند.
- هیچ یک از دو روش بالا توصیه نمی‌گردد وقتی می‌توان یک ویژگی را به سرعت با یک تغییر، در یک قسمت و در یک فایل بر روی چندین قسمت طرح اعمال نمود.

۴-۱۰ انتخاب‌گرهای Tag، Class و ID: سه بلوک اصلی ساختمان CSS

CSS، به عنوان مجموعه ای از قواعد طراحی به مجموعه‌های مختلفی تفکیک می‌گردد که با یکدیگر هم گروه شده اند و Selector نامیده می‌شوند. نحو یا Syntax هر انتخاب‌گر به صورت زیر می‌باشد (شکل ۴-۳۲). شامل نام، آکولاد^۱، ویژگی^۲، علامت دونقطه^۳ یا کالن، مقدار^۴ و سمیکالن^۵ که در انتها با یک آکولاد، پایان می‌یابد.

```
.text-in-box {
  font-weight: bold;
  background-color: blue;
  color: white;
  text-align: center;
  border: 1px solid white;
}
```

شکل ۴-۳۲ Selector's Syntax (HackStack.io, 2015)

سلکتورها بر سه نوعند:

۱. انتخاب‌گرهای TAG، که بر تمام تگ‌های دارای نام یکسان اعمال می‌گردند.
۲. انتخاب‌گرهای CLASS، تنها بر آن دسته از عناصر در فایل HTML اعمال می‌گردد که دارای ویژگی class هم نام هستند. این نام می‌تواند هر چیزی باشد اما بهتر است نامی انتخاب کنیم که دلیل و قصد ما را نشان دهد.
۳. انتخاب‌گرهای ID، بسیار شبیه انتخاب‌گرهای CLASS می‌باشند با این تفاوت که پیشوند^۶ نام ویژگی در CSS برای انتخاب‌گرهای ID به جای نقطه علامت هش تگ (#) می‌باشد و در فایل HTML به جای ویژگی CLASS از ویژگی ID استفاده می‌کنیم. اما مهم ترین تفاوت این می‌باشد که از هر سلکتور ID تنها می‌توان برای یک تگ مشخص استفاده نمود.

^۱ Curly Bracket

^۲ Property

^۳ Colon

^۴ Value

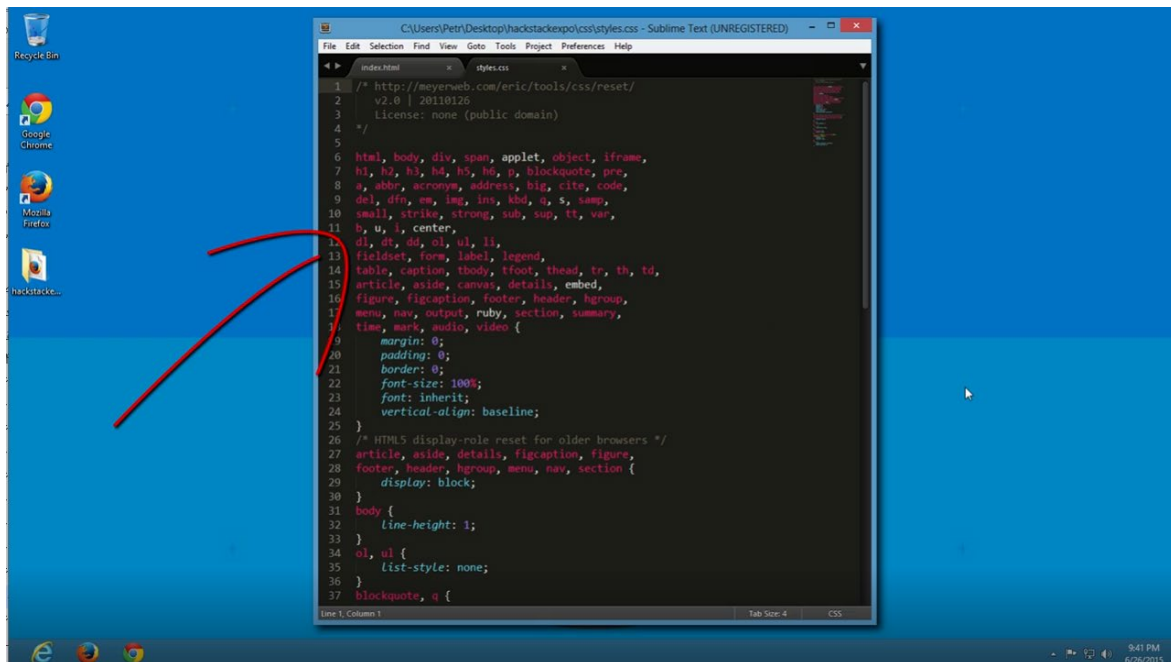
^۵ Semicolon

^۶ Prefix

۱۱-۴ اهمیت RESETS

در این قسمت به بحث Cross-Browser Compatibility^۱ می‌پردازیم که در واقع به این معنی است که با توجه به استفاده از مرورگرهای متفاوت، کدهای HTML و CSS سمت کاربر بایستی به گونه‌ای نوشته شوند تا در تمامی این مرورگرها به یک شکل اجرا و دیده شوند. پیاده‌سازی CSS RESET یکی از راههایی است که به این موضوع کمک می‌کند.

اما CSS RESET دقیقاً چیست؟ مجموعه یا گروهی از قواعد طراحی صفحه یا به عبارتی مجموعه‌ای از کدهای CSS به شکل انتخاب‌گرها، که در آغاز فایل CSS می‌آیند (شکل ۴-۳۳).



شکل ۴-۳۳ CSS RESET (HackStack.io, 2015)

می‌توان از آدرس <https://meyerweb.com/eric/tools/css/reset> یکی از محبوب‌ترین این نمونه کدها را کپی و در ابتدای فایل CSS اضافه کنیم.

۱۲-۴ چگونه از رنگ در CSS استفاده کنیم؟

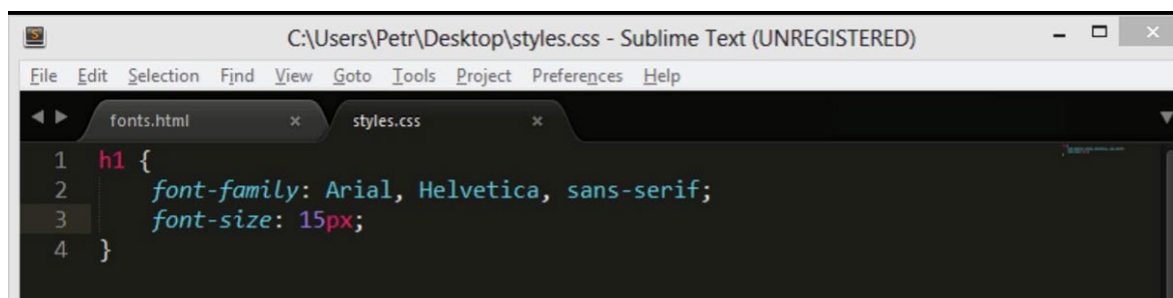
CSS درباره طراحی است و رنگ‌ها بخش بسیار مهم هر طراحی می‌باشند. صرف نظر از نوع Selector، روش‌های مختلفی برای به کاربردن رنگ از طریق Property را پیشنهاد می‌دهد. چندین نوع Property رنگ برای استفاده وجود دارد که هم قابل درک برای انسان و هم منطقی باشد. برای مثال `background-color: red;` مشخص است که برای رنگ پس‌زمینه تخصیص یافته است. نکته مهم این است که شما می‌توانید رنگ را در CSS به وسیله Attribute به سه طریق مشخص کنید که هر کدام برای موقعیت خاصی مفید می‌باشند.

^۱ سازگاری با مرورگر

۱. روش اول استفاده از نام رنگ مورد نظر به صورت کلمه ساده انگلیسی آن می‌باشد.
 ۲. روش دوم استفاده کد هگز^۱ رنگ می‌باشد. کد هگز امکان تعیین دقیق میلیون‌ها رنگ را به ما می‌دهد. این کد رنگ‌ها را می‌توان از منبع رایگان به آدرس <http://colorpicker.com> دریافت نمود.
 ۳. روش آخر استفاده از رنگ‌های RGBA که مخفف Red, Green, Blue و Alpha می‌باشد، است. با ترکیب مقادیر هر کدام بین 0-255 می‌توان شبیه کد هگز میلیون‌ها ترکیب رنگ را به وجود آورد. این روش زمانی که می‌خواهیم رنگ‌هایی را که انتخاب کرده ایم دارای درجاتی از وضوح و شفافیت باشند، به طوری که بتوان پشت رنگ را دید، بسیار کارآمد و مفید است. این جاست که A یا Alpha به کار می‌آید که مقداری بین 0 تا 1 دارد. برای مثال مقدار 0.5 یعنی 50% کدر^۲ بودن. این مقادیر به صورت rgba(red, green, blue, a) در کد ما به کار می‌روند.
- حال از کجا بدانیم که از کدام یک از این سه روش بایستی استفاده کنیم. اگر می‌خواهیم رنگ‌های پایه به کار ببریم از نام انگلیسی ساده آن استفاده می‌کنیم. اگر رنگ خاصی مد نظر باشد از روش کد هگز و اگر رنگ با نسبت وضوح و شفافیت نیاز باشد، روش RGBA را به کار می‌بریم.

۴-۱۳ فونت

همان طور که می‌دانید، فونت‌ها بر روی هر کامپیوتری از قبل نصب گردیده‌اند. به همین دلیل به صورت خودکار می‌توان از آن‌ها در CSS استفاده نمود. برای مثال می‌خواهیم برای یک تگ `<h1>` فونت‌های Arial، Helvetica و sans-serif را به کار ببریم (شکل ۴-۳۴). Property است که نوع فونت مورد استفاده را مشخص می‌کند. اگر دقت کنید متوجه می‌شوید که هر فونت با یک کاما و فاصله از دیگری جدا گردیده که به این وسیله مرورگر تشخیص می‌دهد که Arial اولین انتخاب است و در صورتی که در سیستم بازدیدکننده نصب یا در دسترس نباشد، مرورگر تلاش می‌کند از دومین فونت یعنی Helvetica استفاده کند و در صورتی آن هم در اختیار نباشد فونت سوم، sans-serif به کار می‌رود.



شکل ۴-۳۴ ویژگی فونت (HackStack.io, 2015)

روش دیگر استفاده از Web Font می‌باشد. در مواقعی که فونت مشخص شده در CSS بر روی سیستم بازدیدکننده سایت نصب نباشد می‌توان از این روش استفاده نمود. در این حالت فونت مورد نظر بلافاصله به وسیله مرورگر دانلود می‌گردد. برای تعامل منبع خارجی فونت با فایل HTML، کافی است تگ `<link>` را در قسمت head قرار دهیم (شکل ۴-۳۵).

^۱ HEX Code

^۲ Opaque

```

<!DOCTYPE html>
<html>
<head>
  <title></title>
  <link rel="stylesheet" type="text/css" href="styles.css">
  <link href='http://fonts.googleapis.com/css?family=Open+Sans' rel='
  stylesheet' type='text/css'>
</head>
<body>

```

شکل ۴-۳۵ / استفاده از تگ لینک (HackStack.io, 2015)

زمانی که مرورگر این تگ Link را ببیند، به صورت خودکار فونت را دانلود نموده و با توجه به نوع فونتی که در فایل CSS توسط Selector تعیین شده است آن را نمایش می‌دهد.

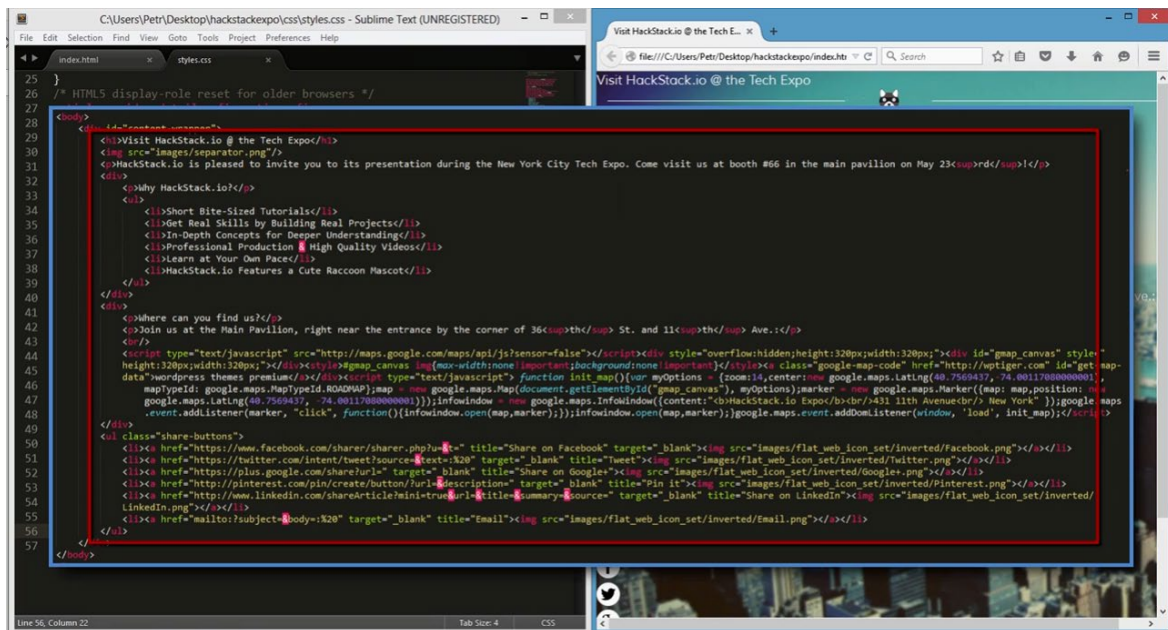
برای این منظور می‌توان به آدرس <https://fonts.google.com> مراجعه نموده، فونت دلخواه را جستجو کرده و قطعه کد آن را طبق شکل قبل به فایل HTML اضافه نمود.

۴-۱۴ با فایل HTML شروع کنیم

حال می‌خواهیم هر آنچه را که تا این جای آموزش فرا گرفته ایم، در عمل به کار ببریم. به فایل HTML بر می‌گردیم و آن را با فایل CSS مرتبط می‌سازیم. به وسیله یک تگ <link> در قسمت head این کار ممکن می‌شود.

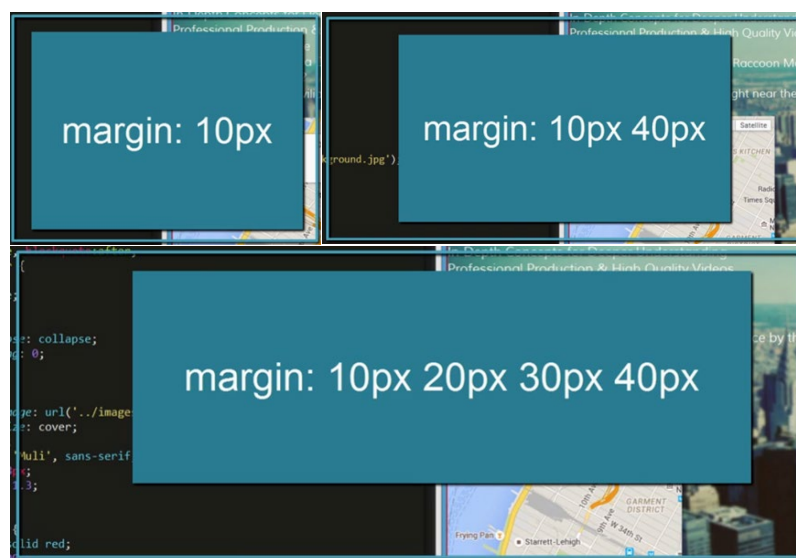
سپس به تنظیمات تگ <body> در فایل CSS و Property های مورد نیاز از قبیل آدرس تصویر پس زمینه، این که آیا تصویر پس‌زمینه در نمایشگر دستگاه‌های مختلف تمامی پس زمینه را صد در صد پوشش می‌دهد، رنگ فونت، نوع فونت، اندازه فونت و فاصله خطوط از یک‌دیگر می‌پردازیم.

بخش بعدی، شکل بخشیدن به Content Wrapper است که در واقع همان تگ <div></div> می‌باشد. معمولاً تمام محتوای صفحه وب را درون خود دارد و تقریباً همه تگ‌های تشکیل دهنده صفحه به صورت تودرتو بین این تگ قرار گرفته است (شکل ۴-۳۶).



شکل ۴-۳ استفاده از تگ <div> در تگ <body> (HackStack.io, 2015)

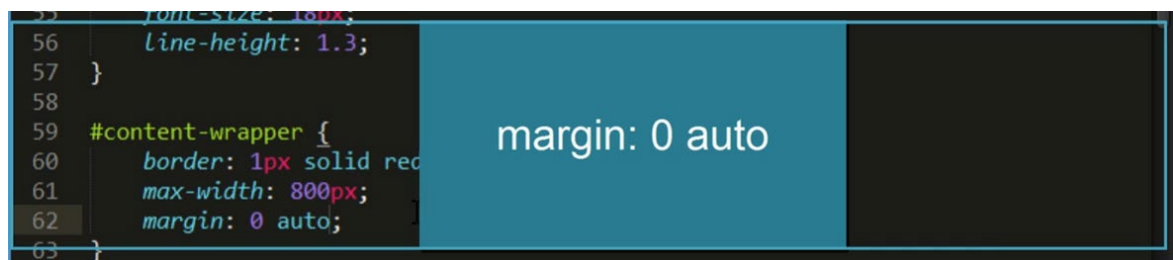
این روش باعث می‌گردد طراحی سهل‌تر صورت گیرد. یکی از مزایای آن این است که محدوده محتوای درون Wrapper نسبت به لبه <div></div> خودش سنجیده می‌شود تا لبه صفحه وب. اولین قدم برای طراحی کردن آن، دادن id می‌باشد تا مرورگر متوجه گردد که قرار است انتخابگر CSS بر این تگ خاص اعمال گردد. بدین منظور Property‌هایی به این id در فایل CSS که عبارتند از border, max-width, margin و padding تخصیص داده می‌شود. در مورد ویژگی margin این نکته لازم به ذکر است که می‌توان به سه نوع آن را اعمال نمود (شکل ۴-۳).



شکل ۴-۳۷ margin با یک مقدار، margin با دو مقدار و margin با سه مقدار (HackStack.io, 2015)

در اولین تصویر، فاصله از همه جهت با لبه صفحه به یک اندازه می‌باشد و در تصویر دوم، اولین مقدار مربوط به فاصله از بالا و پایین و دومین مقدار مربوط به فاصله از سمت راست و چپ می‌باشد. در تصویر سوم، فاصله از لبه صفحه در جهت عمق‌های

ساعت و به ترتیب از بالا، سمت راست، پایین و سمت چپ اعمال می‌گردد. می‌توان از مقدار `margin: 0 auto` برای مواقعی که می‌خواهیم هر آنچه که تعریف کرده ایم در وسط قرار گیرد، استفاده کنیم. مقدار 0 به این معنی است که از بالا و پایین، هیچ فاصله‌ای نباشد و از سمت راست و چپ به یک اندازه فاصله رعایت شود (شکل ۴-۳۸).



شکل ۴-۳۸ / استفاده از `margin: 0 auto` (HackStack.io, 2015)

موضوع بعدی نوشتن CSS برای عنوان صفحه و تصویر جداکننده^۱ می‌باشد. که با تعریف Property برای آنان کامل می‌گردد. سپس یاد می‌گیریم که چگونه فعالانه از CSS برای این که طراحی صفحه وب زیباتر به نظر برسد، استفاده کنیم. با مفاهیم Padding^۲ و چگونگی دادن هم زمان id و class به یک تگ و دادن مقدار به Property ها به شکلی که محتوای درون تگ در تعامل با دیگر اجزا باشد و تغییرات اعمال شده بر این عناصر در آینده با رعایت ارث‌بری^۳ بوده و اصل واکنش‌گرایی در نظر گرفته شود، آشنا می‌شویم. به همین ترتیب برای دو باکس راست و چپ در Wireframe شروع به نوشتن کدهای CSS می‌نماییم.

نحوه قرارگیری باکس‌ها، که می‌بایست در کنار یک دیگر باشند با مفهوم Positioning و Floating^۴ قابل حل است. Float امکان واکنش‌گرا بودن در دستگاه‌های مختلف با اندازه‌های متفاوت را به طراحی می‌دهد. اما با یکی از ایرادات ناخواسته استفاده از Float نیز مواجه می‌شویم. وقتی Float را بر یک تگ اعمال می‌کنیم، موجب Collapse^۵ می‌گردد که به لبریز شدن و بیرون زدن باکس‌های راست و چپ از لبه Content Wrapper منتهی می‌شود (شکل ۴-۳۹).

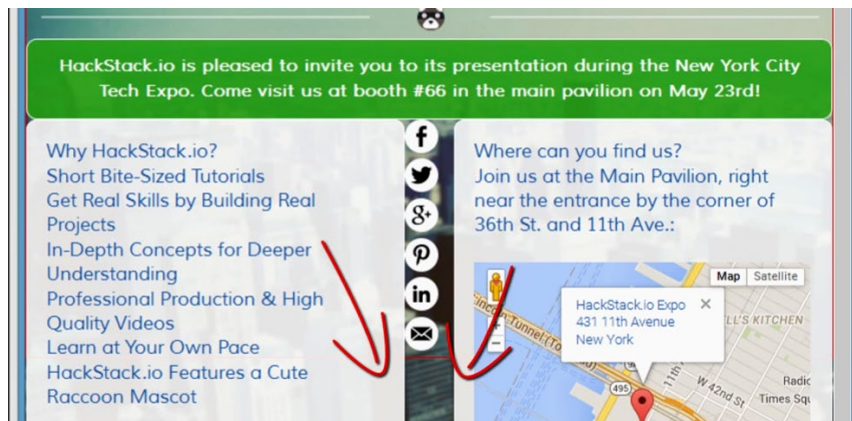
^۱ Separator

^۲ حاشیه داخلی

^۳ Inheritance

^۴ شناوری

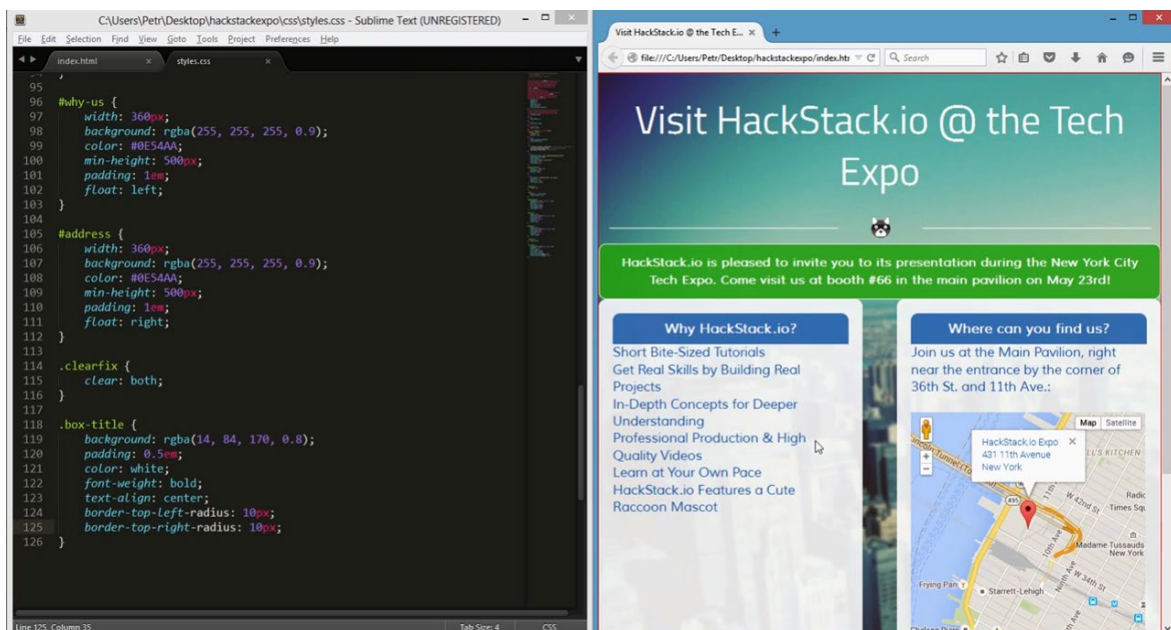
^۵ جمع شدن



شکل ۴-۳۹ به وجود آمدن Collapse بعد از استفاده از Float (HackStack.io, 2015)

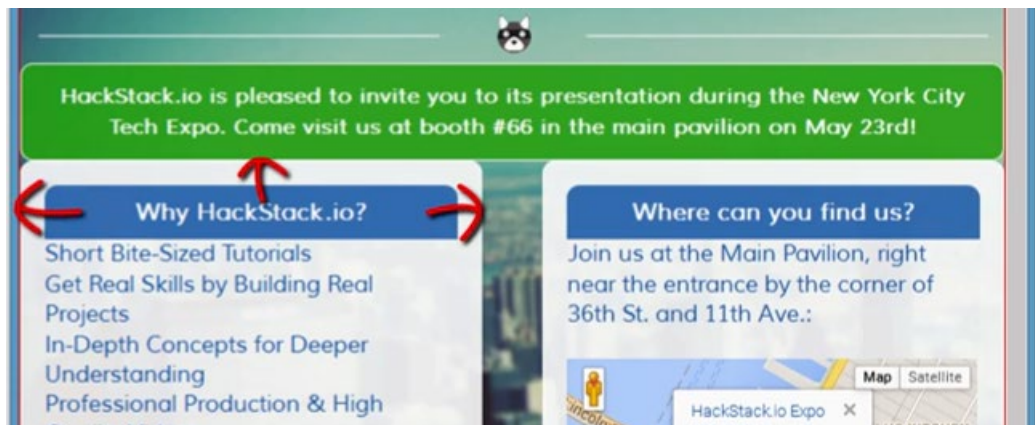
برای رفع این مشکل کافی است ویژگی Clearfix را به تگ <div> مربوط به آیکن‌های رسانه‌های اجتماعی در فایل HTML اضافه کنیم. در فایل CSS با دادن ویژگی clear: both; به کلاس مربوطه مورد برطرف می‌گردد.

مرحله بعد درست کردن باکس عنوان، برای باکس‌های راست و چپ می‌باشد. در فایل HTML انتخاب کلاس مناسب برای هر دو عنوان و ایجاد سلکتور مشابه در فایل CSS و اعمال ویژگی‌های مربوطه شامل background, padding, font-weight, text-align, border-top-left-radius, border-top-right-radius و دادن مقدار به هر یک از آن‌ها منجر به ساخت باکس برای عنوان می‌گردد (شکل ۴-۴۰).



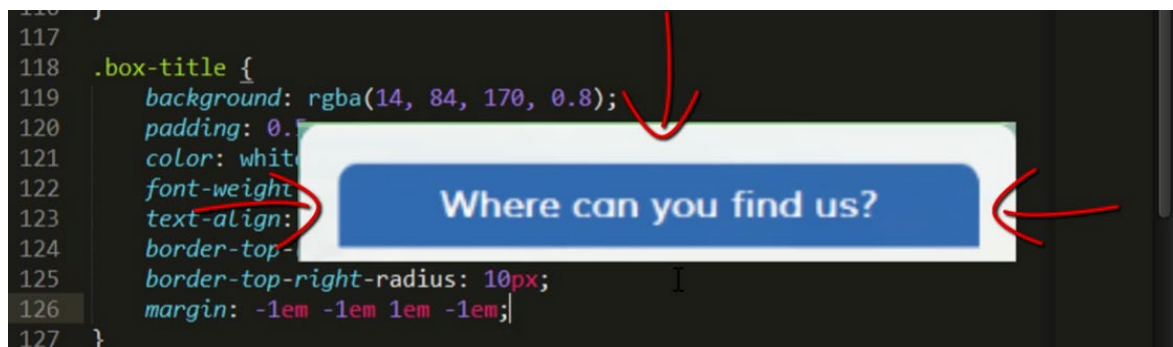
شکل ۴-۴۰ ساخت باکس برای عنوان باکس‌های راست و چپ (HackStack.io, 2015)

می‌خواهیم مرورگر، padding باکس عنوان را با باکس بیرونی در نظر نگیرد اما تنها در سه جهت (شکل ۴-۴۱).



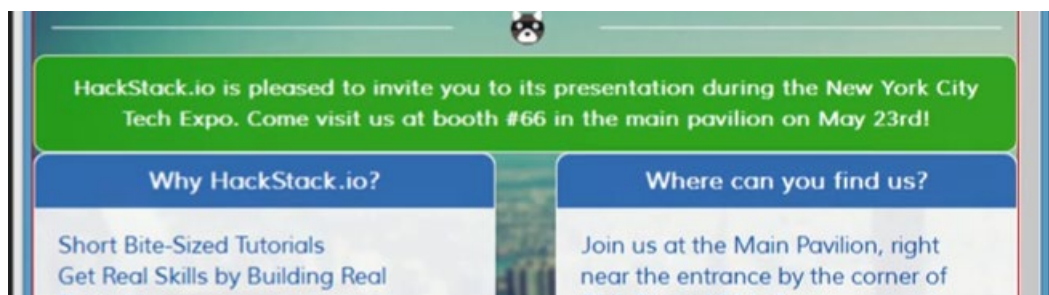
شکل ۴-۴۱ / استفاده از Negative Margin (HackStack.io, 2015)

برای این منظور از negative margin استفاده می‌کنیم. که در این حالت به جای فشار دادن^۱، باکس والد^۲ نسبت به باکس عنوان عمل کشیدن^۳ را اعمال می‌کند (شکل ۴-۴۲).



شکل ۴-۴۲ / استفاده از Negative Margin (HackStack.io, 2015)

شکل نهایی به صورت زیر خواهد بود (شکل ۴-۴۳).



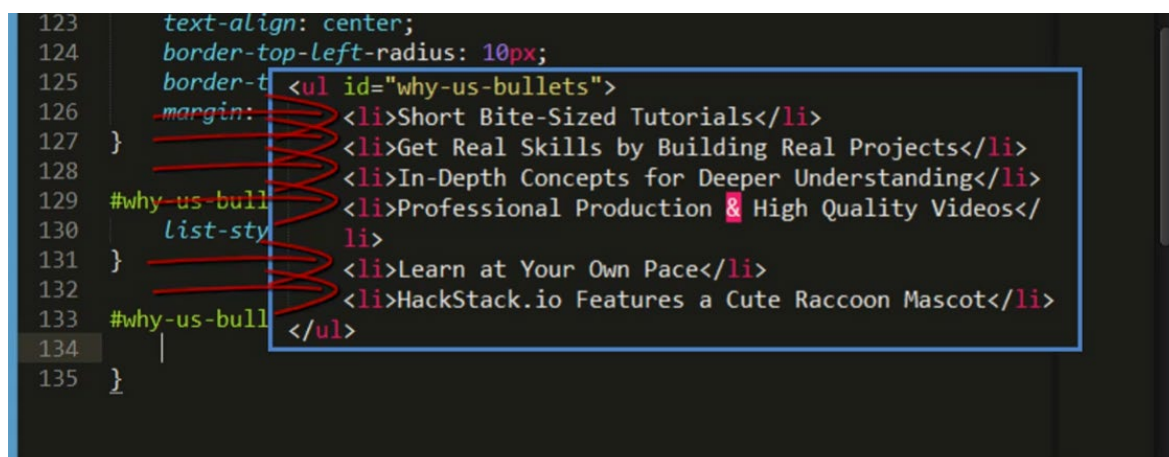
شکل ۴-۴۳ / استفاده از Negative Margin (HackStack.io, 2015)

در خصوص لیست و تگ `` نکته مهم، ساخت نوع خاصی از انتخاب‌گر می‌باشد (شکل ۴-۴۴) که بر تمام تگ‌های `` که به صورت تودرتو درون تگ والد `` قرار دارند، و دارای id یکسان می‌باشند، اعمال شود (شکل ۴-۴۵).

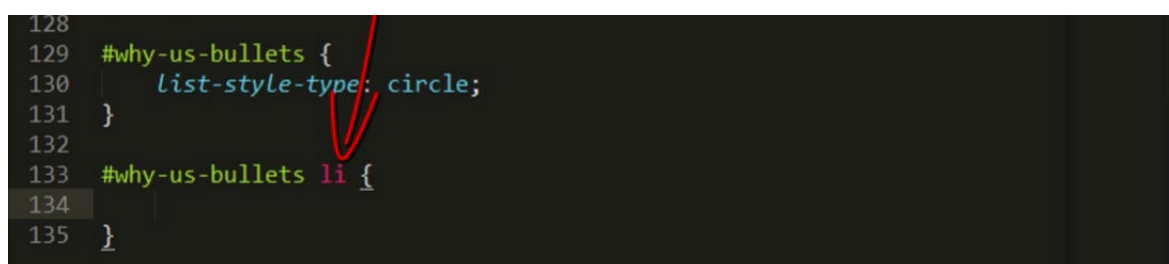
^۱ Push

^۲ Parent

^۳ Pull



شکل ۴-۴ ساخت انتخاب‌گر خاص برای (HackStack.io, 2015)



شکل ۴-۵ ساخت انتخاب‌گر خاص برای لیست (HackStack.io, 2015)

۴-۱۵ طراحی واکنش‌گرا با استفاده از CSS شرطی

حال می‌خواهیم مطمئن شویم که طراحی صفحه وب مان صرف نظر از نوع دستگاهی که بر روی آن به نمایش در می‌آید، کامل و عالی به نظر برسد و تمامی عناصر صفحه در جای خودشان نشان داده شوند. در این جا با کد بسیار کاربردی آشنا می‌شویم. به شکل کلی `{ @media () }` می‌باشد که آن را به عنوان CSS شرطی^۱ می‌شناسیم. چرا شرطی، چون تحت شرایط خاصی اعمال می‌گردد. در داخل پرانتز، شرطی را قرار می‌دهیم که می‌خواهیم از آن به عنوان راه انداز CSS شرطی استفاده کنیم. در این جا چون نگران نوع دستگاه یا اندازه نمایشگر هستیم، از `max-width` به عنوان Property استفاده می‌کنیم. اولاً با توجه به اندازه باکس‌ها در این طراحی مقدار `770px` را انتخاب می‌کنیم ثانیاً و مهم ترین نکته که باید بدانیم این است که مقداری که مشخص نموده‌ایم، تنها زمانی عمل خواهد کرد که اندازه صفحه نمایش `770px` یا کمتر از آن باشد.

چون می‌خواهیم این تنظیمات بر هر دو باکس اعمال شود همان طور که در شکل می‌بینید از دو `id Selector` استفاده نموده‌ایم (شکل ۴-۴۶) که با یک کاما از یک‌دیگر جدا شده اند.

^۱ Conditional CSS

```

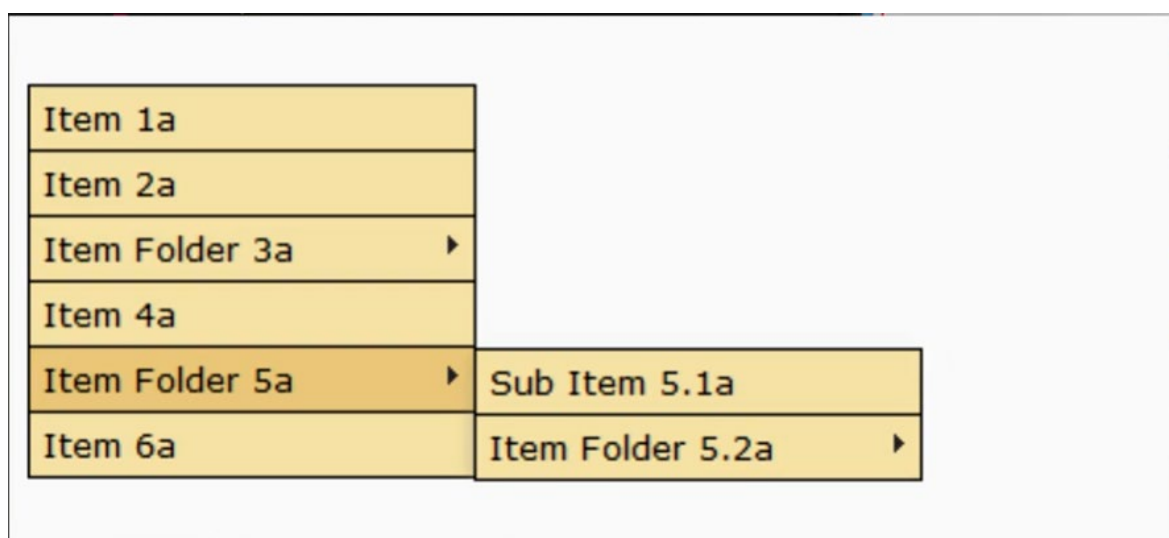
136
137 @media(max-width: 770px) {
138     #why-us, #address {
139         float: none;
140         margin: 1em auto;
141     }
142 }

```

شکل ۴-۴ انتخاب دو id در یک انتخاب گر (HackStack.io, 2015)

۴-۱۶ پیوندهای رسانه اجتماعی

آخرین مطلب باقیمانده، آیکن‌های رسانه‌های اجتماعی می‌باشد. این که به صورت افقی و در وسط صفحه قرار گیرند. اگر به فایل HTML مراجعه کنیم، قبلاً قطعه کدی را برای این منظور به کار برده ایم. که دارای تگ می‌باشد. شاید بد نباشد، بدانید که تگ پایه بیشتر فهرست انتخاب^۱ یا منو صفحات وب می‌باشد (شکل ۴-۴۷).

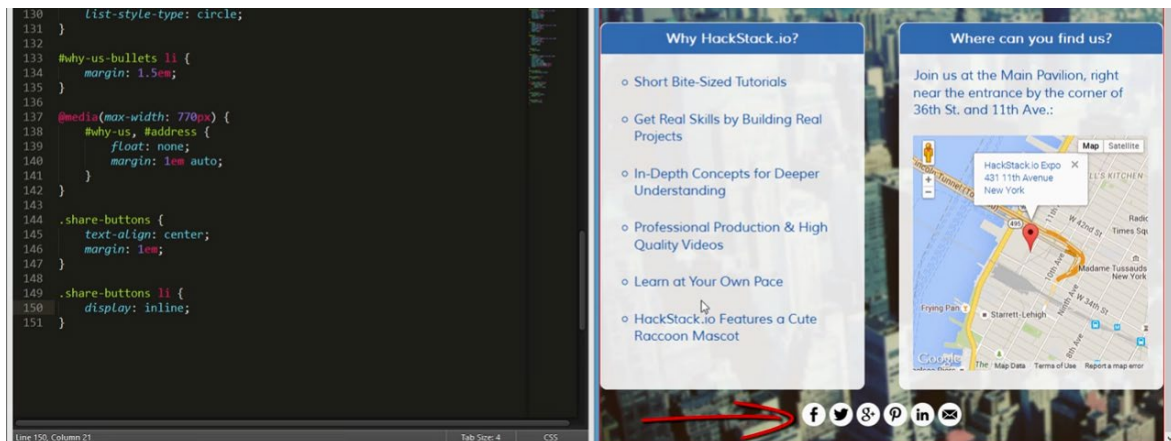


شکل ۴-۴۷ منو با استفاده از تگ (HackStack.io, 2015)

با استفاده از ویژگی‌های text-align و margin هم این نمادها را در وسط قرار می‌دهیم، البته هنوز به صورت عمودی می‌باشند و هم از بالا و پایین به اندازه 1em از باکس‌ها و لبه صفحه فاصله خواهند داشت.

برای نمایش این نمادها به صورت افقی، در فایل CSS با ساخت انتخاب‌گری که فقط بر های کلاس share-buttons اعمال شود به راحتی آن‌ها را به صورت خطی و در یک ردیف قرار می‌دهیم (شکل ۴-۴۸).

^۱ Menu



شکل ۴-۴ تغییر وضعیت تگ `` از عمودی به افقی (HackStack.io, 2015)

۴-۱۷ دوره توسعه^۱ وب سمت کاربر و شبیه‌سازی^۲ Twitter

این بخش از آموزش در تکمیل مطالب قبلی به مفاهیم اصلی HTML و CSS به صورت مفصل‌تری پرداخته است. هم‌چنین بخشی را به آموزش سطح پیشرفته CSS که شامل CSS Flexbox و CSS Grid می‌گردد، اختصاص داده که با آموزش JavaScript توأم گردیده است. در نهایت با انجام پروژه‌ای سعی می‌شود به صورت عملی نیز بتوان آن‌ها را پیاده‌سازی نمود که این امر به یادگیری و تثبیت این مفاهیم کمک شایانی می‌نماید.

۴-۱۸ جمع‌بندی

در این فصل ابتدا به یادگیری پایه‌ای محکم از طراحی وب سمت کاربر پرداختیم و سپس با نمونه‌سازی شروع کردیم و به وسیله HTML5 و CSS3 آن را پیاده‌سازی نمودیم. در ادامه با دو دوره، به یادگیری جاوااسکریپت و هم‌چنین مفاهیم Grid و Flexbox پرداخته و با DOM آشنا شدیم.

فصل بعد به جمع‌بندی این آموزش و هم‌چنین ترسیم نقشه راهی برای استفاده از این کارآموزی و گرفتن نتیجه بهتر از آن در آینده خواهیم پرداخت.

^۱ Development

^۲ Clone

فصل پنجم

جمع‌بندی و پیشنهادها

حال که به انتهای این دوره کارآموزی رسیدیم همان طور که در ابتدای این خودآموز گفته شد به درک بهتر و کاملی از آنچه که قبلا در دروس دانشگاه و دوره های مرتبط فرا گرفته بودیم، رسیده و دانش ما درباره برنامه نویسی وب سمت کاربر که با عمل همراه گردید راه را برای انجام پروژه های آتی فراهم می نماید.

در این فصل به نتیجه گیری درباره هر آنچه که در طول گذراندن این دوره آموخته ایم و ترسیم نقشه راهی برای این که بدانیم در کجای این مسیر قرار داشته و چه آینده ای می توان با طی آن داشت، می پردازیم. برای این که به یک توسعه دهنده وب سمت کاربر تبدیل شویم چه باید کنیم چیزی است که در این فصل به آن خواهیم پرداخت.

۵-۱ چگونه به یک توسعه دهنده وب سمت کاربر تبدیل شویم؟

به طور مختصر توسعه دهنده وب سمت کاربر یا به عبارتی Front-End Developer شخصی است که صفحات وب و نرم افزارهای تحت وب را می سازد. چیزهایی را می سازد که کاربران می توانند، ببینند و این که هر کسی می تواند به یک Front-End Developer تبدیل شود. مسئولیت اصلی توسعه دهنده سمت کاربر، **تعامل با کاربر**^۱ می باشد. به بیان ساده این که صفحه وب چطور به نظر می رسد.

۵-۲ از کجا باید شروع کرد؟

HTML، CSS و JavaScript زبان های پایه ای است که برای ساخت یک صفحه وب لازم است بدانیم. برای این که یک توسعه دهنده سمت کاربر شوید، شروع به یادگیری موضوعات زیر به ترتیبی که آورده شده است، بپردازید:

۱. اولین موضوع، یادگیری زبان HTML می باشد. ساختار هر صفحه وبی با این زبان ساخته می شود.
۲. یادگیری CSS دومین موضوعی است که برای طراحی صفحه وب با رنگ های زیبا و فونت ها نیاز دارید.
۳. پس از یادگیری موارد بالا بهتر است برای ساخت صفحاتی پویا و واکنش گرا، اقدام به یادگیری زبان جاوااسکریپت کنید.
۴. یادگیری کتابخانه های CSS و جاوااسکریپت نظیر Bootstrap، SASS، jQuery، React و همچنین آشنایی با Version Control گیت و انجام تمرین در هر یک از این موارد.
۵. یادگیری RESTful API می تواند به عنوان یک ابزار مهم برای ارتباط Front-End به Back-End^۲ بسیار کمک کننده باشد.

۵-۳ کسب مهارت های مرتبط

برای کسب موفقیت بیش تر در زمینه برنامه نویسی و توسعه ی مهارت، هر توسعه دهنده وبی بهتر است تا با موضوعات زیر نیز آشنا بوده و در آن ها به دانش خود بیفزاید:

۱. آشنایی و یادگیری زبان های برنامه نویسی Python، PHP و SQL
۲. آشنایی با نحوه رفع مشکلات نرم افزاری و سخت افزاری

^۱ User Interface

^۲ سمت - سرورس دهنده

۳. یادگیری مهارت طراحی گرافیک

۴. یادگیری سئو

یادگیری تمامی این موارد نیاز به صرف وقت و تمرینات بسیار در هر یک از موارد ذکر شده دارد اما نتیجه آن بسیار ارزشمند می‌باشد.

مراجع

<https://git.ir/udemy-learn-html5-and-css3-in-3-hours-a-responsive-webpage-tutorial/>, 17 March 2016.

<https://p30download.ir/fa/entry/92061/>, 31 June 2020.

<https://downloadly.ir/elearning/video-tutorials/javascript-the-complete-guide-2021-beginner-advanced-3/>, 29 November 2021.

واژه‌نامه

واژه‌نامه فارسی به انگلیسی

Selector	انتخاب‌گر
SuperScript	بالانویس
Body	بدنه
Collapse	جمع شدن
Margin	حاشیه خارجی
Padding	حاشیه داخلی
Tab	زبان
Cross-Browser Compatibility	سازگاری با مرورگر
Header	سرآیند
Float	شناور بودن
Floating	شناوری
Ordered List	فهرست مرتب
Unordered List	فهرست نامرتب
Font	مجموعه حروف
Positioning	موقعیت‌دهی
Syntax	نحو
Property	ویژگی

واژه‌نامه انگلیسی به فارسی

Attribute	خصوصیت
Back-End	سمت-سرویس دهنده
Background	پس زمینه
Block Elements	عناصر بلوکی
Blueprint	نقشه
Body	بدنه
Box	چارچوب
Browser Title	عنوان مرورگر
Browsers	مرورگرها
Bullet Point	فهرست نقطه‌دار
Cascading Style Sheets	صفحات آبخاری
Case Sensitive	حساس به متن
Cell Phone	گوشی تلفن همراه
Child	فرزند
Client-Side	سمت-کاربر
Client-Side Web Development	توسعه وب سمت-کاربر
Clone	شبیه‌سازی
Coding	کدنویسی
Colon	کالن (نماد دو نقطه)
Conditional CSS	CSS شرطی
Container	حامل
Content	محتوا
Content Wrapper	بسته‌بندی محتوا
Curly Bracket	نماد آکولاد
Desktop	دسک‌تاپ
Development	توسعه
DOM:Document-Object-Model	مدل-شی‌گرا-سند
Domain	دامنه
Elements	عناصر
Exposition	نمایشگاه
Extention	افزونه
File Extention	فرمت فایل
Flexible Box	چارچوب منعطف
Folder	پوشه
Format	قالب

Generate	تولید کردن
Google Map	نقشه گوگل
Header Tag	برچسب سرآیند
HEX Code	کد هگز
Hyper Text Markup Language	زبان نشانه گذاری ابر متن
Icon	نماد
Identification	شناسایی
Identation	دندانه گذاری
Inheritance	ارث بری
Inline Elements	عناصر خطی
Inner	داخلی
Invisible Character	کارکتر مخفی
JavaScript	جاوااسکریپت
Landing Page	صفحه اول سایت
Layout	صفحه آرایشی
Menu	فهرست انتخاب
Nested	تودرتو
Opaque	کدر بودن
Outter	بیرونی
Paired Tags	تگ جفت
Paragraph	بند
Parent	والد
Pavilion	غرفه
Plain Text	متن ساده
Prefix	پیشوند
Property	ویژگی
Pull	کشیدن
Push	فشار دادن
Rendering	تفسیر
Responsive	واکنش گرا
Rich Text Editor	ویرایش گر متن غنی شده
Save	ذخیره کردن
Semicolon	سمیکالن
SEO:Search Engine Optimization	بهینه سازی برای موتور جستجو
Separator	جداکننده
Server-Side	سمت-سرویس دهنده
Snippet	قطعه برنامه

Social Media	رسانه اجتماعی
Space	فضای خالی
Tag	وب
Text Base	مبتهی بر متن
Text Editor	ویرایش گر متن
Theme	تم
Third Party Services	خدمات شخص ثالث
Title	عنوان
Transparency	وضوح
Unpaired Tags	تگ تکی
User Interface	رابط کاربری
Value	مقدار
Web Browser	مرورگر وب
Web Site Hosting Services	خدمات میزبانی وبسایت
Wireframing	نمونه سازی

Abstract

Understanding HTML and CSS can help anyone dealing with the web. Designers can create more engaging and usable websites, web editors can produce better content, marketers can communicate more effectively with their audiences and managers can order better sites and get the most out of their teams.

In this tutorial, we will learn how HTML is used to build web pages. We start by writing the words we want to see on the page and then we add tags and elements to the words that let the browser know what the title is where a paragraph begins and ends and so on. It then introduces tags such as text, list, link, image, table, form, video and miscellaneous elements that are used to build web pages.

Here's how CSS enables us to control the shape and layout of web pages to make them look more interesting using commands. These features are used for example, to control the color of the text, the size and type of the font, the background color and how to add the image to the background of web pages and to control where each of the different elements are located on the page.

After learning HTML and CSS, it is time for JavaScript. This programming language is used to build responsive and dynamic websites. In other words, with JavaScript you can move or change anything on the web page or create animated images and much more. Although such things can be done with CSS but with JavaScript, it is easier and more powerful to do these things. Today, this language is one of the most popular programming languages that is widely used. JavaScript has been built to use for client-side. It means that there is a JavaScript engine inside browsers that allows us to use its code without the need for additional tools or software and thanks to Node.js technology, it is possible to use JavaScript in The server-side is also provided. These cases have made this language very popular and powerful today.



Payame Noor University

**Learning and Practical Use of Web Programming
Languages**

HTML, CSS & JavaScript

An Internship Done for the Award of
Bachelor of Science (BS) in Computer Engineering

Faculty of Engineering
Department of Computer Engineering and Information Technology
Payame Noor University (Tehran Shomal Centre)

Supervisor:
Dr. Seyed Ali Razavi Ebrahimi

By

Mohsen Goli

June 2021