**Html**:

1:در گذشته که از نسخه های دیگر مثل html4, xhtmlاستفاده میشده است، با گذاشتن doctype در واقع اطلاعات اضافی به browser میدادیم که الان برای رندر شدن این کد از چه استانداردی باید استفاده کند. Doctype را نمیتوان به عنوان تگ شناخت.

امروزه دیفالت بیشتر browser ها ، همان html5 میباشد اما باز هم با گذاشتن doctype در اول کد به browser این اطلاعات را می دهیم که باید طبق استاندارد html5 جلو برود و عمل کند در اجرای کد.در واقع برای این است که ورژن html اضافه شده را مشخص کند برای browser ها.

2: *data-\** attributes ها برای اضافه کردن ویژگی ها و اطلاعاتی سفارشی و شخصی سازی شده و خاصی به المنت ها، در سایت ها و اپیکیشن هاست.

در واقع یک روش استاندارد برای اضافه کردن اطلاعات می باشند .

البته که این ویژگی هایی که با پسوند data- شروع میشوند میتوانند توسط کاربر نادیده گرفته شوند.

برای مثال میتوان اطلاعاتی را به سطر های یک لیست اضافه کرد که کاربر با کلیک بر روی آنها بتواند آن اطلاعات را ببیند. مثلا نوع، رنگ ، جنس و ... آن المنت یا نوشته .

3:

در کل بهتر است css در head باشد چرا که ابتدا استایل المنت ها رندر میشوند بعد خودشان، در اینصورت زمانی که صفحه وب لود میشوند تمام المنت ها استایل مربوطه را به خود گرفته اند. در غیر اینصورت اگر کد css در انتهای صفحه باشد. ممکن است برای لحظاتی، یعنی تا زمانی که تگ مروبط به اسایل لود شود، المنت ها بدون استایل بمانند و سپس بعد از اجزای آن خط از کد، استایلشان تغییر کند.

میدانیم که مرورگر ها، تگ ها را به ترتیب و پشت سر هم رندر میکنند، در اینصورت اگر تگ script را در بالای صفحه قرار دهیم، درواقع نمیگذاریم تگ های بعدی لود شوند وکارایی صفحه را پایین می اوریم . اما اگر کد بخش script را به پایین صفحه منتقل کنیم. هنگامی که مرورگر به رندر آن بخش برسد، صفحه ما با تمام المنت ها و استایلش لود شده است، در واقع شمایل سایت بالا آمده است و اینجوری کارایی و پرفورمنس بیشتر و بهتری خواهیم داشت.

**Css**:

1:

Resetting : در این روش تمام استایل های المنت هایی که توسط مرورگر ست شده است، پاک میشوند مانند strong, em, p و ما باید از اول استایل جدیدی به وسیله کد css ، به المنت ها اضافه کنیم .

normalizing": در این روش، استایل ها پاک نمیشوند، بلکه استایل هایی به صورت پیش فرض برای المنت های اصلی در نظر گرفته میشود تا در تمام مرورگر ها شکل ثابت و یکسانی را به خود بگیرند وتنها ما باید css مربوطه به قسمت های باقی مانده که تفاوت دارند را بنویسیم.

معمولا روش دوم انتخاب میشود چون بجای تنظیم و نوشتن css مجدد، css سریع تر و کوچک تری خواهیم داشت.

2:

*box-sizing : این ویژگی مشخص میکند که ایا عرض و ارتفاع (height and width) بر روی padding , border هم اثر بگذارند یا خیر.*

*به صورت دیفالت مقدار این ویژگی برابر با content-box می باشد که به این معناست که فقط height and width شامل content شوند و بر روی padding ,border اثری نگذارند.*

*اما این پراپرتی مقدار دیگری به اسم border-box دارد که به این معناست height and width شامل هم content,paddig ,border شود.*

*3:*

Static : درواقع پوزیشن دیفالت تمام المنت ها static میباشد. یک المنت با موقعیت static ، در موقعیت خاصی قرار نمیگیرد و با جریان کلی صفحه پیش میرود. در واقع در این حالت top,right,left,bottom بر روی موقعیت یک المنت تاثیری نمیگذارد.

Relative: تقریبا شبیه استاتیک میباشد، با این تفاوت که میتوان top,left,right and bottom برای اآن ست کرد. در واقع زمانی که یک المنت، پوزیشن relative به خود میگیرد نسبت به حالت عادی خودش در صفحه قرار میگیرد. فضایی که در صفحه به آن تعلق میگیرد همان فضایی است که در حالت static وجود دارد.

Fixed : در این حالت با توجه به مقداری که برای top,left,bottom and rightبرای المنت مشخص میکنیم، آن در یک موقعیت خاص قرار میگیرد. اما این موقعیت ثابت است و با اسکرول صفحه تغییر نمیکند وقابل مشاهده میباشد.

Absolute : در این حالت المنت نسبت به نزدیک ترین جد خود قرار میگیرد و اگر جدی نداشته باشد از body document استفاده میکند. در واقع المنت میتواند در هر جایی از صفحه قرار بگیرد و با اسکرول جایش تغییر کند. بنابراین ممکن است از دید کاربر خارج شود یا المنت های دیگر را همپوشانی کند.برای آن فضایی در صفحه در نظر گرفته نمیشودو ممکن است در هر جایی از صفحه قرار بگیرد.

4: در این حالت یعنی ابتدا استایل صفحه متناسب با موبایل نوشته میشود وسپس برای صفحات بزرگ تر(view port های بزرگتر). در واقع تمام کد css ای که مینویسیم متناسب با صفحه ی موبایل هست و حال اگر بخواهیم سایت متناسب با صفحات بزرگتر هم باشد یا به عباری multi media باشد از @media استفاده میکنیم و css مربوطه را در این قسمت قرار میدهیم . مثلا اگر سایز صفحه از 300px بیشتر شد حالا بیا و این المنت را به فلان صورت نمایش بده. در این حالت کد ابتدا برای موبایل مینویسیم چون اهمیت و اولویت بیشتری دارد و سپس برای ویوپورت های دیگر

<https://css-tricks.com/front-end-test-element-locators/>

برای قسمت اخر سوال 3 بخش کد ها بنده خیلی سرچ کردم و حدودا 6 ساعت بر روی این تکلیف وقت گذاشتم اما چیزی نیافتم در اخرین لحظات این لینک را در سایت درسی پیدا کردم و فکر میکنم با استفاده ازین ترفند ها بتوانیم خواسته مسئله را انجام دهیم . اما چون قول داده بودم تا شنبه شب بفرستم و هم دیگر تغییر کد ها خیلی زمان میبرد .تغییری ایجاد نکردم .

ممنون میشوم راه حل این قسمت رو برام بنویسید تا یادبگیرم.