6- تفاوت های undefined, null و صفر 0 را بیان کنید.

properties های تعریف نشده در بسیاری از زبان های برنامه نویسی به صورت پیشفرض null در نظر گرفته می شود اما در جاوا اسکریپت undefined در نظرگرفته می شود.اون چیزی که باعث گمراهی بیشتر میشه اینکه شما اگر (x == null) رو چک کنید حتی اگر x در واقع undefined باشد،نتیجه چک true خواهد بود.دلیل این موضع برمیگرده به type coercion که در جاوا اسکریپت اتفاق میفته.

زمان انجام عملیات های ریاضی با null و undefined عملیات های با null value نتیجه integer value خواهد داشت در حالی که هر عملیات ریاضی با undefined نتیجه اون Nan خواهد بود.

7- تفاوت های var , let, const را بیان کنید

متغیرهای var یا دارای حوزه سراسری هستن، یا حوزه تابعی. متغیرهای let و const دارای حوزه بلاکی هستن (Block Scoped).

متغیرهای var می‌تونن دوباره با var تعریف و همچنین مقداردهی بشن. متغیرهای let دوباره نمیتونن تعریف بشن، ولی می‌تونن دوباره مقداردهی بشن. متغیرهایconst ، نه می‌تونن دوباره تعریف و نه دوباره مقداردهی بشن.

متغیرهای var و let و const موقع عملیاتHoisting ، بالای حوزه خودشون میرن، به طوری که متغیرهای var با مقدار undefined پیاده‌سازی میشن. درصورتی که توی let وconst ، متغیرها با هیچ مقداری پیاده‌سازی نمیشن.

وقتی متغیرهای var و let رو تعریف می‌کنیم، می‌تونیم بهشون مقدار ندیم و بعداً این کار رو انجام بدیم. ولی متغیرهای const رو همیشه باید با مقدار، پیاده‌سازی کنیم.

8- تفاوت های کتابخانه و فریم ورک را با مثال بیان کنید

تفاوت اصلی بین کتاب‌خانه و فریم‌ورک در "وارونگی کنترل (inversion of Control)" است. هنگامی که شما یک متد (method) را از کتاب‌خانه فراخوانی می‌کنید، شمایید که بر رویکرد آن کنترل دارید. اما در فریم‌ورک این‌مسئله برعکس است: فریم‌ورک است که کدهای شما را فراخوانی می‌کند.

هنگامی که شما از کتاب‌خانه استفاده می‌کنید، کنترل تمام روند برنامه در دست شماست. شمایید که انتخاب می‌کنید چه‌زمانی و کجا از کتاب‌خانه استفاده‌کنید.

اما در فریم‌ورک شکل‌دیگری از کنترل وجود دارد. فریم‌ورک است که تمام جریان برنامه را در دست می‌گیرد. برخی از مکان‌ها را برای شما فراهم می‌کند تا بتوانید کد خود را به فریم‌ورک متصل کنید و موقع نیاز کدهای شما را فراخوانی می‌کند.

تصور کنید در حال تلاش برای ساخت یک ماشین هستید. با فریم‌ورک تمام مواد ضروری برای ساخت ماشین دراختیار شماست و این‌بستگی به شما دارد چگونه آن‌ها را در کنار هم قرار دهید. فریم‌ورک مانند یک کارخانه عمل می‌کند. با

کدهای از پیش ساخته شده به شما کمک می‌کند محصول خود را بدون فکر کردن به مسائل جزعی و پیکربندی‌های پیچیده، بسازید.

در سمت دیگر، کتابخانه هیچ چیزی را برای شروع در اختیار شما نمی‌گذارد. در مقابلِ فریمورک، ویژگی‌های محدودتری در کتابخانه وجود دارد. همچنین برای استفاده گسترده‌تر از آن‌ها باید از ویژگی‌های ثالث (third-party) دیگر استفاده کنید.

هر کتابخانه، مجموعه‌ای از کلاس‌های (class) تعریف شده‌است. دلیل این امر ساده، استفاده مجدد از کدهاست. برای مثال، کدی که قبلاً توسط توسعه‌دهندگان دیگر نوشته شده است را دریافت‌کنید. کلاس‌ها و متدها معمولاً عملیات خاصی را در یک منطقه خاص، تعریف می‌کنند.

مثلا، چندین کتابخانه برای ریاضیات وجود دارد، که به توسعه‌دهنده اجازه می‌دهند، تابع (function) را بدون پیاده‌سازی مجدد یک الگوریتم و نحوه کار آن، فراخوانی کند.

فریمورک، تمامی جریان (flow) را در کنترل خود دارد و نقاطی از پیش تعریف‌شده وجود دارد که شما باید، با کدهای خود آن‌ها را تکمیل کنید. همچنین فریمورک معمولاً پیچیده‌تر است.

فریمورک اسکلتی را تعریف می‌کند که در آن، برنامه ما ویژگی‌های خاص خود را برای تکمیل‌کردن این‌اسکلت به آن می‌افزاید. از این‌طریق، فریمورک در زمان موردنیاز کدهای شمارا فراخوانی می‌کند.

فایده، این است که توسعه دهندگان، دیگر نیازی به نگرانی در مورد خوب بودن یا نبودن طراحی اسکلت برنامه‌شان ندارند، بلکه تمرکز خود را برروی توسعه آن می‌گذارند.

در هردوی آن‌ها – فریمورک و کتابخانه، APIهای تعریف شده‌ای موجوداست که توسط برنامه‌نویسان، مورد استفاده قرار می‌گیرد. جمع‌بندی: می‌توان به این فکرکرد که کتابخانه، در توسعه ویژگی‌های اپلیکیشن به ما کمک می‌کند و فریمورک اسکلت آن را تشکیل می‌دهد. در این‌میانAPI ، اتصال دهنده‌ای برای استفاده هردوی آن‌ها در کنار هم است.

یک روند توسعه معمولی، با فریمورک شروع می‌شود و کارکرد (function)های بخصوص در کتابخانه‌ها تعریف شده‌است، و از طریق API، خصوصیت‌های توسعه یافته با اسکلت برنامه ادغام و برنامه ما شکل می‌گیرد.

9-بدنه اصلی سند html شامل چه بخش هایی است؟

Head-body