**جلسه هفدهم web1**

1. مطالب جلسه قبل و سوالات تفاوت بین تگ ها و استایل ها و پوزیشن ها ؟
2. شروع استارت پروژه با پوشه بندی صحیح
3. Display:grid
4. آیکون و فونت آوسم و ...
5. ایجاد پالت رنگ – روت
6. Before , after
7. روش های ایجاد سایت مپ و منو و منوی کشویی و همبرگری و دراپ دون
8. سلکتور اکتیو و فوکوس در دکمه ها و تگ سلکت و فرم ها و لینکها و ...
9. پروژه مجتمع در صورتی که صددرصد کد خودمان باشد امکان ارائه به مجتمع فنی دارد
10. انجام پروژه باگ دار و حل آن
11. هرچه تگ یاد گرفته اند با کاربرد و نوع دیسپلی و سئو
12. اشاره به متفاوت بودنشان و تشویقشان
13. تست سرچ کردن و نمره دادن

ایجاد جدول :

**ساختار پایه‌ای یک جدول در HTML:**

* **<table>**: این تگ، کل جدول را محصور می‌کند.
* **<tr>**: هر ردیف جدول را مشخص می‌کند (table row).
* **<th>**: سرستون‌ها (header cells) را مشخص می‌کند که معمولاً برای تیترها استفاده می‌شود.
* **<td>**: داده‌های هر سلول را نمایش می‌دهد (table data).
*  **جداسازی ساختاری**: <tbody> کمک می‌کند تا محتوای اصلی جدول (ردیف‌های داده) از بخش‌های دیگر مانند سرستون‌ها (<thead>) و پاورقی (<tfoot>) جدا شود. این جداسازی به بهبود سازماندهی کد HTML و همچنین کمک به خوانایی آن کمک می‌کند.

بررسی نکات مربوط به Formatting در Table: در HTML، **Formating** در جدول‌ها به معنای اعمال تنظیمات و استایل‌دهی به عناصر داخل جدول است تا ظاهر و چیدمان جدول بهبود یابد و داده‌ها به صورت خوانا و منظم نمایش داده شوند. این فرمت‌بندی شامل استفاده از CSS برای تغییرات بصری در جدول می‌شود.

پراپرتی های مورد استفاده در table :

### 1. border-collapse

پراپرتی border-collapse مشخص می‌کند که آیا حاشیه‌های سلول‌های جدول با یکدیگر ترکیب شوند یا به صورت جداگانه نمایش داده شوند.

* **collapse**:  
  حاشیه‌های سلول‌های مجاور ترکیب می‌شوند و به صورت یک خط نمایش داده می‌شوند. این حالت به شما امکان می‌دهد حاشیه‌های سلول‌ها و ردیف‌ها یکپارچه و به هم پیوسته به نظر برسند.
* **separate**:  
  حاشیه‌های سلول‌ها از هم جدا هستند و فضایی بین آن‌ها وجود دارد. در این حالت، شما می‌توانید از border-spacing برای تعیین فاصله بین حاشیه‌های سلول‌ها استفاده کنید.

2. **border-spacing**

**border-spacing: <length>;**:  
فاصله بین حاشیه‌های سلول‌های جدول را در دو بعد افقی و عمودی تنظیم می‌کند. مقدار این پراپرتی می‌تواند یک عدد (که برای هر دو بعد اعمال می‌شود) یا دو عدد (اولی برای فاصله افقی و دومی برای فاصله عمودی) باشد.

table { border-spacing: 10px 20px; }

مفهوم **Tableless Layout**

مفهوم "Layout" در طراحی وب سایت بدون جدول یا **"Tableless Layout"** به استفاده از CSS برای ایجاد چیدمان (layout) صفحات وب، بدون استفاده از جداول HTML (<table>) اشاره دارد. در گذشته، جداول HTML به طور گسترده‌ای برای ساختار و چینش صفحات وب استفاده می‌شدند. با این حال، این روش مشکلات زیادی به همراه داشت، از جمله:

1. **عدم جداسازی محتوا و نمایش:** جداول برای ارائه اطلاعات جدول‌بندی شده طراحی شده‌اند، نه برای چیدمان صفحه. استفاده از جداول برای چیدمان صفحه باعث می‌شود که ساختار HTML شما پیچیده و گیج‌کننده شود.
2. **کاهش دسترسی (Accessibility):** استفاده از جداول برای چیدمان می‌تواند دسترسی به وب سایت را برای کاربران با ناتوانی‌ها دشوارتر کند، زیرا جداول برای کاربران با ابزارهای کمکی (مثل صفحه‌خوان‌ها) اغلب به درستی تفسیر نمی‌شوند.
3. **مشکلات نگهداری و تغییر:** تغییر چیدمان صفحات وب با استفاده از جداول بسیار دشوارتر و وقت‌گیرتر است، زیرا تغییرات در یک بخش از صفحه می‌تواند بر روی سایر بخش‌ها نیز تأثیر بگذارد.

ایجاد پالت رنگ با روت :

در HTML و CSS، ::root یک شبه‌کلاس (pseudo-class) است که به عنصر ریشه (root element) در یک سند اشاره دارد. این عنصر ریشه معمولاً همان تگ <html> است. استفاده اصلی از ::root زمانی است که شما می‌خواهید متغیرهای CSS را تعریف کنید که در کل سند قابل استفاده باشند.

1. **تعریف متغیرهای رنگ در :root:** شما می‌توانید در بخش :root که نمایانگر ریشه سند است، متغیرهای رنگی خود را تعریف کنید. این متغیرها به صورت سراسری در دسترس خواهند بود.

:root {

--primary-color: #3498db;

--secondary-color: #2ecc71;

--accent-color: #e74c3c;

--background-color: #ecf0f1;

--text-color: #2c3e50;

}

در این مثال:

* + --primary-color: رنگ اصلی (مثلاً برای دکمه‌ها یا لینک‌ها).
  + --secondary-color: رنگ ثانویه.
  + --accent-color: رنگ برای تاکید یا جلب توجه (مثلاً برای دکمه‌های مهم).
  + --background-color: رنگ پس‌زمینه.
  + --text-color: رنگ متن‌ها.

1. **استفاده از متغیرهای رنگ در استایل‌ها:** پس از تعریف متغیرها، می‌توانید آنها را در هر کجا که نیاز به رنگ داشته باشید، استفاده کنید.

body {

background-color: var(--background-color);

color: var(--text-color);

}

**چرا از :root و متغیرهای CSS استفاده کنیم؟**

* **مدیریت آسان:** اگر بخواهید رنگ‌های پروژه‌تان را تغییر دهید، کافیست مقادیر متغیرها را در یک جا تغییر دهید.
* **ثبات در طراحی:** استفاده از متغیرهای رنگی باعث می‌شود که تمام بخش‌های پروژه با یک پالت رنگی ثابت طراحی شوند.
* **دسترسی‌پذیری:** می‌توانید با تغییر مقادیر متغیرها، تم‌های مختلف (مثل حالت تاریک و روشن) را به راحتی پیاده‌سازی کنید

**تنظیمات مرتبط با تم (Theme):**

* اگر وبسایت شما دارای چند تم (مثل تم روشن و تم تاریک) است، می‌توانید متغیرهای مربوط به هر تم را در ::root تعریف کرده و آن‌ها را بر اساس کلاس‌های مختلف یا حالت‌های مختلف تغییر دهید.

:root {

--background-color: white;

--text-color: black;

}

.dark-theme {

--background-color: black;

--text-color: white;

}

body {

background-color: var(--background-color);

color: var(--text-color);

}

در این مثال، تغییر تم با اضافه کردن کلاس dark-theme به عنصر body باعث می‌شود رنگ‌ها بر اساس تم تغییر کنند.