**جلسه چهاردهم 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. animation
4. Transform , transition
5. Display:grid
6. آیکون و فونت آوسم و ...
7. تمرین بررسی یک سایت و تشریح آن
8. انجام پروژه باگ دار و حل آن
9. 14تا سوال از هرکس و پاسخ درست را یک نفر دیگر بدهد و سوال نوشتن من و حدس
10. هرچه تگ یاد گرفته اند با کاربرد و نوع دیسپلی و سئو
11. اشاره به متفاوت بودنشان و تشویقشان
12. انجام پروژه تیمی و توسعه توسط دیگری و چند صفحه ای
13. تست سرچ کردن و نمره دادن

مفهوم واژه ها:

Transform :

Transition :

Translate :

Transparent:

@keyframe و animate :

 **Keyframes**: مسئول تعریف نقاط کلیدی و تغییرات وضعیت عنصر در طول زمان هستند.

 **Animation**: پراپرتی‌هایی که شما را قادر می‌سازد تا انیمیشن‌ها را به عناصر HTML اعمال کنید و کنترل کنید که چگونه انیمیشن اجرا شود.

**پراپرتی‌های کلیدی:**

* **animation-name**: نام انیمیشنی که در @keyframes تعریف شده است.
* **animation-duration**: مدت زمان انیمیشن (مثلاً: 2s یعنی 2 ثانیه).
* **animation-timing-function**: توزیع سرعت انیمیشن (مثلاً: ease, linear, ease-in, ease-out, ease-in-out).
* **animation-delay**: زمانی که باید قبل از شروع انیمیشن صبر شود.
* **animation-iteration-count**: تعداد دفعات تکرار انیمیشن (مثلاً: infinite برای تکرار بی‌نهایت).
* **animation-direction**: جهت انیمیشن (مثلاً: normal, reverse, alternate, alternate-reverse).
* **animation-fill-mode**: تعیین می‌کند که چه اتفاقی بعد از پایان یا قبل از شروع انیمیشن بیفتد (مثلاً: forwards, backwards).
* **animation-play-state**: کنترل پخش یا توقف انیمیشن (paused یا running).

**Transform :**

### ****ویژگی**** transform

ویژگی **transform** در CSS یکی از ویژگی‌های قدرتمند برای اعمال تغییرات بصری بر روی عناصر HTML است. این ویژگی به شما امکان می‌دهد که عناصر را بدون تغییر در ساختار واقعی سند (بدون تغییر در **flow** صفحه)، تغییر دهید، بچرخانید، مقیاس دهید، و حتی کج کنید. این کار به صورت دوبعدی (۲D) یا سه‌بعدی (۳D) انجام می‌شود.

### ****نحوه استفاده از**** transform

ویژگی transform به شما امکان می‌دهد که تغییرات مختلفی را روی یک عنصر اعمال کنید. این تغییرات شامل موارد زیر است:

1. **translate (جابجایی)**:
   * این تابع به شما اجازه می‌دهد یک عنصر را در امتداد محورهای X و Y (و در صورت نیاز Z) جابجا کنید.
   * مثال:

css

Copy code

div {

transform: translate(50px, 100px); /\* 50px به راست و 100px به پایین \*/

}

1. **rotate (چرخش)**:
   * این تابع برای چرخاندن یک عنصر حول نقطه مرکزی آن استفاده می‌شود.
   * مثال:

div {

transform: rotate(45deg); /\* چرخش 45 درجه‌ای در جهت عقربه‌های ساعت \*/

}

1. **scale (مقیاس‌دهی)**:
   * این تابع برای تغییر اندازه عنصر استفاده می‌شود. می‌توانید عنصر را بزرگتر یا کوچکتر کنید.
   * مثال:

div {

transform: scale(1.5); /\* بزرگ کردن عنصر به اندازه 1.5 برابر \*/

}

1. **skew (کج کردن)**:
   * این تابع برای کج کردن یک عنصر در امتداد محورهای X و Y استفاده می‌شود.

div {

transform: skew(20deg, 10deg); /\* کج کردن 20 درجه‌ای در محور X و 10 درجه‌ای در محور Y \*/

}

1. **matrix**:
   * این تابع پیچیده‌تر است و اجازه می‌دهد که شما ترکیبی از تغییرات **translate**، **rotate**، **scale** و **skew** را هم‌زمان اعمال کنید.
   * مثال:

div {

transform: matrix(1, 0.5, -0.5, 1, 30, 20); /\* ترکیبی از تغییرات مختلف \*/

}

### ****نکات مهم در مورد**** transform

1. **ترکیب چندین transform**:
   * شما می‌توانید چندین تغییر transform را به یک عنصر اعمال کنید. برای این کار، کافیست هر تغییر را با یک فاصله از هم جدا کنید:

div {

transform: translate(50px, 100px) rotate(45deg) scale(1.5);

}

1. **نقطه مبنا (Transform Origin)**:
   * **transform-origin**: این پراپرتی به شما اجازه می‌دهد نقطه‌ای که حول آن تغییرات transform انجام می‌شود را تعیین کنید. به طور پیش‌فرض این نقطه، مرکز عنصر است.

div {

transform: rotate(45deg);

transform-origin: top left; /\* چرخش حول نقطه بالا-چپ عنصر \*/

}

1. **عملکرد در حالت سه‌بعدی (3D Transforms)**:
   * در صورت نیاز می‌توانید از تغییرات سه‌بعدی مانند **rotateX**، **rotateY**، **translateZ** و **scaleZ** نیز استفاده کنید.
   * مثال:

div {

transform: rotateX(30deg) translateZ(50px);

}

1. **پراپرتی perspective**:
   * این پراپرتی عمق نمایشی را تعیین می‌کند و به شما اجازه می‌دهد که تاثیرات سه‌بعدی واقع‌گرایانه‌تری ایجاد کنید.

div {

perspective: 500px;

}

### ****کاربردهای معمول**** transform

* **ایجاد انیمیشن‌های پویا**: ترکیب ویژگی transform با ویژگی‌های انیمیشن یا گذار (transition) به شما امکان می‌دهد که انیمیشن‌های پویا و جذابی ایجاد کنید.
* **استایل‌دهی به دکمه‌ها و المان‌های UI**: با استفاده از transform می‌توانید جلوه‌های جالبی به دکمه‌ها و المان‌های UI بدهید، مثلاً دکمه‌ها را در هنگام حرکت ماوس به زیبایی بزرگتر کنید.
* **طراحی رابط‌های کاربری سه‌بعدی**: با استفاده از قابلیت‌های سه‌بعدی transform، می‌توانید رابط‌های کاربری سه‌بعدی ایجاد کنید.

Transition :

پراپرتی‌های transition در CSS برای ایجاد انیمیشن‌های نرم و یکنواخت در هنگام تغییر ویژگی‌های یک المان استفاده می‌شوند. این پراپرتی‌ها به شما امکان می‌دهند تا به سادگی تغییرات را زمانی که یک استایل خاص (مانند رنگ، اندازه، موقعیت و...) تغییر می‌کند، مدیریت کنید. بیایید به انواع پراپرتی‌های transition و کاربردهای آن‌ها نگاهی بیندازیم:

**1. transition-property**

این پراپرتی مشخص می‌کند که کدام ویژگی یا ویژگی‌ها باید به صورت تدریجی تغییر کنند.

* **مثال**:

css

Copy code

transition-property: background-color, width;

این کد مشخص می‌کند که تغییرات در background-color و width باید با یک انتقال تدریجی همراه باشند.

**2. transition-duration**

این پراپرتی مدت زمان انتقال را تعیین می‌کند، یعنی مدت زمانی که طول می‌کشد تا ویژگی از حالت اولیه به حالت جدید خود تغییر کند. مقدار این پراپرتی می‌تواند به صورت ثانیه (s) یا میلی‌ثانیه (ms) باشد.

* **مثال**:

css

Copy code

transition-duration: 2s;

این کد می‌گوید که تغییرات مشخص شده در transition-property باید طی 2 ثانیه اتفاق بیافتد.

**3. transition-timing-function**

این پراپرتی سرعت تغییرات را در طول انتقال مشخص می‌کند. به عبارت دیگر، تعیین می‌کند که آیا انتقال باید به صورت خطی، با شتاب یا کاهش سرعت انجام شود.

* **مقادیر معمول**:
  + ease: انتقال با سرعت کاهش و افزایش نرم اتفاق می‌افتد (پیش‌فرض).
  + linear: انتقال با سرعت ثابت اتفاق می‌افتد.
  + ease-in: انتقال به تدریج سریع‌تر می‌شود.
  + ease-out: انتقال به تدریج آهسته‌تر می‌شود.
  + ease-in-out: انتقال با سرعت آهسته شروع می‌شود، سپس سریع می‌شود و دوباره آهسته می‌شود.
* **مثال**:

css

Copy code

transition-timing-function: ease-in-out;

این کد می‌گوید که انتقال به صورت آهسته شروع و پایان یابد، ولی در وسط سریع‌تر شود.

**4. transition-delay**

این پراپرتی مشخص می‌کند که انتقال باید بعد از چه مدت زمانی شروع شود. این مقدار می‌تواند به صورت ثانیه (s) یا میلی‌ثانیه (ms) باشد.

* **مثال**:

css

Copy code

transition-delay: 1s;

این کد می‌گوید که انتقال باید 1 ثانیه بعد از اعمال استایل شروع شود.

**5. transition (شکل مختصر)**

این پراپرتی ترکیبی است که تمام موارد بالا را در یک خط کد خلاصه می‌کند. ترتیب مقادیر باید به این صورت باشد: transition-property, transition-duration, transition-timing-function, transition-delay.

* **مثال**:

css

Copy code

transition: background-color 2s ease-in-out 1s;

این کد به مرورگر می‌گوید که تغییر در رنگ پس‌زمینه باید 2 ثانیه طول بکشد، از نوع ease-in-out باشد و بعد از 1 ثانیه تأخیر شروع شود.

**مثال کلی:**

css

Copy code

.box {

width: 100px;

height: 100px;

background-color: red;

transition: width 2s ease, background-color 1s linear;

}

.box:hover {

width: 200px;

background-color: blue;

}

* در این مثال، وقتی که موس روی .box قرار می‌گیرد، عرض آن به آرامی در مدت 2 ثانیه تغییر می‌کند و رنگ پس‌زمینه نیز در 1 ثانیه به رنگ آبی تغییر می‌کند.

**کاربرد:**

* **انیمیشن‌های ساده**: ایجاد افکت‌های بصری زیبا و نرم برای تعامل بهتر با کاربر.
* **تغییرات ناگهانی**: جلوگیری از تغییرات ناگهانی و ایجاد تجربه کاربری بهتر.
* **رابط کاربری**: کاربردهای فراوان در طراحی رابط کاربری، مانند دکمه‌های تعاملی، منوها، تصاویر، و غیره.

1. مفهوم parallax :

فرض کنید شما در حال رانندگی در یک جاده هستید. به اطراف نگاه می‌کنید و می‌بینید که درختان کنار جاده نسبت به کوه‌های دوردست سریع‌تر حرکت می‌کنند. این به دلیل اختلاف سرعت حرکت اجسام در فواصل مختلف نسبت به شما است. این افکت در وب سایت‌ها نیز قابل پیاده‌سازی است و به آن افکت Parallax گفته می‌شود.

**زمینه (Background) و محتوا (Content)**:

1. مفهوم single page (SPA):

 **وب‌سایت‌های سنتی (چند صفحه‌ای)**:

* در وب‌سایت‌های سنتی، هر بار که شما روی یک لینک کلیک می‌کنید، یک صفحه جدید از سرور بارگذاری می‌شود.
* این فرآیند ممکن است باعث وقفه‌ها و تأخیرهایی در تجربه کاربری شود.

 **وب‌سایت تک‌صفحه‌ای (Single Page Application)**:

* در وب‌سایت‌های تک‌صفحه‌ای، همه چیز در یک صفحه واحد بارگذاری می‌شود.
* وقتی شما روی یک لینک یا دکمه کلیک می‌کنید، به جای بارگذاری مجدد کل صفحه، فقط بخش‌هایی از محتوا به روز می‌شوند.
* این فرآیند باعث می‌شود که تجربه کاربری بسیار سریع‌تر و روان‌تر باشد.

### فریمورک‌ها و کتابخانه‌ها:

برای ساخت SPAها، معمولاً از فریمورک‌ها و کتابخانه‌های جاوااسکریپت مثل React، Angular، یا Vue.js استفاده می‌شود که به تسهیل مدیریت و به روز رسانی بخش‌های مختلف صفحه کمک می‌کنند.

**Image map**

یک **Image Map** در HTML به شما اجازه می‌دهد تا نواحی خاصی از یک تصویر را به لینک‌های مختلف اختصاص دهید. به عبارتی، می‌توانید یک تصویر را به چندین ناحیه قابل کلیک تقسیم کنید که هر ناحیه به آدرس URL متفاوتی هدایت می‌شود.

### توضیح:

* **<img>**: تصویر اصلی که می‌خواهید نواحی کلیک‌پذیر روی آن ایجاد کنید.
* **usemap**: ویژگی‌ای که به یک نقشه (map) لینک می‌دهد.
* **<map>**: نقشه تصویر که شامل نواحی کلیک‌پذیر است.
* **<area>**: تعریف هر ناحیه با شکل و مختصات خاص و لینک مربوط به آن.

### انواع شکل‌ها:

* **rect**: مستطیل (coords: x1,y1,x2,y2)
* **circle**: دایره (coords: x,y,radius)
* **poly**: چندضلعی (coords: x1,y1,x2,y2,x3,y3,...)