CENNI DI: 3D CSS E CSS Animation

Un’altra faccia del CSS:

Slide 3 - CSS Perspective vs CSS Transform:perspective;

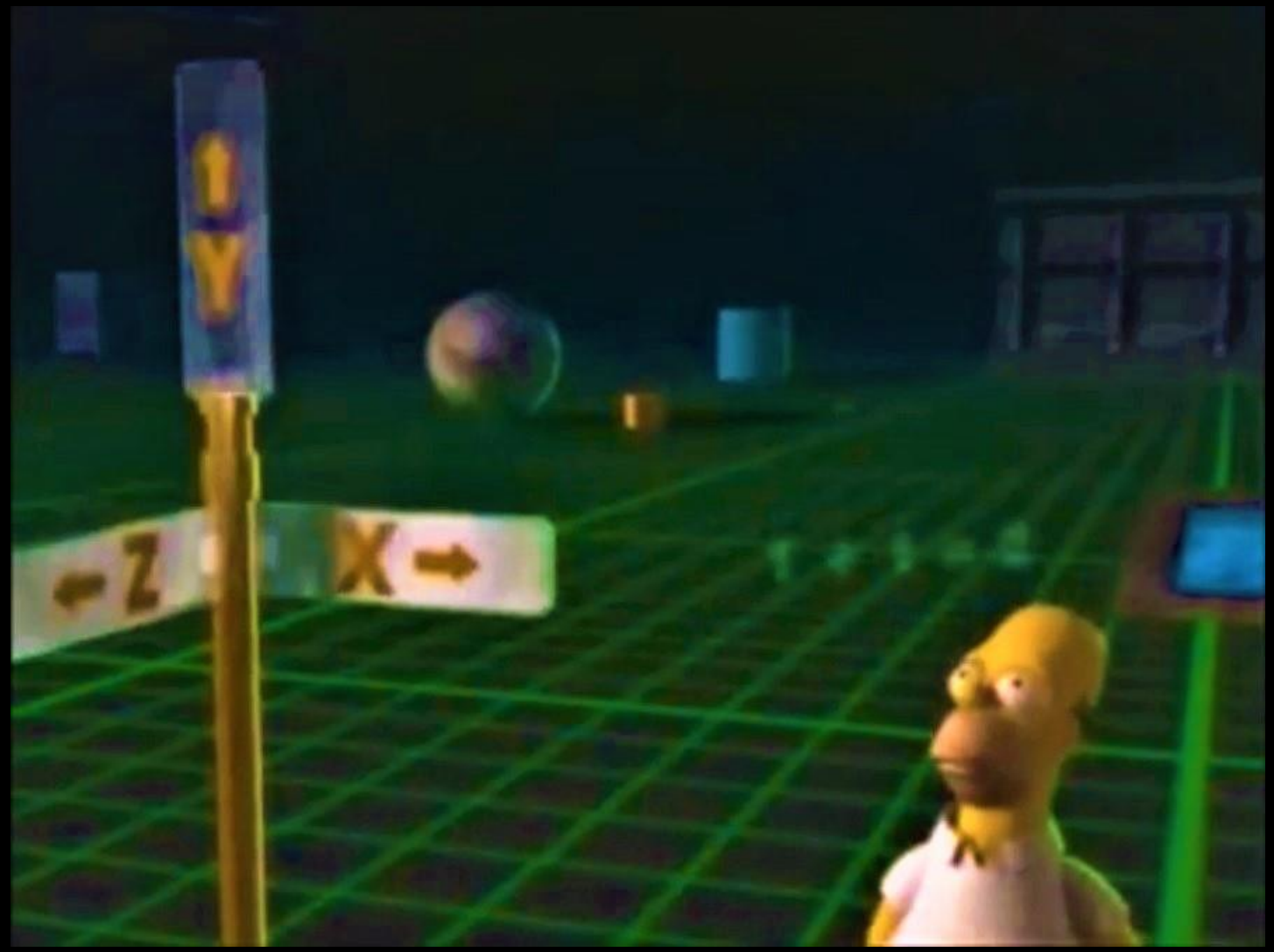
Slide 4 - CSS 3D Transform;

Slide 6 - CSS Transition+Transformation;

Slide 9 - CSS Animation.

3D CSS: CI SENTIAMO PERDUTI COME HOMER?

Niente paura, anche lui ne è uscito (in qualche modo ☺)



SE VUOI SCOPRIRE COME VA A FINIRE:

https://www.youtube.com/watch?v=KphaTbU7-Iw&t=101s

CSS: PROPRIETA’ perspective E transform: perspective

Serve a dare profondità a determinati elementi:

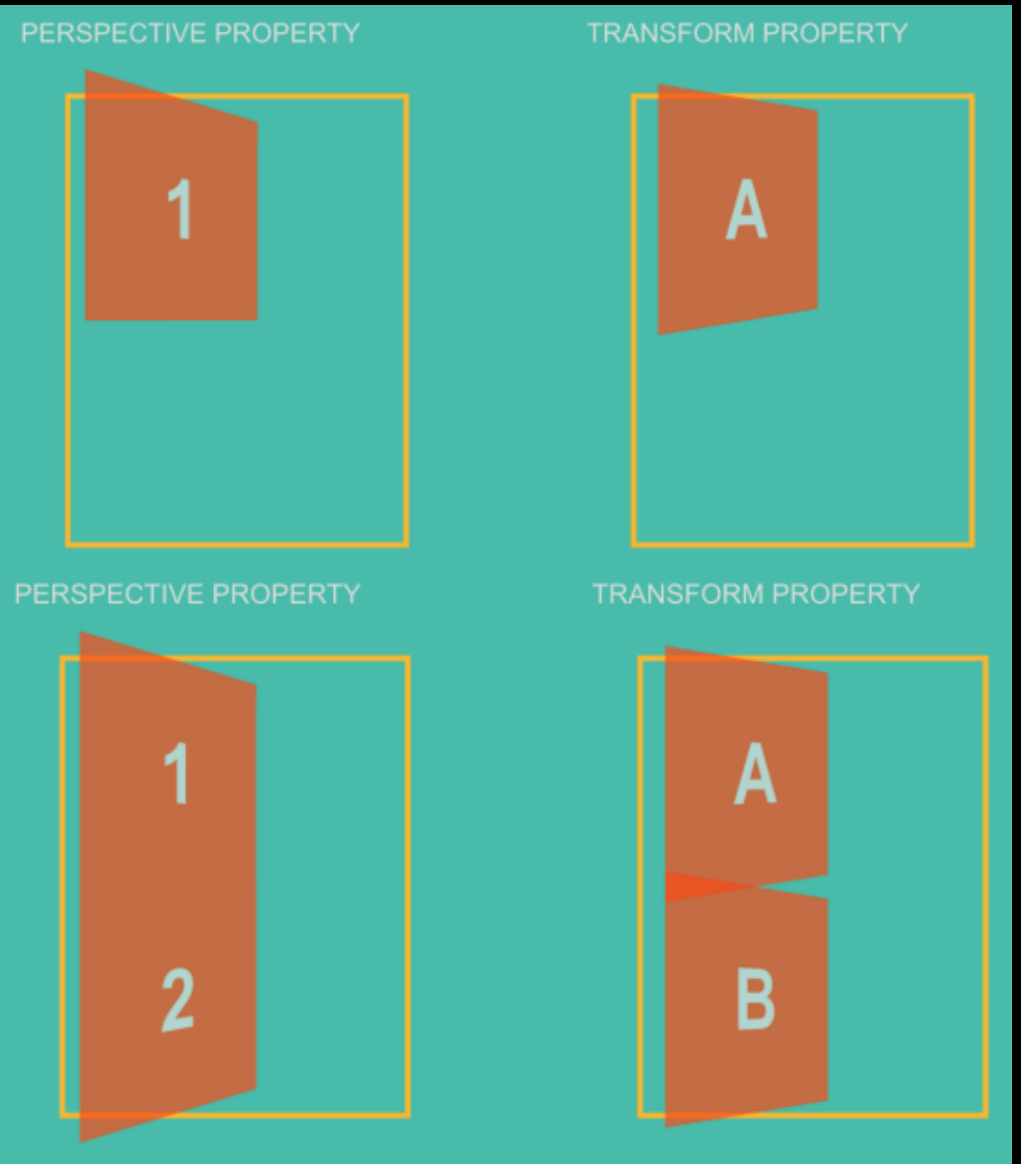
La proprietà PERspective usa una logica PERversa (finte risate): quanto vuoi che l’elemento in questione si avvicini o allontani dal tuo naso? 10px di perspective daranno un effetto 3D più spinto al tuo elemento rispetto a 1000px, perché l’elemento disterà solo 10 px dal tuo naso e non 1000px.

Perspective (1) VS Perspective+Transform (2) (esempi nella slide 5):

(1) Quando assegnata come proprietà semplice ad un elemento, – es= perspective: 600px; – non è l’elemento stesso ad ottenere l’effetto prospettiva ma suo figlio.

(2) Se invece inseriamo una perspective anticipata da transform
– es = transform: perspective (600px); – sarà l’elemento stesso ad ottenere l’effetto prospettiva.

ESEMPI: Perspective e transform perspective UNITE A rotatey(angle)



PREMESSA: CONSIDEREREMO SOLO LE PROPRIETA' CHE FORNISCONO L'ASPETTO 3D AI NOSTRI div

---

ESEMPIO perspective PROPERTY

*Per container si intende il riquadro giallo, one è il blocco al suo interno

.container { perspective: 600px; }

.one {transform: rotateY(45deg); }

ESEMPIO transform PROPERTY

*Per A si intende il blocco all'interno del container (in questo caso il nostro container non possiede proprietà specifiche per l'effetto 3D)

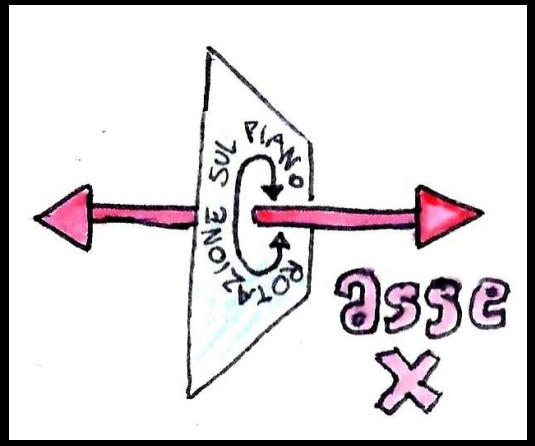
.A { transform: perspective(600px) rotateY(45deg); }

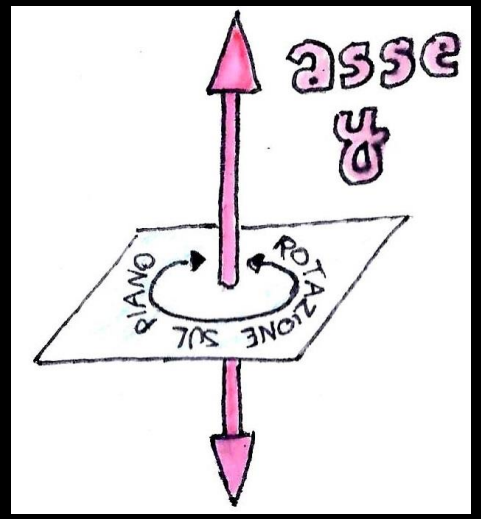
NELL'IMMAGINE SOTTOSTANTE, AGGIUNGENDO UN div IDENTICO A QUELLI VISTI PRIMA ALL'INTERNO DEL CONTAINER, SI RENDE EVIDENTE UN FATTO: PER CIO' CHE RIGUARDA L'EFFETTO 3D NEL CASO DI SINISTRA I 2 BLOCCHI SI TRASLANO PROPRIO COME SE IL LORO CONTENITORE FOSSE UNA SCATOLA VISTA DALL'ALTO (IMMAGINIAMO ANCHE LE PARETI DI QUESTA SCATOLA). NEL CASO DI DESTRA I NOSTRI div SEGUONO UN 3D LORO CARATTERISTICO E IL CONTAINER LI TIENE SEMPLICEMENTE AL SUO INTERNO.

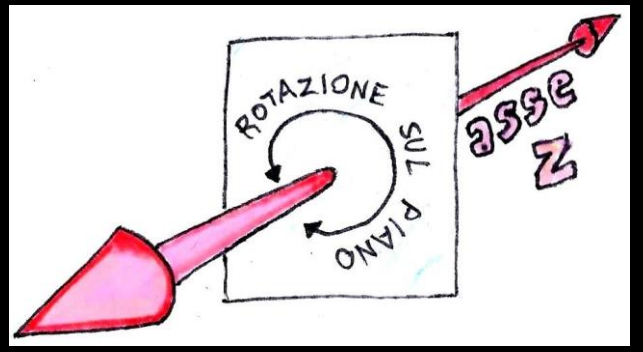
CSS 3D transforms – CENNI DI GEOMETRIA SPICCIOLA

Traslazioni: il piano di proiezione, con il nostro elemento sopra, si sposta lungo l'asse.

Rotazioni: L'elemento scivola sul piano di proiezione dell'asse.







	TRASLAZIONE X	TRASLAZIONE Y	TRASLAZIONE Z
Proprietà CSS unitarie	Transform: translateX(*x*)	Transform: translateY(*y*)	Transform: translateZ(*z*)
	Valori in pixel. Più aumentate i pixel e più l'elemento si sposta lungo l'asse.		
Shortcut	Transform: translate3d(*x,y,z*) → valori in pixel		
	ROTAZIONE X	**ROTAZIONE Y**	**ROTAZIONE Z**
Proprietà CSS unitarie	Transform: rotateX(*angle*)	Transform: rotateY(*angle*)	Transform: rotateZ(*angle*)
	Valori in deg(gradi) (es 90deg, 180deg, 270deg, 3600deg…)		
Shortcut	Transform: rotate3d(*x,y,z,angle*)→valori da 0a1 che fungono da moltiplicatori per l'ultimo valore (deg)		

CSS Transition+Transformation 1/3

POSSIAMO UTILIZZARE QUESTA COMBINAZIONE PER CREARE UN EFFETTO IN ALMENO DUE CASI: POSSIAMO CREARE UN EFFETTO IN CASO DI :hover (1) OPPURE POSSIAMO FARE IN MODO CHE IL NOSTRO ELEMENTO SFOGGI UN EFFETTO AUTONOMAMENTE (2).

(1) EFFETTO IN CASO DI :hover → esempio pratico (vedi anche HTML allegato):

```
div {
  width: 100px;          /* LARGHEZZA INIZIALE*/
  height: 100px;         /* ALTEZZA INIZIALE*/
  background: red;
  -webkit-transition: width 2s, height 2s, -webkit-transform 2s; /* Safari */
  transition: width 2s, height 2s, transform 2s;          /* TEMPI DI CRESCITA DI LARGHEZZA, ALTEZZA E TRASFORMAZIONE */
}

div:hover {
  width: 300px;          /* LARGHEZZA A FINE TRANSIZIONE*/
  height: 300px;         /* ALTEZZA A FINE TRANSIZIONE*/
  -webkit-transform: rotate(180deg); /* Safari */
  transform: rotate(180deg);      /* EFFETTO DELLA TRASFORMAZIONE*/
}
```

CSS Transition+Transformation 2/3

(2) L'ELEMENTO AGISCE DA SOLO → esempio pratico (vedi anche HTML allegato):

```
.div {
    background-color: blue;
    transform: rotateX(360deg);
    transition-delay: 1.5s;
    transition: ease 1s;   */shortcut per indicare transition-timing function +  transition-duration/*
}
```

CSS Transition+Transformation 3/3

TRANSITION-TIMING-FUNCTION

La transition-timing-function di default in CSS è ease (linear considerato noioso).

```
.examples {

  transition-timing-function: linear;
  transition-timing-function: cubic-bezier(0, 0, 1, 1);

  transition-timing-function: ease;
  transition-timing-function: cubic-bezier(0.25, 0.1, 0.25, 1);

  transition-timing-function: ease-in-out;
  transition-timing-function: cubic-bezier(0.42, 0, 0.58, 1);

  transition-timing-function: ease-in;
  transition-timing-function: cubic-bezier(0.42, 0, 1, 1);

  transition-timing-function: ease-out;
  transition-timing-function: cubic-bezier(0, 0, 0.58, 1);

}
```

Di lato: le transition-timing-functioning preimpostate con le relative cubic bezier

Le curve di bezier sono totalmente personalizzabili e ci sono anche appositi strumenti.
Qui si trova un ottimo strumento: https://cubic-bezier.com/

CSS Animation

Inseriamo all'interno del nostro elemento l'animation-name, che riporta a una @keyframe a cui diamo istruzioni su come gestire il tempo di durata:

```
.div {animation-name: example;
   animation-duration: 8s;
   animation-delay: 4.5s;
}

@keyframes example {
 0%   {background-color:; left:; top:;} */fondamentale impostare il punto di partenza per evitare spiacevoli effetti scomparsa improvvisi/*
 25%  {background-color:; left:; top:; transform: rotatez() translateZ();}
 50%  {background-color:; left:; top:; transform: rotatez() translateZ();}
 75%  {background-color:; left:; top:; transform: rotatez() translateZ();}
 100% {background-color:; left:; top:;}
```

FONTI

https://medium.com/@janetbird/a-little-bit-of-css-perspective-e042d1c2539d

https://www.w3schools.com/

https://css-tricks.com/ease-out-in-ease-in-out/

https://3dtransforms.desandro.com/perspective