



Finanziato
dall'Unione europea
NextGenerationEU



Ministero dell'Istruzione
e del Merito



Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA



SOFTWARE DEVELOPER

Fondamenti di UX/UI Design e HTML CSS

Docente: Loredana Frontino

Titolo argomento: Scopriamo le proprietà CSS

Object fit

Fondamenti di UX/UI design e HTML CSS – Modulo 13

Object fit

La proprietà object fit definisce come un elemento risponde alle width e height del loro content box.

Viene utilizzata per immagini, video e altre tipologie di formati media.

Questa proprietà permette di definire come viene gestito l'adattamento dell'elemento multimediale al box che lo contiene.



Fondamenti di UX/UI design e HTML CSS – Modulo 13

Proprietà object-fit - valori

fill

contain

cover

none

scale-down

Fondamenti di UX/UI design e HTML CSS – Modulo 13

Proprietà object-fit - valori

fill

contain

cover

none

scale-down

Il contenuto dell'elemento viene ridimensionato per riempire completamente l'area di contenuto del container, effettuando una distorsione potenziale per poter riempire completamente il contenitore, perdendo le sue proporzioni originali.



Fondamenti di UX/UI design e HTML CSS – Modulo 13

Proprietà object-fit - valori

fill

Il contenuto dell'elemento viene ridimensionato mantenendo la stessa proporzione dell'elemento e si adatta agli elementi del content box.

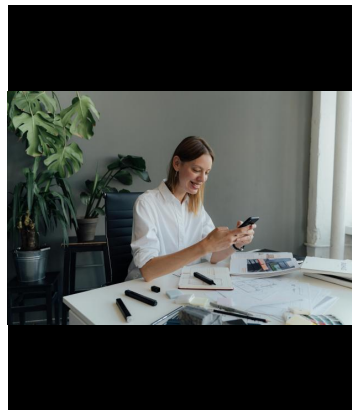
contain

Questo adattamento genera spesso dei margini nell'elemento, che possono essere verticali o orizzontali a seconda del rapporto tra elemento e content box.

cover

none

scale-down



Fondamenti di UX/UI design e HTML CSS – Modulo 13

Proprietà object-fit - valori

fill

Il contenuto dell'elemento viene ridimensionato mantenendo la stessa proporzione dell'elemento e si adatta agli elementi del content box, ma in questo caso l'adattamento può generare un taglio dell'immagine.

contain

cover

none

scale-down



Fondamenti di UX/UI design e HTML CSS – Modulo 13

Proprietà object-fit - valori

fill

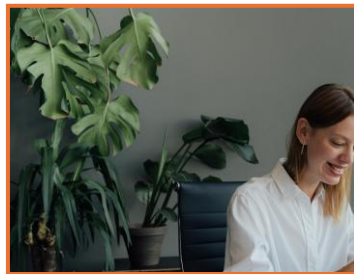
L'immagine ignora width e height dell'elemento padre e mantiene le sue dimensioni originali.

contain

cover

none

scale-down



Fondamenti di UX/UI design e HTML CSS – Modulo 13

Proprietà object-fit - valori

fill

Confronta le dimensioni intrinseche dell'elemento con le dimensioni del suo contenitore.

contain

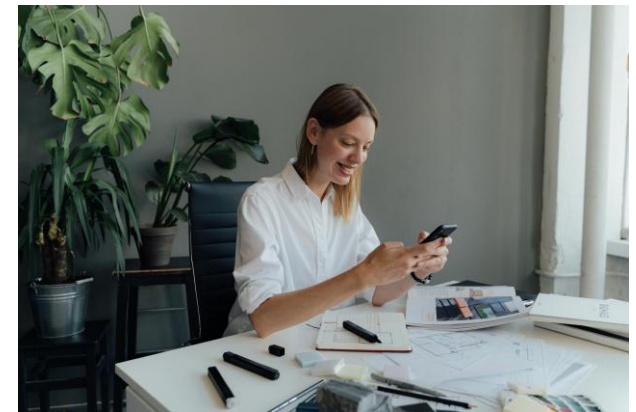
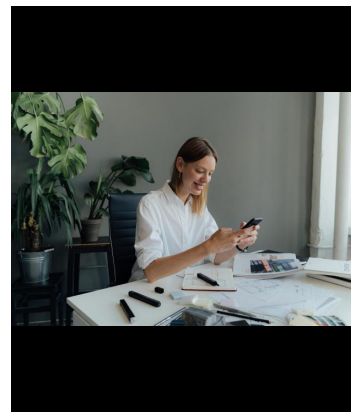
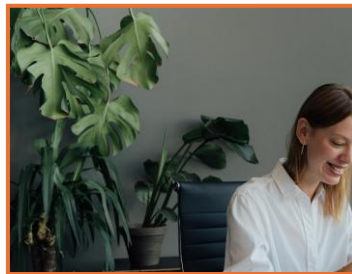
Poi applica le regole della proprietà none (se l'elemento è più piccolo del suo contenitore) oppure contain (se

cover

l'elemento è più grande del suo contenitore)

none

scale-down



Fondamenti di UX/UI design e HTML CSS – Modulo 13

Object fit

Ci sono anche altre proprietà associate che permettono di definire quale porzione dell'elemento prendere in considerazione quando si effettua l'adattamento dell'elemento al suo contenuto.

Object-position

Image-orientation

Image-rendering

Image-resolution

Fondamenti di UX/UI design e HTML CSS – Modulo 13

Esercitazioni

Esercizio 1

In un file HTML inserisci:

5 div con all'interno ognuno un'immagine in cui:

- L'immagine avrà una max-width definita in px; width e height impostate al 100%;
- I div invece devono essere disposti uno accanto all'altro, sfruttando le potenzialità del display flex;

Sperimentare con le proprietà object-fit:

- Impostare per ognuna delle immagini un tipo di object fit differente e osservarne il comportamento

Box shadow

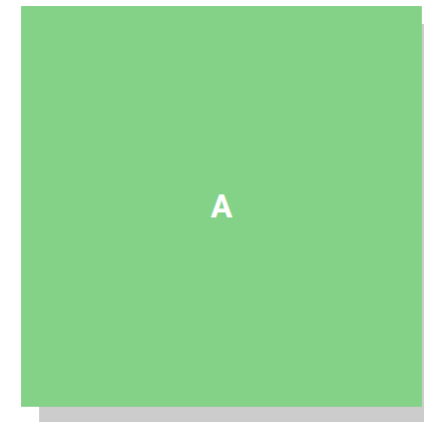
Fondamenti di UX/UI design e HTML CSS – Modulo 13

Proprietà box-shadow

Utilizzata per disegnare un effetto di ombra attorno a un elemento.

I parametri che definiscono la proprietà sono offset X e Y, sfocatura, raggio di diffusione e colore.

```
.shadow {  
  box-shadow: 9px 9px rgba(0, 0, 0, 0.2);  
}
```

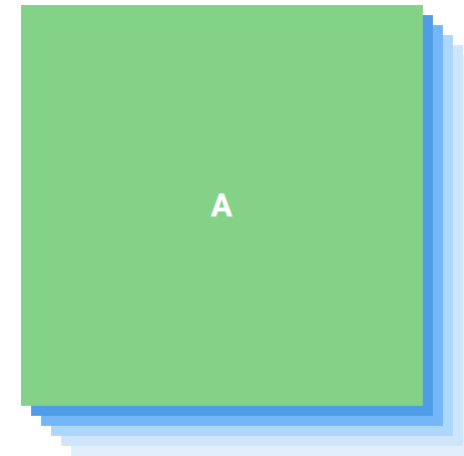


Fondamenti di UX/UI design e HTML CSS – Modulo 13

Proprietà box-shadow

Possono essere utilizzate anche più box-shadow
sullo stesso elemento contemporaneamente.

```
box-shadow: 5px 5px 0px 0px #289fed,  
10px 10px 0px 0px #5fb8ff,  
15px 15px 0px 0px #a1d8ff,  
20px 20px 0px 0px #cae6ff,  
25px 25px 0px 0px #e1eeff,  
5px 5px 15px 5px rgba(0, 0, 0, 0);
```



Fondamenti di UX/UI design e HTML CSS – Modulo 13

Esercitazioni

Esercizio 2

Partendo dal file usato per l'esercizio 1

Impostare alla prima immagine una box shadow con offset x e y di 3px e una sfocatura di 15px con un colore a piacere.

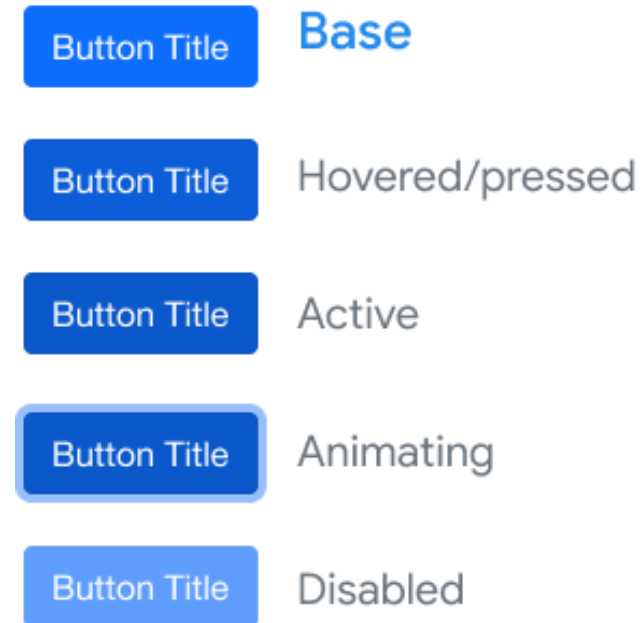
Transitions

Fondamenti di UX/UI design e HTML CSS – Modulo 13

Proprietà Transition

La proprietà transition CSS è una potente funzionalità CSS che consente di controllare i passaggi intermedi in una modifica della proprietà CSS, creando transizioni fluide e graduali tra i diversi stati di un element.

Viene comunemente utilizzato per migliorare l'esperienza dell'utente fornendo un feedback visivo quando un elemento cambia stato, ad esempio quando ci si passa sopra con il mouse, di da il focus o è active.



Fondamenti di UX/UI design e HTML CSS – Modulo 13

Proprietà Transition

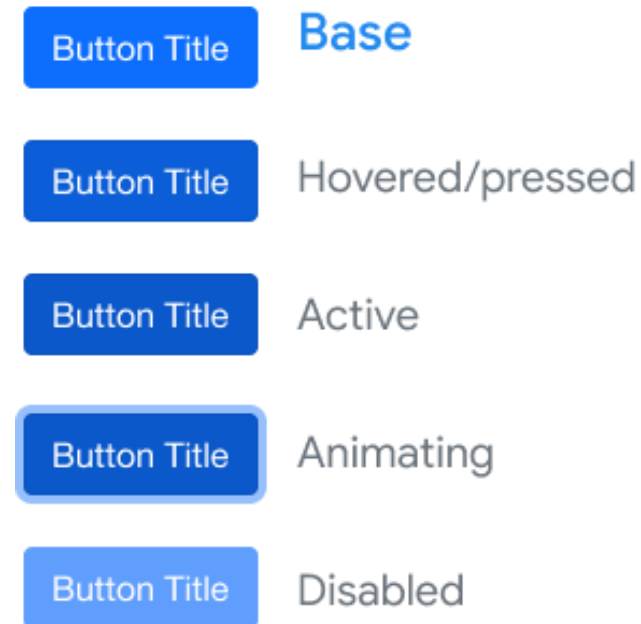
La proprietà transition è l'abbreviazione di 4 proprietà:

transition-property: la proprietà su cui verrà applicato l'effetto della transazione;

transition-duration: la durata della transazione;

transition-timing-function: definisce come vengono calcolati i valori intermedi della transazione (linear, ease-in, ease-out, etc.)

transition-delay: specifica il tempo di attesa prima che parta la transazione (in secondi o millisecondi)



Fondamenti di UX/UI design e HTML CSS – Modulo 13

Proprietà Transition

```
.transition-element{  
    width: 100px;  
    height: 100px;  
    background-color: blue;  
    transition: width 2s ease-in-out;  
}  
  
. transition-element:hover {  
    background-color: orange;  
    width: 200px;  
    height: 200px;  
}
```

Fondamenti di UX/UI design e HTML CSS – Modulo 13

Proprietà Transition - multipla

```
.transition-element{  
    width: 100px;  
    height: 100px;  
    background-color: blue;  
  
    transition: width 2s ease-in-out, height 2s ease-in-out;  
}  
  
. transition-element:hover {  
    background-color: orange;  
    width: 200px;  
    height: 200px;  
}
```

Fondamenti di UX/UI design e HTML CSS – Modulo 13

Proprietà Transition - multipla

Le transizioni multiple sono utili per creare animazioni complesse e migliorare la user experience.

Un eccessivo uso potrebbe però distrarre e ridurre le performance.

Fondamenti di UX/UI design e HTML CSS – Modulo 13

Esercitazioni

Esercizio 3

In un nuovo file HTML o partendo dal file usato per l'esercizio 1

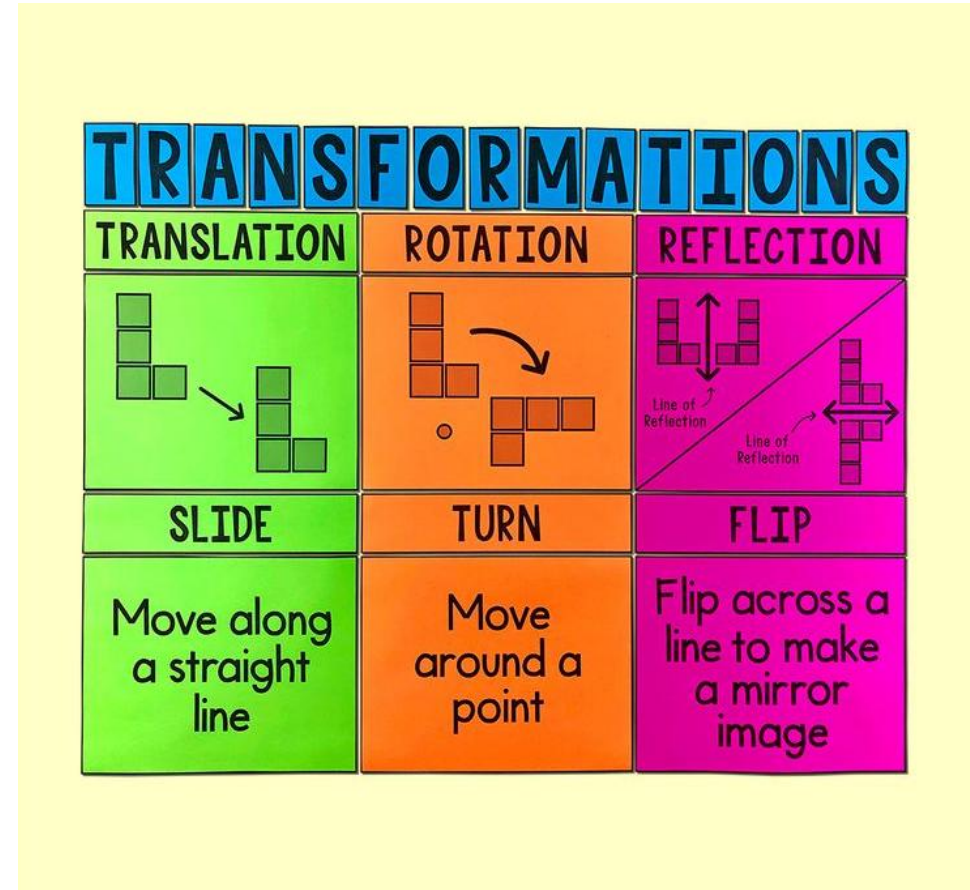
- Creare un box che all'hover si sposta da un punto A (0,0) a un punto B (0,50px) in modo graduale con una transazione ease;
- Creare un bottone che di base è blu, quando diventa active è rosso con una transazione ease-out, mentre quando è in hover diventa celeste con una transazione lineare di 0.4s .
- Crea poi un box colorato messo sopra a un elemento di testo "Testo a comparsa" che all'hover del box lo porta a una opacity di 0 con una durata della transazione di 3s.

Transformations

Fondamenti di UX/UI design e HTML CSS – Modulo 13

Proprietà transformation

La proprietà transformation consente di manipolare dinamicamente l'aspetto degli elementi della pagina web per consentire modifiche di posizione, dimensione e orientamento senza influire sul flusso del documento.



Fondamenti di UX/UI design e HTML CSS – Modulo 13

Proprietà transformation

La proprietà che viene applicata è transform a cui possono essere associati i seguenti valori:

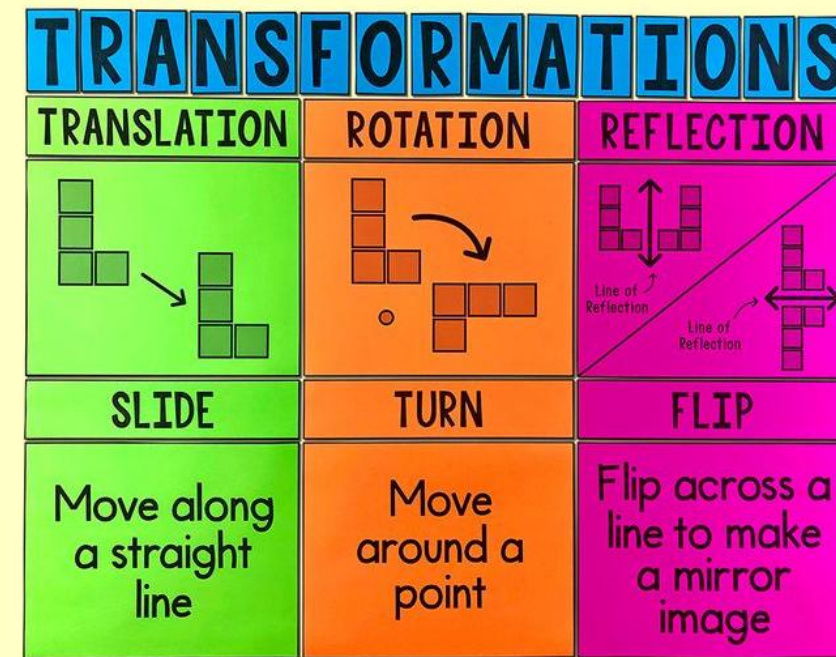
translate: sposta l'elemento in base agli assi x e y;

scale: modifica le dimensioni dell'elemento;

rotate: ruota l'elemento specificato un angolo;

skew: inclina un elemento lungo gli assi x e y.

Poi c'è la proprietà **transform-origin** che specifica il punto attorno al quale vengono applicate le trasformazioni.

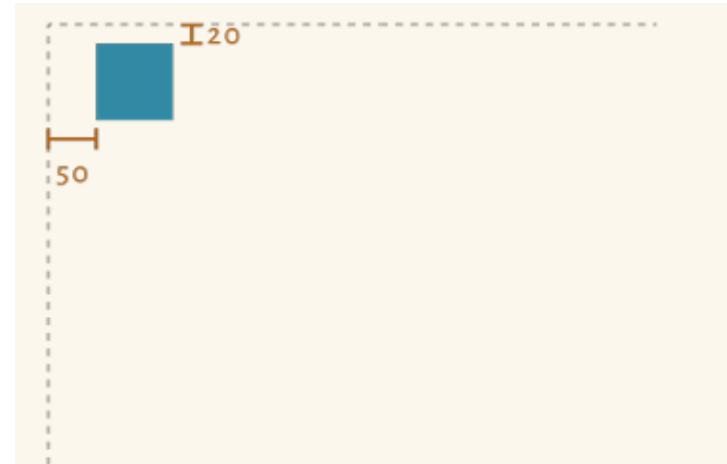


Fondamenti di UX/UI design e HTML CSS – Modulo 13

Proprietà transform: translate

Esempio

```
.transition-element{  
  width: 100px;  
  height: 100px;  
  background-color: blue;  
  transition: transform 2s ease-in-out;  
}  
  
.transition-element:hover {  
  transform: translate(50px, 20px);  
}
```

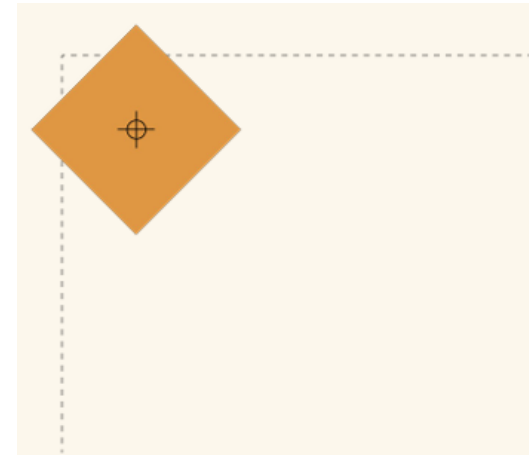


Fondamenti di UX/UI design e HTML CSS – Modulo 13

Proprietà transform: rotate

Esempio

```
.transition-element{  
  width: 100px;  
  height: 100px;  
  background-color: blue;  
  transition: transform 2s ease-in-out;  
}  
. transition-element:hover {  
  transform: rotate(45deg);  
}
```

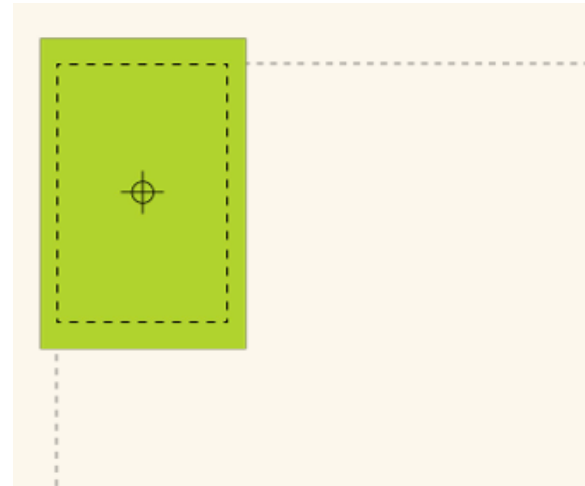


Fondamenti di UX/UI design e HTML CSS – Modulo 13

Proprietà transform: scale

Esempio

```
.transition-element{  
  width: 100px;  
  height: 100px;  
  background-color: blue;  
  transition: transform 2s ease-in-out;  
}  
  
. transition-element:hover {  
  transform: scale(1.4);  
}
```



Fondamenti di UX/UI design e HTML CSS – Modulo 13

Proprietà transform - multipla

Esempio

```
.transition-element{  
  width: 100px;  
  height: 100px;  
  background-color: blue;  
  transition: transform 2s ease-in-out;  
}  
  
. transition-element:hover {  
  transform: scale(1.4) translate(50px, 20px) rotate(45deg);  
}
```

Fondamenti di UX/UI design e HTML CSS – Modulo 13

Esercitazioni

Esercizio 4

Nel file HTML utilizzato per l'esercizio 3

- Creare un box 1 che effettua una rotazione di 45 gradi all'hover
- Creare un div che contiene un'immagine che viene zoomata di 1.5 quando si passa in hover
- Creare un box 2 con all'interno un element di testo centrato all'interno del box ed applicare un'inclinazione del box all'hover all'hover
- Creare un box come il primo dell'esercizio 3 ma applicando le proprietà di transformations e notare se ci sono differenze a livello visivo e a livello di codice

("Creare un box che all'hover si sposta da un punto A (0,0) a un punto B (0,50px) in modo graduale con una transazione ease");

Animations

Fondamenti di UX/UI design e HTML CSS – Modulo 13

Animazioni in CSS

Le animazioni CSS sono uno strumento potente per creare animazioni fluide e complesse direttamente in CSS, controllando ogni aspetto della loro esecuzione.

È fondamentale definire la sequenza di animazione tramite la regola `@keyframes` e poi richiamarla con `animation-name`.



Fondamenti di UX/UI design e HTML CSS – Modulo 13

@keyframe

Questa regola definisce la sequenza di fotogrammi (o "keyframe") che compongono l'animazione.

All'interno di @keyframes, si specifica lo stato dell'elemento in diversi momenti dell'animazione utilizzando percentuali (da 0% a 100%) o le parole chiave from (che equivale a 0%) e to (che equivale a 100%).

Per ogni fotogramma, si definiscono le proprietà CSS che si desidera animare e i loro valori in quel preciso momento.

```
@keyframes beat {  
  0% { transform: scale(1); }  
  25% { transform: scale(1.1); }  
  50% { transform: scale(0.9); }  
  100% { transform: scale(1); }  
}
```

```
.shape:hover {  
  animation: beat 2s infinite;  
}
```

Fondamenti di UX/UI design e HTML CSS – Modulo 13

animation

Questa proprietà (o le sue singole parti) applica l'animazione definita con @keyframes a un elemento HTML e ne controlla vari aspetti di esecuzione.

```
@keyframes beat {  
  0% { transform: scale(1); }  
  25% { transform: scale(1.1); }  
  50% { transform: scale(0.9); }  
  100% { transform: scale(1); }  
}  
  
.shape:hover {  
  animation: beat 2s infinite;  
}
```

Fondamenti di UX/UI design e HTML CSS – Modulo 13

Esercitazioni

Esercizio 5

Nel file HTML utilizzato per l'esercizio 3

- Creare un box A in cui verrà applicata un'animazione di rimbalzo (denominata bounce) in cui si definisce che deve passare da una posizione 0 a una posizione -80px sull'asse delle Y e fare in modo che l'animazione duri un secondo e si iteri in modo infinito con un ease-in-out
- *Creare un box B in cui applicare un'animazione di rotazione (Denominata rotate) in cui si definisce una rotazione da 0 a 360 gradi, con una durata di 3 secondi e che viene ripetuto 2 volte;*
- *Creare un box C in cui si genera un'animazione in cui si applica un'opacità che va da 1 (0) a 0.3 (50%) a 0.8 (100%) e allo stesso tempo viene applicato uno scale che parte da 1 e arriva a 1.4 (100%), della durata di 2 secondi e che viene ripetuto 5 volte*

Link utili

[Aspect-ratio](#)

[object-fit](#)

[Text-shadow](#)

[Box-shadow](#)

[Transition-property, transition-duration, transition-delay, transition-timing-function](#)

[Guida interattiva per le transizioni](#)

[CSS Animations](#)

[Creative CSS Animations](#)