

# JavaScript: modificare la visibilità degli elementi con `display:none`

La proprietà CSS `display` non può essere animata, quindi non è soggetta né alle transizioni né alle animazioni CSS. L'effetto che si ottiene è quello di un cambio di stato immediato, ad esempio tra lo stato `none` e `block`.

Questo cambio repentino di fatto annulla qualsiasi animazione o trasformazione che abbiamo impostato per ottenere un effetto simile al metodo `.show()` di jQuery. Ma c'è una soluzione.

Il seguente codice CSS ci offre un esempio di quello che abbiamo appena detto:

```
#element {
    width: 150px;
    height: 150px;
    background-color: green;
    transform: scale( 0 );
    transition: all 500ms linear;
    display: none;
}

#element.visible {
    transform: scale( 1 );
}
```

L'elemento è nascosto tramite `display: none` in quanto la trasformazione CSS annulla solo le sue proporzioni. Se modificassimo tale

dichiarazione applicando contemporaneamente la classe CSS `visible` perderemmo completamente l'effetto creato dalla transizione.

Quindi dobbiamo impostare un ritardo (timeout) prima di applicare la suddetta classe:

```
(function() {
    document.addEventListener(
    'DOMContentLoaded', function() {
        const show = document.querySelector(
    '#show' ),
        element =
    document.querySelector( '#element' );

        show.addEventListener(
    'click', function() {
            element.style.display
    = 'block';

            setTimeout(function() {
                element.className =
    'visible';

            }, 300);

        }, false);

    });

})();
```