

CSS Transform

CSS Demo: transform

RESET

transform: `matrix(1, 2, 3,`


transform: `translate(120px,`

transform: `scale(2, 0.5);`

transform: `rotate(0.5turn);`

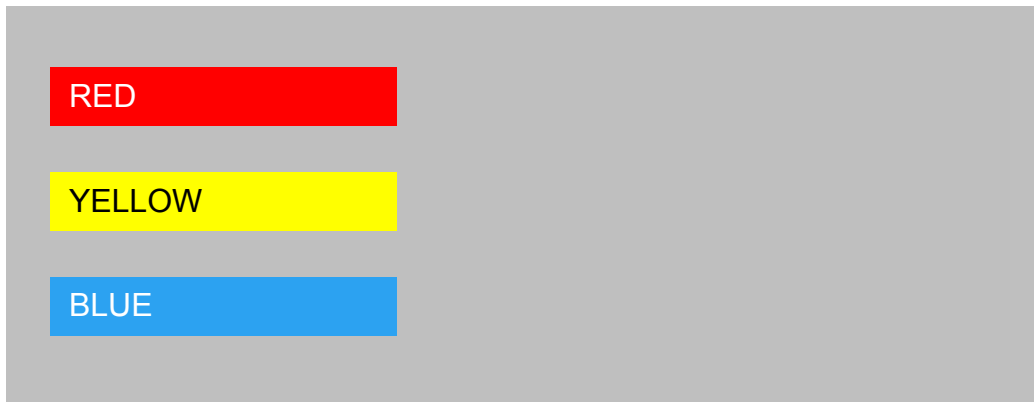
transform: `skew(30deg, 20deg)`

transform: `scale(0.5) trans`

A Firefox logo with a vibrant, multi-colored gradient (yellow, orange, red, blue) applied to its circular shape. The logo is positioned on the right side of the demo interface, representing the visual result of the CSS transform applied to the default state.

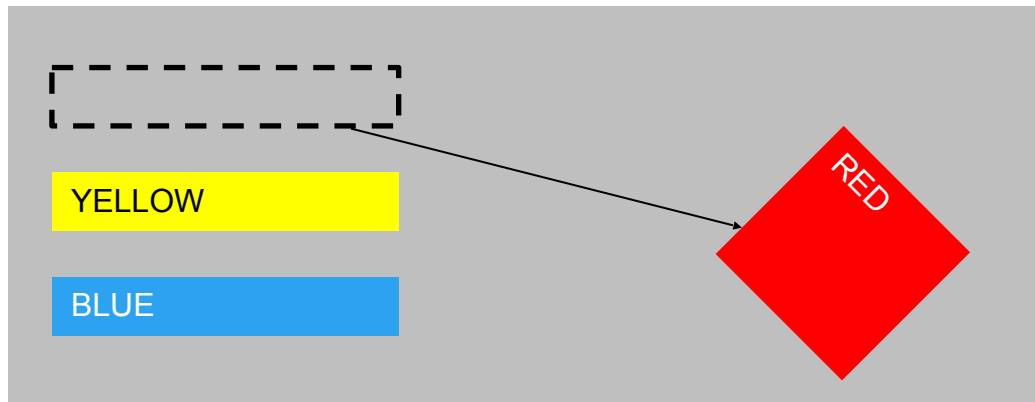
CSS transform - individual transform properties

Transformerer elementers placering, rotation og skalering med udgangspunkt i deres oprindelige placering, rotation og skalering



CSS transform - individual transform properties

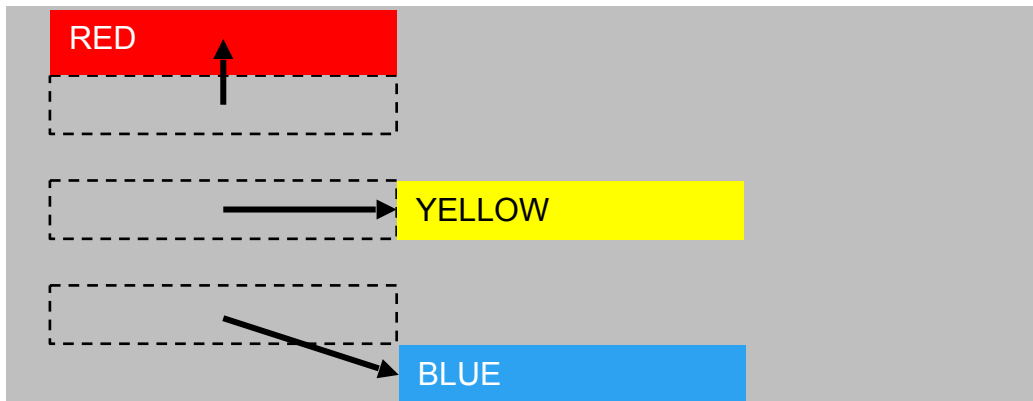
Transformerer elementers placering, rotation og skalering med udgangspunkt i deres oprindelige placering, rotation og skalering



- translate – flytter
- scale – skalerer
- rotate – roterer

Translate

translate flytter horisontalt og/eller vertikalt



```
#redbox {  
    translate: 0 -100%;  
}
```

Flytter boksen op, svarende til boksens egen højde

```
#yellowbox {  
    translate: 100% 0;  
}
```

Flytter boksen til højre, svarende til boksens egen bredde

```
#bluebox {  
    translate: 100% 100%;  
}
```

Flytter boksen til højre og ned

translate() - CSS: Cascading S

developer.mozilla.org/en-US/docs/Web/CSS/transform-function/translate

MDN Plus now available in your country! Support MDN and make it your own. Learn more

mdn web docs

References

Guides

MDN Plus

Theme

Already a subscriber?

Get MDN Plus

References > translate()

English (US)

In this article

Try it

Syntax

Examples

Specifications

Browser compatibility

See also

translate()

The `translate()` [CSS function](#) repositions an element in the horizontal and/or vertical directions. Its result is a `<transform-function>` data type.

Try it

CSS Demo: translate()


RESET

transform: `translate(0);`

transform: `translate(42px`

transform: `translate(-2.1i`

transform: `translate(3ch,`



CSS

Tutorials

CSS basics

▶ CSS first steps

▶ CSS building blocks

▶ Styling text

▶ CSS layout

scale

scale skalerer størrelse



```
#redbox {  
    scale: 1;  
}
```

skalering til "samme størrelse" = 100%

```
#yellowbox {  
    scale: 0.5 1;  
}
```

horizontal skalering til halv størrelse = 50%

```
#bluebox {  
    scale: 0.5 3;  
}
```

horizontal skalering til halv størrelse = 50%
vertikal skalering til 3x størrelsen = 300%

scale() - CSS: Cascading Style

developer.mozilla.org/en-US/docs/Web/CSS/transform-function/scale

mdn web docs

ReferencesGuidesMDN Plus

Theme

Already a subscriber?

Get MDN Plus

References > scale()

English (US)

In this article

Try it

Syntax

Accessibility concerns

Examples

Specifications

Browser compatibility

See also

scale()

The `scale()` [CSS function](#) defines a transformation that resizes an element on the 2D plane. Because the amount of scaling is defined by a vector, it can resize the horizontal and vertical dimensions at different scales. Its result is a [<transform-function>](#) data type.

Try it

CSS Demo: scale()


RESET

transform: scale(1);

transform: scale(0.7);

transform: scale(1.3, 0.4);

transform: scale(-0.5, 1)



CSS

Tutorials

CSS basics

▶ CSS first steps

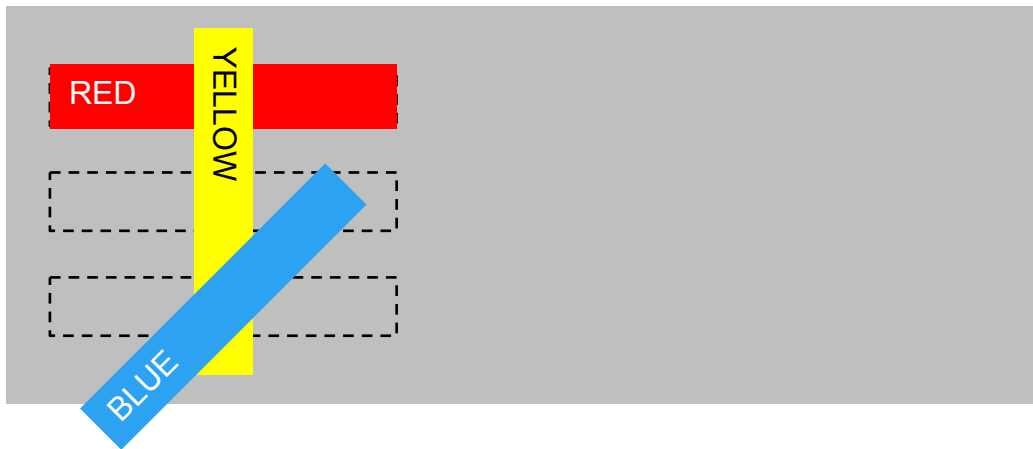
▶ CSS building blocks

▶ Styling text

▶ CSS layout

rotate

translate flytter horisontalt og/eller vertikalt



```
#redbox {  
    rotate: 0;  
}
```

Ingen rotation

```
#yellowbox {  
    rotate: 90deg;  
}
```

rotation 90 grader med uret

```
#bluebox {  
    rotate: -45deg;  
}
```

rotation 45 grader mod uret

rotate() - CSS: Cascading Styl

developer.mozilla.org/en-US/docs/Web/CSS/transform-function/rotate

mdn web docs

ReferencesGuidesMDN Plus

Theme

Already a subscriber?

Get MDN Plus

References > rotate()

English (US)

In this article

Try it

Syntax

Examples

Specifications

Browser compatibility

See also

rotate()

The `rotate()` [CSS function](#) defines a transformation that rotates an element around a fixed point on the 2D plane, without deforming it. Its result is a `<transform-function>` data type.

Try it

CSS Demo: rotate()


RESET

`transform: rotate(0);`

`transform: rotate(90deg);`

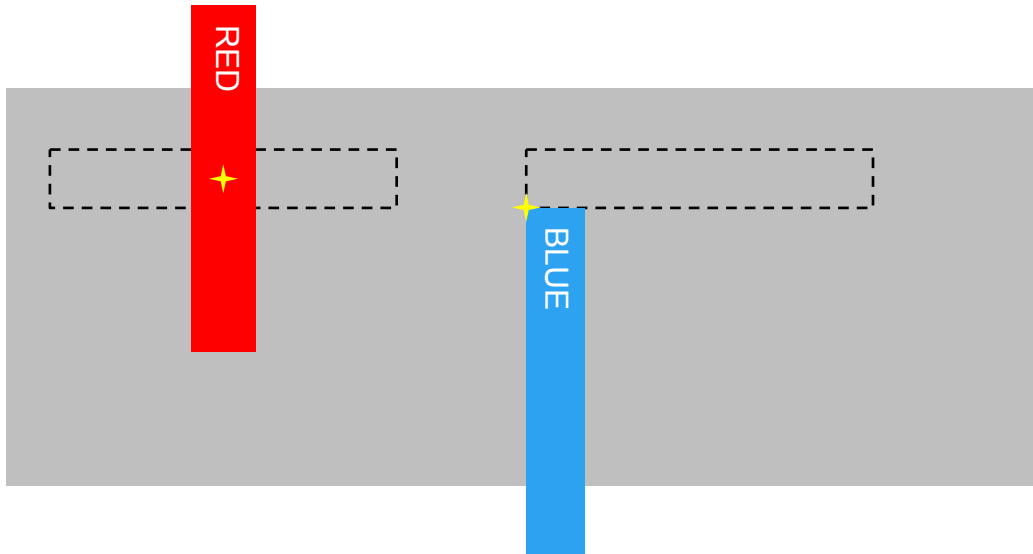
`transform: rotate(-0.25turn);`

`transform: rotate(3.142rad);`



transform-origin

Flytter omdrejningspunktet



```
#redbox {  
    rotate: 90deg;  
}
```

Omdrejningspunktet ligger default i midten

```
#bluebox {  
    transform-origin: 0 100%;  
    rotate: 90deg;  
}
```

Omdrejningspunktet er flyttet til nederste venstre hjørne

Kan også skrives:

```
#bluebox {  
    transform-origin: left  
    bottom;  
    rotate: 90deg;  
}
```

transform-origin - CSS: Cascad...

developer.mozilla.org/en-US/docs/Web/CSS/transform-origin

new transform in your country. Support more and make it your own. Learn more.

mdn web docsReferencesGuidesMDN PlusThemeSearchAlready a subscriber?Get MDN Plus

References > transform-originEnglish (US)

table-layout

▶ text-*

top

touch-action

▼ transform-*

transform

transform-box

transform-origin

transform-style

▶ transition-*

translate

unicode-bidi

user-select

vertical-align

visibility

white-space

widows

transform-origin

The `transform-origin` CSS property sets the origin for an element's transformations.

Try it

CSS Demo: transform-originRESET

transform-origin: center;

transform-origin: top left

transform-origin: 50px 50px

transform-origin: bottom left

Den ”gamle” måte

I vil sikkert støde på den i eksempler i finder på nettet, derfor nævner vi den her:

DEN NYE MÅDE:

```
#redbox {  
    translate: 0 -100%;  
}
```

```
#yellowbox {  
    scale: 0.5 1;  
}
```

```
#bluebox {  
    rotate: 90deg;  
}
```

DEN GAMLE MÅDE:

```
#redbox {  
    transform: translate(0 -100%);  
}
```

```
#yellowbox {  
    transform: scale(0.5 1);  
}
```

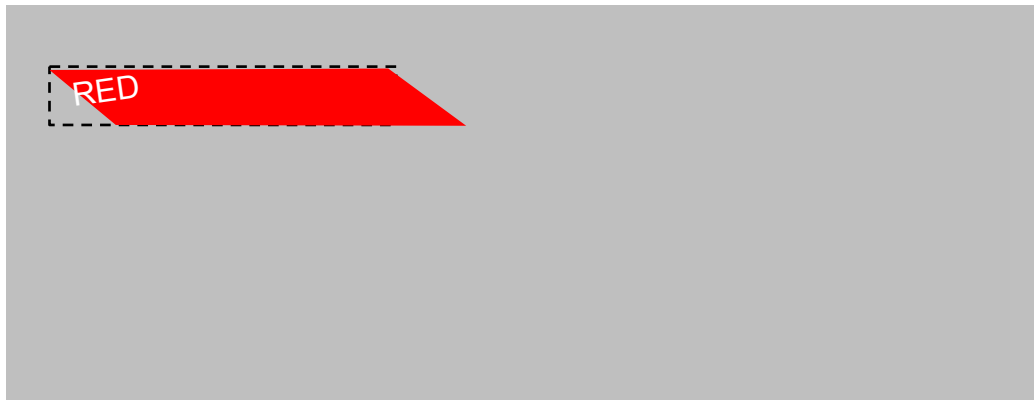
```
#bluebox {  
    transform: rotate(90deg);  
}
```

Den ”gamle” måde bruges stadig

Enkelte transforms har endnu ikke fået deres egen individuelle transform property, fx skew

DEN GAMLE MÅDE:

```
#redbox {  
  transform: skew(30deg);  
}
```



Læs mere her:

<https://web.dev/css-individual-transform-properties/>

<https://www.stefanjudis.com/blog/order-in-css-transformation-transform-functions-vs-individual-transforms/>