

# CSS Transform functions

Beyond box placement, more detailed visual controls

- Shift visuals X/Y
- Rotation
- Scale (zoom in/out)
- 3d interactions

# Translate (shift x/y coords)



Some Paragraph Text

```
.kitten {  
  transform: translate( 2rem, 5rem );  
}
```



Some Text

# Translate details

- Values can be all units, including percentages
- Negative values are fine (when meaningful)
- Creates a positioned element
  - like `position` other than `static`
- `translateX()` and `translateY()`
  - also `translateZ()`
- pass multiple to `transform` as space-separated

```
.some-selector {  
  transform: translate( 1rem, 5rem);  
}  
.other-selector { /* Same effect as above */  
  transform: translateX( 1rem ) translateY( 5rem );  
}
```

# Translate standalone property

`translate` was so popular in `transform`

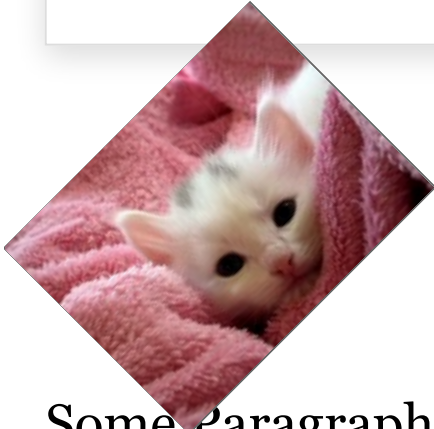
- It got a spin-off series!

```
.some-selector {  
  translate: 1rem 5rem;  
}  
.other-selector { /* Same effect as above */  
  transform: translate( 1rem, 5rem );  
}
```

- Standalone property
  - Makes mixing transforms easier
- No commas

# CSS Rotation

```
.kitten {  
  transform: rotate( -45deg );  
}
```



Some Paragraph Text

# Rotate Details

- Units are degrees, rads, grads, or turns
  - positive (clockwise-tilt) or negative
- `rotate()` is `rotateZ()`
  - exist: `rotateX()`, `rotateY()`, or `rotateZ()`
- Easier to work with one transform at a time
- inline vs block effects are exaggerated
  - rotating a block will still have full width

# Combining Transformations

```
img.kitten { /* Note block vs inline-block */  
  display: inline-block; /* Consider impacts of wrapping */  
}  
  
p.kitten-label {  
  display: inline-block;  
  color: red;  
  background-color: #FFFFFFF60;  
  border: 2px dashed red;  
  transform:  
    translateX(-173px)  
    translateY(-65px)  
    rotate(-45deg);  
}
```



# rotate also exists as standalone property

```
img.kitten { /* Note block vs inline-block */  
  display: inline-block; /* Consider impacts of wrapping */  
}  
  
p.kitten-label {  
  display: inline-block;  
  color: red;  
  background-color: #FFFFFFF60;  
  border: 2px dashed red;  
  translate: -173px -65px;  
  rotate: -45deg;  
}
```





# Standalone rotate options

To rotate on a different axis, list the axis first:

```
rotate: x 45deg;  
rotate: y 45deg;  
rotate: z 45deg;
```

# Scale

Resizes (zoom in/out)

- `scale()`
  - one arg: Scale both x and y by that multiplier
  - two arg: Scale X by first arg, Y by second

Notice that like all transformations, document flow is not impacted by visuals

# Kitten grows scales

```
#kitten-unscaled {  
  display: inline-block;  
}  
  
#kitten-scale-up {  
  display: inline-block;  
  transform: scale(2);  
}  
  
#kitten-scale-down {  
  display: inline-block;  
  transform: scale(0.5);  
}
```



# scale has a spin-off as a separate property

```
#kitten-unscaled {  
  display: inline-block;  
}  
  
#kitten-scale-up {  
  display: inline-block;  
/* transform: scale(2); */  
  scale: 2;  
}  
  
#kitten-scale-down {  
  display: inline-block;  
/* transform: scale(0.5); */  
  scale: 0.5;  
}
```

# Rarely used transformations

I've only introduced the common ones

- `rotate3d()`
- `scale3d()`
- `translate3d()`
- `matrix()`
- `matrix3d()`
- `perspective()`