Transitions

- Permite modelar a mudança de propriedades entre os elementos, tornando-os animados.

Alterando o tamanho da

Alterando a posição da

Transitions

- Sintaxe:
  - transition: **nome_evento | duração | efeito**
  - transition: background-color 1s ease-in
- Efeitos: ease-in, ease-out e ease-in-out
  - ease-in: começa devagar e acelera até o fim
  - ease-out: o oposto, começa rápido e desacelera até o fim
  - ease-in-out: começa devagar, acelera no meio e reduz no fim

Transitions

- Exemplo simples

```
<a class="target">Passe o mouse aqui !!!</a>
```

```
.target {
  font-size: 14px;
  transition: font-size 4s 1s;
}

.target:hover {
  font-size: 36px;
}
```

Transitions

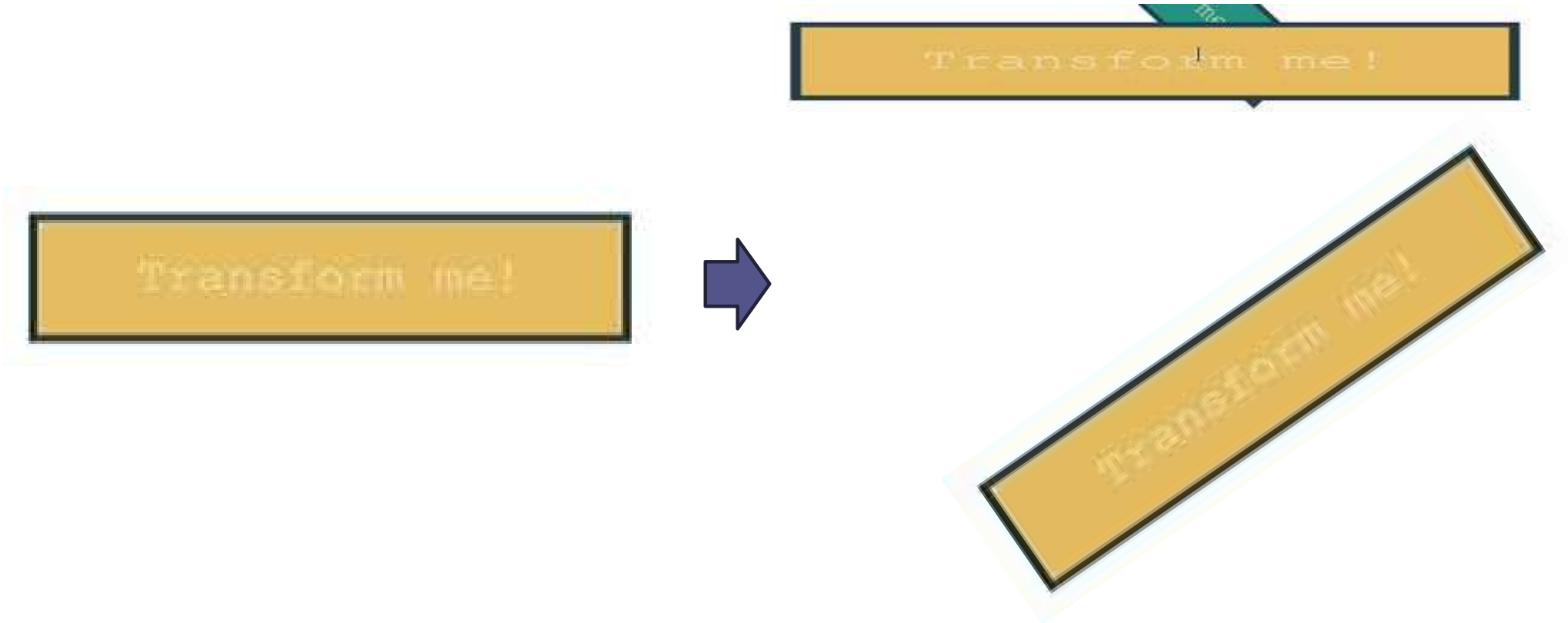
- Mostrar os códigos: 3.css-new-material\2.Other_Properties\Transition

Transitions

- Exemplos de transições: https://easings.net/

Transformations

- O comando *transform* permite você girar, dimensionar, inclinar ou posicionar um elemento.



Transformations

- Propriedade
  - rotate()

Rotaciona o elemento

```
div {
  border: solid red;
  transform: translate(30px, 20px) rotate(20deg);
  width: 140px;
  height: 60px;
}
```

Move o elemento

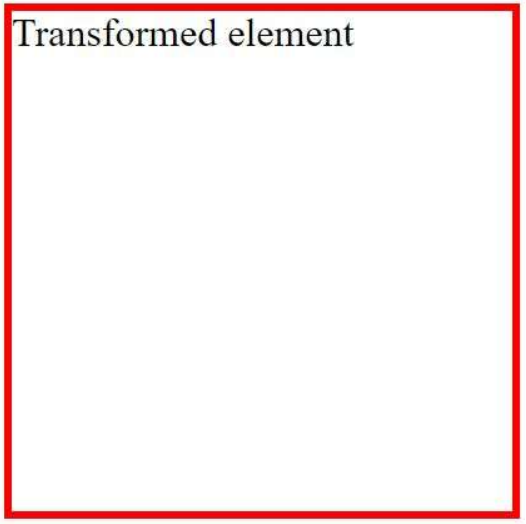
```
<div>Transformed element</div>
```

Transformations

- Propriedade
  - scale()
    - Permite aumentar ou diminuir o elemento

```
div {
  border: solid red;
  width: 200px;
  height: 200px;
}

#transformed{
  border: solid red;
  transform: translate(200px, 200px) scale(2);
  width: 200px;
  height: 200px;
}
```

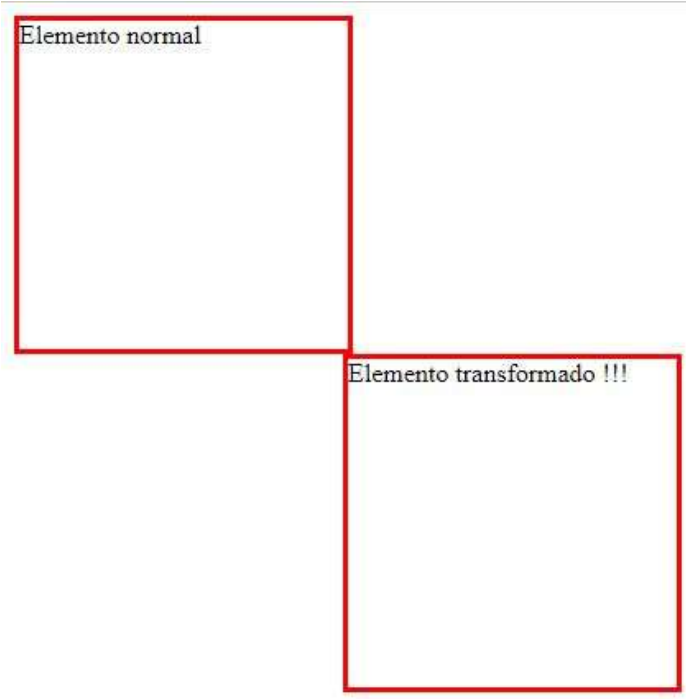
```
<div>Elemento tamanho normal</div>

<div id="transformed">Transformed element</div>
```



Transformations

- Propriedade
  - translate(x,y), translateX(x), translateY(y)
    - Permite movimentar um elemento

```
div {
  border: solid red;
  width: 200px;
  height: 200px;
}

#transformed{
  border: solid red;
  transform: translate(200px, 200px);
  width: 200px;
  height: 200px;
}
```



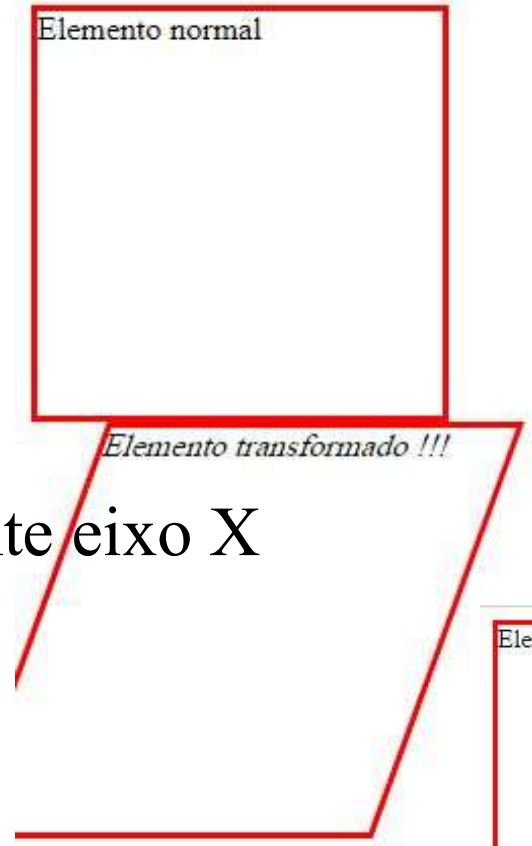

```
<div>Elemento normal</div>

<div id="transformed">Elemento transformado !!!</div>
```

Transformations

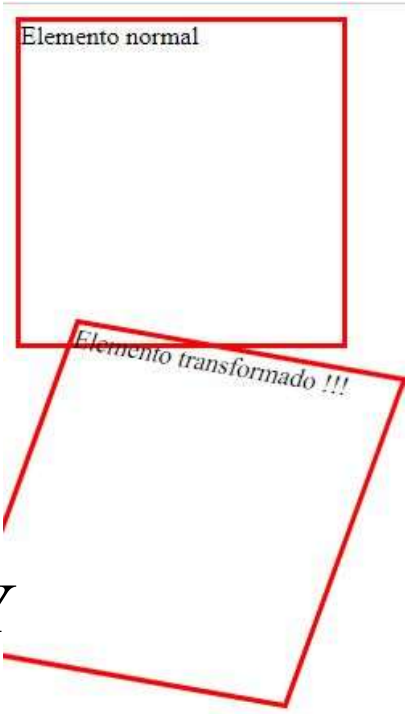
- Propriedade
  - skew(grausX, grausY)
    - Permite inclinar o elemento

```
div {
  border: solid red;
  width: 200px;
  height: 200px;
}

#transformed{
  border: solid red;
  transform: skew(-20deg);
  width: 200px;
  height: 200px;
}
```

```
<div>Elemento normal</div>

<div id="transformed">Elemento transformado !!!</div>
```



Somente eixo X



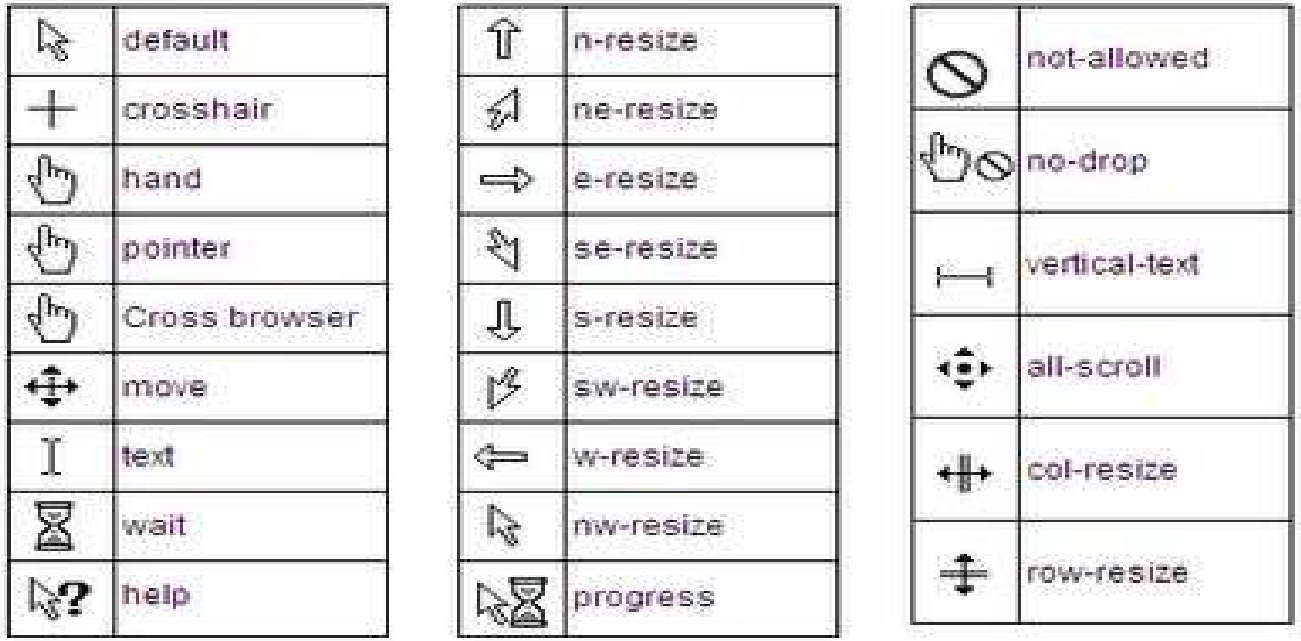
Eixos X e Y

CSS: cursor

- Propriedade
  - cursor: pointer;

  - Obs.: Também é possível criar cursores personalizados!

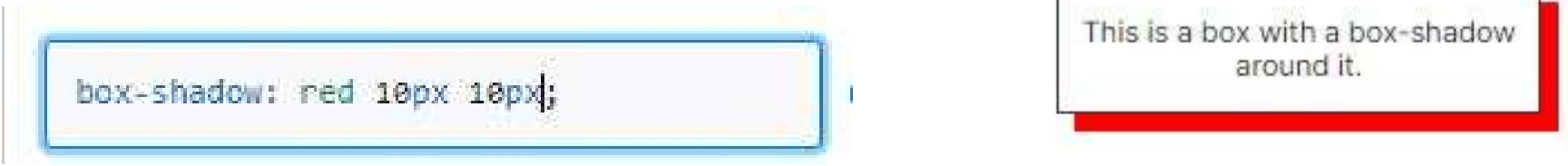
https://developer.mozilla.org/en-US/docs/Web/CSS/cursor



CSS: cursor

- Propriedade
  - box-shadow: COR TAM1 TAM2;
  - Também permite a inclusão de mais 2 tamanhos, TAM3 e TAM4, representando efeito borrado e disperso.

https://developer.mozilla.org/en-US/docs/Web/CSS/box-shadow



Que tal juntarmos tudo isso e fazermos um efeito bem legal?

- Quais componentes CSS Podemos utilizar?

```css
body {
  font-family: 'Roboto', sans-serif;
  display: flex;
  align-items: center;
  justify-content: center;
  height: 100vh;
  background-color: #151b29;
}

button {
  background: none;
  color:#ffa260;
  border: 2px solid;
  padding: 1em 2em;
  font-size: 1em;
  transition: color 0.25s, border-color 0.25s, box-shadow
0.25s,transform 0.25s;
}

button:hover {
  border-color: #f1ff5c;
  color: white;
  box-shadow: 0 0.5em 0.5em -0.4em #f1ff5c;
  transform: translateY(-0.9em);
  cursor: pointer;
}
```

```html
<body>
    <button>Enviar dados</button>
</body>
```

Botão estilizado

- HTML
  - <button>Enviar dados</button>
- CSS

```
body {
  font-family: 'Roboto',
sans-serif;
  display: flex;
  align-items: center;
  justify-content: center;
  height: 100vh;
  background-color:
#151b29;
}
```

```
button {
  background: none;
  color:#ffa260;
  border: 2px solid;
  padding: 1em 2em;
  font-size: 1em;
  transition: color 0.25s, border-color 0.25s,
box-shadow 0.25s,transform 0.25s;
}

 button:hover {
   border-color: #f1ff5c;
   color: white;
   box-shadow: 0 0.5em 0.5em -0.4em #f1ff5c;
   transform: translateY(-0.9em);
   cursor: pointer;
 }
```

Background-image

- É possível definir mais de uma imagem no background

```
    background-image:  url("circle.png"),
url("star.png");
    background-position: top left, bottom right;
    background-repeat: no-repeat, no-repeat;
```

Only Background

```
background: url(star.png)
repeat-x;
```


```
background: url(star.png)
repeat-y;
```

Only Background

Preencha todo o espaço (ou 70% dele)

```
background: center/cover url("star.png");
background: center/70% url("star.png");
```

float

- Faz com que os elementos flutuem na página de acordo com determinada política
- O mesmo design pode ser obtido usando elementos posicionados de forma absoluta ou float.
  - Posicionamento float é uma solução flexível para layouts multi-colunas e permite que elementos “limpem” elementos ao seu lado.
- Funciona para o elemento onde for declarada.
- Box retirado da sua posição normal do fluxo do documento.
- O elemento seguinte “ocupa a vaga” onde estava o elemento que está usando float.

float

- Um elemento float normalmente deve ter uma largura (width) específica declarada com um valor diferente de ‘auto’.
- Valores possíveis:
  - left
    - O elemento flutua para a esquerda.
  - right
    - O elemento flutua para a direita
  - none (padrão)
    - O elemento não flutua e será exibido de acordo com o fluxo normal do navegador.
  - inherit
    - O valor deve ser herdado do elemento pai.

float

- Exemplo:

```
p {
  float: left;
}
```

Flutuar para a esquerda.
Deixa elementos um ao lado do outro.

float

- Exemplo:

```
img {
   float: right;
}
```

Flutuar a imagem para a direita dentro do elemento onde ela está inserida.

```
<div id='mydiv'>
<img id='minhaimg' src='https://www.freecodecamp.org/news/content/images/2019/10/Ekran-Resmi-2019-10-04-23.09.24.png' width="400" height="200">
</div>
```

```
#mydiv {
    width: 600px;
    height: 400px;
    border: 3px solid black;
}

#minhaimg {
    float: right;
}
```

float

```
<span class='destaque’>
meu outro exemplo de CSS
</span>
```

```
span.destaque {
    width: 40%;
    /* tamanho da caixa span */
    float: right;
    background-color: #888;
    border: 1px solid red;
}
```

clear

- Especifica que nenhum elemento float possa ficar à esquerda ou direita (ou ambos) de um elemento em bloco.
- O elemento com essa propriedade será movido para baixo até se livrar do elemento em bloco ao seu lado.

clear

- Valores possíveis:
  - left
    - Não permite elementos float do lado esquerdo.
  - right
    - Não permite elementos float do lado direito.
  - both
    - Não permite elementos float em nenhum lado (esquerdo ou direito).
  - none (padrão)
    - permite elementos float em ambos os lados (esquerdo e direito)
  - inherit
    - Valor herdado do elemento pai.

```
<!DOCTYPE html>
<html>
<head>
    <style>
        img {
            float: left;
        }

        p.clear {
            clear: both;
        }
    </style>
</head>
<body>

    <h1>The clear Property</h1>

    <img src="w3css.gif" width="100" height="132">
    <p>This is some text. This is some text. This is some text. This is some text. Th
is is some text. This is some text.
    </p>
    <p class="clear">This is also some text. This is also some text. This is also som
e text. This is also some text. This is also some text. This is also some text.</p>
    <p><strong>Remove the "clear" class to see the effect.</strong></p>

</body>
</html>
```

Antes vs Depois: clear

The clear Property

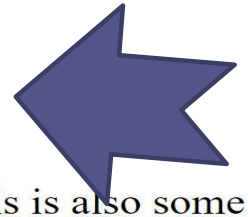
This is some text. This is some text. This is some text. This is some text. This is some text. This is some text.

This is also some text. This is also some text. This is also some text. This is also some text. This is also some text. This is also some text.

Remove the "clear" class to see the effect.

The clear Property

This is some text. This is some text. This is some text. This is some text. This is some text. This is some text.



This is also some text. This is also some text. This is also some text. This is also some text. This is also some text. This is also some text.

Remove the "clear" class to see the effect.

Layout

- Dividir a página em áreas usando o elemento div ou section do HTML.
- Cada área deve ter um nome bem representativo e não vinculado a informações sobre o estilo ou posicionamento a ser aplicado.
  - O estilo e o posicionamento podem ser modificados e o nome dado pode perder o sentido original.
  - Exemplos de “bons” nomes:
    - geral, container, menu, conteudo, destaque, rodape, footer, blogs, topo, centro, center, etc.

Layout

- Estruturar as áreas que esperamos no nosso layout.

```
<div id="container">
  <div id="header"></div>
  <div id="left"></div>
  <div id="content"></div>
  <div id="right"></div>
  <div id="footer"></div>
</div>
```

Classificação dos layouts

- Número de colunas
- Largura da página
  - Fixa
    - A largura é fixa e não se expande ou se encolhe de acordo com a janela do navegador.
      - Melhor controle do design.
  - Líquida
    - O conteúdo da página se expande para se ajustar à pagina de acordo com as dimensões da janela do navegador.
  - Elástica
    - A largura é fixa, mas as margens se expandem ou se encolhem de acordo com a janela do navegador.
    - Normalmente posiciona o conteúdo no centro da página.

Layout de largura fixa

```
#container {
  border: 1px solid black;
  width: 500px;
  background-color: rgb(255,108, 180)
}
```

Mesmo que você redimensione o tamanho da janela, será de 500px o elemento!

Layout de largura líquida

```
#container {
   border: 1px solid black;
   width: 80%;
   min-width: 300px;
}
```

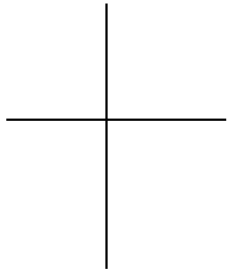
Vai ocupar 80% do espaço reservado ao elemento.

Layout de largura elástica

```
/* largura elastica */

* {
    font-size: 15px;
}

#container {
    border: 1px solid black;
    width: 30em;
    /* de acordo com o tamanho da fonte */
}
```

Centralização do layout

```
#container {
    border: 3px solid black;
    width: 50%;
    margin: 20px;
}
```



```
#container {
  border: 1 px solid black;
  width: 500px;
  margin: auto;
}
```

- **margin: 25px 50px 75px 100px;**
- top margin is 25px
- right margin is 50px
- bottom margin is 75px
- left margin is 100px

margin: auto centraliza os elementos de forma horizontal

Propriedades adicionais

- min-width
  - Define a largura mínima do elemento.
- min-height
  - Define a altura mínima do elemento.

```
div {
    max-width: 500px;
    height: 100px;
    background-color: powderblue;
}
```

```
<h2>Set the max-width of an element</h2>
<p>This div element has a height of 100px and a max-width of
 500px:</p>
<div></div>
<p>Resize the browser window to see the effect.</p>
```

- Encerramento CSS - Aula 4