



PROFESSOR: Daniel Caetano
DISCIPLINA: CCT0735 – Tecnologia para Internet I

MATERIAL DO PLANO DE AULA ESTÁCIO - AULA 09

FONTE: Plano de Aula – Estácio

4.7 Transições

Normalmente, quando o valor de um CSS é alterado, o resultado da renderização é instantâneo, esta regra permite atualizar uma propriedade suavemente do estado antigo para o novo estado.

Exemplo de utilização:

Esta regra define a mudança para o valor ?opacity? em um período de 2 segundos.

```
div {  
    transition-property: opacity;  
    transition-duration: 2s;  
}
```

Esta regra define que a cor do elemento por default é azul. Quando o mouse for posicionado sobre o elemento, ele mudará sua cor para verde em um período de tempo de 2 segundos.

```
li {  
    transition: background-color linear 1s;  
    background: blue;  
}  
li:hover {  
    background-color: green;  
    transition-duration: 2s; /* applies to the transition *to* the :hover  
state */  
}
```

4.8 Overflow

Geralmente o conteúdo de uma caixa pode não estar completamente alinhando com as dimensões da mesma. Veja o exemplo:

```
<html>
<head>
<style>
div { width : 100px; height: 100px;
border: thin solid red;
}
blockquote { width : 125px; height : 100px;
margin-top: 50px; margin-left: 50px;
border: thin dashed black
}
cite { display: block;
text-align : right;
border: none
}
</style>
</head>
<div>
<blockquote>
```

O elemento `blockquote` ultrapassa os limites do elemento `div`. No entanto, é permitido que o conteúdo do elemento `blockquote` seja visualizado, pois o valor inicial da regra **overflow** é 'visible'. Para manter o elemento `blockquote` dentro da `div` basta alterar a regra do `css` da `div` para **overflow:hidden**. Veja o exemplo abaixo:

```
<html>
<head>
<style>
div { width : 100px; height: 100px;
border: thin solid red;
```

```
overflow: hidden;  
  
}  
  
blockquote { width : 125px; height : 100px;  
margin-top: 50px; margin-left: 50px;  
border: thin dashed black  
  
}  
  
cite { display: block;  
text-align : right;  
border: none  
  
}  
  
</style>  
  
</head>  
  
<div>  
  
<blockquote>
```

5.4 Transformações

Este módulo (não normativo) permite alterar as coordenadas de um Box dentro do HTML.

Coordenadas:

X - Eixo horizontal para a direita

Y - Eixo vertical para baixo

Z - Eixo perpendicular em direção a tela.

Propriedades:

transform-origin: Controla a origem da transformação.

perspective: Controla a perspectiva em um modelo tridimensional criando a ilusão de profundidade.

perspective-origin: define o ponto de origem da propriedade perspective.

transform-style: Controla a renderização dos elementos filhos no espaço 3D.

Backface-visibility: Define a visibilidade de um elemento caso um outro elemento seja colocado atrás.

Funções previstas para transformações:

- perspective()
- rotate()
- rotateX()
- rotateY()
- rotateZ()
- rotate3d()
- translate()
- translateX()
- translateY()
- translateZ()
- translate3d()
- scale()
- scaleX()
- scaleY()
- scaleZ()
- scale3d()
- skew()
- skewX()
- skewY()
- matrix()
- matrix3d()

onde:

- perspective() - tem o mesmo efeito de profundidade da propriedade perspective
- rotate() - Rotaciona elementos
- translate() - Desloca um Box no plano xy
- scale() - Expande ou contrai um Box
- skew() - Inclina um Box
- matrix() - Função 'resumida' que permite aplicar uma ou várias funções anteriores

Exemplos:**Posicionamento**

```
<html>
<head>
<style type="text/css">
div {
background-color: #71ABC4;
height: 200px;
width: 300px;
transform: translate(100px, 100px); /* posiciona o box na posicao
100,100 */
}
</style>
</head>
<body>
<div>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
</p>
</div>
```

```
</body>
```

```
</html>
```

Rotação

```
<html>
```

```
<head>
```

```
<style type="text/css">
```

```
div {
```

```
background-color: #71ABC4;
```

```
height: 200px;
```

```
width: 300px;
```

```
transform-origin: 50px 250px; /* define ponto de origem */
```

```
transform: rotate(45deg); /* rotaciona 45graus */
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div>
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
```

```
</p>
```

```
</div>
```

```
</body>
```

```
</html>
```

Rotação, translação e escala

```
<html>
```

```
<head>
```

```
<style type="text/css">
```

```
div {
background-color: #71ABC4;
height: 200px;
width: 300px;
height: 100px; width: 100px;
transform: translate(80px, 80px) scale(1.5, 1.5) rotate(45deg);
}
</style>
</head>
<body>
<div>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
</p>
</div>
</body>
</html>
```

Perspectiva

```
<html>
<head>
<style type="text/css">
div {
background-color: #71ABC4;
height: 200px;
width: 300px;
transform: translate(100px, 100px); /* posiciona o box na posicao
100,100 */
```

```
}
```

```
.container {
```

```
background-color: #fff;
```

```
perspective: 500px;
```

```
border: 1px solid black;
```

```
}
```

```
.transformed {
```

```
transform: rotateY(50deg);
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="container">
```

```
<div class="transformed">
```

```
  Lorem ipsum dolor sit amet, consectetur adipiscing elit.
```

```
</div>
```

```
</div>
```

```
</body>
```

```
</html>
```

Perspectiva com Preserve-3d

```
<html>
```



```
<style>
div {
height: 150px;
width: 150px;
}
.container {
perspective: 500px;
border: 1px solid black;
}
.transformed {
transform-style: preserve-3d;
transform: rotateY(50deg);
background-color: blue;
}
.child {
transform-origin: top left;
transform: rotateX(40deg);
background-color: lime;
}
</style>
<div class="container">
<div class="transformed">
<div class="child"></div>
</div>
</div>
</html>
```

Interceção

```
<html>

<style>

div {

height: 150px;

width: 150px;

}

.container {

background-color: rgba(0, 0, 0, 0.3);

transform-style: preserve-3d;

perspective: 500px;

}

.container > div {

position: absolute;

left: 0;

}

.container > :first-child {

transform: rotateY(45deg);

background-color: orange;

top: 10px;

height: 135px;

}

.container > :last-child {

transform: translateZ(40px);

background-color: rgba(0, 0, 255, 0.75);

top: 50px;
```

```
height: 100px;
}
</style>

<div class="container">
<div>a</div>
<div>b</div>
</div>
</html>
```

5.5 Animações

Este módulo permite a mudança de valores das propriedades de modo que a mudança do estado do elemento seja realizada de forma suave em um espaço de tempo especificado.

Regra @keyframes

Permite a mudança gradual de um valor a outro em um tempo específico.

Exemplo 1:

Permite a mudança de do estado da cor vermelha para a amarela, ato término volta para o estado original.

```
<html>
<head>
<style>
div {
width: 100px;
height: 100px;
background-color: red;
-webkit-animation-name: exemplo; /* Chrome, Safari, Opera */
-webkit-animation-duration: 4s; /* Chrome, Safari, Opera */
```

```
animation-name: exemplo;

animation-duration: 4s;

}

/* Chrome, Safari, Opera */
@-webkit-keyframes exemplo {
  from {background-color: red;}
  to {background-color: yellow;}
}

/* Standard syntax */
@keyframes exemplo {
  from {background-color: red;} /* cor de origem */
  to {background-color: yellow;} /* para cor de destino */
}

</style>

</head>

<body>

<div></div>

</body>

</html>
```

Exemplo 2

Outra forma de mudar o estado da propriedade.

```
<html>

<head>

<style>

div {

width: 100px;
```

```
height: 100px;

background-color: red;

-webkit-animation-name: exemplo; /* Chrome, Safari, Opera */
-webkit-animation-duration: 4s; /* Chrome, Safari, Opera */

animation-name: exemplo;

animation-duration: 4s;

}

/* Chrome, Safari, Opera */
@-webkit-keyframes exemplo {

0% {background-color: red;}

25% {background-color: yellow;}

50% {background-color: blue;}

100% {background-color: green;}

}

/* Standard syntax */
@keyframes exemplo {

0% {background-color: red;}

25% {background-color: yellow;}

50% {background-color: blue;}

100% {background-color: green;}

}

</style>

</head>

<body>

<div></div>

</body>
```

```
</html>
```

Exemplo 3

Movimentação de elementos

```
<html>
```

```
<head>
```

```
<style>
```

```
div {
```

```
width: 100px;
```

```
height: 100px;
```

```
background-color: red;
```

```
position: relative; /* adicionado para troca de posicao */
```

```
-webkit-animation-name: exemplo; /* Chrome, Safari, Opera */
```

```
-webkit-animation-duration: 4s; /* Chrome, Safari, Opera */
```

```
animation-name: exemplo;
```

```
animation-duration: 4s;
```

```
}
```

```
/* Chrome, Safari, Opera */
```

```
@-webkit-keyframes exemplo {
```

```
0% {background-color:red; left:0px; top:0px;}
```

```
25% {background-color:yellow; left:200px; top:0px;}
```

```
50% {background-color:blue; left:200px; top:200px;}
```

```
75% {background-color:green; left:0px; top:200px;}
```

```
100% {background-color:red; left:0px; top:0px;}
```

```
}
```

```
/* Standard syntax */
```

```
@keyframes exemplo {
```

```
0% {background-color:red; left:0px; top:0px;}
25% {background-color:yellow; left:200px; top:0px;}
50% {background-color:blue; left:200px; top:200px;}
75% {background-color:green; left:0px; top:200px;}
100% {background-color:red; left:0px; top:0px;}
}
</style>
</head>
<body>
<div></div>
</body>
</html>
```

Exemplo 4

Loop com 3 interações

```
<html>
<head>
<style>
div {
width: 100px;
height: 100px;
background-color: red;
position: relative; /* adicionado para troca de posicao */
-webkit-animation-name: exemplo; /* Chrome, Safari, Opera */
-webkit-animation-duration: 4s; /* Chrome, Safari, Opera */
animation-name: exemplo;
animation-duration: 4s;
```

```
animation-iteration-count: 3;

}

/* Chrome, Safari, Opera */
@-webkit-keyframes exemplo {
0% {background-color:red; left:0px; top:0px;}
25% {background-color:yellow; left:200px; top:0px;}
50% {background-color:blue; left:200px; top:200px;}
75% {background-color:green; left:0px; top:200px;}
100% {background-color:red; left:0px; top:0px;}
}

/* Standard syntax */
@keyframes exemplo {
0% {background-color:red; left:0px; top:0px;}
25% {background-color:yellow; left:200px; top:0px;}
50% {background-color:blue; left:200px; top:200px;}
75% {background-color:green; left:0px; top:200px;}
100% {background-color:red; left:0px; top:0px;}
}

</style>

</head>

<body>

<div></div>

</body>

</html>
```


FONTE: Plano de Aula – Estácio

Adaptado por: Daniel Caetano