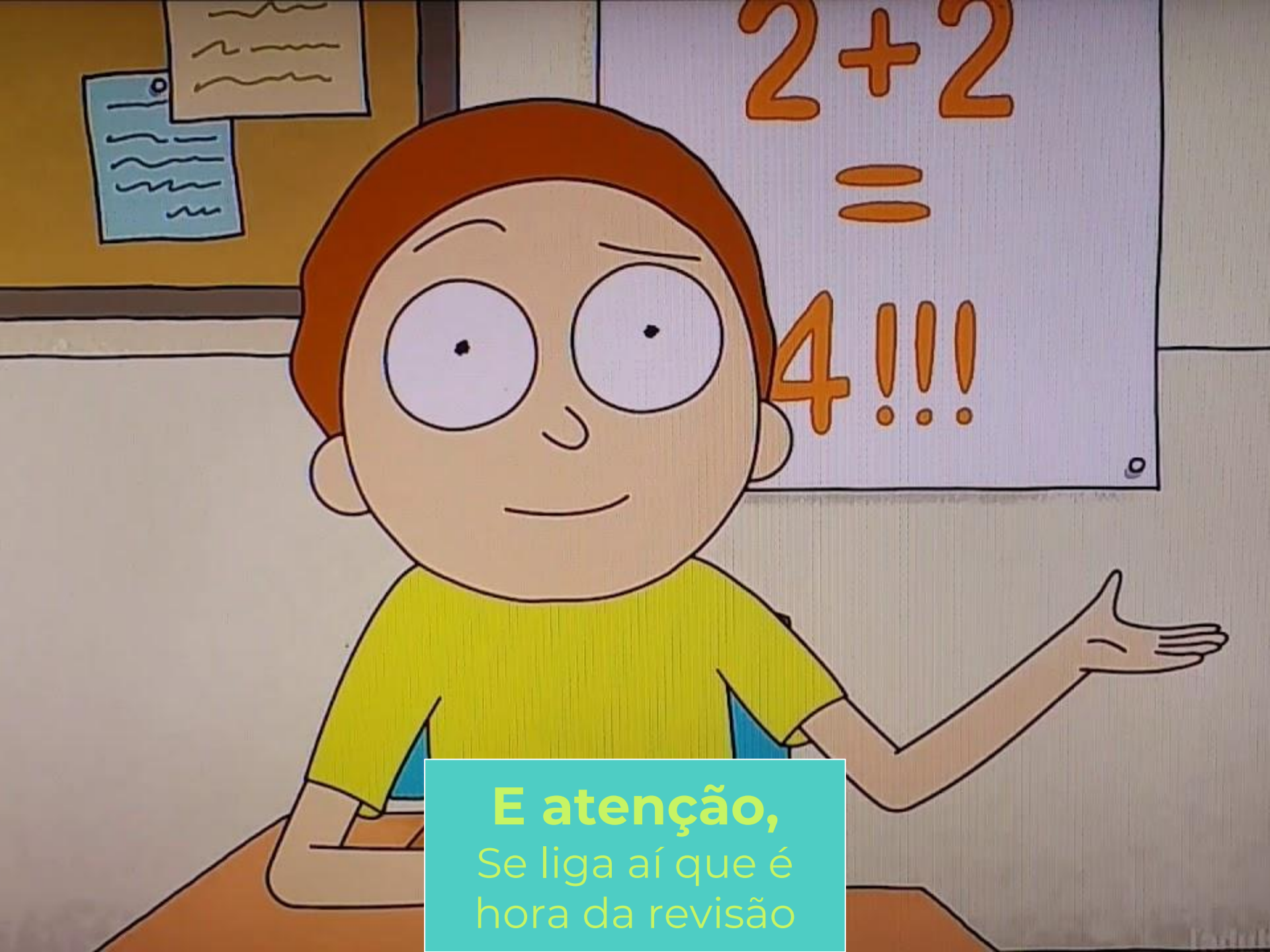


Olá!



E atenção,
Se liga aí que é
hora da revisão

Lembrando...

1. Hover, Opacity
2. Border
 - a. Border-width, border-color, border-radius, border shorthand,
3. Box Shadow
4. Position
 - a. Static, relative, absolute, sticky, fixed
5. Z-index
6. Visibilidade
 - a. Display:none x visibility: hidden
7. Float, Box-sizing

CSS

Aula 07

1.

CSS Gradients

Começaremos a ver a utilização de transição suave de cores no CSS.


Gradients



É possível, com CSS3, colocar gradientes nas cores utilizadas, partindo de uma cor a outra. Existem duas formas de realizar isso:

- Linear gradients: pra cima, baixo, esquerda, direita, diagonal
- Radial gradients: definido a partir do seu centro

Linear Gradients



Gradientes lineares são construídos indicando duas cores (ao qual será aplicada a transição gradativa). É possível definir também um ângulo, ponto de partida e direção, por exemplo.

Linear Gradients - direção



Por padrão, o gradiente irá do topo para o inferior de onde está se aplicando a cor. Outros valores podem ser dados para mudar esse direcionamento:

- To left
- To right
- To bottom
- To bottom right (diagonal)

Linear Gradients - direção



```
#grad {  
  background: red; /* For browsers that do not support gradients */  
  background: -webkit-linear-gradient(left, red , yellow); /* For Safari 5.1 to 6.0 */  
  background: -o-linear-gradient(right, red, yellow); /* For Opera 11.1 to 12.0 */  
  background: -moz-linear-gradient(right, red, yellow); /* For Firefox 3.6 to 15 */  
  background: linear-gradient(to right, red , yellow); /* Standard syntax */  
}
```

Linear Gradients - outros parâmetros



Para os gradientes, também podem ser usados outros parâmetros, como por exemplo:

- Angulação
- Múltiplas cores
- Transparência

Linear Gradients - outros parâmetros



■ Angulação

```
#grad {  
  background: red; /* For browsers that do not support gradients */  
  background: -webkit-linear-gradient(-90deg, red, yellow); /* For Safari 5.1 to 6.0 */  
  background: -o-linear-gradient(-90deg, red, yellow); /* For Opera 11.1 to 12.0 */  
  background: -moz-linear-gradient(-90deg, red, yellow); /* For Firefox 3.6 to 15 */  
  background: linear-gradient(-90deg, red, yellow); /* Standard syntax */  
}
```

Linear Gradients - outros parâmetros



- Múltiplas cores

```
#grad {  
  background: red; /* For browsers that do not support gradients */  
  background: -webkit-linear-gradient(red, yellow, green); /* For Safari 5.1 to 6.0 */  
  background: -o-linear-gradient(red, yellow, green); /* For Opera 11.1 to 12.0 */  
  background: -moz-linear-gradient(red, yellow, green); /* For Firefox 3.6 to 15 */  
  background: linear-gradient(red, yellow, green); /* Standard syntax */  
}
```

Linear Gradients - outros parâmetros



■ Transparência

```
#grad {  
  background: red; /* For browsers that do not support gradients */  
  background: -webkit-linear-gradient(left, rgba(255,0,0,0), rgba(255,0,0,1)); /*Safari 5.1-6*/  
  background: -o-linear-gradient(right, rgba(255,0,0,0), rgba(255,0,0,1)); /*Opera 11.1-12*/  
  background: -moz-linear-gradient(right, rgba(255,0,0,0), rgba(255,0,0,1)); /*Fx 3.6-15*/  
  background: linear-gradient(to right, rgba(255,0,0,0), rgba(255,0,0,1)); /*Standard*/  
}
```

Linear Gradients - repetição



Gradientes lineares podem ser repetidos.

```
#grad {  
  background: red; /* For browsers that do not support gradients */  
  /* Safari 5.1 to 6.0 */  
  background: -webkit-repeating-linear-gradient(red, yellow 10%, green 20%);  
  /* Opera 11.1 to 12.0 */  
  background: -o-repeating-linear-gradient(red, yellow 10%, green 20%);  
  /* Firefox 3.6 to 15 */  
  background: -moz-repeating-linear-gradient(red, yellow 10%, green 20%);  
  /* Standard syntax */  
  background: repeating-linear-gradient(red, yellow 10%, green 20%);  
}
```

Radial Gradients



São gradientes definidos a partir do centro do elemento.
Podem ser aplicados limitadores de cores e múltiplas cores.

```
#grad {  
  background: red; /* For browsers that do not support gradients */  
  background: -webkit-radial-gradient(red 5%, yellow 15%, green 60%); /* Safari 5.1-6.0 */  
  background: -o-radial-gradient(red 5%, yellow 15%, green 60%); /* For Opera 11.6-12.0 */  
  background: -moz-radial-gradient(red 5%, yellow 15%, green 60%); /* For Firefox 3.6-15 */  
  background: radial-gradient(red 5%, yellow 15%, green 60%); /* Standard syntax */  
}
```

Radial Gradients - formato



Também podem ser adicionados formatos ao gradiente.

```
#grad {  
  background: red; /* For browsers that do not support gradients */  
  background: -webkit-radial-gradient(circle, red, yellow, green); /* Safari */  
  background: -o-radial-gradient(circle, red, yellow, green); /* Opera 11.6 to 12.0 */  
  background: -moz-radial-gradient(circle, red, yellow, green); /* Firefox 3.6 to 15 */  
  background: radial-gradient(circle, red, yellow, green); /* Standard syntax */  
}
```


Radial Gradients - formato



Também podem ser feitas repetições para este tipo de gradiente.

```
#grad {  
  background: red; /* For browsers that do not support gradients */  
  /* For Safari 5.1 to 6.0 */  
  background: -webkit-repeating-radial-gradient(red, yellow 10%, green 15%);  
  /* For Opera 11.6 to 12.0 */  
  background: -o-repeating-radial-gradient(red, yellow 10%, green 15%);  
  /* For Firefox 3.6 to 15 */  
  background: -moz-repeating-radial-gradient(red, yellow 10%, green 15%);  
  /* Standard syntax */  
  background: repeating-radial-gradient(red, yellow 10%, green 15%);  
}
```



Exercício 1

1. Usando o Codepen, crie quatro divs (aplique margens para distanciar uma da outra) e faça o seguinte:
 - a. Defina um estilo de borda, cor, além de colocar uma altura para cada div.
 - b. Para cada uma das divs, aplique um background com diferentes tipos de gradientes

2.

Transições CSS

Iremos aprender a fazer transições com elementos no CSS.

Transitions



Transições em CSS3 nos permite alterar valores de propriedades durante um determinado período, com determinados efeitos.

Precisa ser definido o valor da duração (pois o padrão é 0) e a propriedade CSS ao qual deseja-se adicionar o efeito.

```
-webkit-transition: width 2s; /* Safari */  
transition: width 2s;
```

Transition delay

Define um atraso para o início da transição aplicada.

```
div {  
    -webkit-transition-delay: 1s; /* Safari */  
    transition-delay: 1s;  
}
```

Transition timing function

Define a variação de aceleração da transição.

```
/* For Safari 3.1 to 6.0 */
#div1 {-webkit-transition-timing-function: linear;}
#div2 {-webkit-transition-timing-function: ease;}
#div3 {-webkit-transition-timing-function: ease-in;}
#div4 {-webkit-transition-timing-function: ease-out;}
#div5 {-webkit-transition-timing-function: ease-in-out;}
```

```
/* Standard syntax */
#div1 {transition-timing-function: linear;}
#div2 {transition-timing-function: ease;}
#div3 {transition-timing-function: ease-in;}
#div4 {transition-timing-function: ease-out;}
#div5 {transition-timing-function: ease-in-out;}
```

Transition shorthand



É possível definir todas as propriedades vistas usando o formato shorthand.

```
div {  
  transition: width 2s linear 1s;  
}
```




Exercício 2

1. Usando o Codepen, crie um quadrado, que terá um background determinado.
2. Ao passar o mouse, esse quadrado deverá triplicar de tamanho e sua cor de fundo deverá apresentar uma transparência.
3. A transição deverá ocorrer em dois segundos.
4. O início da transição deverá acontecer após 2s.

3.

Transformações 2D

Iremos aprender a
fazer
transformações
2D com
elementos no
CSS.

Transform



A propriedade transform consegue operar transformações no elemento com CSS. Existem os seguintes tipos de transformações.

- `translate()`
- `rotate()`
- `skewX()`
- `skewY()`
- `matrix()`
- `scale()`

Transform - translate



O translate faz com que o elemento se desloque de acordo com o eixo x e y passado.

```
div {  
  -ms-transform: translate(50px, 100px); /* IE 9 */  
  -webkit-transform: translate(50px, 100px); /* Safari */  
  transform: translate(50px, 100px);  
}
```

Transform - scale



O scale faz com que o elemento seja redimensionado, de acordo com os parâmetros dados para altura e largura.

```
div {  
    -ms-transform: scale(2, 3); /* IE 9 */  
    -webkit-transform: scale(2, 3); /* Safari */  
    transform: scale(2, 3);  
}
```

Transform - skewX e skewY

O skew inclina um elemento de acordo com o grau dado, a partir do eixo x ou eixo y. Ambos podem ser escritos de uma única vez com skew(xdeg, ydeg);

```
div {  
    -ms-transform: skewX(20deg); /* IE 9 */  
    -webkit-transform: skewX(20deg); /* Safari */  
    transform: skewX(20deg);  
}
```

Transform - matrix



O matrix combina todas as transformações em um único método.

```
matrix(scaleX (), skewY (), skewX (), scaleY (), translateX (), translateY ())
```




Exercício 3

1. No codePen, crie um quadrado, que deve possuir uma cor de fundo
2. Ao passar o mouse sobre o quadrado, o mesmo deverá rotacionar em 180 graus, em uma transição de 2s.

4.

Transformações 3D

Iremos aprender a fazer transformações 3D com elementos no CSS.

Transform



A propriedade transform também consegue operar transformações 3D no elemento com CSS. A seguir, alguns dos exemplos de tipos de transformações 3D.

- rotateX
- rotateY
- rotateZ

Transform - rotate



O rotate faz rotações 3D a partir dos eixos X, Y e Z, como podemos ver no exemplo abaixo:

```
div {  
  -webkit-transform: rotateX(150deg); /* Safari */  
  transform: rotateX(150deg);  
}
```

```
div {  
  -webkit-transform: rotateY(130deg); /* Safari */  
  transform: rotateY(130deg);  
}
```

```
div {  
  -webkit-transform: rotateZ(90deg); /* Safari */  
  transform: rotateZ(90deg);  
}
```

Transform - perspective



Definir o ponto de vista de onde o elemento é visto.

A propriedade perspectiva define quantos pixels um elemento 3D é colocado a partir da visão.

Esta propriedade permite-lhe mudar a perspectiva sobre a forma como elementos 3D são vistos.

Ao definir a propriedade perspectiva de um elemento, são os elementos filho que recebem a vista em perspectiva, não o próprio elemento..

Perspective

```
div {  
    -webkit-perspective: 500px; /* Chrome, Safari, Opera */  
    perspective: 500px;  
}
```

Backface-visibility



Esconder a parte traseira de um elemento, quando girado

```
div {  
  -webkit-backface-visibility: hidden; /* Chrome, Safari, Opera */  
  backface-visibility: hidden;  
}
```


5.

Animações CSS

Iremos aprender a fazer animações com elementos no CSS.

CSS Animations



Uma animação permite que um elemento mudar gradualmente a partir de um estilo para outro.

Você pode alterar o máximo de propriedades CSS que você quer, como muitas vezes quiser.

Para usar animação CSS3, você deve primeiro especificar alguns quadros-chave para a animação.

Keyframes definem quais estilos o elemento terá em determinados momentos.

CSS Animations - keyframes



Ao especificar estilos CSS dentro da @keyframes, a animação irá gradualmente mudar a partir do estilo atual para o novo estilo em determinados momentos.

Para se ter uma animação para o trabalho, você deve ligar a animação a um elemento.

```
@keyframes example {  
  from {background-color: red;}  
  to {background-color: yellow;}  
}  
  
div {  
  width: 100px;  
  height: 100px;  
  background-color: red;  
  animation-name: example;  
  animation-duration: 4s;  
}
```

CSS Animations - keyframes



Também é possível usar porcentagem. Usando porcentagem, você pode adicionar tantas mudanças de estilo como queira.

```
@keyframes example {  
  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;}  
}  
  
div {  
  width: 100px;  
  height: 100px;  
  position: relative;  
  background-color: red;  
  animation-name: example;  
  animation-duration: 4s;  
}
```

CSS Animations - propriedades

- Animation-name: define o nome da animação (referencia o keyframe)
- Animation-duration: duração da animação
- Animation-delay: tempo de espera para iniciar a animação
- Animation-iteration-count: define quantas vezes a animação será executada (pode ser infinito com o valor "infinite")
- Animation-direction: define se a animação será executada no sentido inverso, por exemplo.

CSS Animations - shorthand



Pode ser usado shorthand para a animation, definindo suas propriedades em uma única propriedade CSS.

```
div {  
    animation: example 5s linear 2s infinite  
    alternate;  
}
```



Desafio

1. No codePen, crie uma div cujo background é uma imagem de um campo de futebol
2. Crie uma outra div, dentro da primeira, e adicione um estilo de forma que ela fique circular (será a bola).
3. Faça uma animação que deverá:
 - a. Ser infinita
 - b. Trocará gradativamente a cor da bola, a cada vez que ela se movimenta
 - c. Deverá se movimentar apenas por um dos lados do campo, sem sair dos limites do mesmo.



Dúvidas?



github.com/drcabral



twitter.com/DrCabrales



diogo.cabral.dev@gmail.com