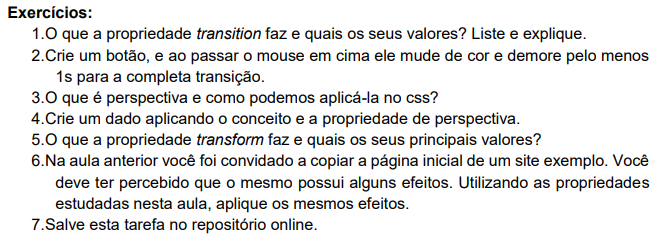
Atividade 10



Q1) A propriedade “transition” é uma abreviação para “transition-property”, “transition-duration”, “transition-timing-function” e transition-delay.

Por exemplo, podemos escrever:

Transition: margin-right 2s ease-in-out 0.5s;

Nesse caso, quem passará por uma transição será a margin-direita, a transição terá uma duração de 2s, ease-in-out implica que a transição será mais lenta no início, acelera e por fim reduz novamente no final e a animação começará após um atraso de 0,5s.

Outro exemplo: transition: margin-right 2s, color 1s; duas propriedades passarão por uma transição: margin, numa duração de 2s e a cor do texto 1s.

Ela permite definir como será a transição entre dois estados de um elemento. É possível definir diferentes estados usando pseudo-classes como :hover ou :active ou dinamicamente por meio do JavaScript.

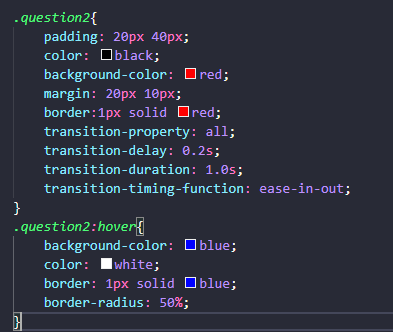
Transition-delay-> Especifica o tempo que deve ser esperado até iniciar o efeito de transição quando um valor de estado muda.

Transition-duration-> Determina o tempo de duração da animação de transição, por padrão esse valor é 0s, isto é, não ocorre animação, ao adicionarmos um valor positivo, passa a ocorrer uma animação de transição. Quando mais de uma característica deve passar pela mesma transição, podemos usar o valor “all” no lugar de apontar o tempo de cada transição de propriedade.

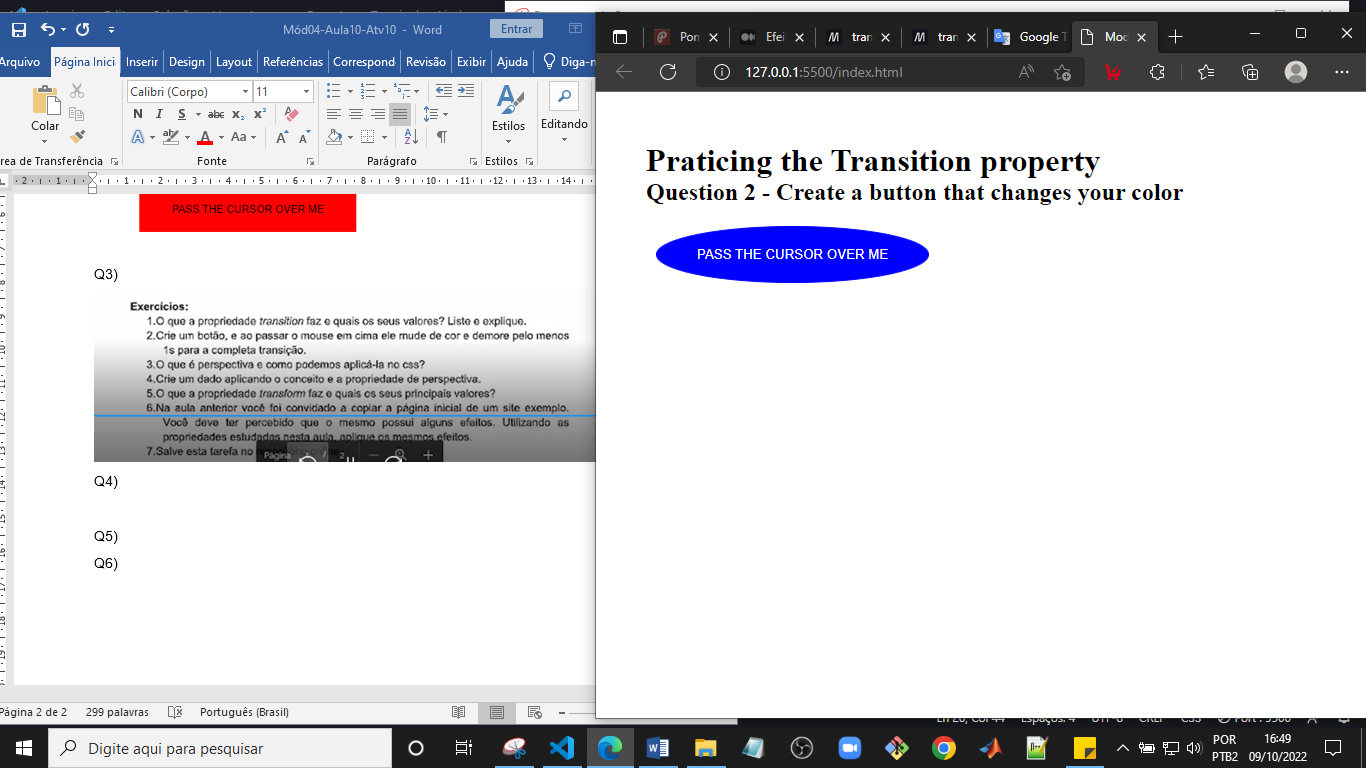
Transition-property-> Define a propriedade que passará por um efeito de transição.

Transition-timing-function -> Define como os valores intermediários são calculados para as propriedades do CSS sendo afetadas por um efeito de transição. Por exemplo, o valor “linear” faz a animação de transição ocorrer de forma linear. É comum fazer uso da cubic-bezier, que permite gerar transições bem interessantes, como acelerar no começo e no final, mas uma transição gradual no meio tempo. O valor “ease” equivale a cubic-bezier(0.25,0.1,0.25,1.0) , aumenta a velocidade em direção ao meio da transição e desacelera para o final.

Q2)

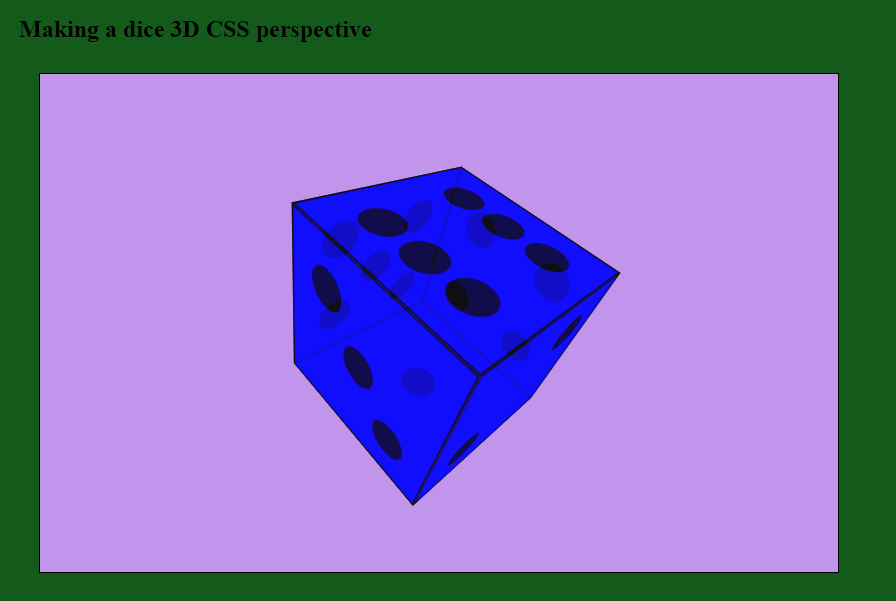






Q3) A propriedade perspective determina a distância entre o plano z = 0 e o usuário com o objetivo de dar a um elemento 3D um pouco de perspectiva. Exemplo de uso: perspective:800px; O valor default é “none”.

Q4) Usando perspective, transform e @keyframes



Q5) A propriedade “transform” permite modificar o espaço coordenado do modelo de formatação CSS. Usando ele é possível deslocar elementos, rotacionar, inclinar, aplicar fator de escala, entre outras funcionalidades. Seus principais valores são: matrix(), translate(),scale(), rotate(), skew(). É possível aplicar uma combinação desses, além disso, é comum fazer animações combinando isso com transition.

Q6)

Já fiz, basta enviar o arquivo da aula anterior.

Q7)