Enxurrada de Bits

Estado CSS e animações

Estados CSS

Recurso do css que nos permite aplicar estilos em resposta à uma mudança de estado de um elemento

Pseudo-classes

As pseudo-classes são códigos adicionados a **seletores** para que especifique um estado especial do elemento selecionado, podendo ser automático ou de acordo com certa interação do usuário

Elas permitem aplicar um estilo a um elemento, não apenas em relação ao conteúdo do documento, mas também em relação a fatores externos como o histórico de navegação, o status do seu conteúdo, ou a posição do mouse

Sintaxe

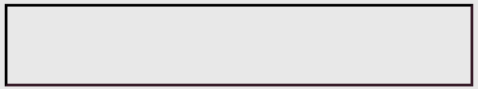
A sintaxe da pseudo-classe se apresenta da seguinte forma:


```
seletor: pseudo-classe {
    propriedade: valor;
}
```

Pseudo-classes

➔ :hover

As propriedades são aplicadas quando o ponteiro do mouse está em cima do elemento



➔ :focus

As propriedades são aplicadas quando um elemento recebe foco, geralmente sendo clicado

➔ :active

As propriedades são aplicadas enquanto o mouse estiver clicando no elemento

Pseudo-classes

- :link

As propriedades são aplicadas no estado inicial do link, quando o link ainda não foi visitado.

- :visited

As propriedades são aplicadas quando o link já foi visitado anteriormente, verificando se o link já foi visitado (observando o histórico).

Pseudoelementos

Os pseudoelementos são utilizados para estilizar parte específica de um elemento selecionado

A sintaxe da pseudoelementos se apresenta da seguinte forma:


```
seletor:: pseudoelemento {
    propriedade: valor;
}
```

Pseudoelementos

➔ ::before

Adiciona um conteúdo extra antes do elemento, com sua própria estilização

Conteúdo antes ➔ parágrafo

HTML:

```
<p class="antes">
    parágrafo
</p>
```

CSS:

```
.antes::before {
  content: "Conteúdo antes ➔ ";
  color: orange;
}
```

➔ ::after

Adiciona um conteúdo extra depois do elemento, com sua própria estilização

parágrafo ➔ Conteúdo depois

HTML:

```
<p class="depois">
    parágrafo
</p>
```

CSS:

```
.depois::after{
  content: " ➔ Conteúdo depois";
  color: red;
}
```

Pseudoelementos

- ::placeholder

Utilizado geralmente junto com a tag <input> ele estiliza um espaço reservado

- ::selection

Utilizado para aplicar estilos à parte de um documento que foi destacada pelo usuário (como clicar e arrastar o mouse pelo texto)

Enxurrada de Bits CEFET-MG

Animações

Recurso nativo do css que nos permite fazer animações utilizando somente código CSS

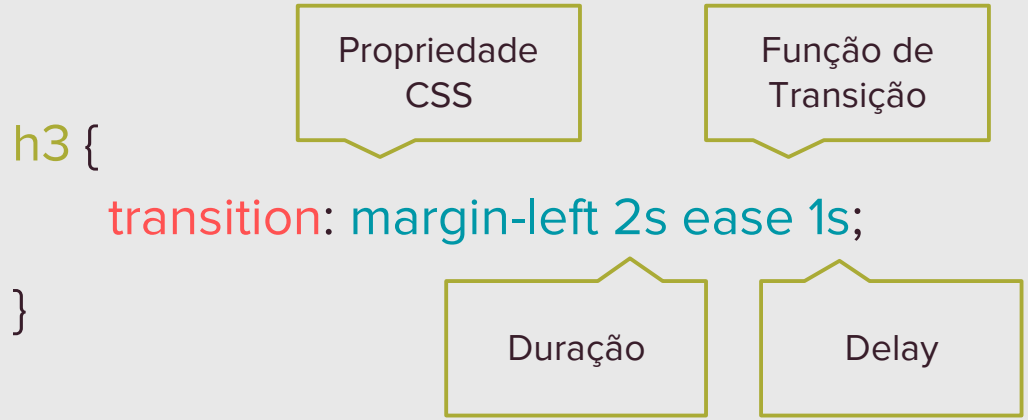
transition

Permite definir a transição entre dois estados de um elemento e é um atalho para:

- transition-property: all;
  quais propriedades CSS devem ter a transição
- transition-duration: 0s;
  duração da transição
- transition-timing-function: ease;
  função de interpolação
- transition-delay: 0s;
  tempo de atraso até que se comece a transição

Sintaxe

A sintaxe que mais usamos para a propriedade *transition* é:

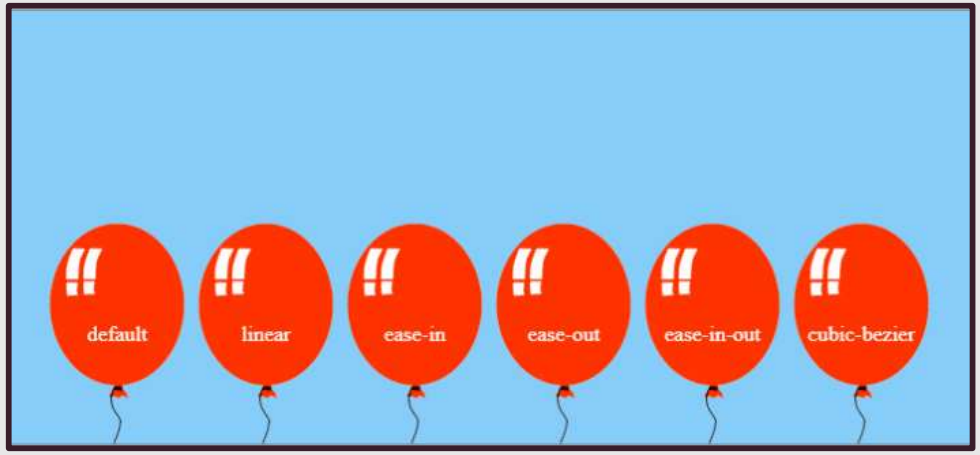


Função de interpolação

As timing functions determinam uma função para definir como os valores intermediários das propriedades são calculados.

Sendo:

- default (ease)
- linear
- ease-in
- ease-out
- ease-in-out
- cubic-bezier



Observe melhor em: https://fegemo.github.io/cefet-front-end/classes/css6/#23

transition

Geralmente é associado com os seletores de estados, nesse caso, associamos o transition com o hover:

Exemplo:


```
transition-property: width,height,transform,background,font-size,color, padding;
transition-duration: 3s,2s,3s,2s,3s,2s;
transition-delay:0s, 1s, 0s, 0s, 0s, 0s;
```

transform

Transforma um elemento na tela, podendo mudar suas coordenadas, alargá-lo, afiná-lo, rodar o elemento, etc

- translateY e translateX
- scaleX e scaleY
- rotateX, rotateY e rotateZ

translate, scale, rotate...

```
img {
    transform: valor;
}
```

Scale

transform

Temos que ter atenção às unidades de medida de cada atributo:

- translateY( ) e translateX( )
  - Valores em px, em, pt, cm, mm
- scaleX( ) e scaleY( )
  - Valores em números reais (float)
- rotateX( ), rotateY( ) e rotateZ( )
  - Valores em deg, grad, rad, turn

translate

animation

Uma animação CSS é composta por:

- Uma definição de quadros de animação (@keyframes)
- Uma configuração de parâmetros da animação (animation)

@keyframes - definem o valor das propriedades que variam ao longo da animação.

animation - configura o tempo da animação, a ordem de execução e o nome da sequência de quadros a ser usada.

animation

Aplica uma animação entre estilos e é um atalho para:

- animation-name: none;
  - nome dado ao **@keyframes** a ser usado
- animation-duration: 0s;
  - duração da animação
- animation-delay: 0s;
  - tempo de atraso (tempo a ser gasto antes da animação começar)
- animation-direction: normal;
  - determina a direção da animação (normal, reverse, alternate, alternate-reverse)
- animation-iteration-count: 1;
  - quantas vezes rodar (número, infinite)
- animation-timing-function: ease;
  - função de interpolação
- animation-play-state: running;
  - estado da animação (running, pause)

Sintaxe

Existem diversas sintaxes da propriedade *animation*, vamos a mais simplificada:

Nome da sua animação

Função de interpolação

```
seletor {
    animation: nome tempo timing-function;
}
@keyframes nome {
    0% {
        propriedade: valorX;
    }
    100% {
        propriedade: valorY;
    }
}
```

Valor da propriedade no estado inicial

Valor da propriedade no estado final

Podemos fazer estados intermediários como 25%, 40%, 70%

Estados CSS e Animações

Fazendo uma junção dos conteúdos dessa aula podemos fazer várias coisas bacanas, como no exemplo:


```
div {
  padding: 20px;
  background-color: green;
  font-family: Arial;
}
div:hover {
  animation-name: example;
  animation-duration: 4s;
  animation-iteration-count: infinite;
}
@keyframes example {
  0% { background-color: green;  color:yellow }
  25% { background-color: yellow;  color:blue }
  50% { background-color: blue;  color:white }
  75% { background-color: white;  color: green }
  100% { background-color: green;  color:yellow }
}
```

Referências

Material da disciplina de Programação Web do prof. Flávio Coutinho:
https://fegemo.github.io/cefet-web/classes/css3/#30
https://fegemo.github.io/cefet-web/classes/css5/#17

Documentação da Mozilla Developer Network:
https://developer.mozilla.org/pt-BR/docs/Web/CSS/Pseudo-classes

https://developer.mozilla.org/pt-BR/docs/Web/CSS/animation#:~:text=A%20propriedade%20CSS%20abreviada%20animation,e%20animation%2Dplay%2Dstate%20.

https://developer.mozilla.org/pt-BR/docs/Web/CSS/CSS_Animations/Usando_anima%C3%A7%C3%B5es_CSS

Leiam

visibility:
https://developer.mozilla.org/pt-BR/docs/Web/CSS/visibility
https://fegemo.github.io/cefet-web/classes/css3/#26

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

posicionamentos:
https://fegemo.github.io/cefet-front-end/classes/css4/#6

Dúvidas?

Jornal do bairro (parte 2)

Na atividade de hoje, vamos continuar o jornal do bairro.
Hoje, você deve fazer:

Aplicar CSS para deixar o site com um visual interessante

a. Utilize as propriedades que aprendemos para deixar o estilo do seu gosto
   Exemplo: background-color, color, font-size, etc
a. Use as propriedades de estado para deixar o menu dentro do nav interativo, ou seja, que varia conforme as ações do usuário
   Exemplo: Mude a cor caso o mouse passe por cima do elemento e caso o elemento seja clicado
a. Faça com que o cursor do mouse mude para aquela “mãozinha”, indicando que o elemento é clicável

Jornal do bairro (parte 2)

d. Em alguma notícia do jornal, coloque um link para uma notícia de outro jornal (*ex:* CNN, O Globo, etc). Você deve utilizar as propriedades de estado :link e :visited para mudar a cor do link caso ele não tenha sido visitado nenhuma vez ou tenha sido visitado alguma vez
e. Faça uma animação legal com o título do seu jornal do bairro utilizando as tags *transition*, *transform* e *animation*
   Exemplo: Faça a cor ficar variando e mude a posição dele de forma fluída quando o mouse passar em cima