Interatividade na Web

Front End Web

Msc. Lucas G. F. Alves
e-mail: lucas.g.f.alves@gmail.com

Senac

Planejamento de Aula

Revisão

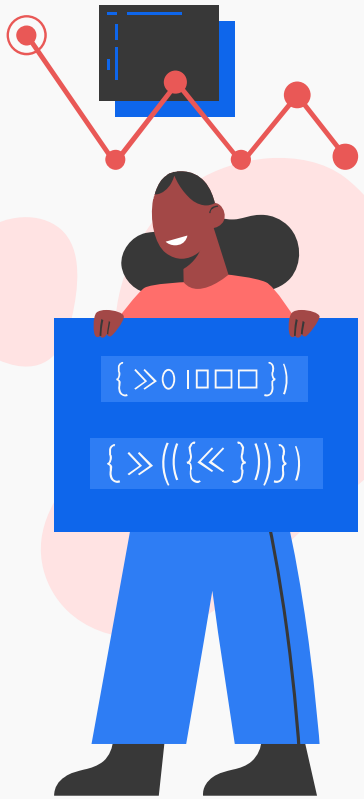
CSS Transitions

CSS Transforms

Media Types e Media Queries

Viewport

Exercícios



Revisão

CSS Avançado

Com CSS, os desenvolvedores front-end utilizam diversas técnicas de estilização.

Mesmo assim algumas coisas eram impossíveis de se conseguir utilizando somente CSS.

O CSS3 agora permite coisas antes impossíveis como elementos com cor ou fundo gradiente, sombras e cantos arredondados. Antes só era possível atingir esses resultados com o uso de imagens e às vezes até com um pouco de JavaScript.

A redução do uso de imagens traz grandes vantagens quanto à performance e quantidade de tráfego de dados necessária para a exibição de uma página.

CSS Avançado

Seletores avançados

Os seletores CSS mais comuns e tradicionais são os que já vimos: por ID, por classes e por descendência.

Seletor tag -> header{} **Seletor ID** -> #cabecalho **Seletor classe** -> .divp

Seletor Dependência -> #rodape img {} **Seletor Atributo** -> input[type="text"] {}

Com o CSS3, há muitos outros seletores novos que são bastante úteis.

Já vimos alguns, como os seletores de atributo que usamos anteriormente.

CSS Avançado

Seletor de irmãos (~)

Tem como objetivo selecionar de uma certa maneira todos os elementos após o subtítulo.
Ex.:

```
<article>
    <h1>Título</h1>
    <p>Início</p>

    <h2>Subtítulo</h2>
    <p>Texto</p>
    <p>Mais texto</p>
</article>
```

```
h2 ~ p {
    font-style: italic;
}
```

CSS Avançado

Seletor de irmãos Adjacentes (+)

Tem como objetivo selecionar apenas o parágrafo imediatamente seguinte ao subtítulo.
Ex.:

```
<article>
    <h1>Título</h1>
    <p>Início</p>

    <h2>Subtítulo</h2>
    <p>Texto</p>
    <p>Mais texto</p>
</article>
```

```
h2 + p {
    font-variant: small-caps;
}
```

CSS Avançado

Seletor de Filho Direto (>)

Tem como objetivo selecionar apenas o parágrafo filho seguinte ao subtítulo.
Ex.:

```
<article>
    <h1>Título principal</h1>


    <section>
    <h1>Título da seção</h1>
    </section>
</article>
```

```
/* vai pegar todos os h1 da página */
h1 { color: blue;}

/* vai pegar todos os h1 do article, incluindo de
dentro da section */
article h1 { color: blue; }

/* vai pegar só o h1 principal, filho direto de article e
não os netos */
article > h1 { color: blue; }
```

CSS Avançado

Negação

Tem como objetivo escrever um seletor que pega elementos que não batem naquela regra..
Ex.:

```
<p>Texto</p>
<p>Outro texto</p>
<p class="especial">Texto especial</p>
<p>Mais texto</p>
```

```
p:not(.especial) {
    color: gray;
}
```

A sintaxe do :not() recebe como argumento algum outro seletor simples (como classes, IDs ou tags).

({(({≫})) ≪ }

Pseudo-Classes

Pseudo-classes são como classes CSS já pré-definidas para nós. É como se o navegador já colocasse certas classes por padrão em certos elementos, cobrindo situações comuns.

Há duas pseudo-classes do CSS3, uma que representa o primeiro elemento filho de outro (first-child) e o último elemento filho (last-child). Essas classes já estão definidas, não precisamos aplicá-las em nosso HTML.

Ex.:

```
<ul>
    <li>Primeiro item</li>
    <li>Segundo item</li>
    <li>Terceiro item</li>
    <li>Quarto item</li>
</ul>
```

```
li:first-child {
    color: red;
}

li:last-child {
    color: blue;
}
```

Pseudo-Classes

nth-child

É um seletor genérico do CSS3 que permite passar o índice do elemento. Por exemplo, podemos pegar o terceiro item com:

```
li:nth-child(3) { color: yellow; }
```

O mais interessante é que o nth-child pode receber uma expressão aritmética para indicar quais índices selecionar.

```
li:nth-child(2n) { color: green; } /* elementos pares */
li:nth-child(2n+1) { color: blue; } /* elementos impares */
```

Pseudo-Classes

Pseudo-Classes de Estado

O CSS possui excelentes pseudo-classes que representam estados dos elementos.

Antes utilizamos javascript para mudar o estado de um elemento.

Agora temos pseudos-classes de estados que fazem isso.
Ex: Uma que representa o momento que o usuário está com o mouse em cima do elemento, a :hover.

```
/* seleciona o link no exato momento em que passamos o mouse por cima dele */
 a:hover { background-color:#FF00FF; }
```

Pseudo-Classes

Pseudo-Classes de Estado

```
/* seleciona todas as âncoras que têm o atributo "href", ou seja, links */
 a:link { background-color:#FF0000; }

/* seleciona todos os links cujo valor de "href" é um endereço já visitado */
a:visited { background-color:#00FF00; }

/* seleciona o link no exato momento em que clicamos nele */
a:active { background-color:#0000FF; }
```

Pseudo Elementos

São elementos que não existem no documento mas podem ser selecionados pelo CSS. É como se houvesse um elemento lá!
Ex.:

```
<p>Pseudo Elementos</p>
```

```
p:first-letter {
    font-size: 200%;
}

p:first-line {
    font-style: italic;
}
```

Pseudo Elementos

Novos Conteúdos

Um outro tipo de pseudo-elemento mais poderoso que nos permite gerar conteúdo novo via CSS é o **after** e **before**.

[Link 1] [Link 2] [Link 3]

HTML

```
<a href="...">Link1</a>
<a href="...">Link2</a>
<a href="...">Link3</a>
```

CSS

```
a:before {
        content: '[ ';
}
a:after {
        content: ' ]';
}
```

Exercícios

1) Alterar a página de sobre voces, a contato.html modificando as primeiras letras dos parágrafos fiquem em negrito. Altere o arquivo contato.css e use a pseudo-classe :first-letter pra isso. p:first-letter { font-weight: bold; }

2) Remover apenas a identação do primeiro parágrafo da página. Usando seletor de irmãos adjacentes. Acrescente ao contato.css: h1 + p { text-indent: 0;}

3) Alterar o visual da primeira linha do texto com :first-line. Por exemplo, transformando-a em small-caps usando a propriedade font-variant: p:first-line {font-variant: small-caps;}

4) Agora no index.css. Temos (.menu-opcoes) que é uma lista de links. Altere seus estados quando o usuário passar o mouse (:hover) e quando clicar no item (:active). Adicione ao arquivo index.css: .menu-opcoes a:hover { color: #007dc6;} .menu-opcoes a:active { color: #867dc6; }

Exercícios

5) Fazer um menu que abre e fecha em puro CSS. Temos o .menu-conteudo na esquerda da página com várias categorias de aulas. Adicionar subcategorias que aparecem apenas quando o usuário passar o mouse.

HTML

```
<li>
        <a href="#">Aula 1</a>
        <ul>
                <li><a href="#">Exercício 1</a></li>
                <li><a href="#">Exercício 2</a></li>
                <li><a href="#">Exercício 3</a></li>
        </ul>
</li>
```

CSS

```
.menu-conteudo li ul {
        display: none; /*por padrão escondida*/
}
.menu-conteudo li:hover ul {
        display: block; /*no mouse aparece*/
}
.menu-conteudo ul ul li {
        background-color: #dcdcdc;
 }
```

Exercícios

6) Colocar um traço na frente para diferenciar os links dos menus. Podemos alterar o HTML colocando os traços - algo visual e não semântico -, ou podemos gerar esse traço via CSS com pseudo elementos. Utilise o :before para injetar um conteúdo novo via propriedade content no CSS: .menu-conteudo li li a:before {content: '- ';}

7) Na página Contato, abra-a no navegador. No exercício 3, as primeiras linhas foram colocadas em small-caps usando o seletor p:first-line. Mas todos os parágrafos foram afetados. Para resolver utilize a negação. Com o seletor :not() do CSS3:
:not(p) + p:first-line { font-variant: small-caps; }

CSS3 Border Radius

No CSS3 foi adicionado as bordas arredondadas via CSS. Até então, a única forma de obter esse efeito era usando imagens, o que deixava a página mais carregada e dificultava a manutenção.

Com o CSS3 há o suporte a propriedade border-radius que recebe o tamanho do raio de arredondamento das bordas. Por exemplo:

```
div {
    border-radius: 5px;
}
```

CSS3 Border Radius

Também pode passar valores diferentes para cantos diferentes do elemento:

```
.a{ /* todas as bordas arredondadas com um raio de 15px */ border-radius: 15px; }
.b{ /*borda superior esquerda e inferior direita com 5px borda superior direita e inferior esquerda com 20px*/ border-radius: 5px 20px; }
.c{ /*borda superior esquerda com 5px borda superior direita e inferior esquerda com 20px borda inferior direita com 50px */ border-radius: 5px 20px 50px; }
.d{ /*borda superior esquerda com 5px borda superior direita com 20px borda inferior direita com 50px bordar inferior esquerda com 100px */ border-radius: 5px 20px 50px 100px; }
```

CSS3 Text Shadow

Outro efeito do CSS3 são as sombras em textos com text-shadow. Sua sintaxe recebe o deslocamento da sombra e sua cor:

```
p {
    text-shadow: 10px 10px red;
}
```

Ou ainda receber um grau de espalhamento (blur):

```
p {
    text-shadow: 10px 10px 5px red;
}
```

É possível até passar mais de uma sombra ao mesmo tempo para o mesmo elemento:

```
text-shadow: 10px 10px 5px red, -5px -5px 4px red;
```

CSS3 box-shadow

Agora também é possível colocar sombras em qualquer elemento com box-shadow. A sintaxe é parecida com a do text-shadow:

```
box-shadow: 20px 20px black;
```

Podemos ainda passar um terceiro valor com o blur:

```
box-shadow: 20px 20px 20px black;
```

CSS3 box-shadow

Diferentemente do text-shadow, o box-shadow suporta ainda mais um valor que faz a sombra aumentar ou diminuir:

```
box-shadow: 20px 20px 20px 30px black;
```

Por fim, é possível usar a keyword inset para uma borda interna ao elemento:

```
box-shadow: inset 0 0 40px black;
```

Opacidade RGBA

No CSS2 é possível mudar a opacidade de um elemento para que ele seja mais transparente com o uso da propriedade opacity.

```
p { opacity: 0.3; }
```

No CSS3, além das cores hex normais (#FFFFFF pro branco), podemos criar cores a partir de seu valor RGB, passando o valor de cada canal (Red, Green, Blue) separadamente (valor entre 0 e 255):

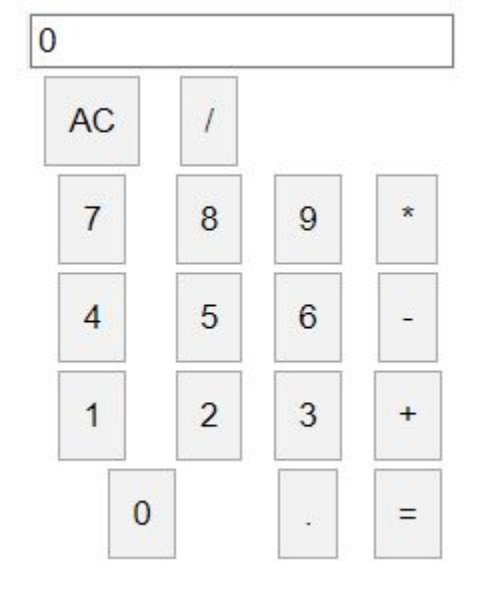
```
/* todos equivalentes */
color: #FFFFFF;  color: white;  color: rgb(255, 255, 255);
```

Porém, existe uma função chamada RGBA que recebe um quarto argumento, o chamado canal Alpha. Na prática, seria como o opacity daquela cor (um valor entre 0 e 1):

```
/* branco com 80% de opacidade */
color: rgba(255,255,255, 0.8);
Ex,: p { background: rgba(0,0,0,0.3); color: white; }
```

Opacidade RGBA

1) Criar uma calculadora com todas as funções básicas, utilizando table para estruturar os botões. Além de resolver as questões aritméticas com funções em javascript.



CSS3: Gradientes

CSS3: Gradientes

No CSS3 tem declaração de gradientes sem precisar usar imagens.

Existe suporte a gradientes lineares e radiais, inclusive com múltiplas paradas.

A sintaxe básica é:

```
.linear { background: linear-gradient(white, blue); }
.radial { background: radial-gradient(white, blue); }
```

Pode usar gradientes com angulações diferentes e diversas paradas de cores:

```
.gradiente { background: linear-gradient(45deg, #f0f9ff 0%, #cbebff 47%, #a1dbff 100%); }
```

CSS3: Gradientes

Outras sintaxes:
Na versão atual da especificação o primeiro argumento indica a direção do gradiente:

```
.linear { background: linear-gradient(to bottom, white, blue); }
/* Gradiente do branco para o azul vindo de cima para baixo*/
```

Na versão suportada antes do rascunho dos gradientes:

```
.linear {background: -webkit-linear-gradient(top, white, blue);
        background: -moz-linear-gradient(top, white, blue);
        background: -o-linear-gradient(top, white, blue);}
```

Versões bem mais antigas do WebKit

```
.linear {background: -webkit-gradient(linear, left top, left bottom,
color-stop(0%, white), color-stop(100%, blue));}
```

Progressive Enhancement

Progressive Enhancement

Muitas pessoas não gostam das versões atuais de navegadores e utilizam navegadores antigos como IE8 ou mais antigos, onde algumas propriedades novas não são aceitas.

Mas temos que implementar sempre o mais inovador ao usuário, como resolver?

Fazer funcionar em qualquer navegador sem prejudicar os antigos.
Graceful degradation foi o nome da primeira técnica a pregar isso.
Este removia funcionalidades não suportadas e degradava “graciosamente”.

Hoje se utiliza Progressive enhancement com uma ideia parecida mas ao contrário, se desenvolve funcionalidades normalmente e acrescentando pequenas melhorias mesmo que só funcionem nos navegadores modernos.

Exercícios

Exercícios

1) Colocar alguns efeitos nos painéis com border-radius, box-shadow e text-shadow.

```
.painel { border-radius: 4px; box-shadow: 1px 1px 4px #999; }
.painel h2 { text-shadow: 3px 3px 2px #FFF; }
```

2) Utilizar bordas internas aos elementos. Aplicar aos painéis de aula. Basta usar a opção inset:

```
box-shadow: inset 1px 1px 4px #999;
```

3) Utilizar border-radius em algumas bordas específicas.

```
.busca { border-top-left-radius: 4px; border-top-right-radius: 4px; }
```

4) Utilizar cores com canal Alpha para translucência de 80%.

```
.painel h2 { text-shadow: 1px 1px 2px rgba(255,255,255,0.8); }
```

5) Aplicar gradientes aos paineis de aulas.

```
.aula1{background: linear-gradient(#f5dcdc, #bebef4);}
.aula2{background: linear-gradient(#dcdcf5, #f4bebe);}
```

CSS3 Transitions

CSS3 Transitions

Com transitions, é possível animar o processo de mudança de algum valor do CSS.

Por exemplo: temos um elemento na posição top:10px e, quando passarmos o mouse em cima (hover), queremos que o elemento mude para top:30px.

O CSS básico muda de uma vez sem transição:

```
#teste { position: relative; top: 0; }
#teste:hover { top: 30px; }
```

Com transição animando durante 2 segundos:

```
#teste:hover {transition: top 2s; }
```

CSS3 Transitions

Por padrão a transição é linear, mas tem outros tipos de transições, como:

- linear - velocidade constante na animação;
- ease - redução gradual na velocidade da animação;
- ease-in - aumento gradual na velocidade da animação;
- ease-in-out - aumento gradual, depois redução gradual na velocidade da animação;
- cubic-bezier(x1,y1,x2,y2) - curva de velocidade para animação customizada (avançado);

```
#teste:hover { transition: top 2s ease; }
#teste { position: relative; top: 0; color: white; } #teste:hover { top: 30px; color: red; transition: top 2s, color 1s ease; } /* incluindo cores*/
#teste:hover { transition: all 2s ease; } /*o efeito para todas as propriedades*/
```

CSS3 Transforms

CSS3 Transforms

Nova especificação para alterar propriedades visuais dos elementos, como ângulo, mostrá-lo em uma escala maior ou menor ou alterar a posição do elemento sem sofrer interferência de sua estrutura.

Translate

```
.header { /* Move o elemento no eixo horizontal */ transform: translateX(50px); }

#main { /* Move o elemento no eixo vertical */ transform: translateY(-20px); }

footer { /* Move o elemento nos dois eixos (X, Y) */ transform: translate(40px, -20px); }
```

CSS3 Transforms

Rotate

```
#menu-conteudo { transform: rotate(-10deg); }
```

Scale

```
#aula1 li { /* Alterar a escala total do elemento */ transform: scale(1.2); }
```

```
#aula2 li { /* Alterar a escala vertical e horizontal do elemento */ transform: scale(1, 0.6); }
```

CSS3 Transforms

Skew

footer { /* Distorcer o elemento no eixo horizontal */ transform: skewX(10deg); }
#social { /* Distorcer o elemento no eixo vertical */ transform: skewY(10deg); }

Mais de um transform junto:

html { transform: rotate(-30deg) scale(0.4); }

Exercícios

Exercícios

1) Quando passar o mouse em algum exercício dos painéis de destaques, mostrar uma sombra por trás. Utilize também transição para que essa sombra apareça suavemente:
.painel li:hover { box-shadow: 0 0 5px #333; transition: box-shadow 0.7s; }

2) Colocar agora um fundo branco no elemento. Anime esse fundo também, fazendo um efeito tipo fade. Indicar que todas as propriedades devem ser animadas.
.painel li:hover { background-color: rgba(255,255,255,0.8); box-shadow: 0 0 5px #333; transition: 0.7s; }

3) Quando passar o mouse em cima do item do painel, aumentar o elemento em 20%.
.painel li:hover { transform: scale(1.2); }

4)Alterar o comando anterior para fazer rotacionar suavemente em 5 graus .
.painel li:hover { transform: scale(1.2) rotate(-5deg); }

CSS media types

CSS media types

São suportes de regras de layout diferentes, que podem ser declarados ao se invocar um arquivo CSS:

```
<link rel="stylesheet" href="site.css" media="screen" />
<link rel="stylesheet" href="print.css" media="print" />
<link rel="stylesheet" href="handheld.css" media="handheld" />

@media screen { body { background: blue; color: white; } }
@media print { body { background: white; color: black; } }
```

O media type screen determina a visualização normal, no Desktop. O media type print com regras de impressão. O media type handheld, é voltado para dispositivos móveis.

Só que o handheld foi desenvolvido para celulares antigos, e os novos smartphone, tablets, televisões, já visualizam a página inteira com todos os detalhes.

({(({ » }) « }

CSS3 media queries

CSS3 media queries

É uma nova forma de adaptar o CSS.

```
<link rel="stylesheet" href="base.css" media="screen">
<link rel="stylesheet" href="mobile.css" media="(max-width: 480px)">
```

Outra forma de declarar os media types é separar as regras dentro do mesmo arquivo CSS:

```
@media screen { body { font-size: 16px; } }
@media (max-width: 480px) { body { font-size: 12px; } }
```

Viewport

Viewport

Por padrão, iPhones, Androids e afins costumam considerar o tamanho da tela visível, chamada de viewport como grande o suficiente para comportar os Sites Desktop normais.

A Apple criou então uma solução que depois foi copiada pelos outros smartphones, que é configurar o valor que julgamos mais adequado para o viewport:

<meta name="viewport" content="width=320">

Ou utilizar um valor device-width definido pelo fabricante.

<meta name="viewport" content="width=device-width">

Exercícios

Exercícios

1) Adaptar nossa home page (index.html) para mobile. Escrever nosso CSS de adaptação em um novo arquivo, chamado mobile.css. Crie esse arquivo e o importe no head do index.html.

```
<meta name="viewport" content="width=device-width">
<link rel="stylesheet" href="css/mobile.css" media="(max-width: 320px)">
```

2) A página hoje tem o tamanho fixo em 940px e é centralizada (com o uso do seletor .container). No mobile.css.

```
.container { width: 96%; }
```

3) Ajustar os elementos do topo da página.

```
header h1 { text-align: center; }
header h1 img { max-width: 50%; }
.carrinho { display: none; }
.menu-conteudo { position: static; text-align: center; }
```

Exercícios

4) Aumentar o espaço entre os link.

```
.menu-opcoes ul li { display: inline-block; margin: 5px; }
```

5) Ajustar a seção de busca, o menu da esquerda e a imagem de destaque

```
.busca, .menu-departamentos, .destaque img { margin-right: 0; width: 100%; }
```

6) Ajustar os painéis de destaques de aulas.

```
.painel { width: auto; }
.painel li { width: 30%; }
.painel img { width: 100%; }
```

7)Ajustar a media query para aplicar o CSS de adaptação a qualquer tamanho de tela menor.

```
<link rel="stylesheet" href="css/mobile.css" media="(max-width: 939px)">
```

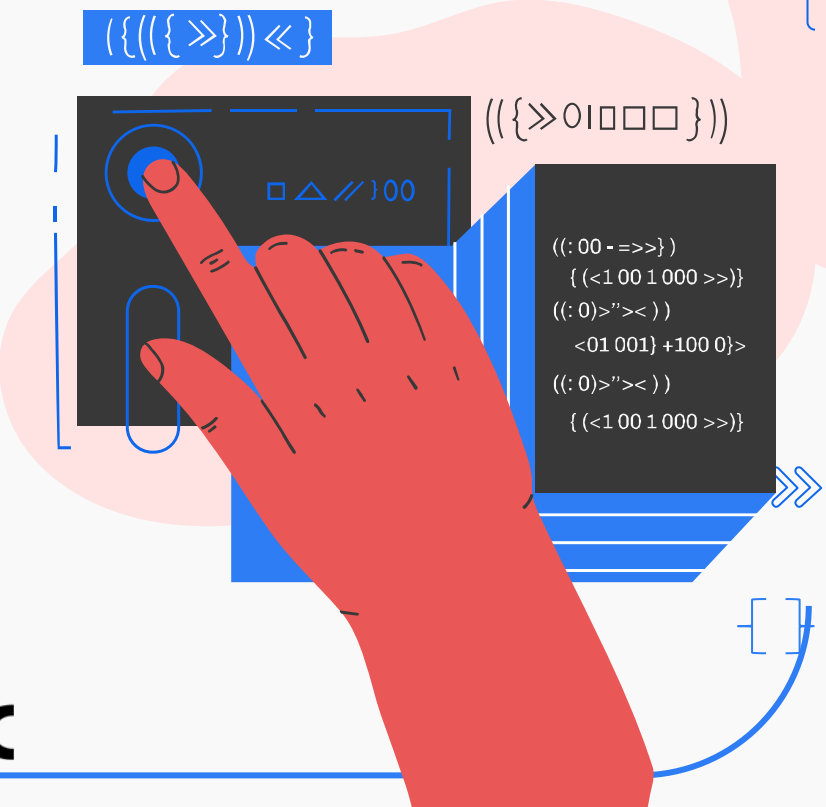
Professor

Lucas G. F. Alves

Senac

Obrigado!

E-mail :lucas.g.f.alves@gmail.com



Senac