HTML e CSS

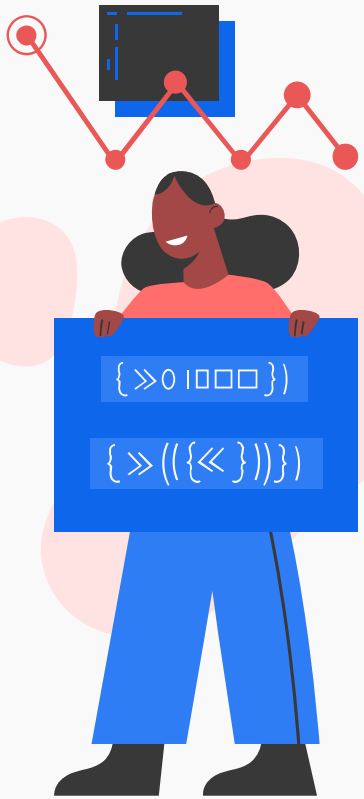
Front End Web

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

Senac

Planejamento de Aula

- Revisão HTML e CSS
- HTML Semântico
- CSS Reset
- Position
- Exercícios



Revisão

Propriedades Tipográficas e fontes

As fontes podem ser alteradas com o uso da propriedade **font-family**.

A propriedade **font-family** pode receber seu valor com aspas (nome do arquivo de fonte) ou sem aspas (família da fonte).

Por padrão, os navegadores exibem texto no tipo “**serif** ”.

As fontes mais conhecidas são “**Times**” e “**Times New Roman**”. Elas são chamadas de fontes serifadas pelos pequenos ornamentos em suas terminações.

Site com lista de fontes: https://wavian.com/font-list.html

Propriedades Tipográficas e fontes

Font-family

Pode-se alterar a família de fontes como “**sans-serif** ” (sem serifas), que contém, por exemplo, as fontes “**Arial**” e “**Helvetica**”.

Pode-se também declarar uma família de fontes “monospace” como, por exemplo, a fonte “**Courier**”.

Exemplo:

```
h1 { font-family: serif; }
h2 { font-family: sans-serif;}
p { font-family: monospace; }
```

Propriedades Tipográficas e fontes

Pode-se testar se uma fonte existe no computador.

As fontes mais comuns são consideradas “seguras” por serem bem populares.
Exemplo: body { font-family: "Arial", "Helvetica", sans-serif; }

1) O navegador verificará se a fonte “Arial” está disponível e a utilizará.
2) Caso a fonte “Arial” não esteja disponível, o navegador verificará a próxima fonte a “Helvetica”.
3) Caso o navegador não encontre também, ele solicita qualquer fonte da família “sans-serif ”.

Outra tag que serve para manipular a fonte, é a font-style, que define o estilo da fonte que pode ser: normal (normal na vertical), italic (inclinada) e oblique (oblíqua)

Alinhamento e decoração de texto

Text-align

Para alinhamento de texto é utilizada a tag text-align.

p { text-align: right; }

O exemplo determina que os parágrafos tenham o texto alinhado para a direita.

Pode-ser também determinar que seja alinhado ao centro ao definirmos o valor center, ou então definir que o texto vai ocupar toda a largura do elemento aumentando o espaçamento entre as palavras com o valor justify.

Alinhamento e decoração de texto

Por padrão o texto é alinhado à esquerda, com o valor left.

É possível configurar também uma série de espaçamentos de texto com o CSS:

```
p {
        line-height: 3px; /* tamanho da altura de cada linha */
        letter-spacing: 3px; /* tamanho do espaço entre cada letra */
        word-spacing: 5px; /* tamanho do espaço entre cada palavra */
        text-indent: 30px; /* tamanho da margem da primeira linha do texto */
    }
```

Bordas

Border

É disponível uma série de opções para definição de bordas.

Podemos, para cada borda de um elemento, determinar sua cor, seu estilo de exibição e sua largura.
Por exemplo:

```
body {
    border-color: red;
    border-style: solid;
    border-width: 1px;
}
```

Espaçamento

Padding

A propriedade padding é utilizada para definir uma margem interna (distância entre o limite do elemento, sua borda, e seu respectivo conteúdo) e tem as subpropriedades listadas a seguir:

- padding-top
- padding-right
- padding-bottom
- padding-left

Essas propriedades aplicam uma distância entre o limite do elemento e seu conteúdo acima, à direita, baixo e à esquerda respectivamente. Sentido horário.

Espaçamento

Padding

Pode-se definir todos os valores de uma única vez.

Se passado somente um valor para a propriedade padding, esse mesmo valor é aplicado em todas as direções.

```
p { padding: 10px; }
```

Se passados dois valores, o primeiro será aplicado acima e abaixo (mesmo valor para padding-top e padding-bottom) e o segundo será aplicado à direita e à esquerda (padding-right e padding-left).

```
p { padding: 10px 15px; }
```

Espaçamento

Padding

Pode ser passado três valores, o primeiro será aplicado acima (padding-top), o segundo será aplicado à direita e à esquerda (padding-right e padding-left) e o terceiro valor será aplicado abaixo do elemento (padding-bottom)

```
p { padding: 10px 20px 15px; }
```

Se forem quatro valores, serão aplicados respectivamente a padding-top, padding-right, padding-bottom e padding-left.

```
p { padding: 10px 20px 15px 5px; }
```

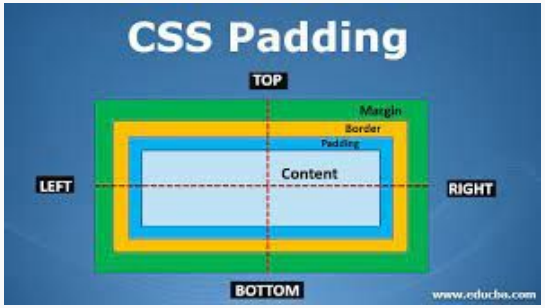
({(({≫})) ≪ }

Margem

Margin

A propriedade margin é parecida padding, porém ela adiciona espaço após o limite do elemento, ou seja, é um espaçamento além do elemento em si.

- margin-top
- margin-right
- margin-bottom
- margin-left



Para permitir que o navegador defina qual será a dimensão da propriedade padding ou margin conforme o espaço disponível na tela: definimos o valor auto.

```
p { margin: 0 auto; }
```

Dimensões

Um elemento pode ser dimensionado tanto pelas tags de posicionamento, mas também pelas propriedades de HEIGHT será definido a altura e WIDTH a largura do objeto, por exemplo:

height e width — definem dimensões de um elemento (auto, inherit, comprimento em pixels ou %)

max-height e max-width — definem dimensões máximas de um elemento (auto, inherit, comprimento em pixels ou %)

min-height e min-width — definem dimensões mínimas de um elemento (auto, inherit, comprimento em pixels ou %)

Ocultando / exibindo

Comandos:

Retira a elemento da visualização
display:none;
Apenas esconde o elemento, porém o espaço que ele ocupa fica visível.
— visibility: hidden;
Define elemento como visível num bloco (padrão)
— display:block;
Define elemento como visível numa linha (conteúdo disposto em linha)
— display:inline;

Ocultar e re-exibir elementos é muito útil para layouts com abas ou trocas alternadas de elementos de conteúdo.

Seletores de ID

É possível aplicar propriedades definindo um valor de seu atributo id. Para isso, o seletor deve iniciar com o caractere “#” seguido do valor correspondente.

```
#cabecalho {  color: white; text-align: center; }
```

O seletor acima fará com que todos os elementos dentro de cabecalho tenha seu texto renderizado na cor branca e centralizado.

Como o atributo id deve ter valor único no documento, o seletor deve aplicar suas propriedades declaradas somente àquele único elemento e, por cascata, a todos os seus elementos filhos.

Seletores Hierárquicos

Pode-se utilizar um seletor hierárquico que permite aplicar estilos aos elementos filhos de um elemento pai:

```
#rodape img {
        margin-right: 35px;
        vertical-align: middle;
        width: 94px;
}
```

No exemplo, o estilo será aplicado apenas nos elementos img filhos do elemento com id=rodape.

Fluxo de documento: Float

Float

A propriedade float permite que tiremos um certo elemento do fluxo vertical do documento o que faz com que o conteúdo abaixo dele flua ao seu redor.

Aplicando float a uma imagem fará com que o conteúdo do parágrafo flua ao redor da nossa imagem.

Houve uma perturbação do fluxo HTML, e a imagem parece existir fora do fluxo.

Fluxo de documento: Float

Float

Propriedades

clear — desvincula um elemento do efeito de floating (left, right, both, none, inherit)

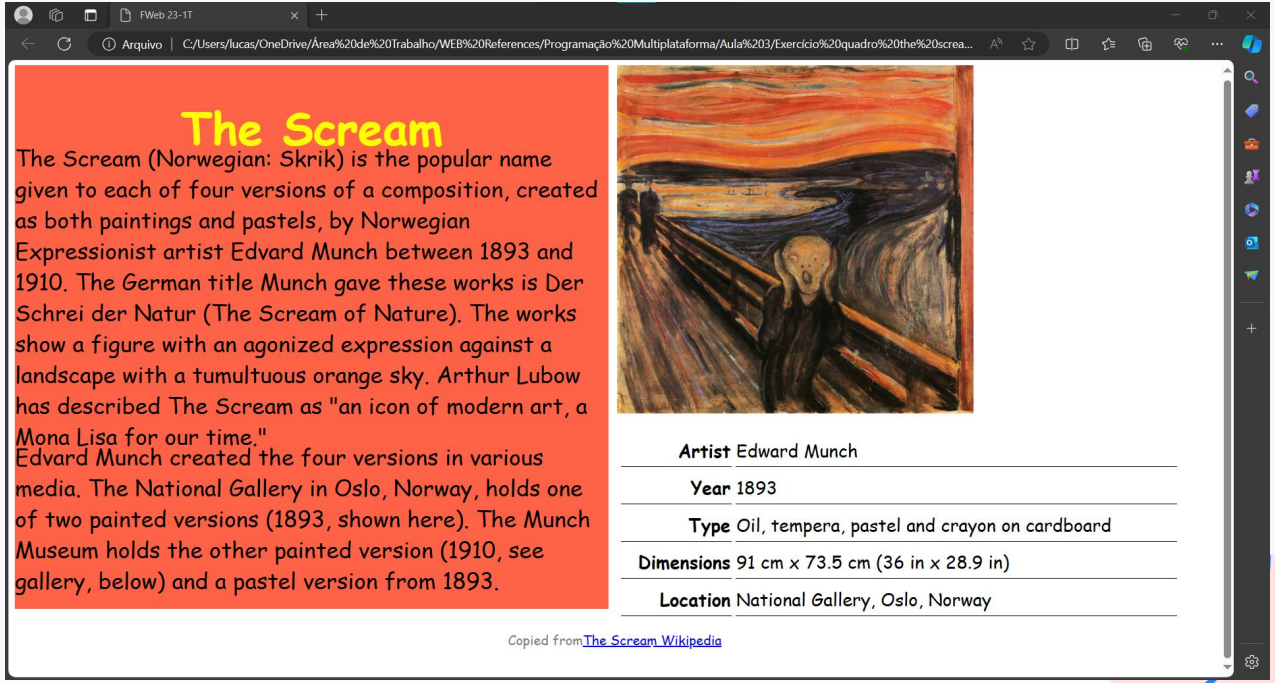
float — especifica uma flutuabilidade para um elemento (left, right, none, inherit)

Exercícios CSS

1) Recriar essa página, e colocar link para a página em conteudo.html.

Fontes:
'Sedgwick Ave', cursive;

'Caveat', cursive;



HTML semântico e posicionamento no CSS

HTML semântico

As tags que usamos antes - header, section e footer - são tags novas do HTML5.

Antigamente, seria criado apenas três div, uma para cada parte da página, e talvez colocando ids diferentes para cada uma.

A função do HTML é fazer a marcação do conteúdo da página, representar sua estrutura.

Já o CSS é cuidar da apresentação final e dos detalhes de design.

O HTML precisa ser claro e limpo, focado em marcar o conteúdo.

HTML semântico

Um HTML semântico carrega significado independente da sua apresentação.

Um usuário cego poderia usar um leitor de tela para ouvir sua página. Neste caso, a estrutura semântica do HTML é essencial para ele entender as partes do conteúdo.

É muito comum usar um <h1> com um texto que represente o título da nossa página.

Mas e pra colocar uma imagem de logo?
Quando o texto for lido para um cego, queremos essa mensagem lida. Quando o Google indexar, queremos que ele associe nossa página com o texto escrito e não com uma imagem.

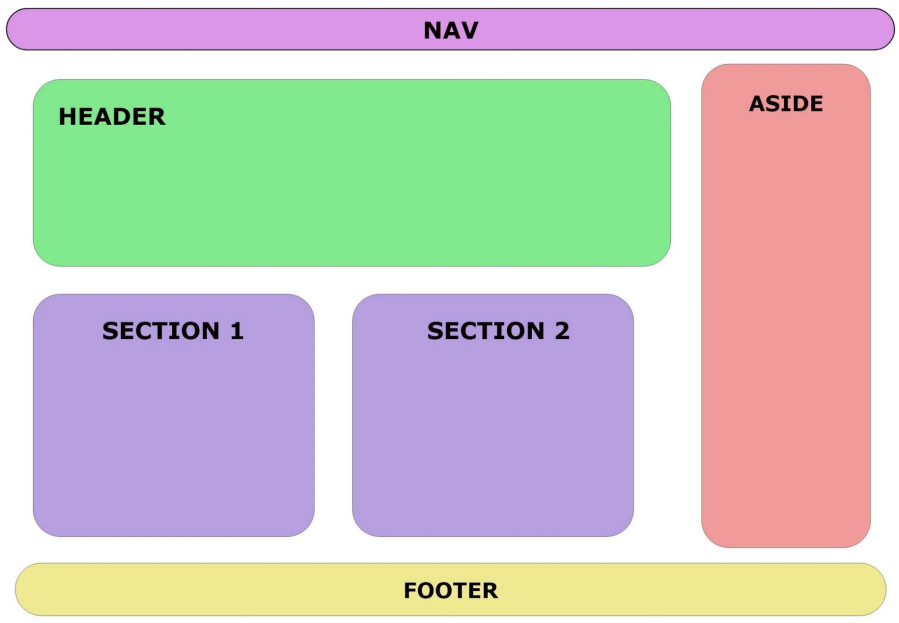
```
<h1><img src="img/logo.png" alt="site-tal"></h1>
```

Tags Semânticas

Tags semânticas são tags que possuem um significado, que dão sentido a informação de texto ao navegador e buscadores.

São elas:

<body>
    NAV
    HEADER
    MAIN
        SECTION
    FOOTER
    ASIDE
</body>



Estilização com classes

Para estilizar os elementos é necessário definir o CSS de cada coisa.

Já vimos **seletor de tag** e por **ID**. Ou seja, pra estilizar nosso menu <nav>, podíamos fazer:

nav { ... }

Mas imagine que podemos ter muitos NAV na página e queremos ser mais específicos. O ID é uma solução:

<nav id="menu-opcoes"> </nav>

E, no CSS:
#menu-opcoes { ... }

Estilização com classes

Classes.

O código é semelhante mas usa o atributo class no HTML e o ponto no CSS:

<nav class="menu-opcoes"> </nav>

E, no CSS:

.menu-opcoes { … }

Mas quando será usar ID ou CLASS?

Estilização com classes

Classes.

Ambos fariam seu trabalho nesse caso.

Mas IDs são mais fortes e devem ser únicos na página.

Embora o menu seja único agora, no futuro, o mesmo menu pode ser criado em outro ponto da página, mais pra baixo?

Usar classes facilita reuso de código e flexibilidade.

Estilização com classes

Um elemento pode ter mais de uma classe, aplicando estilos de várias regras do CSS:

```
<nav class="menu-opcoes menu-cabecalho"> ... </nav>
.menu-opcoes {
// código de um menu de opcoes
// essas regras serão aplicadas ao nav
}
.menu-cabecalho {
// código de um menu no cabeçalho
// essas regras TAMBÉM serão aplicadas ao nav
}
```

No caso do ID, não. Cada elemento só tem um id, único.

Estilização com classes

Utilizar classes é reaproveitar aquele elemento em mais de um ponto depois.

Vamos fazer isso no carrinho também:
 Nenhum item no carrinho de compras

Pode ser interessante criar uma classe que determina a centralização horizontal de qualquer elemento, sem interferir em suas margens superior e inferior, como no exemplo a seguir:

```
.container {
margin-right: auto;
margin-left: auto;
}
```

Estilização com classes

Agora, é só adicionar a class “container” ao elemento, mesmo que ele já tenha outros valores para esse atributo:

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

    <p>Conteúdo que deve ficar centralizado</p>

</div>
```

Exercícios

Exercícios

1) No arquivo index. Criar o cabeçalho utilizando as tags semânticas <header>, <nav>, <ul>, <li>, etc. Crie links para as páginas no menu. E use <h1> para representar o título da página com o logo acessível.

```
<header class="container">
    <h1><img src="img/logo.png" alt="nome_voces"></h1>
    <p class="carrinho"> Nenhum item no carrinho de compras </p>
    <nav class="menu-opcoes">
        <ul>
            <li><a href="conteudo.html">Conteúdo</a></li>
            <li><a href="blog.html">Blog</a></li>
            <li><a href="contato.html">Contato</a></li>
        </ul>
    </nav>
</header>
```

Exercícios

2) Ajuste as cores e alinhamento dos textos. Coloque o ícone da sacola com CSS através de uma imagem de fundo do parágrafo:

.carrinho { background: url(../img/carrinho.png) no-repeat top right; font-size: 14px; padding-right: 35px; text-align: right; width: 140px; }

.menu-opcoes ul { font-size: 15px; }

.menu-opcoes a { color: #003366; }

body { color: #333333; font-family: "Lucida Sans Unicode", "Lucida Grande", sans-serif; }

CSS Reset

CSS Reset

O navegador utiliza uma série de estilos padrão, que são diferentes em cada um dos navegadores.

Para evitar problemas de quebra de layout por navegador alguns desenvolvedores e empresas criaram alguns estilos que chamamos de CSS Reset.

- Setar um valor básico para todas as características do CSS, sobrescrevendo totalmente os estilos padrão do navegador.

Começando sempre do mesmo ponto para todos os casos, o que nos permite ter um resultado muito mais sólido em vários navegadores.

CSS Reset

A opções para resetar os valores do CSS.

HTML5 Boilerplate

O HTML5 Boilerplate fornecer um ponto de partida para quem desenvolve um novo projeto com HTML5.

Tem uma série de técnicas para aumentar a compatibilidade da nova tecnologia com navegadores um pouco mais antigos e o código é totalmente gratuito.

Em seu arquivo “style.css”, estão reunidas diversas técnicas de CSS Reset. Apesar de consistentes, algumas dessas técnicas são um pouco complexas.

CSS Reset

YUI3 CSS Reset

Criado pelos desenvolvedores front-end do Yahoo!, uma das referências na área, esse CSS Reset é composto de 3 arquivos distintos.

O primeiro deles, chamado de Reset, muda todos os valores possíveis para um valor padrão, onde até mesmo as tags <h1> e <small> passam a ser exibidas com o mesmo tamanho.

O segundo arquivo é chamado de Base, padroniza margens e dimensões dos elementos.

O terceiro é chamado de Font, onde o tamanho dos tipos é definido para que tenhamos um visual consistente inclusive em diversos dispositivos móveis.

CSS Reset

Eric Meyer CSS Reset

Há também o famoso CSS Reset de Eric Meyer, que pode ser obtido em http://meyerweb.com/eric/tools/css/reset/.

É apenas um arquivo com tamanho bem reduzido.

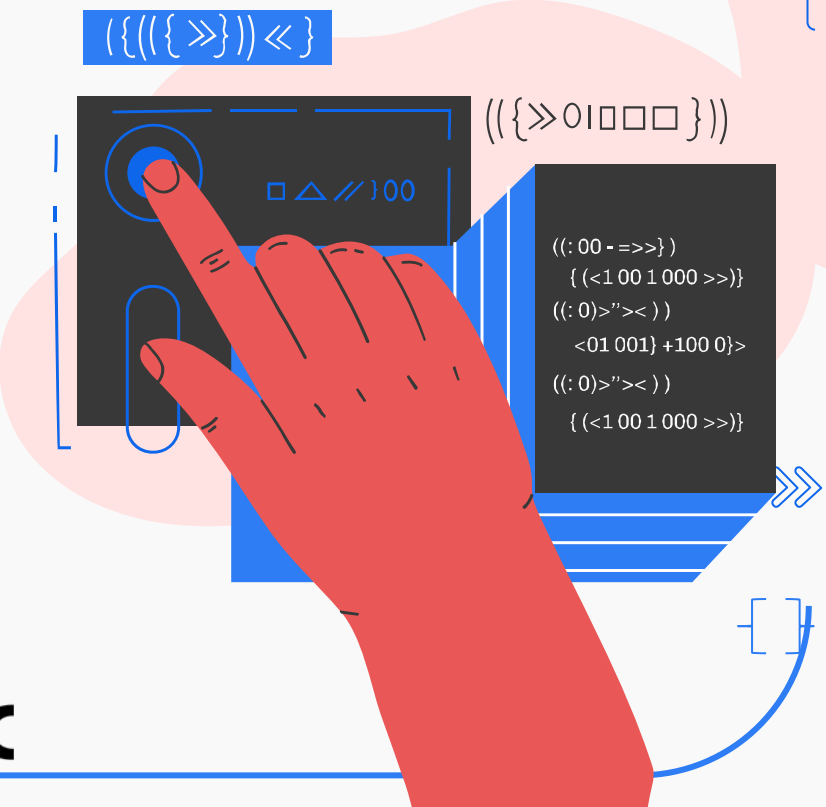
Professor

Lucas G. F. Alves

Senac

Obrigado!

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



Senac