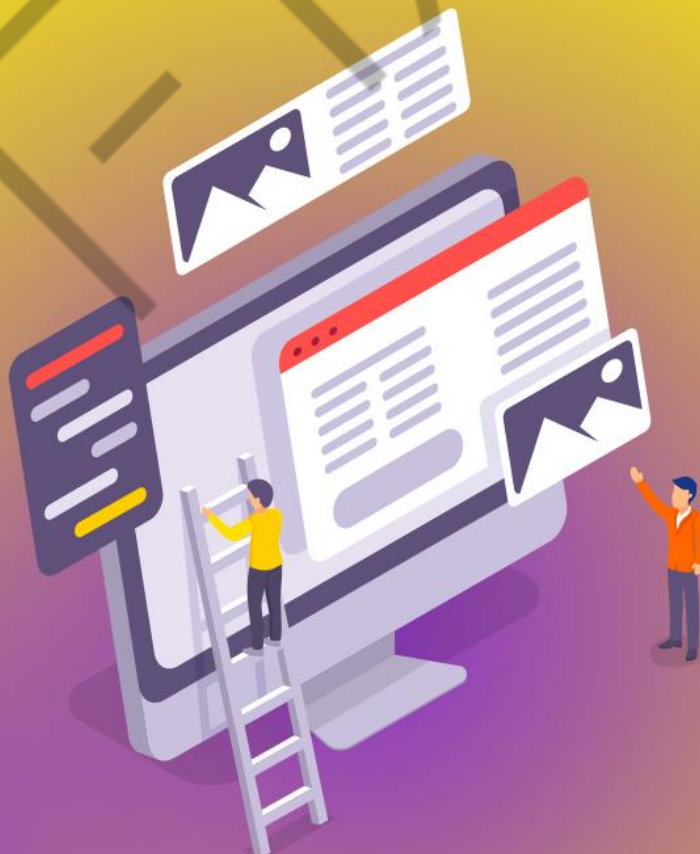


WEB DEVELOPMENT - PARTE 1

ELEMENTOS BÁSICOS HTML - *Introdução CSS*

ISRAEL MARQUES CAJAI JUNIOR



2

LISTA DE FIGURAS

Figura 2.1 – Representação dos cabeçalhos no <i>browser</i>	6
Figura 2.2 – Representação da informação em níveis de cabeçalhos no <i>browser</i> ...	7
Figura 2.3 – Parte da página principal do site da Split – uso de <h1> e <h2>	8
Figura 2.4 – Representação de cabeçalhos e parágrafos <i>browser</i>	9
Figura 2.5 – Parte da página principal da Xbox – uso de parágrafo.....	9
Figura 2.6 – Representação de parágrafo com no <i>browser</i>	10
Figura 2.7 – Representação de uma lista desordenada no <i>browser</i>	11
Figura 2.8 – Representação de uma lista ordenada no <i>browser</i>	12
Figura 2.9 – Representação de uma lista de definição no <i>browser</i>	12
Figura 2.10 – Representação de uma lista aninhada no <i>browser</i>	13
Figura 2.11 – Parte da página principal do game God of War – menu com lista.....	14
Figura 2.12 – Representação de imagens no <i>browser</i>	15
Figura 2.13 – Texto alternativo sendo exibido no <i>browser</i>	15
Figura 2.14 – Parte da página do game Shadow of the Colossus - imagens.....	16
Figura 2.15 – Resultado da regra CSS inline	18
Figura 2.16 – Exibição da regra CSS interna	19
Figura 2.17– Representação CSS Externo	20

LISTA DE CÓDIGOS-FONTE

Código-fonte 2.1 – Estrutura de cabeçalhos	6
Código-fonte 2.2 – Estruturando informação em níveis de cabeçalhos.....	7
Código-fonte 2.3 – Exemplo de cabeçalhos e parágrafos	8
Código-fonte 2.4 – Exemplo de parágrafo com 	10
Código-fonte 2.5 – Exemplo de uma lista desordenada	11
Código-fonte 2.6 – Exemplo de uma lista ordenada	11
Código-fonte 2.7 – Exemplo de uma lista de definição	12
Código-fonte 2.8 – Exemplo de uma lista de aninhada	13
Código-fonte 2.9 – Exemplo de inserção de imagens	15
Código-fonte 2.10 – Exemplo de formatação CSS	17
Código-fonte 2.11 – Exemplo de formatação CSS <i>inline</i>	18
Código-fonte 2.12 – Exemplo de formatação CSS interna	19
Código-fonte 2.13 – Exemplo de link com arquivo CSS	20

SUMÁRIO

2 ELEMENTOS BÁSICOS HTML – INTRODUÇÃO CSS	5
2.1 <i>Headings</i>	5
2.2 Parágrafos	8
2.3 Listas	10
2.3.1 Lista desordenada	10
2.3.2 Lista ordenada	11
2.3.3 Lista de definição	12
2.4 Imagens	14
2.5 CSS	17
2.5.1 Inserindo CSS na página	18
REFERÊNCIAS	21

2 ELEMENTOS BÁSICOS HTML – INTRODUÇÃO CSS

Tudo o que é visto na página web foi programado para que tivesse o design, o aspecto desejado, do parágrafo ao tamanho do texto, de listas até a posição em que uma imagem deve ficar, tudo é pensado e programado. Nas próximas seções, descubra como usar tudo isso a seu favor e qual a melhor forma para apresentar seu conteúdo em uma página web.

2.1 Headings

Os *headings* (cabeçalhos) devem ser usados para inserir títulos e subtítulos em seus documentos, ajudando na organização e na hierarquia da informação. Originalmente, há seis níveis de cabeçalho: <h1>, <h2>, <h3>, <h4>, <h5> e <h6>, sendo que o nível <h1> tem um maior destaque por usar uma fonte maior em relação aos outros, enquanto o nível <h6> tem o menor destaque por usar uma fonte menor em relação aos outros.

O <h1> deve ser o elemento do texto mais visível da página, ou seja, o título principal. O Google também entende que esse é o conteúdo principal, então é uma boa prática termos apenas um elemento de cabeçalho <h1> por página.

Quando precisamos definir um subtítulo, devemos utilizar um <h2>. Se houver outras seções dentro do <h2>, devemos usar o <h3>, e assim sucessivamente. Sempre que inserirmos qualquer cabeçalho, devemos efetuar o seu fechamento.

Vamos testar o código? Para isso utilize uma das ferramentas indicadas no capítulo anterior Notepad++, Atom ou Sublime. Abra a ferramenta, insira o código html, salve o arquivo com extensão nomequalquer.html e depois abra a pasta onde o arquivo foi salvo, clique duas vezes sobre ele, pronto, o navegador irá abrir e você poderá testar seu código.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Títulos</title>
  </head>
  <body>
    <h1>Nível de Cabeçalho 1</h1>
```

```
<h2>Nível de Cabeçalho 2</h2>
<h3>Nível de Cabeçalho 3</h3>
<h4>Nível de Cabeçalho 4</h4>
<h5>Nível de Cabeçalho 5</h5>
<h6>Nível de Cabeçalho 6</h6>
</body>
</html>
```

Código-fonte 2.1 – Estrutura de cabeçalhos
Fonte: Elaborado pelo autor (2017)

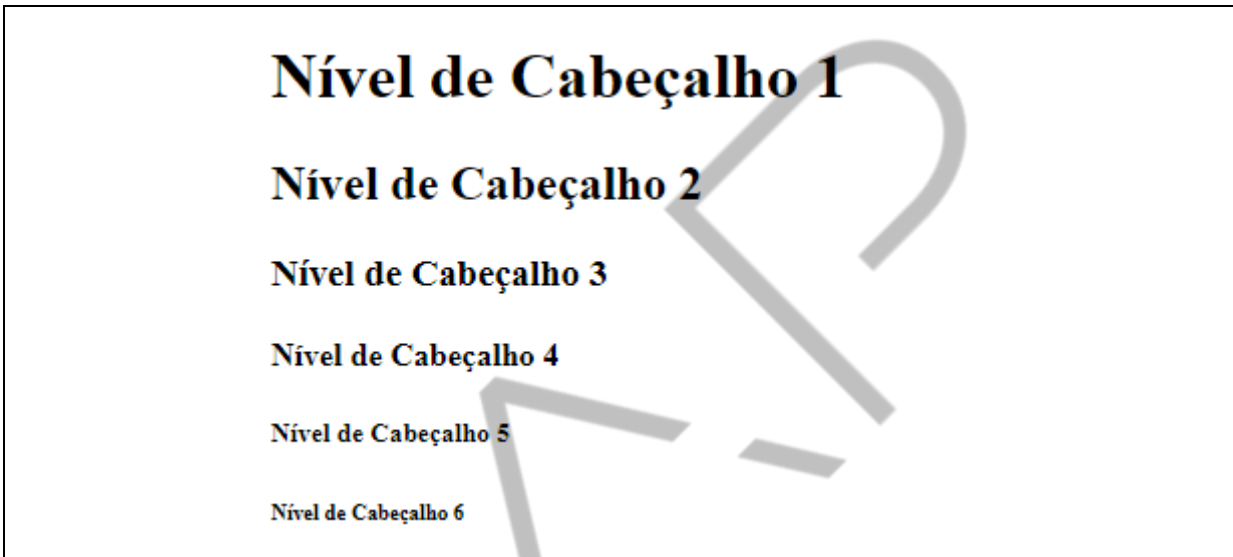


Figura 2.1 – Representação dos cabeçalhos no *browser*
Fonte: Elaborado pelo autor (2017)

Imagine que estamos fazendo uma página mostrando nossos jogos preferidos. Utilizando os níveis de cabeçalhos, poderíamos hierarquizar as informações da seguinte forma:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Títulos</title>
  </head>
  <body>
    <h1>Meus jogos preferidos</h1>
    <h2>Melhores jogos para plataforma A</h2>
      <h3>Jogo 1</h3>
      <h3>Jogo 2</h3>
      <h3>Jogo 3</h3>
    <h2>Melhores jogos para plataforma B</h2>
      <h3>Jogo 1</h3>
      <h3>Jogo 2</h3>
      <h3>Jogo 3</h3>
    <h2>Melhores jogos para plataforma C</h2>
```

```
<h3>Jogo 1</h3>  
<h3>Jogo 2</h3>  
<h3>Jogo 3</h3>  
  
</body>  
</html>
```

Código-fonte 2.2 – Estruturando informação em níveis de cabeçalhos
Fonte: Elaborado pelo autor (2017)

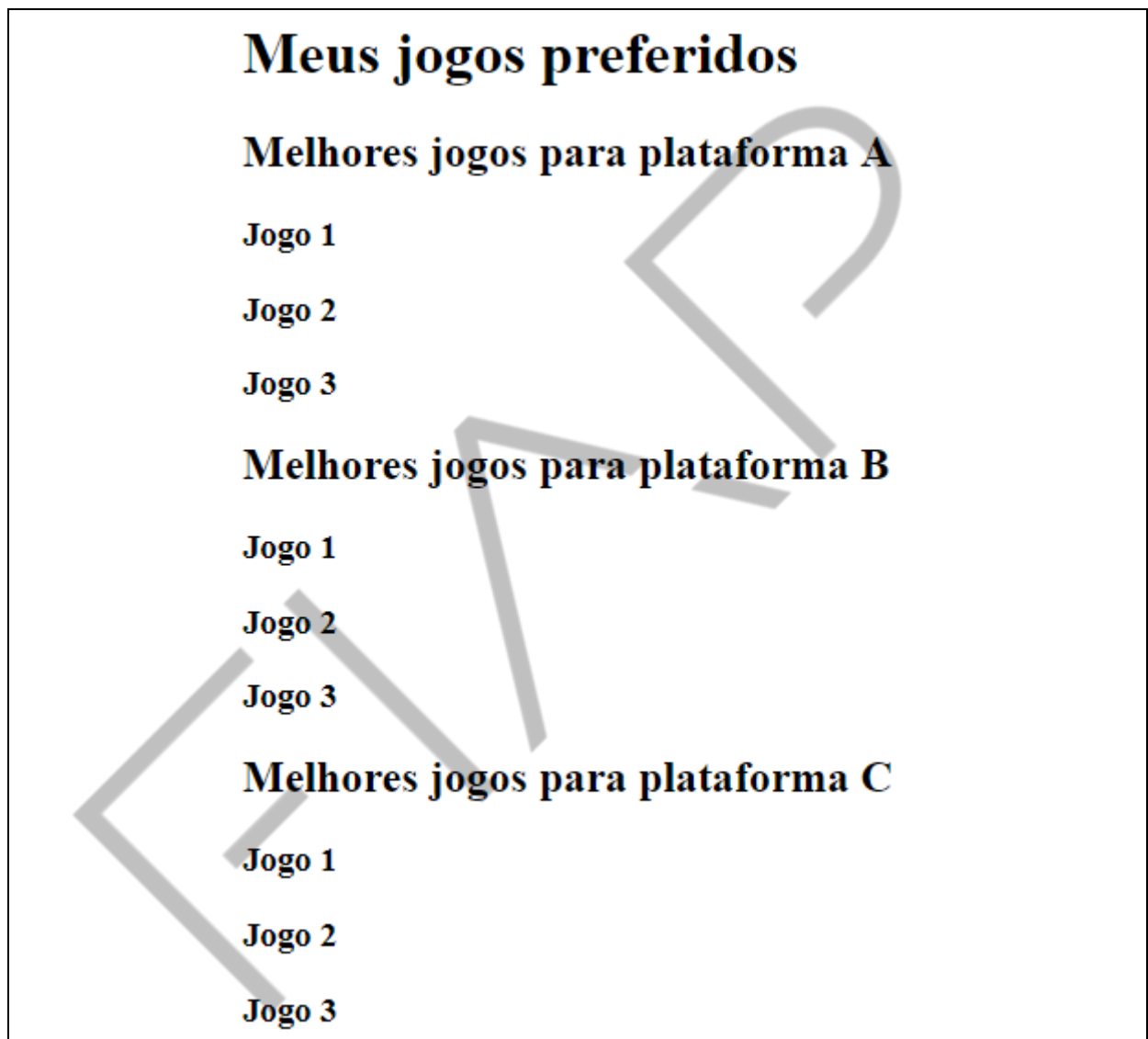


Figura 2.2 – Representação da informação em níveis de cabeçalhos no *browser*
Fonte: Elaborado pelo autor (2017)

Na página principal do site da **SPLIT**, (<http://splitstudio.com.br/>), um grande estúdio de animação digital do Brasil com clientes totalmente relevantes, essas *tags* são usadas para exibir algum título importante. No exemplo abaixo, foi usado o elemento `<h1>` para o título da área de contato e o elemento `<h2>` para o número do telefone da empresa.

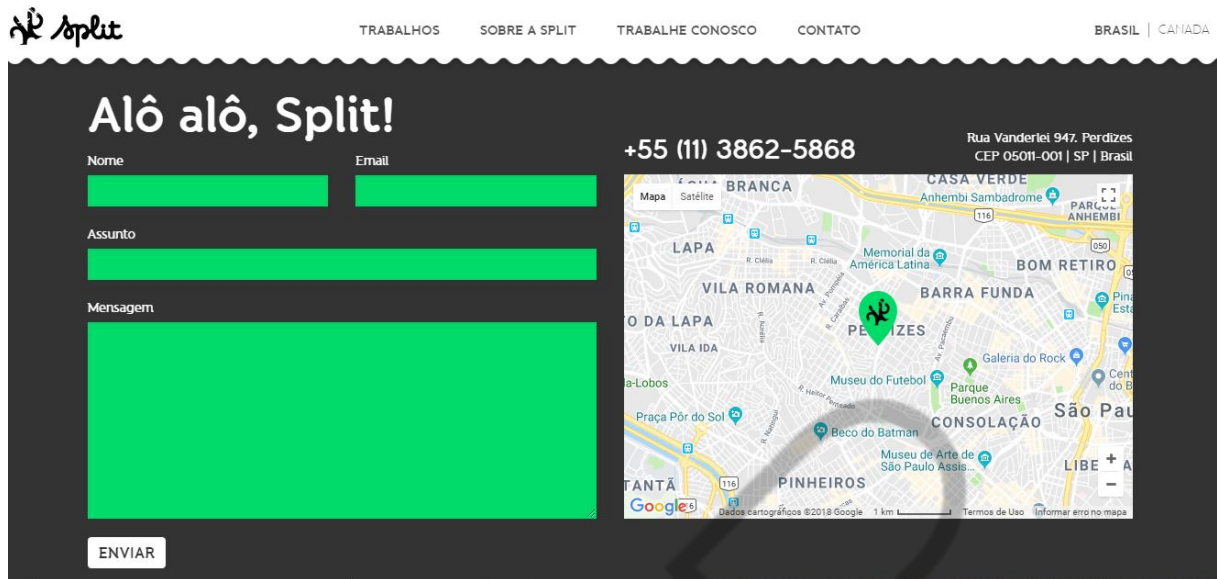


Figura 2.3 – Parte da página principal do site da Split – uso de <h1> e <h2>
Fonte: Split Studio (2018)

2.2 Parágrafos

A tag <p> irá definir um bloco de texto como sendo um parágrafo. Nosso browser exibirá cada parágrafo em uma quantidade determinada de linhas e dará um espaçamento padrão entre eles. Não se esqueça de fechar o parágrafo no final da linha, em seu código.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Parágrafos</title>
</head>
<body>
  <h1>Meus jogos preferidos</h1>
  <h2>Jogo 1</h2>
  <p>Lorem ipsum dolor sit amet, consectetur
adipisicing elit, sed do eiusmod tempor incididunt ut labore
et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud
exercitation ullamco laboris nisi ut aliquip ex ea commodo
consequat.</p>
  <h2>Jogo 2</h2>
  <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit,
sed do eiusmod tempor incididunt ut labore et dolore magna
aliqua. Ut enim ad minim veniam, quis nostrud exercitation
ullamco laboris nisi ut aliquip ex ea commodo consequat.</p>
</body>
</html>
```

Código-fonte 2.3 – Exemplo de cabeçalhos e parágrafos
Fonte: Elaborado pelo autor (2017)

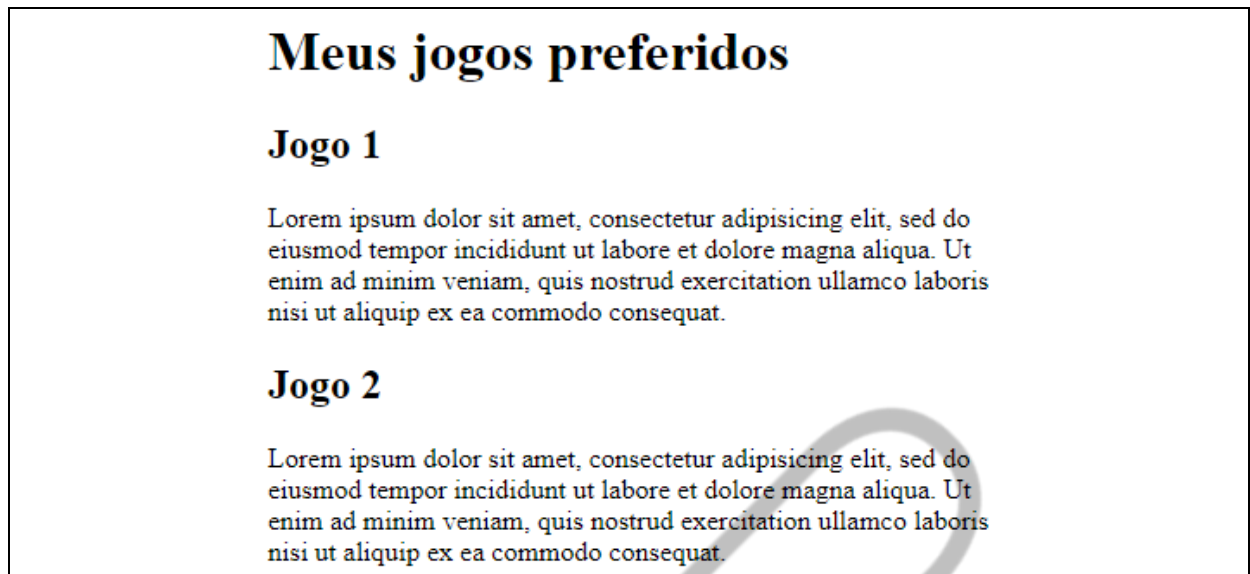


Figura 2.4 – Representação de cabeçalhos e parágrafos *browser*
Fonte: Elaborado pelo autor (2017)

Já no site do Xbox, (<https://www.xbox.com/pt-BR>), foi inserido um parágrafo curto fazendo uma chamada para os jogos do Xbox One.



Figura 2.5 – Parte da página principal da Xbox – uso de parágrafo
Fonte: Xbox (2018)

Podemos também forçar uma quebra de linha utilizando a *tag* `
`. Quando o browser encontrar essa marcação, o conteúdo que estiver à direita da *tag* será exibido uma linha abaixo.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Parágrafos</title>
</head>
<body>
  <h1>Meus jogos preferidos</h1>
  <h2>Jogo 1</h2>
  <p>Lorem ipsum dolor sit amet, consectetur adipisicing
elit, sed do eiusmod tempor incididunt ut labore et dolore
magna aliqua.<br /> Ut enim ad minim veniam, quis nostrud
exercitation ullamco laboris nisi ut aliquip ex ea commodo
consequat.</p>
</body>
</html>
```

Código-fonte 2.4 – Exemplo de parágrafo com

Fonte: Elaborado pelo autor (2017)

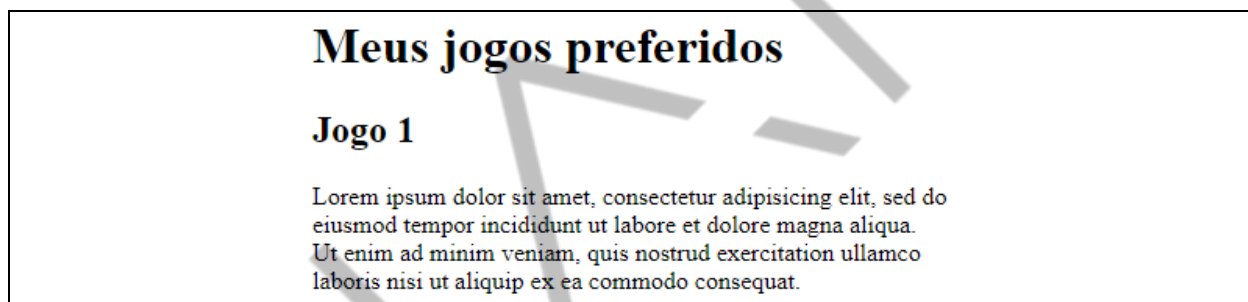


Figura 2.6 – Representação de parágrafo com
 no *browser*
Fonte: Elaborado pelo autor (2017)

2.3 Listas

Como fazemos no mundo real, o HTML também pode ser usado para a criação de listas, apresentando a vantagem de possuir três opções, como listadas a seguir.

2.3.1 Lista desordenada

Cada elemento recebe um marcador (um pequeno símbolo posicionado no lado esquerdo) que indica os respectivos itens da lista. Para criar essa lista, usamos a tag , na qual inserimos cada item da lista na tag .

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Lista desordenada</title>
</head>
```

```
<body>
  <h1>Lista desordenada</h1>
  <ul>
    <li>Jogo 1</li>
    <li>Jogo 2</li>
    <li>Jogo 3</li>
  </ul>
</body>
</html>
```

Código-fonte 2.5 – Exemplo de uma lista desordenada
Fonte: Elaborado pelo autor (2017)

Lista desordenada

- Jogo 1
- Jogo 2
- Jogo 3

Figura 2.7 – Representação de uma lista desordenada no *browser*
Fonte: Elaborado pelo autor (2017)

2.3.2 Lista ordenada

Cada elemento da lista é numerado, ou seja, possui uma ordem. Pode ser um número decimal, um número romano ou uma letra. Para criar essa lista, usamos a tag ``, na qual inserimos cada item da lista na tag ``.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Lista ordenada</title>
</head>
<body>
  <h1>Lista ordenada</h1>
  <ol>
    <li>Jogo A</li>
    <li>Jogo B</li>
    <li>Jogo C</li>
  </ol>
</body>
</html>
```

Código-fonte 2.6 – Exemplo de uma lista ordenada
Fonte: Elaborado pelo autor (2017)

Lista ordenada

1. Jogo A
2. Jogo B
3. Jogo C

Figura 2.8 – Representação de uma lista ordenada no *browser*
Fonte: Elaborado pelo autor (2017)

2.3.3 Lista de definição

Conjunto de termos e suas respectivas definições. Para criar essa lista usamos a tag `<dl></dl>`, na qual inserimos cada item a ser definido na tag `<dt></dt>` e sua definição em uma tag `<dd></dd>`.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Lista de definição</title>
</head>
<body>
  <h1>Lista de definição</h1>
  <dl>
    <dt>HTML</dt>
    <dd>HyperText Markup Language</dd>
    <dt>CSS</dt>
    <dd>Cascading Style Sheets</dd>
  </dl>
</body>
</html>
```

Código-fonte 2.7 – Exemplo de uma lista de definição
Fonte: Elaborado pelo autor (2017)

Lista de definição

HTML
HyperText Markup Language
CSS
Cascading Style Sheets

Figura 2.9 – Representação de uma lista de definição no *browser*
Fonte: Elaborado pelo autor (2017)

Podemos, ainda, ter as chamadas listas aninhadas, ou seja, uma lista dentro de outra.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Listas aninhadas</title>
</head>
<body>
  <h1>Listas aninhadas</h1>
  <ul>
    <li>Jogo 1</li>
    <li>Jogo 2
      <ul>
        <li>Personagem</li>
        <li>Força</li>
        <li>Velocidade</li>
      </ul>
    </li>
    <li>Jogo 3</li>
  </ul>
</body>
</html>
```

Código-fonte 2.8 – Exemplo de uma lista de aninhada
Fonte: Elaborado pelo autor (2017)

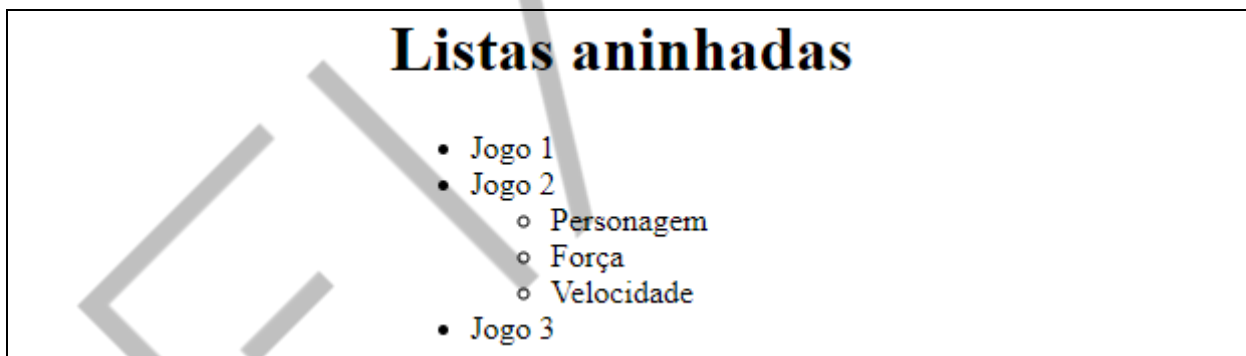


Figura 2.10 – Representação de uma lista aninhada no *browser*
Fonte: Elaborado pelo autor (2017)

A utilização de listas é muito comum em praticamente todos os sites, principalmente para a criação de menus de navegação, como veremos um pouco mais à frente. Abaixo, temos o exemplo do site da nova versão do aclamado game *God Of War*, (<https://godofwar.playstation.com/>), o menu apresentado na parte superior da página foi feito com as listas do HTML.

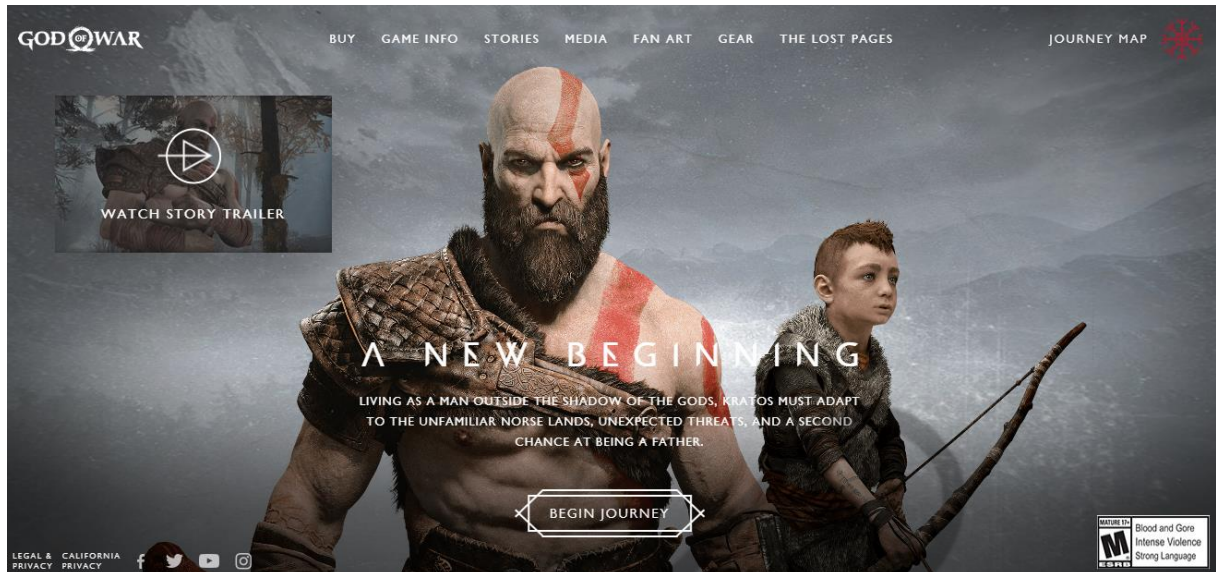


Figura 2.11 – Parte da página principal do game God of War – menu com lista
Fonte: God of War (2018)

2.4 Imagens

O HTML também permite a inclusão de imagens em suas páginas. Para isso, usamos a tag ``, que dispõe de alguns atributos que podem melhorar a exibição no navegador. A tag `` não possui fechamento, mas podemos, no final da linha, inserir um sinal de barra (/), que indica o fim da tag. Os atributos permitidos são:

- `src`: indica, ao navegador, a localização do arquivo da imagem.
- `alt`: é um texto alternativo que será exibido caso a imagem não apareça. Esse texto deve ser uma descrição precisa da imagem, pois alguns *softwares* de leitura de tela são usados por pessoas com deficiência visual, e mecanismos de busca também fazem uso dele.
- `title`: é um texto que será exibido assim que você posicionar o ponteiro do mouse sobre a imagem, trate-o como informação adicional sobre a imagem. Praticamente todos os elementos HTML possuem esse atributo.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Inserindo imagens</title>
</head>
<body>
  <h2> Imagens </h2>
```

```
  
  
  
</body>  
</html>
```

Código-fonte 2.9 – Exemplo de inserção de imagens

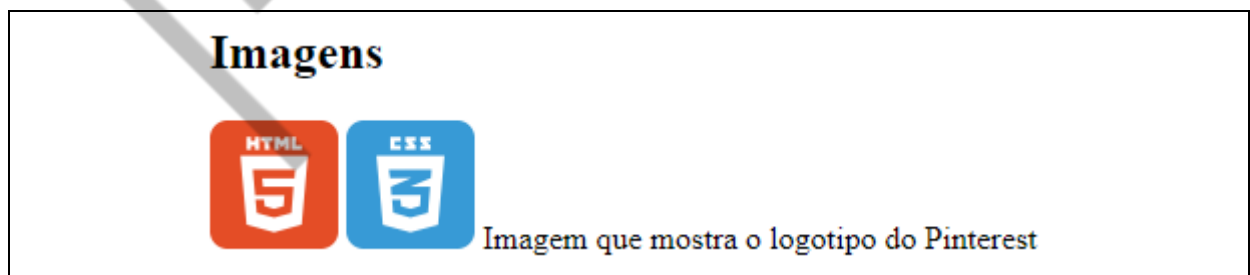
Fonte: Elaborado pelo autor (2017)

Figura 2.12 – Representação de imagens no *browser*

Fonte: Elaborado pelo autor (2017)

Como as três imagens são pequenas, uma fica ao lado da outra, caso elas fossem maiores, uma ficaria abaixo da outra. Imagem é um elemento que permite novos conteúdos ao seu lado, desde que tenha espaço para isso.

A figura abaixo mostra o que acontece quando o navegador não consegue exibir uma imagem na página. Perceba que o texto inserido no atributo alt será visualizado, portanto, ele é de suma importância e você não deve deixar de preenchê-lo.

Figura 2.13 – Texto alternativo sendo exibido no *browser*

Fonte: Elaborado pelo autor (2017)

O clássico game *Shadow of the Colossus*, (<https://www.playstation.com/pt-br/games/shadow-of-the-colossus-ps4>), possui imagens fantásticas, então, nada melhor do que aproveitá-las. O site oficial faz uso desse recurso, criando várias galerias de imagens.

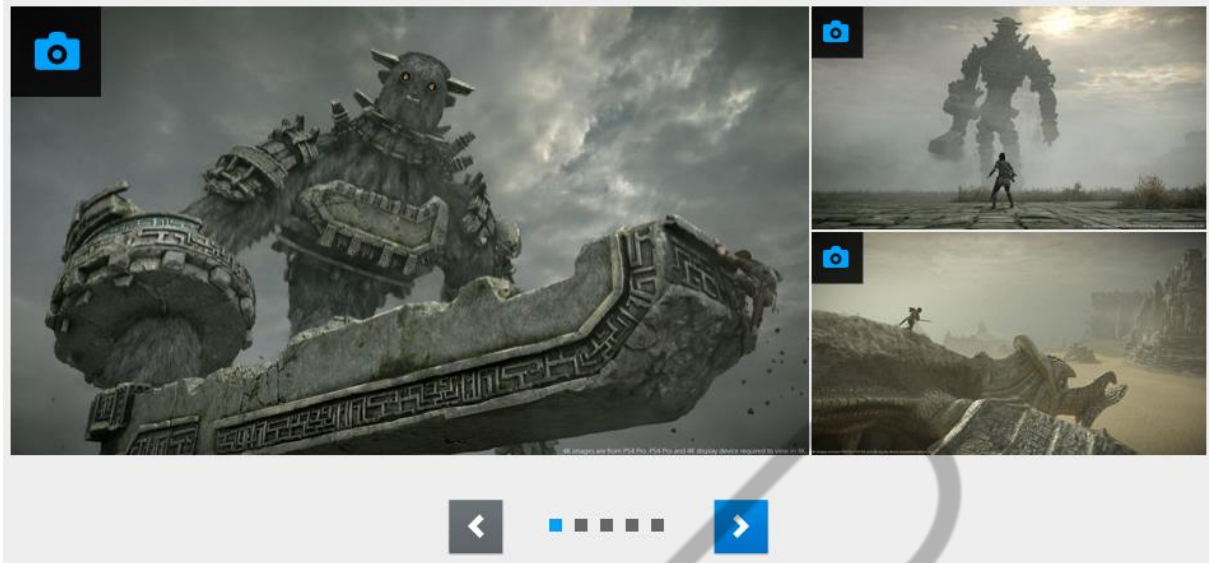


Figura 2.14 – Parte da página do game Shadow of the Colossus - imagens
Fonte: Play Station (2018)

Dicas para imagens:

- Use imagens no padrão gif, jpg, png ou svg.
- Imagens devem ter um tamanho padrão, portanto, sempre pegue ou crie imagens do mesmo tamanho.
- Vamos supor que, na sua página, você tenha uma área delimitada de 400px x 400px para inserir uma imagem, faça com que ela também tenha exatamente esse tamanho, utilizando um *software* gráfico, pois, se você aumentar ou reduzir o tamanho da imagem, ela poderá ficar distorcida e/ou estendida, dando uma má impressão ao usuário.
- Use sempre a resolução correta, a maioria das telas de computador só exibe páginas em 72 pixels por polegada, sendo assim, se você salvar uma imagem em uma resolução maior, ela apenas levará mais tempo para carregar e não ficará com melhor definição.
- Imagem é exibida, pois é feita uma requisição, ou seja, se sua página tem 50 imagens, serão feitas 50 requisições ao servidor. Isso poderá retardar o seu carregamento e o usuário pode não esperar.

2.5 CSS

A CSS (*Cascading Style Sheets* – Folhas de Estilo em Cascata) é um dos mais poderosos recursos de formatação para páginas Web. Utilizamos CSS para estilizar qualquer elemento HTML por meio do que chamamos de regras, ou seja, determinamos como uma determinada *tag* deverá ser exibida no navegador. CSS cuida da parte visual da página e é uma recomendação W3C para o desenvolvimento de *sites*. Atualmente, estamos usando CSS3, mas já está em estudo a versão 4.

Para criarmos uma regra CSS, precisamos de três elementos:

Seletor { propriedade: valor; }

- **Seletor** é o elemento que desejamos formatar, pode ser uma tag, um id ou uma classe.
- **Propriedade** é aquilo que desejamos formatar no seletor, pode ser a cor, o tamanho, a altura, a largura, enfim, algum atributo do seletor.
- **Valor** é como queremos que o seletor fique, ou seja, se vai ficar na cor azul, se vai ficar mais alto, se vai ficar mais largo etc.

Toda regra CSS sempre será escrita dessa forma e deve obedecer à ordem que foi apresentada, bem como a utilização de chaves { } para separar o seletor da regra, o uso de dois pontos (:) para separar a propriedade do valor e ponto e vírgula (;) para informar que aquela formatação acabou.

Podemos ter, na mesma regra, a formatação de várias propriedades de um seletor. Exemplo: queremos que o título h1 da página fique com o texto na cor azul, com tamanho de 40px.

```
h1 {color: blue;  
    size: 40px;}
```

Código-fonte 2.10 – Exemplo de formatação CSS
Fonte: Elaborado pelo autor (2017)

A regra acima fará a formatação estipulada, aplicando essa formatação a todas as *tags* h1 da página. Se tivermos cinco elementos h1 na mesma página, todos eles serão afetados; não se preocupe caso você deseje formatar apenas um dos elementos, mais adiante aprenderemos como controlar melhor as regras CSS.

2.5.1 Inserindo CSS na página

Existem três formas de inserirmos CSS em nossas páginas, como apresentado a seguir.

2.5.1.1 CSS inline

A formatação do elemento será feita diretamente na linha do código HTML que desejamos estilizar por meio do atributo **<style>**. Essa forma trará prejuízo ao desenvolvimento e não deverá ser utilizada, pois:

- mistura código de formatação CSS com conteúdo HTML;
- deixa poluído o código da página;
- gera dificuldade na manutenção no código;
- formata apenas o elemento em que foi inserido o atributo **<style>** e
- por questão de boa prática a formatação, a CSS deve ser separada do conteúdo HTML.

```
<h1 style="color:blue; font-size:40px;"> Meu jogo </h1>
```

Código-fonte 2.11 – Exemplo de formatação CSS *inline*
Fonte: Elaborado pelo autor (2017)

Observe que inserimos, na tag h1, o atributo **style**, que irá efetuar a sua formatação, deixando a cor do texto azul, com tamanho de 40px.



Meu jogo

Figura 2.15 – Resultado da regra CSS inline
Fonte: Elaborado pelo autor (2017)

2.5.1.2 CSS interno

É chamado assim, pois o código CSS é inserido em uma tag **<style></style>** que geralmente é armazenada dentro do elemento **<head>**. A formatação será exclusiva e aplicada somente ao documento que possuir a *tag* com suas respectivas

regras, ou seja, se seu o *site* possuir outras páginas, elas deverão receber o mesmo código para que fiquem iguais.

Use essa forma se seu o site tiver apenas uma página, mesmo assim, não é considerada uma boa prática, pois misturamos conteúdo HTML com formatação CSS. Na *tag* `<style>`, podemos definir que os estilos serão descritos em CSS, para isso, definimos o seu atributo `type` como sendo “text/css”.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Usando CSS interno</title>
<style type="text/css">
  h2{
    color: red;
    font-size: 40px;
  }
</style>
</head>
<body>
  <h2>Meu jogo</h2>
</body>
</html>
```

Código-fonte 2.12 – Exemplo de formatação CSS interna
Fonte: Elaborado pelo autor (2017)



Figura 2.16 – Exibição da regra CSS interna
Fonte: Elaborado pelo autor (2017)

2.5.1.3 CSS externo

Com certeza, é a forma mais inteligente de inserirmos CSS em uma página. Para isso, criamos um arquivo separado do HTML que terá as regras de formatação que queremos para cada elemento do documento, esse arquivo deverá possuir a extensão CSS.

Outra grande vantagem é que poderemos linkar esse arquivo a vários outros documentos HTML, permitindo, assim, uma padronização de estilos e uma manutenção de código mais eficiente, pois basta alterar um único arquivo, no caso o

CSS, que a alteração replicará para todas as páginas que estão ligadas a ele; essa é considerada uma boa prática de desenvolvimento web.

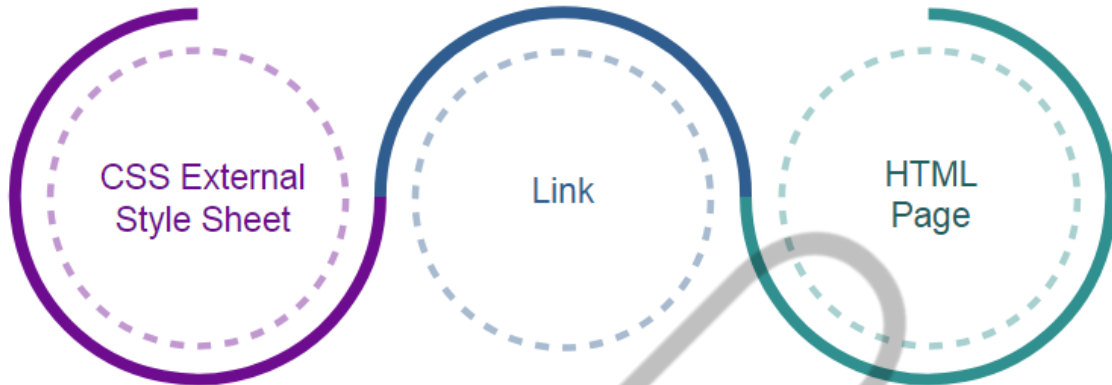


Figura 2.17– Representação CSS Externo
Fonte: FIAP (2018)

Para linkarmos um arquivo CSS a uma ou mais páginas HTML, basta inserir a seguinte linha de comando na seção **<head>** de cada documento:

```
<head>  
<link rel="stylesheet" type="text/css" href="arquivo.css"  
</head>
```

Código-fonte 2.13 – Exemplo de link com arquivo CSS
Fonte: Elaborado pelo autor (2017)

Nessa linha, temos os seguintes atributos:

- Link rel: relação entre a página HTML e o arquivo que está sendo chamado. Quando indicamos que é um arquivo CSS, seu valor deverá ser stylesheet.
- Type: determina o tipo de documento que está sendo chamado, nesse caso, o valor deverá ser text/css.
- Href: determina o local de origem do arquivo e o seu nome, o valor deverá ser o nome do seu arquivo CSS que contém as regras de formatação.

A hora é de praticar esses conceitos. Para treinar, aproveite este conteúdo e monte um site, defina a estrutura ou, pelo menos, um primeiro protótipo de como ficaria a página de divulgação do seu conteúdo.

REFERÊNCIAS

DUCKETT, J. **HTML&CSS projete e construa websites**. Rio de Janeiro: Alta Books, 2014.

FERREIRA, E.; EIS, D. **HTML5 – Curso W3C Escritório Brasil**. 2014. Disponível em: <<http://www.w3c.br/pub/Cursos/CursoHTML5/html5-web.pdf>>. Acesso em: 12 nov. 2020.

SILVA, M. S. **HTML5 – A linguagem de marcação que revolucionou a Web**. São Paulo: Novatec, 2011.

TERUEL, E. C. **HTML5 – Guia prático**. São Paulo: Erica, 2011.