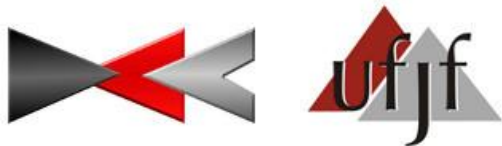

Código fonte, codificação e linguagens de marcação de texto

UFJF - DCC202 - Desenvolvimento Web

Prof. Igor Knop igor.knop@ufjf.br



<https://bit.ly/3McsF0k>

Objetivos

- Saber controlar a codificação de arquivos texto;
- Entender o que são linguagens de marcação de texto;
- Criar documentos estruturados com linguagem de marcação.

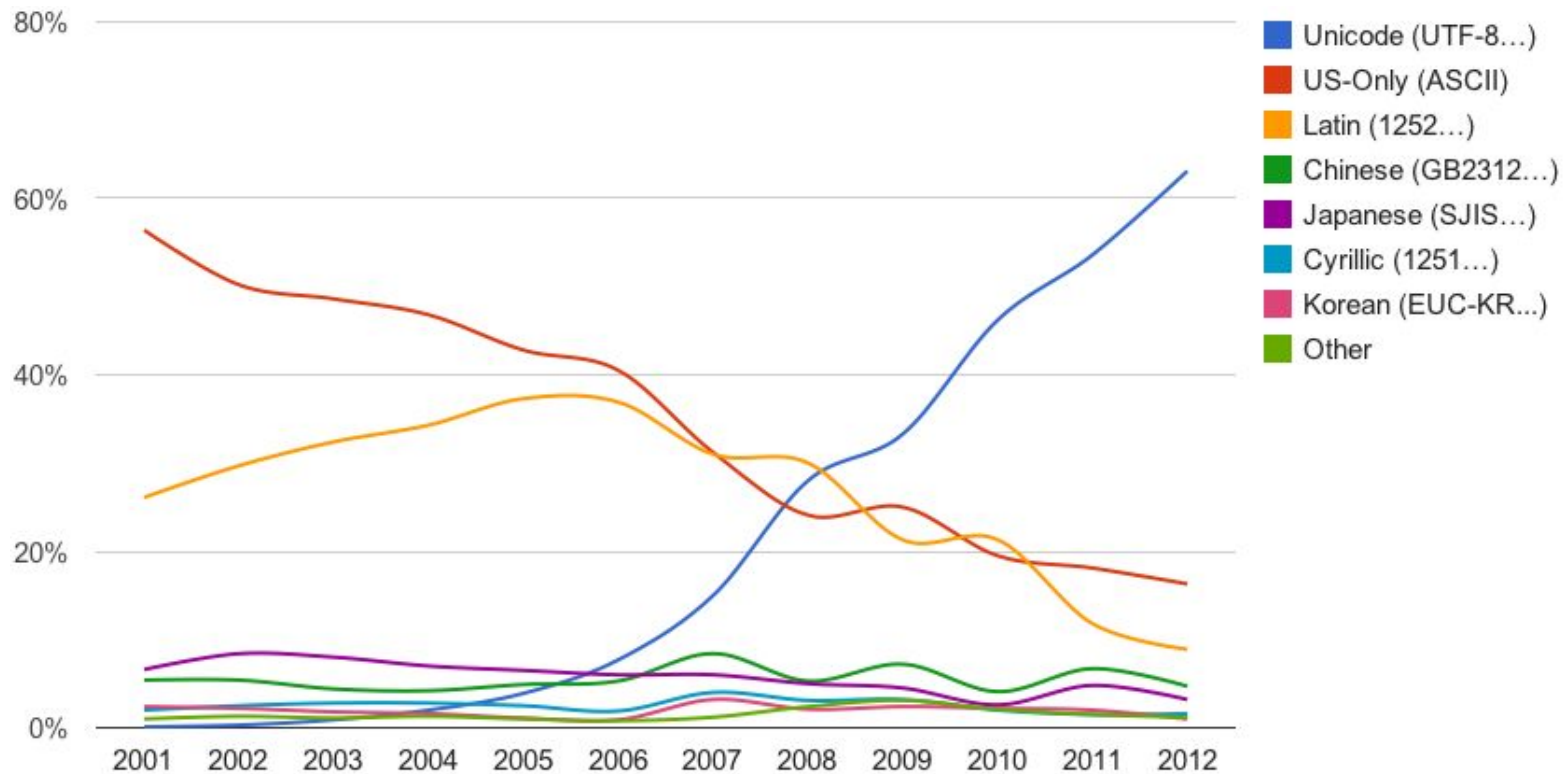
Arquivos em Texto Puro (plain text files)

- Arquivos Texto puro/plain text: são os que contêm dados representados apenas por símbolos legíveis e nenhum outro código de formatação ou estilo;
- É formado por uma sequência de bytes representando caracteres;
- Caracteres especiais exigem mais de um Byte;
- A regra que diz exatamente quantos Bytes um caractere especial possui chama-se **codificação** ou **encoding**;
- Editores de texto puro geram o arquivo texto em uma determinada codificação;
- Um leitor deve saber em qual codificação o arquivo foi criado para uma visualização correta.

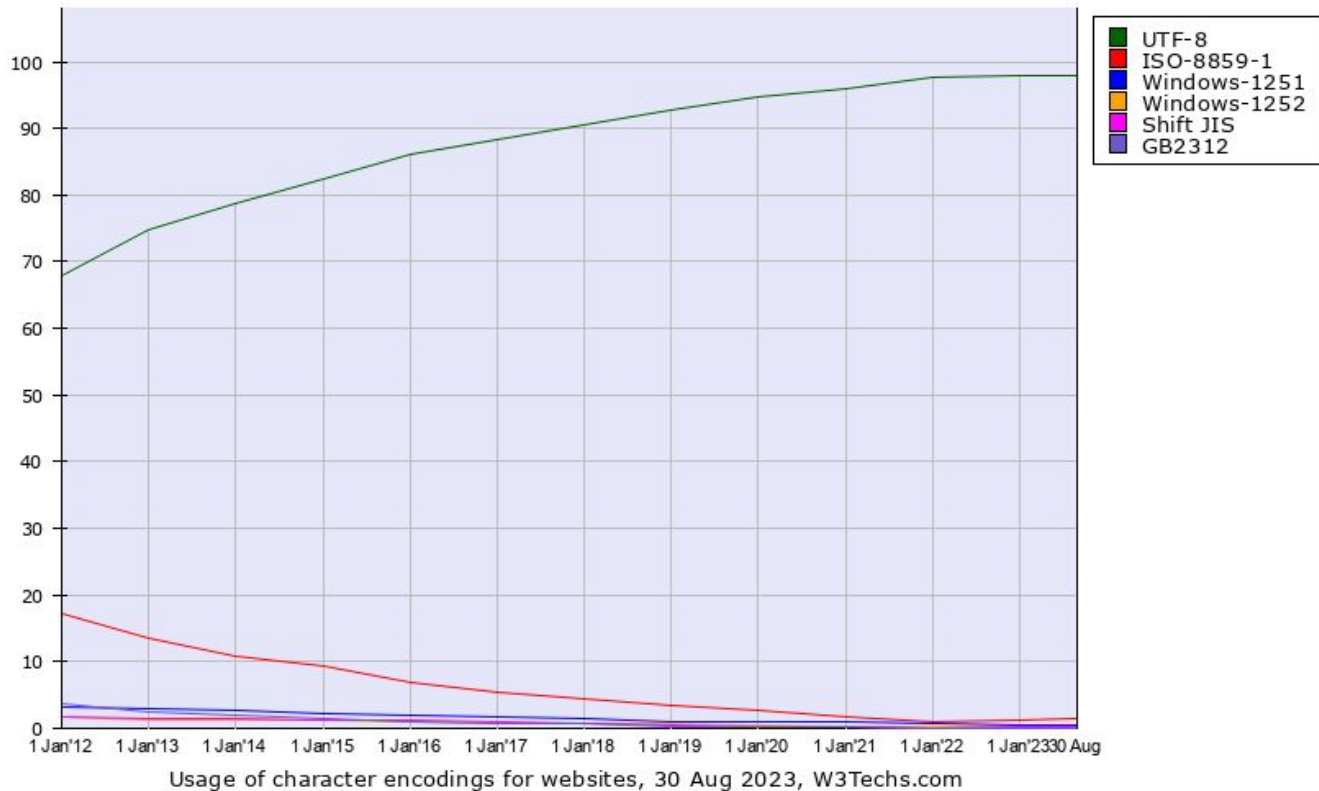
Unicode

- Criado para fornecer conteúdo em qualquer linguagem humana existente
- Fornece um meio de associar um número único a cada símbolo
- Evita conflito entre computadores, programas e documentos
- É um padrão internacional amplamente adotado e tido como padrão em vários sistemas operacionais modernos
- É recomendado pela W3C como a codificação mínima necessária que um agente de usuário (navegador) deve ter suporte

Unicode



Unicode hoje



fonte: https://w3techs.com/technologies/history_overview/character_encoding/ms/y

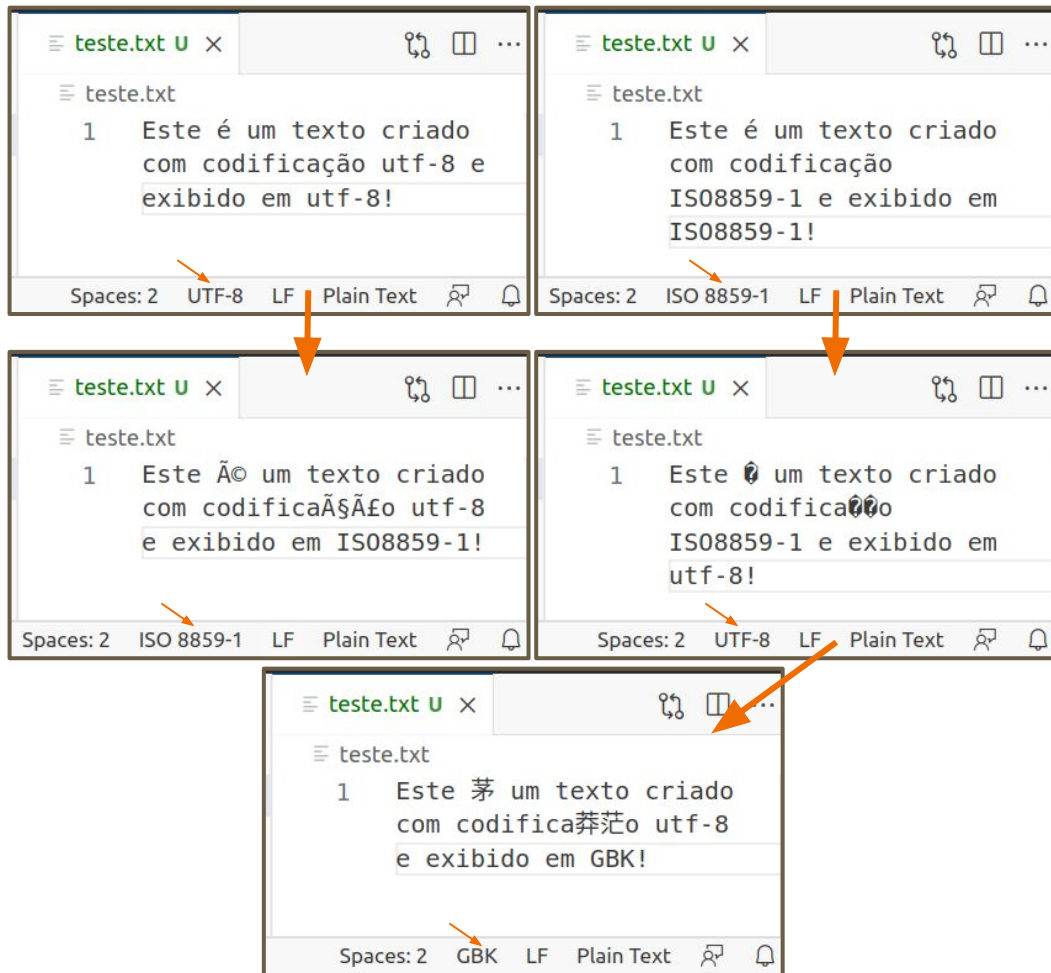
O editor escreve. O leitor lê.

É importante conhecer em qual codificação o seu editor está criando o arquivo texto.

Se o leitor tentar abrir o arquivo com uma codificação diferente, o texto aparece com caracteres estranhos.

Boa parte dos problemas de codificação na web e bancos de dados pode ser resolvida ao observar se a codificação está sendo mantida ou convertida corretamente.

Nesta disciplina: vamos usar o utf-8 para editar os arquivos e informar ao navegador (leitor) qual é a codificação das páginas.



Linguagem de marcação

- É uma forma de se criar documentos estruturados utilizando apenas texto puro.
- Trechos do texto recebem uma marcação.
- Essa marcação possui um significado associado.
- Quando esse texto marcado é processado, os seus elementos podem ter exibição ou função redefinidos de acordo com a aplicação.
- É uma forma de facilitar a interpretação do conteúdo para o computador sem usar linguagem natural.

Exemplo abstrato: análise gramatical de uma frase

O gato subiu no telhado

Em português, podemos fazer uma análise gramatical:

- Sujeito: o gato
- Predicado: subiu no telhado
- Verbo: subiu
- Objeto direto: no telhado

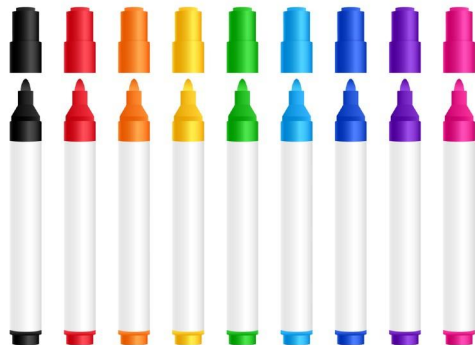
Mas como indicar isso para o computador usando linguagem de marcação?

Exemplo abstrato: análise gramatical de uma frase

O gato subiu no telhado

Em português, podemos fazer uma análise gramatical:

- Sujeito: o gato
- Predicado: subiu no telhado
- Verbo: subiu
- Objeto direto: no telhado



Mas como indicar isso para o computador usando linguagem de marcação?

Linguagem de marcação (2)

Para descrever uma estrutura usando um arquivo texto precisamos usar um sistema de marcação. As linguagens de marcação de texto se baseiam em etiquetas embutidas no próprio texto puro.

Sistema de etiquetas marcar o início e fim

- Etiqueta de início/abertura:
<etiqueta>
- Etiqueta de fim/fechamento:
</etiqueta>
- Etiqueta vazia:
<etiqueta></etiqueta>
<etiqueta />

Exemplo: marcando a frase

Primeiro vamos marcar a frase: onde ela começa e onde termina.

- **Início e fim da frase**
- Sujeito: o gato
- Predicado: subiu no telhado
- Verbo: subiu
- Objeto direto: no telhado

<frase>0 gato subiu no telhado</frase>

Exemplo: marcando o sujeito

Primeiro vamos marcar a frase: onde ela começa e onde termina.

- Início e fim da frase
- **Sujeito: o gato**
- Predicado: subiu no telhado
- Verbo: subiu
- Objeto direto: no telhado

<frase><sujeito>O gato</sujeito> subiu no telhado</frase>

Exemplo: marcando o predicado

Primeiro vamos marcar a frase: onde ela começa e onde termina.

- Início e fim da frase
- Sujeito: o gato
- **Predicado: subiu no telhado**
- Verbo: subiu
- Objeto direto: no telhado

```
<frase><sujeito>O gato</sujeito> <predicado>subiu no telhado</predicado></frase>
```

Exemplo: marcando o verbo

Primeiro vamos marcar a frase: onde ela começa e onde termina.

- Início e fim da frase
- Sujeito: o gato
- Predicado: subiu no telhado
- **Verbo: subiu**
- Objeto direto: no telhado

```
<frase><sujeito>O gato</sujeito> <predicado><verbo>subiu</verbo> no  
telhado</predicado></frase>
```

Exemplo: marcando o objeto direto

Primeiro vamos marcar a frase: onde ela começa e onde termina.

- Início e fim da frase
- Sujeito: o gato
- Predicado: subiu no telhado
- Verbo: subiu
- **Objeto direto: no telhado**

```
<frase><sujeito>O gato</sujeito> <predicado><verbo>subiu</verbo>  
  <objetoDireto>no telhado</objetoDireto></predicado></frase>
```


Exemplo: reorganizando a estrutura

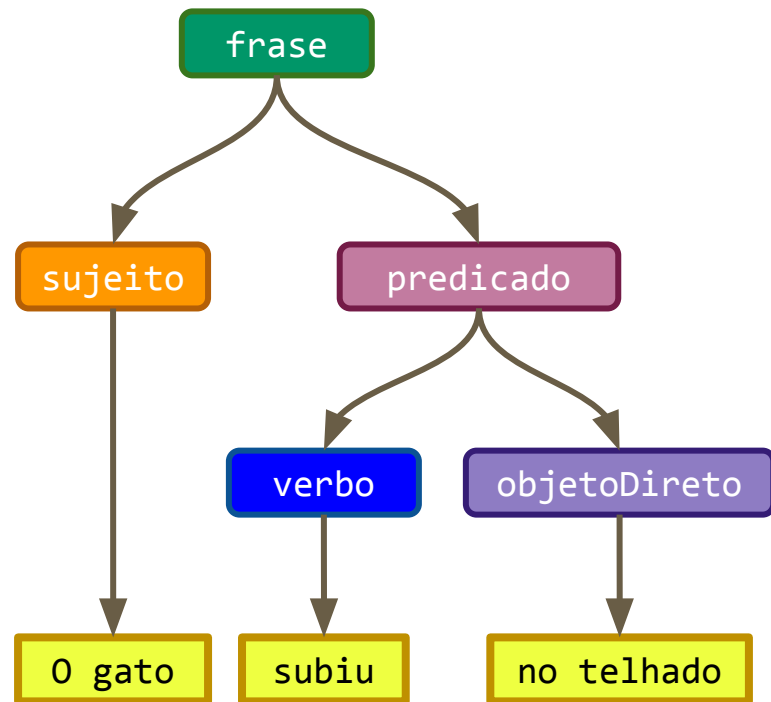
```
<frase>  
  <sujeito>O gato</sujeito>  
  <predicado>  
    <verbo>subiu</verbo>  
    <objetoDireto>no telhado</objetoDireto>  
  </predicado>  
</frase>
```

O código pode ser reorganizado usando espaços em branco, quebras de linha e tabulações para ficar mais legível por humanos.

Os navegadores e interpretadores sabem lidar com os espaços em branco e não vai prejudicar o seu funcionamento.

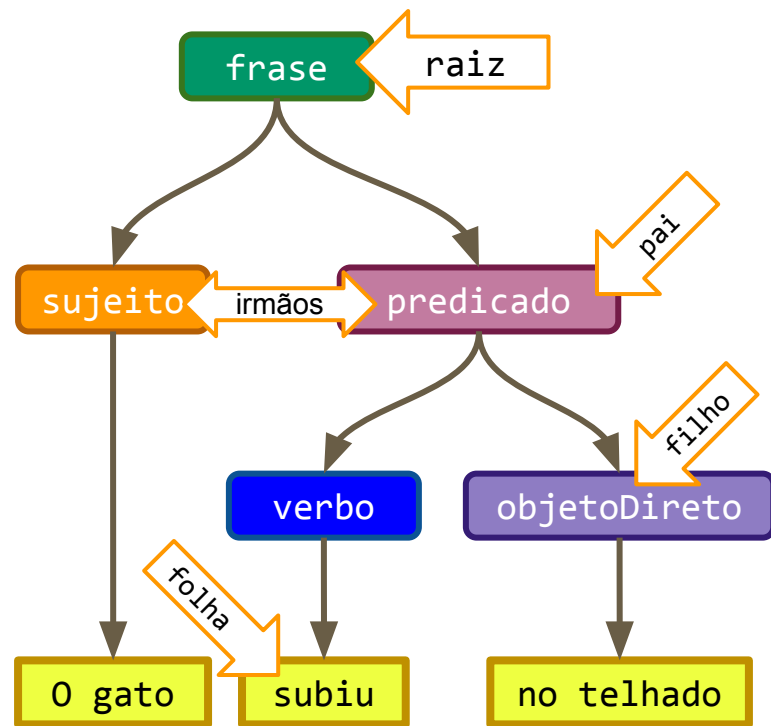
Exemplo: reorganizando a estrutura como árvore

```
<frase>  
  <sujeito>O gato</sujeito>  
  <predicado>  
    <verbo>subiu</verbo>  
    <objetoDireto>no telhado</objetoDireto>  
  </predicado>  
</frase>
```



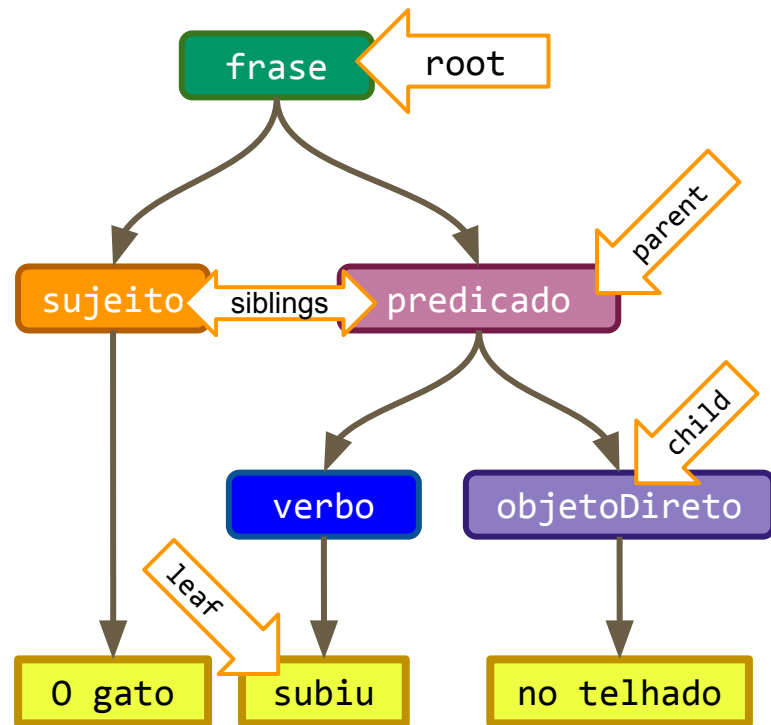
Exemplo: reorganizando a estrutura como árvore

```
<frase>  
  <sujeito>O gato</sujeito>  
  <predicado>  
    <verbo>subiu</verbo>  
    <objetoDireto>no telhado</objetoDireto>  
  </predicado>  
</frase>
```



Exemplo: reorganizando a estrutura como árvore

```
<frase>  
  <sujeito>O gato</sujeito>  
  <predicado>  
    <verbo>subiu</verbo>  
    <objetoDireto>no telhado</objetoDireto>  
  </predicado>  
</frase>
```



Linguagem de marcação: atributos

Os chamados atributos permitem acrescentar dados a um elemento HTML.

- Devem sempre ser definidos na etiqueta inicial;
- Devem ter um identificador ou nome;
- Podem ser associados um valor, representado por um símbolo de igual "="
- Recomenda-se envolver o valor em aspas duplas ou simples e é obrigatório quando este incluir espaços em branco;
- Etiquetas podem ter vários atributos simultaneamente, separados por espaço.

```
<etiqueta atributo=5></etiqueta>
```

```
<etiqueta atributoUm="Mês 5" atributoDois='Dia 12'></etiqueta>
```

A linguagem de marcação define parte dos atributos

Atributos são considerados dados extras de um elemento:

- Alguns mudam o comportamento ou exibição do elemento, mas não são conteúdo!
- A linguagem de marcação vai definir alguns atributos para cada elementos e outros gerais. Por exemplo, o HTML define:
 - atributo **id** para representar uma identidade de um elemento no documento. Seu valor deve ser único na página!
 - atributo **class** representará uma ou mais famílias a qual o elemento pertence. Vários elementos podem estar na mesma família!
 - vários atributos prefixados com **data-** para definir dados específicos da aplicação;
 - vários atributos referentes à acessibilidade do conteúdo o aplicação.

HTML: Estrutura básica

```
<!DOCTYPE html>
<html lang="pt">
  <head>
    <meta charset="UTF-8" />
    <title></title>
  </head>
  <body>

  </body>
</html>
```

Nesta disciplina, sempre que nos referirmos a “documento HTML” ou “documento HTML mínimo”, estaremos nos referindo à estrutura ao lado.

O documento terá uma extensão de arquivos .html e teremos que ter consciência de qual codificação o editor está configurado, preferindo o utf-8.

Vamos discutir cada trecho a seguir...

HTML: Estrutura básica (2)

```
<!DOCTYPE html>
<html lang="pt">
  <head>
    <meta charset="UTF-8" />
    <title></title>
  </head>
  <body>

  </body>
</html>
```

Descreve como o navegador deve interpretar o seu código.

Caso ausente, o navegador entra em modo de compatibilidade (*quirks mode*) e renderiza seu site como uma página antiga.

HTML: Estrutura básica (3)

```
<!DOCTYPE html>
<html lang="pt">
  <head>
    <meta charset="UTF-8" />
    <title></title>
  </head>
  <body>

  </body>
</html>
```

Nó raiz do documento. Marca todo o texto que fará parte do documento HTML.

Qualquer código fora dele, está fora da especificação (mas o navegador vai tentar corrigir!).

Podemos adicionar um atributo lang especificando a linguagem em todo o documento. Isso ajuda ao navegador a sugerir tradução ou correção de texto.

HTML: Estrutura básica (4)

```
<!DOCTYPE html>
<html lang="pt">
  <head>
    <meta charset="UTF-8" />
    <title></title>
  </head>
  <body>

  </body>
</html>
```

"Cabeça" do documento: guarda dados sobre os dados (metadados) que não fazem parte do conteúdo.

Vários desses dados estão diretamente relacionados em como sua página é exibida, mas não é um lugar para inserir conteúdo.

A cabeça conversa com o seu navegador e não com o seu leitor!

HTML: Estrutura básica (5)

```
<!DOCTYPE html>  
<html lang="pt">  
  <head>  
    <meta charset="UTF-8" />  
    <title></title>  
  </head>  
  <body>  
  
    </body>  
</html>
```

Define a codificação que o documento foi escrito.

Essa codificação deve seguir a mesma que você selecionar no seu editor de textos.

Se houver diferença entre a codificação do seu editor e este metadado, o seu navegador vai exibir caracteres estranhos em alguns lugares.

HTML: Estrutura básica (6)

```
<!DOCTYPE html>
<html lang="pt">
  <head>
    <meta charset="UTF-8" />
    <title></title>
  </head>
  <body>

  </body>
</html>
```

Metadado que define o título do documento.

É o texto que aparece na aba/título do navegador.

Ele pode e será usado para dar uma direção para o leitor encontrar sua página.

HTML: Estrutura básica (7)

```
<!DOCTYPE html>
<html lang="pt">
  <head>
    <meta charset="UTF-8" />
    <title></title>
  </head>
  <body>

  </body>
</html>
```

Elemento que envolve todo o conteúdo do documento que será apresentado no navegador do usuário.

O corpo do documento traz todo o conteúdo final da sua página ou aplicação.

O corpo conversa com seu leitor, tudo o que ele vê é o corpo do documento.

HTML: Estrutura básica* (8)

```
<!DOCTYPE html>
<html lang="pt">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport"
      content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
  </head>
  <body>
  </body>
</html>
```

Um metadado que não é considerado básico, mas é muito importante para acessibilidade e uso em dispositivos móveis é o **viewport**.

Ele diz ao navegador como desenhar a sua página ou aplicativo.

Você pode especificar um tamanho fixo ou, com o **width=device-width** falar que quer usar a largura do dispositivo.

O **initial-scale=1.0** define o *zoom* inicial.

Para saber mais...

- MOZILLA DEVELOPER NETWORK. Iniciando com HTML. 2020. Disponível em: https://developer.mozilla.org/pt-BR/docs/Aprender/HTML/Introducao_ao_HTML/Getting_started
- FRANCIS, Mark Norman. The basics of HTML. 2008. Available on Internet: <http://dev.opera.com/articles/view/12-the-basics-of-html>
- Basic Structure of a Web Page. Available on Internet: <http://reference.sitepoint.com/html/page-structure>
- HTML from the Ground Up. Available on Internet: <http://www.youtube.com/watch?v=KIYcgPZqITk>