

## **Um pouco de HTML5 e XHTML 2.0**

O conteúdo não é completo, é só uma pequena referência do que vem por aí.

# XHTML2

## Necessidades

- Que o XHTML fosse o mais genérico possível, em comparação ao XML;
- Menos apresentação e mais estrutura: uso de folhas de estilo para definir a apresentação;
- Mais usabilidade: tornar a linguagem mais simples de escrever e o documento resultante mais simples de usar;
- Mais acessibilidade;
- Melhor internacionalização;
- Mais independência de dispositivos;
- Integração com a Web Semântica.

## O que há mudanças

- Mais controle da estrutura: novo elemento `section` para ajudar a definir a estrutura do documento;
- O elemento `hr` agora é `separator`;
- As quebras de linha `br` agora viraram o elemento `l` que encapsula cada linha de texto. Capacidade de auto-numerar linhas, colorir linhas alternadamente;
- Permissão para incluir nas tags `p` listas e tabelas, o que antes não era permitido;
- Elemento `n1` para listas de navegação;
- Imagens: qualquer elemento pode ter um atributo `src`, que especifica um objeto a ser carregado ao invés do conteúdo. Se este não puder ser carregado, o conteúdo original encapsulado é carregado ao invés do atributo `src`;
- O modelo de conteúdo do elemento `table` foi limpo e simplificado;
- Atributo `edit` criado para substituir os elementos `ins` e `del`;
- Qualquer elemento agora pode conter o atributo `href`;
- O atributo `role` foi adicionado com conjunto de valores pré-definidos para detalhar mais a função do elemento;
- O XHTML2 usa `XMLEvents` para manipular os eventos, e o elemento `script` agora se chama `handler`;
- `XForms` agora é parte integral do XHTML2;
- O modelo de quadros (frames) não é mais especificado no XHTML2, é especificado separadamente em `XFRAMES`;
- Acrônimos agora também serão usados com o elemento `abbr`, ao invés de `acronym`;
- Elemento `blockcode` para blocos de código;
- O elemento `a` ainda existe, mesmo que todos os elementos agora possam ter o atributo `href`, mesmo que âncoras possam ser marcadas com o atributo `id`;
- Embora o elemento `object` consiga fazer tudo o que o `img` fazia, este ainda foi mantido para facilitar a transição;
- Mesmo com a criação do elemento `h` para cabeçalhos, os elementos `hx` ainda foram mantidos, e podem causar alguma confusão.

# HTML5

## Necessidades

- Melhora na Semântica dos documentos;
- Foco também voltado às aplicações Web.

## O que há de mudanças

- Web Application APIs
  - API de desenho 2D com elemento `canvas`;
  - API de reprodução de vídeo e áudio, com os elementos `video` e `audio`;
  - API de aplicações offline (semelhante ao Google Gears);
  - API para permitir que Aplicações Web registrem por si próprias certos protocolos e MIME Types;
  - API de edição para ser usada em conjunto com o atributo `contenteditable`;
  - API de arrastar e soltar com o elemento `draggable`;
  - API de *networking* para permitir comunicação direta usando TCP;
  - API de histórico do *browser*, para construção de aplicações que não quebrem o botão Voltar;
  - API de mensagens entre documentos;
  - Eventos enviados pelo servidor em combinação com o elemento `event-source`.
- Elemento `figure` para representar figuras e imagens com uma legenda descritiva, agrupando o elemento `img` para a imagem e `caption` para a legenda;
- Elemento `m` para marcar palavras em um documento, como por exemplo em uma página de resultados de busca;
- Elemento `time` para data e hora e `measure` para medidas;
- Elemento `datagrid` para representar listas hierárquicas e tabelas de dados interativas;
- Elemento `command` para comandos que o usuário pode invocar;
- Elemento `event-source` para capturar eventos enviados pelo servidor;
- Elemento `output` para conter qualquer tipo de saída de dados, como resultados de um cálculo ou operação;
- Elemento `progress` para representar o progresso de uma tarefa, como efetuar um download.
- Elementos como `section`, `article`, `header`, `footer`, e também um elemento `dialog` para diálogos;
- Atributo `ping` para rastreamento sem depender de redirecionamentos via Javascript;

- Deve-se indicar, usando uma tag meta de tipo `generator`, quando foi usado um editor WYSIWYG para gerar a página.

## **Semelhanças entre os dois**

XHTML 2 e HTML5 não podem conter estes elementos: `basefont`, `big`, `font`, `s`, `strike`, `tt`, e `u`. XHTML2 também remove `small`, `b`, `i` e `hr`, enquanto o HTML5 redefine-os com significados não-visuais.

### **Listas de navegação**

XHTML 2 usará o elemento `nl` e o HTML5 usará o elemento `nav`. O elemento `nl`, do XHTML 2 é um elemento de lista – e portanto terá como filhos elementos `li`, e o elemento `nav` do HTML5 deverá incluir uma lista não ordenada `ul` e então elementos filhos `li`.

### **Mais semântica**

No XHTML 2 haverá o atributo `role`, que permitirá definir em um conjunto de valores pré-definidos a função semântica de um elemento. O HTML5 tomou uma abordagem semelhante à dos *microformats*, usando palavras reservadas como valor do atributo `class`.

### **Formulários mais avançados**

XHTML 2 usará uma nova abordagem, referenciando a especificação XForms, onde a estrutura do formulário e a exibição dos controles são feitas separadamente. Já no HTML5, a estrutura básica dos elementos de formulário já existentes será mantida, com a adição de novos tipos de `input`, como campos para URI, e-mail, e campos numéricos.