**Programa DIO**

**Formação HTML Web Developer**

**Trabalhando com Formulários**

Formatando Textos

* Tag itálico **< i >** - muito utilizado em citações, termos regionalizados (gírias).
* Tag negrito **< b >** ou **< strong >** - “bold” deixa mais escuro o texto e a espessura da linha mais grossa.
* Tag sublinhado **< u >** - “underline”

Obs. essas tags podem ser combinadas, ex. ficar negrito + sublinhado.

* Tag **< mark >** - marca texto
* Tag **< sup >** - deixa o texto uma margem um pouco superior a linha do texto.
* Tag **< sub >** - deixa o texto uma margem um pouco abaixo a linha do texto.
* Tag **< blockquote >** - um espaçamento que se dá ao texto, geralmente se utiliza quando vai colocar uma explicação e/ou citação.

Semântica

É a forma ao qual se estrutura o HTML de modo que olhando determinado trecho do código, é possível saber do que se trata.

Por exemplo tem se as seguintes tags: <b> e o <strong>, ambos deixam determinado texto ou trecho dele em negrito. Ou seja, para o usuário não há diferença, mas no código HTML ao se olhar um trecho dele contendo <b> e o mesmo trecho contendo <strong>, percebe-se que o “strong” dá mais ênfase no trecho.

Dia: portanto, se quiser dar mais ênfase no código coloque “strong”, se apenas quiser deixar em negrito usa “b”.

Em questão de acessibilidade, por exemplo pessoas “com cegueira” o robô ao se fazer leitura do texto, e ler no código a palavra em negrito cujo HTML foi escrito com “strong”, o robô dará uma ênfase maior para que o usuário escute.

* Tag **< font >** - basicamente tem a função de formatar a letra do texto.

Não é comum uso desta tag para mudar a fonte do texto, normalmente se usa o CSS para isso, que é mais indicado.

Atributos da tag < font >

Obs. a tag “font” sozinha ela não executa nenhuma função.

Obs. há tag que precisa de atributos para que funcione.

* Color - < font **color=””**> - hexadecimal/rgb/hsl/lba/sla ou nome da cor

Obs. vários navegadores suportam nome de cores.

* Face - < font color=”red” **face=”Arial”**> - atributo face é utilizado para se colocar o nome da fonte ao qual se quer modificar o texto.

Obs. dependendo do tipo de fonte que se utiliza, não será possível a leitura da fonte, pois, como o HTML roda no navegador, ele entende que o usuário tenha determinada fonte já instalada no computador. Se não tiver fonte instalada ele o navegador irá rodar o padrão “Sans Serife”.

Dica, desse modo, é bom colocar mais de um tipo de fonte no HTML, assim, o usuário poderá ter alguma delas; E não sintaxe, separa-se por “virgulas”.

Ex. < font color=”red” **face=”Arial, Tahoma”**>

Obs. Quando há nome de fonte composto, para que o navegador possa entender que tudo é um nome de fonte, precisa colocar mais uma aspa.

Ex. < font color=”red” **face=”Arial, ‘Trebuchet MS’”**>

Obs. Tag font está caindo em desuso, pois, é possível setar a fonte no CSS, que é o mais indicado para isso.

Tag **< div >** e **< span >**

São consideradas tags “coringas”, utilizadas para estrutura página HTML, pois, não carregam peso semântico. Então elas servem para simplesmente delimitar um espaço e depois via CSS, pode-se através desse espaço manipular eles.

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Div – é uma tag do tipo dispay block – é aquela que ocupa a horizontal inteira de onde está inserida

Tag span – tag span não ocupa o espaço horizontal inteiro da tela como ao que a div faz.

Obs. no CSS não consegue alterar cor de uma letra se estiver em um block. Resolve-se colocando uma tag span, abraçando determinado trecho do texto, torna-se possível manipular pelo CSS.

Tag **< label >**

Como se fosse a tag span só que ela geralmente fica de um lado do campo. Ou seja, o label de determinado campo.

Tag **< fieldset >**

Possui uma finalidade mais voltada para estrutura do HTML, mas possui uma aparência mais especifica, portanto, poderá auxiliar o código ficar um pouco mais didático.

Por exemplo, se há um projeto muito grande, há possibilidade em separar em blocos onde cada conjunto possa representar dados de mesma espécie, gênero ou família.

Obs. essa tag por si só, somente delimita o espaço inserindo um retângulo.

Portanto, há necessidade de se aliar essa tag com outras.

Exemplo:

Tag **< legend >**

Colocar um texto inserido na linha da caixa delimitada pela tag fieldset.

Tag **< embed >**

Não se usa mais esta tag, mas há site que contém ela em sua estrutura.

Interface gráfica do usuário

Descrição gerada automaticamente com confiança média

Interface gráfica do usuário, Texto, Email, Site

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Quando foi se evoluindo a necessidade de se melhorar os sites, profissionais se questionaram em como fazer para que cada pedaço do site fosse carregado de forma independente.

Pois, todas as informações na conexão da internet dos anos 90, era muita informação, para ser carregado na página.

Assim, verificou-se que havia necessidade de carregar trechos do HTML de maneira separada. Para evitar travamento da págima devido ao tráfico cada ve maior na página.

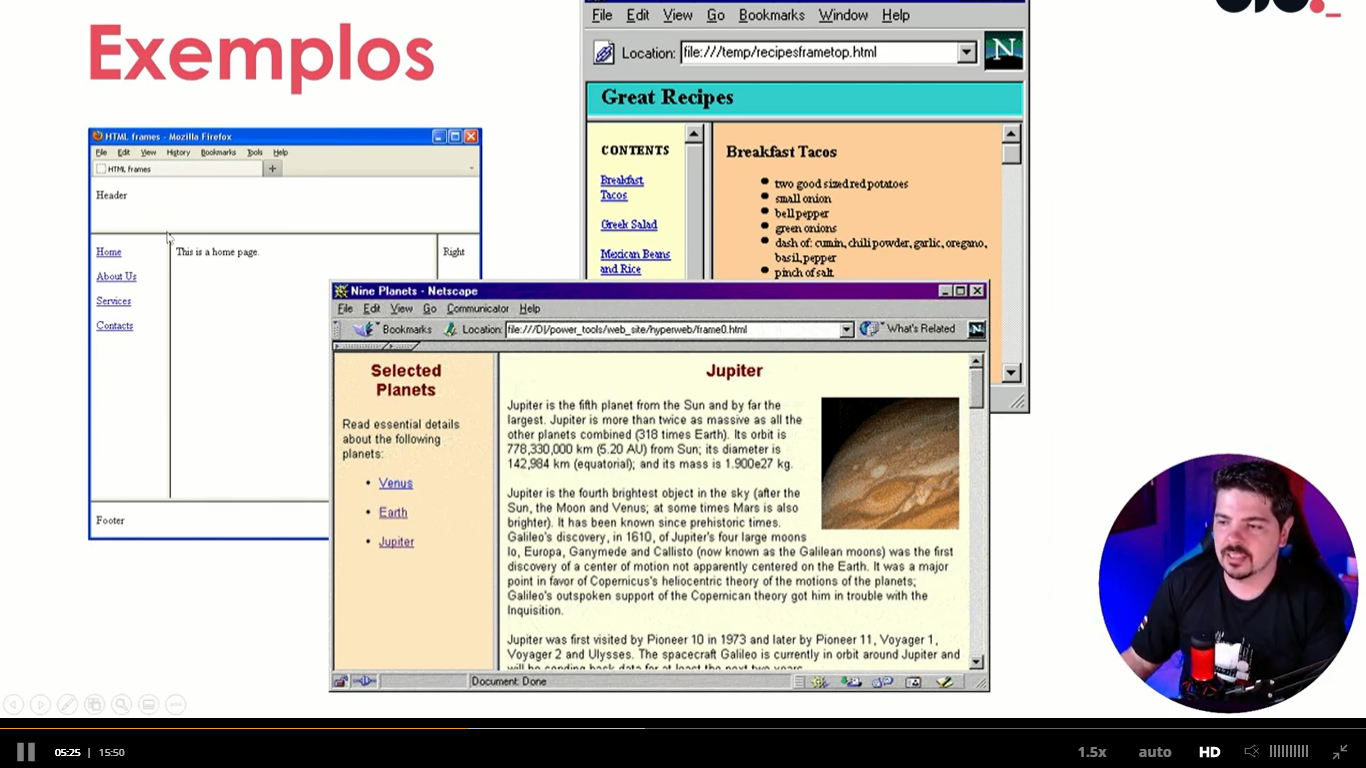
Então qual foi a estratégia para melhorar experiência do usuário?

Diagrama

Descrição gerada automaticamente

Foi criado os frameset (s), que eram estrutura de quadros. Onde cada frame era uma página HTML.

Assim, na época o pensamento era, se usuário clica-se em determinado frame, o conteúdo carregado era somente o que estava dentro dele, e assim, não precisaria recarregar toda estrutura do site novamente.

Exemplos dos framesets:

Mas ainda sim era muito pesado carregar a página, a exemplo, se a imagem de júpiter fosse um vídeo, somente exibiria o conteúdo da página depois que o vídeo terminasse de ser carregado.

Então foi -se percebendo a necessidade de carregar diferentes pedaços do HTML, era mais relacionado a áudios, imagens e mídias.

Antigamente era o flash players: animações – hoje os navegadores não possuem suporte para ele.

E também Java Applets – seriam os plugins java. Para acessar alguns sites antigamente, ex. bancos, havia necessidade de se instalar plugins java. Por conta da necessidade de carregar de modo independente seções da página, veio essa parte dos plugins java. Assim, aos navegadores teriam programas instalados a parte que ficava sendo inserido na página.

Interface gráfica do usuário

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Assim surgiu a tag embed – ele veio para ajudar carregar de forma independente determinada parte do HTML. Assim, ele pega um conteúdo externo e plugar dentro do site para ser exibido.

O embed é utilizado para vídeos arquivos em flash.

Há tag object mesma funcionalidade para embed, mais ele é mais voltado para aplicativo.

Obs. em termo de acessibilidade , semanticamente, se um robô acessar o site, e ver a tag embed, ele não saberá o conteúdo que terá dentro da tag, pois, ele pode ser uma imagem, áudio ou vídeo. Então, semanticamente essa tag não é muito forte.

Pois, para vídeo, por exemplo ao invés da tag embed, utiliza-se a **Tag < vídeo >** (atributos: **controls** e **autoplay**)e o vídeo ele abre e fecha e tem uma **Tag < source >** que se coloca o caminho do vídeo (fonte) .

Tag **< iframe >**

Obs. precisa-se tomar cuidado com essa tag no quesito segurança do site.

Ele permite abrir uma janela na página exibindo conteúdo de uma outra página inteira se quiser, como também vídeos do youtube até google maps.

<iframe src=”site”></iframe>

Pode-se também fazer com que outros sites não fação isso com o seu site.

Obs. quando se acessa um conteúdo através de um iframe, conta a visita para o outro site e também.

Resenha sobre cores

Aproximadamente os navegadores aceitam ca. de 140 cores ou nome de cores.

Referência: <https://www.w3schools.com/colors/colors_names.asp>

Obs. há navegadores que suportam algumas cores e outros não.

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Obs. RGBA e o HSLA – o último número representa a transparência.

Obs. HLS – vai de 0 a 360 onde o 0 é vermelho, e os outros 2 números não são cores , representa saturação que é (quão nítido é a cor) e o último representa o brilho e quando se coloca o “A”, pode-se definir a transparência (0 a 1 %).

Referência: https://www.w3schools.com/colors/colors\_hsl.asp

Hexadecimal é de 0 a 15

A partir do 10 ele se transformam em números

10 = A

11 = B

12 = C

13 = D

14 = E

15 = F

Por padrão os 2 primeiros são vermelho, após verde e após azul

Obs. sempre precedido de hash tag “**#”**. Exemplo

#RRGGBB

#FF0000 – cor vermelha

Ou #255 0 0