

BOOK NOTE TAKING

HTML5 & CSS3

BY GUSTAVO GUANABARA

Note Taking Builder

Ramon De Sá Santos

Created On

09, October, 2024

| Módulo 01 - Chapter One

Questões chave:

- (1) É importantíssimo manter o foco nas aulas durante o curso.
- (2) É de grande importância realizar a prática constante do que se é aprendido.
- (3) Guardar as informações de maior relevância em um caderno ou note 's app.
- (4) Endereços essenciais: [repositório público] [1] github.com/gustavoguanabara
[2] gustavoguanabara.github.io
- (5) É necessário manter uma certa constância e paciência ao aprender o que se está sendo exposto no curso, dê tempo ao seu próprio tempo, não há necessidade de “maratonar” o curso.

| Esse curso é para mim?

Questionamentos Comuns:

1. “Me interessa por aprender a criar sites, mas não sei por onde começar”;
2. “Já tentei, mas acho confuso ‘decorar’ tantos comandos”;
3. “Tenho boa base, mas quero me aprofundar nas especificações”;
4. “Consigo adaptar códigos prontos, mas não sei criar os meus próprios sites”;
5. “Estou tendo essa disciplina e estou com dificuldades”;
6. “Além de aprender a criar sites, quero deixá-los online”.

| Melhores livros para aprender HTML5 & CSS3

Bibliografia recomendada:

1. E-book do curso HTML e CSS3;
2. Referências on-line: **W3C Standards, created by World Wide Web Consortium**;
3. Livros:
 - [1] **HTML5**, Mark Pilgrim, editora O'Reilly;
 - [2] **CSS**, Eric Meyer, editora O'Reilly;
 - [3] **Flexbox in CSS**, Estelle Weyl, editora O'Reilly;
 - [4] **Grid Layout in CSS**, editora O'Reilly.
 - [5] **HTML5 e CSS3**, editora Alta Books, de Elizabeth Castro e Bruce Hyslop, guia prático e ilustrado;
 - [6] **Curso de Design Gráfico**, editora GG, por David Dabner e outros.

BOOK NOTE TAKING

HTML5 & CSS3

GUSTAVO GUANABARA

| Como a Internet chega na minha casa?

- A Internet nasceu para atender as necessidades dos países do bloco principal da Guerra Fria, para atender às demandas estratégicas dos EUA e da URSS;
- Sputnik, apenas um satélite de tecnologia para teste, criado pela União Soviética;
- DARPA (1969), criada por Dwight Eisenhower. Existe até os dias atuais.
- Os EUA tiveram a ideia de criar uma “teia” de informações entre os quatro grandes computadores, militares e não militares, da época.
- Protocolos eram uma forma de linguagem tecnológica para sistematizar e unificar essa rede, “teia” de dados.
- Bob Kahn, pesquisador que criou o protocolo TCP, tinha problemas com a identificação de pontos de informações.
- Vint Cerf, também pesquisador que criou um protocolo de identificação de máquinas (IP).
- Formou-se, então, em 1972, TCP IP, que existe até os dias atuais, em uma versão bem mais atualizada.
- ARPANET: A rede militar se tornou MILNET, a científica (Universidades) NSFNET, e por fim, as redes comerciais.
- Interconnect Networking [Internetworking] > Internet (Evolução do conceito).
- Praticamente, todos os continentes estão unidos por cabos submarinos, promovendo assim a distribuição da Internet.
- Vídeo de como a Internet funciona: <https://youtu.be/TNQsmPf24go>
- O protocolo http:// e a linguagem HTML, foram criadas pelo cientista Tim Berners-Lee.
- Para acessar a World Wide Web é necessário um navegador, o **Mosaic** (compatível com o http://) foi o primeiro, criado por Marc Andreessen.
- HTML - sigla para **HyperText Markup Language**.

| Como a Internet funciona?

- O computador é um equipamento eletrônico, que não é dotado de inteligência, mas é capaz de processar informações muito rapidamente.
- Como um equipamento eletrônico, um computador ele funciona através de sinais, o conjunto binário [0,1] pela computação, são na verdade um conjunto de ondas quadradas (sinais elétricos); são na verdade, dígitos binários [binary digit (bit)], a computação reúne um conjunto de oito bits (byte), porção mínima que se represente um dado, uma informação.
- Código multi byte UTF-8, utiliza de um a quatro bytes, para representar letras e símbolos.

BOOK NOTE TAKING

HTML5 & CSS3

GUSTAVO GUANABARA

- Seguindo a lógica de representação de dados, 1024 bytes equivalem a 1 kilobyte (base 2, 2^{10}), assim, 1024 kilobytes equivalem a 1 megabyte, 1024 megabytes equivalem a 1 gigabyte, e ainda, 1024 gigabytes equivalem a 1 terabyte.
- MEGABYTES (armazenamento) MB (maiúsculos).
- MEGABITS (transmissão, rede) Mb (“b” minúsculo).

| Como nos conectamos?

- Nosso computador é denominado “cliente” porque recebe as informações de um “servidor” (a Internet, necessariamente). O computador se conecta a um cabeamento por intermédio de um **modem** (processo de modulação-demodulação), (sistema de TV, antena, fios telefônicos).
- Site para descobrir o ip de Internet: **www.ip location.net** (ao reiniciar o roteador, o IP vai mudar).
- A Internet possui uma “agenda eletrônica”, lugar que faz a ligação entre os nomes e números, é o servidor DNS (Domain Name System).
- E as rotas? Por ser uma rede finita, geralmente, o caminho habitual pode estar congestionado ou mesmo pode ter deixado de funcionar, por isso as solicitações do cliente não seguem um padrão prático, tomando assim outras rotas para o acesso às informações.

| O que é domínio e hospedagem?

- **Situação 1.0** - Os arquivos estão no servidor (S); ao pesquisar no dispositivo cliente o endereço (url), por exemplo, `gustavo guanabara.github.io`, vai ser acionado um número fonte de consulta no servidor DNS, que vai promover o acesso ao servidor do HTML e CSS. Os códigos e arquivos HTML e CSS vão ser inicialmente guardados no dispositivo cliente. Os códigos e arquivos HTML e CSS vão ser acessados somente se estiverem em um servidor.
- **Situação 2.0** - O domínio (`github.io`) é necessário para promover o acesso de outros usuários (não podem existir dois domínios com o mesmo nome), um domínio é pago anualmente e o valor varia entre 30 e 50 reais. Há a necessidade de TLDs. Já a Hospedagem é um espaço para armazenar arquivos. A hospedagem varia em questão de qualidade: espaço para armazenar arquivos, pago mensalmente, garante espaço, memória, além de recursos.
- **Situação 3.0** - as nomenclaturas `.com`, `.net`, `.io`, `.online`, `.store` (domínios genéricos, as GTLDs - Top Level Domain). Há também os CCTLDs que identificam a localidade (país): BR, UK, US. Após a barra (/) as informações subsequentes são os caminhos.

BOOK NOTE TAKING

HTML5 & CSS3

GUSTAVO GUANABARA

| A diferença entre HTML, CSS e JavaScript

- Lembrar: “a linguagem HTML”, e “as CSS” (plural e feminino) (as folhas de estilo);
- A HTML e as CSS não são em essência linguagens de programação. Eu desenvolvo/crio sites em HTML e com as CSS.
- HTML - Hypertext Markup Language (Hypertext, clica em uma área e é redirecionada para outra área). HTML é uma linguagem de marcação.
- CSS - Cascading Style Sheets (Folhas de Estilo em Cascata).
- A linguagem HTML é focada em conteúdo. Esses conteúdos podem ser textos, imagens, vídeos, tabelas, etc.
- As CSS são utilizadas para atrativos visuais para o público. Seu foco, portanto, é na configuração de designs. São feitas com as CSS as cores, sombras, tamanhos, posicionamento em um site (desenvolvimento web).
- Já o JavaScript (JS) é uma linguagem de programação que tem o foco nas interações (interatividade usuário/site). Com o JavaScript é possível criar menus interativos, animações, popups, validações, etc.
- Conteúdo em HTML - Conteúdos: Exemplo de título; para transformar um conteúdo em título, é necessário usar um TAG específica (Tags em HTML, são informações colocadas entre colchetes angulares).

Exemplo de abertura e fechamento de Tag (sem espaço):

<h1>exemplo de título</h1>

Exemplo de abertura e fechamento de Parágrafo (sem espaço, novamente):

<p>Exemplo de parágrafo</p>

Exemplo de Tag para se adicionar uma foto ao site:

**** (não possui fechamento de Tag, mas a maioria das Tags possuem fechamento).

1. **Src & Alt:** são parâmetros.
 2. **“Foto.png” e “Exemplo de foto”:** são valores.
- **Estilo CSS (Anatomia):**
 1. Assim: **h1 {**

font-family: Arial;
font-size: 20pt;
Color: blue;

}
 2. Assim o **h1** (seletor);
 3. Informações abaixo das chaves (declaração);
 4. **Font-family, Font-size, Color** (propriedade);

BOOK NOTE TAKING

HTML5 & CSS3

GUSTAVO GUANABARA

5. Arial, 20pt, blue (valor);

- Estrutura básica de um documento HTML5

```
<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport"
      content="width=device-width ,
      initial-scale=1.0">
    <title>Document</title>
  </head>
  <body>
    <h1>Olá, Mundo !</h1>
  </body>
</html>
```

- A área HEAD é a área de configurações;
- A área BODY é a área de “corpo”, onde fica o site;
- A codificação linha `<meta charset="UTF-8">` implica em identificar o site com compatibilidade em acentos gráficos (para a Língua Portuguesa);
- A codificação linha meta-name... o site vai ocupar 100% da “parte branca” e vai ser visualizado em uma escala de 100%, sem zoom, padrão.
- A função do navegador (Chrome, exemplo) é basicamente analisar linha a linha e gerar um resultado visual da análise do código.

| Diferenças Entre Front-end, Back-end & Full Stack

- Site estático - é o termo que define quando a experiência visual de um site é a mesma para todos os clientes que solicitarem as tecnologias (HTML, CSS, JavaScript).
- Qualquer um dos navegadores disponíveis e de sucesso na atualidade, que funcionam em um dispositivo mobile e em um PC, por exemplo, são capazes de executar as três tecnologias, são portanto, tecnologias client-side (client-side developer ou Front-end). (Desenvolvedor(res) de tecnologias que rodam “ao lado do cliente”)

BOOK NOTE TAKING

HTML5 & CSS3

GUSTAVO GUANABARA

- Por outro lado, existem tecnologias que rodam “ao lado do servidor”, server-side (server-side developer ou Back-end).
- Basicamente o desenvolvedor **Front-End**, será responsável por gerar a parte visual e interativa do site criado, além de se preocupar com a experiência causada no usuário (user experience), dessa forma, os trabalhos entre um **Designer** e um desenvolvedor Front-end eles se unem em alguns pontos.
- Já o desenvolvedor **Back-end** está mais interessado nas interações entre o código criado com as “entranhas” do servidor. As linguagens mais aplicadas no desenvolvimento Back-end são: PHP (a mais famosa); JavaScript (Node); as linguagens Sharp e Python; Além das linguagens Ruby e Java.
- Hoje em dia, faz-se necessário utilizar um **banco de dados** atrelado ao desenvolvimento Front-end, já que, por exemplo, uma loja virtual só funciona com a associação entre as duas técnicas de desenvolvimento.
- Desenvolvedor **Full Stack** é a associação de bagagem e domínio entre as carreiras de Front-end e Back-end.

| Copyright versus Creative Commons

- Resumidamente, no momento de escolher o formato de imagem para o seu site, tenha preferência sempre pelo formato **JPEG** (Joint Photographic Experts Group) com uma compactação entre 30% e 50%. Já o formato **PNG** (Portable Network Graphics) só deverá ser usado **quando houver a necessidade de transparência em uma imagem**.
- Uma imagem **1920x1080** de **3MB** não vai ficar mais leve se você mudar a largura dela no seu **CSS**. Muito pelo contrário. Seu navegador vai levar um bom tempo para carregar o arquivo do seu servidor e vai exibir a imagem minúscula na tela. E no que isso te prejudica? Os mecanismos de busca como o Google penaliza sites lentos e pesados, retirando-os da primeira página de buscas. Quem aqui quer criar um site que não aparece nos resultados do Google?