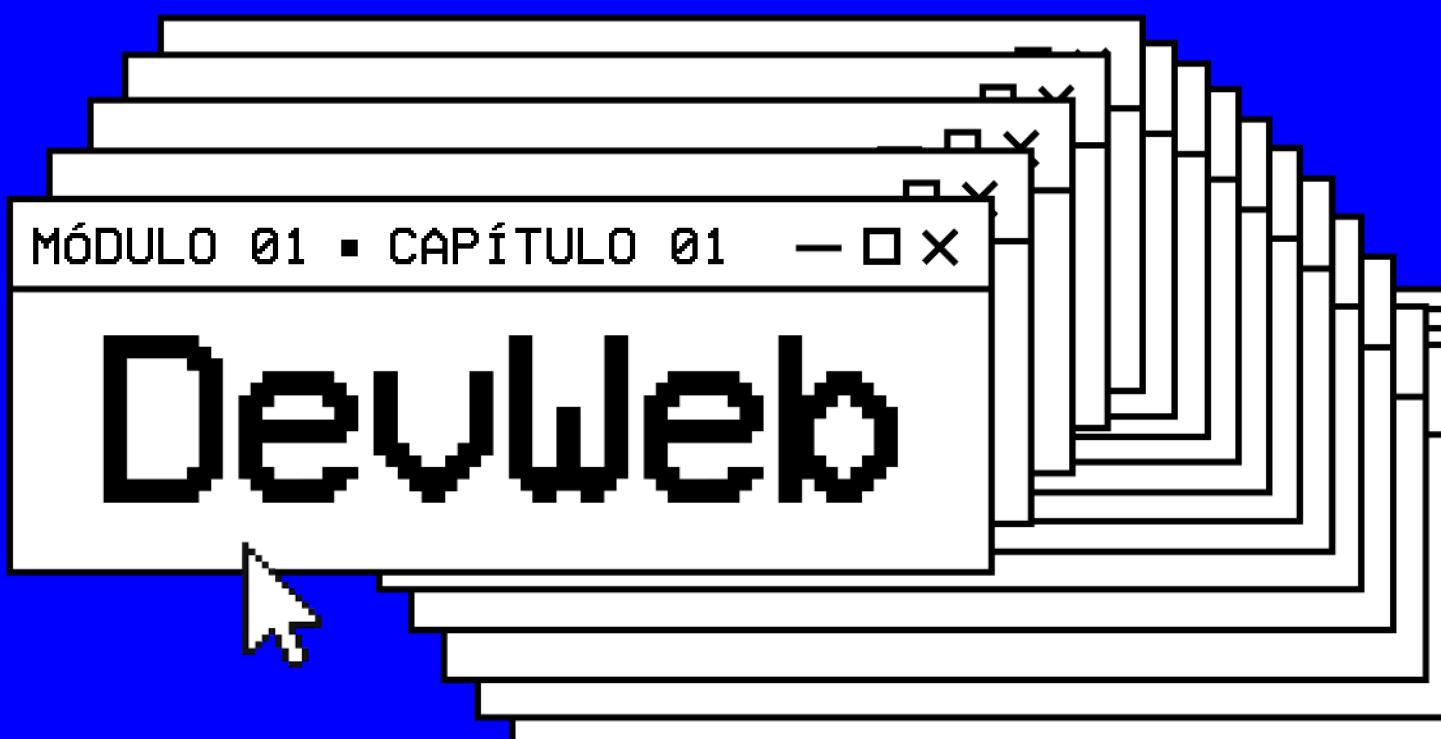
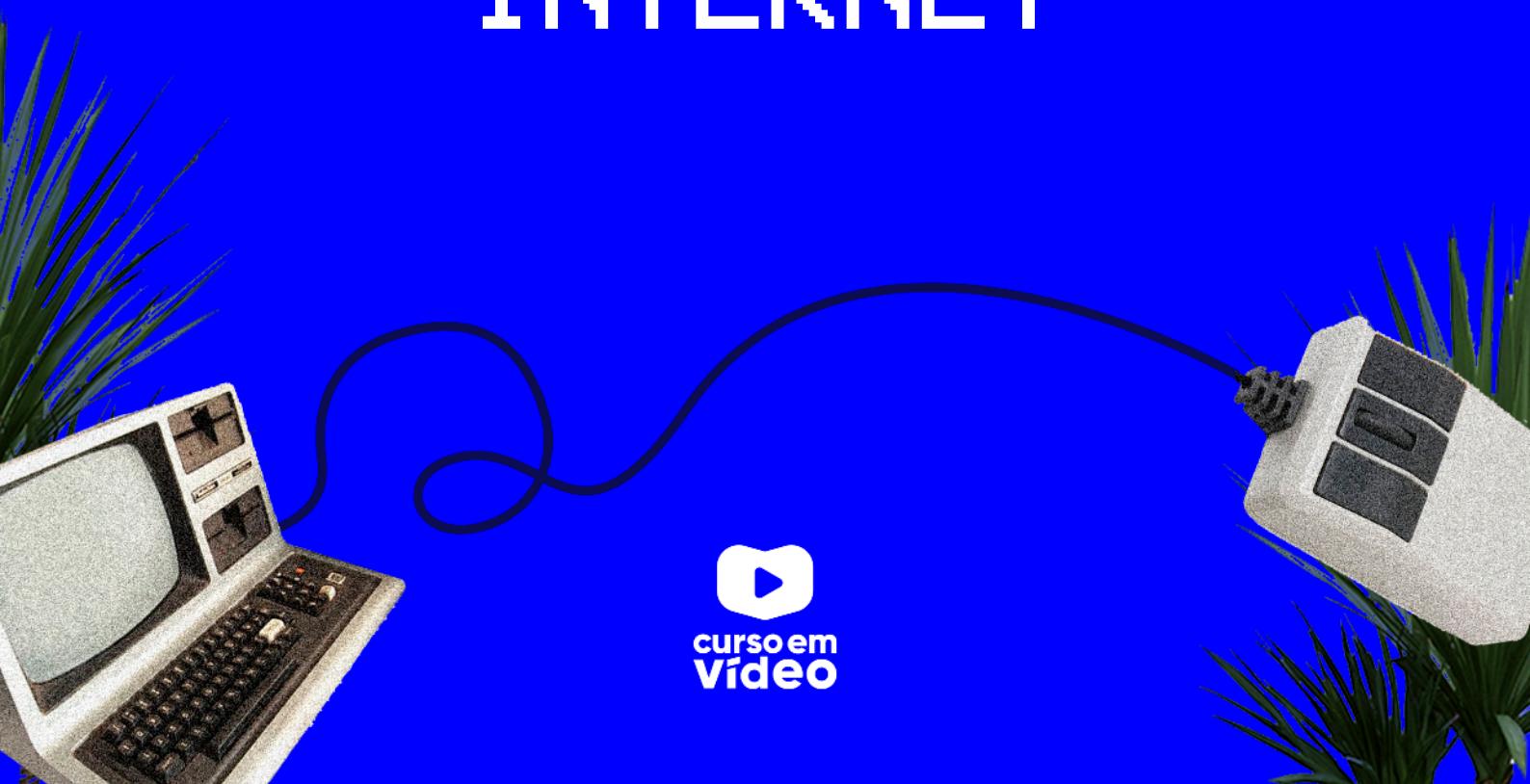


CURSO DE INFORMÁTICA PARA INTERNET

Gustavo Guanabara



HISTÓRIA DA INTERNET



M01C01

veja em vídeo



HISTÓRIA DA INTERNET

Todos nós temos uma História. Quando entendemos de onde viemos, conseguimos compreender a evolução até chegar onde estamos. E com a Internet não foi diferente. Tudo começa logo após a Segunda Guerra Mundial, com uma treta gigante entre Estados Unidos e União Soviética, que antes eram aliados mas agora se tornam inimigos. Acompanhe o desenrolar dessa novela e entenda como uma Guerra deu origem à maior rede do mundo. Vamos lá?



Você tem todo o direito de usar esse material para seu próprio aprendizado. Professores também podem ter acesso a todo o conteúdo e usá-lo com seus alunos. Porém todos que usarem esse material - seja para qual for a finalidade - deverão manter a referência ao material original, criado pelo **Prof. Gustavo Guanabara** e disponível no endereço do seu repositório público <https://github.com/gustavoguanabara/>. Este material não poderá ser utilizado em nenhuma hipótese para ser replicada - integral ou parcialmente - por autores/editoras para criar livros ou apostilas, com finalidade de obter ganho financeiro com ele.

— □ ×

A Internet veio da Guerra (infelizmente)

Depois da Segunda Guerra, EUA e URSS começaram a ter seus desentendimentos, dando origem à **Guerra Fria** em 1949. Neste contexto, em que os dois blocos ideológicos e politicamente antagônicos exerciam enorme controle e influência no mundo, qualquer mecanismo, qualquer inovação, qualquer ferramenta nova poderia contribuir nessa disputa liderada pela **União Soviética** e pelos **Estados Unidos**: as duas superpotências compreendiam a eficácia e a necessidade absoluta dos meios de comunicação.



APRENDA MAIS: Quer aprender mais sobre a Guerra Fria? Dá uma olhada aqui nesse vídeo de 9 minutos e com certeza você vai entender mais sobre essa treta toda.

Canal Descomplica: <https://youtu.be/cAwsLa04HGQ?t=49>

Nessa perspectiva, o governo dos Estados Unidos temia um ataque russo às bases militares. Um ataque poderia trazer a público informações sigilosas, tornando os EUA vulneráveis.

Então foi idealizado um modelo de troca e compartilhamento de informações que permitisse a descentralização das mesmas. Assim, se o Pentágono fosse atingido, as informações armazenadas ali não estariam perdidas. Era preciso, portanto, criar uma rede, a **ARPANET**, criada pela DARPA, sigla para **Defence Advanced Research Projects Agency**.



O ataque inimigo nunca aconteceu, mas o que o Departamento de Defesa dos Estados Unidos não sabia era que dava início ao maior fenômeno midiático do século 20', único meio de comunicação que em apenas 4 anos conseguiria atingir cerca de 50 milhões de pessoas.

O começo de tudo

A **ARPANET** funcionava através de um sistema conhecido como *chaveamento de pacotes*, que é um sistema de transmissão de dados em rede de computadores no qual as informações são divididas em pequenos pacotes, que por sua vez contém:

- trecho dos dados
- o endereço do destinatário
- informações que permitiam a remontagem da mensagem original.

Em 29 de Outubro de 1969 ocorreu a transmissão do que pode ser considerado o **primeiro E-mail da história**. O texto desse primeiro e-mail seria "LOGIN", conforme desejava o Professor Leonard Kleinrock da Universidade da Califórnia em Los Angeles (UCLA), mas o computador no Stanford Research Institute, que recebia a mensagem, parou de funcionar após receber a letra "O".



A ARPANET no início, em 1969, só tinha 4 pontos

Já na década de 1970, a tensão entre URSS e EUA diminui. As duas potências entraram definitivamente naquilo em que a história se encarregou de chamar de **Coexistência**

— □ ×

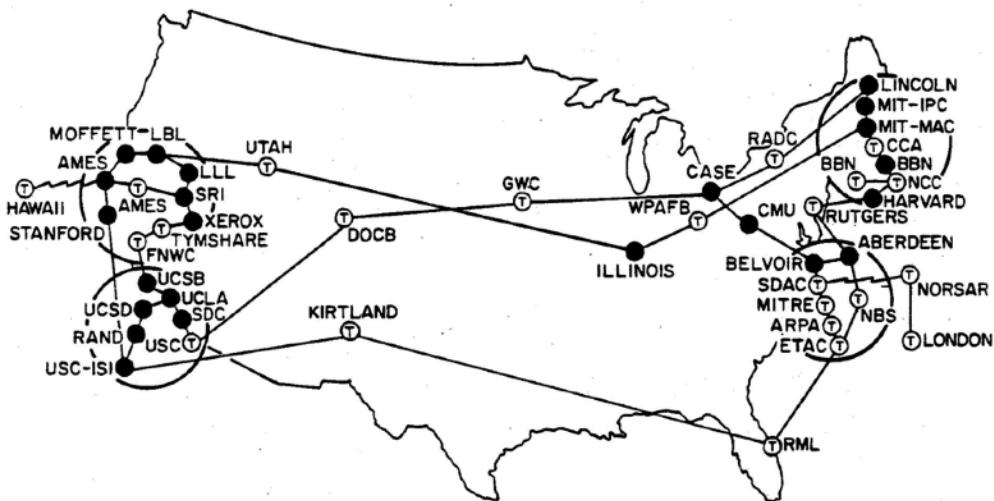
Cursos grátis de tecnologia
que te preparam para o
mercado de trabalho

RECODE

Scannable QR code linking to RECODE's free technology courses.

Pacífica. Não havendo mais a iminência de um ataque imediato, o governo dos EUA permitiu que pesquisadores que desenvolvessem, nas suas respectivas universidades, estudos na área de defesa pudessem também entrar na ARPANET.

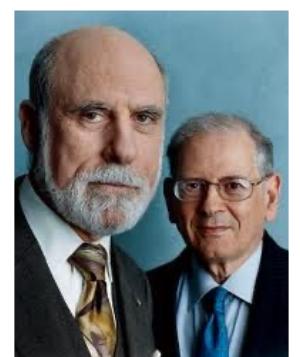
Com isso, a ARPANET começou a ter dificuldades em administrar todo este sistema, devido ao grande e crescente número de localidades universitárias contidas nela. Dividiu-se então este sistema em dois grupos, a **MILNET**, que possuía as localidades militares e a **nova ARPANET**, que possuía as localidades não militares. O desenvolvimento da rede, nesse ambiente mais livre, pôde então acontecer. Não só os pesquisadores como também os alunos e os amigos dos alunos, tiveram acesso aos estudos já empreendidos e somaram esforços para aperfeiçoá-los.



Já na década de 70, depois da abertura da rede para pesquisadores

Além desses backbones, existem os criados por empresas particulares. A elas são conectadas redes menores, de forma mais ou menos anárquica. É basicamente isto que consiste a Internet, que não tem um dono específico.

Com a entrada de muitos pontos na rede e com métodos de comunicação diferentes entre eles, alguma atitude tinha que ser tomada, já que o antigo protocolo **NCP** já não estava mais aguentando. *Robert Kahn* da DARPA e ARPANET recrutaram *Vint Cerf* da Universidade de Stanford para trabalhar com ele nesse problema. Em 1973, eles logo trabalharam com uma reformulação fundamental, onde as diferenças entre os protocolos de rede eram escondidas pelo uso de um protocolo inter-redes comum, e, ao invés da rede ser a responsável pela confiabilidade, como no ARPANET, os hospedeiros ficaram como responsáveis.



A especificação do protocolo resultante contém o primeiro uso atestado do termo internet, como abreviação de **internetworking**; então a palavra começou como um adjetivo, ao invés do nome que é hoje. Com o papel da rede reduzida ao mínimo, ficou possível a junção de praticamente todas as redes, não importando suas características, assim, resolvendo o problema inicial de Kahn. O DARPA concordou em financiar o projeto de desenvolvimento do software, e depois de alguns anos de trabalho, a primeira demonstração de algo sobre gateway entre a rede de Packet

Radio na Baía de SF área e a ARPANET foi conduzida. Decorrentes das primeiras especificações do TCP em 1974, **TCP/IP** emergiu em meados do final de 1978, em forma quase definitiva. Em 1º de janeiro de 1983, data conhecida como **Flag Day**, o protocolo TCP/IP se tornou o único protocolo aprovado pela ARPANET, substituindo o antigo protocolo NCP.

O cientista **Tim Berners-Lee** (foto ao lado), do **CERN**, criou a **World Wide Web**, a linguagem **HTML** e o protocolo **HTTP** em 1992. Essa linguagem simples, mas eficiente, era usada para a criação dos sites com o conceito de hipertexto (documentos ligados entre si).

A empresa norte-americana **Netscape** criou o protocolo **HTTPS** (HyperText Transfer Protocol Secure), possibilitando o envio de dados criptografados para transações comerciais pela internet.



APRENDA MAIS: A História da Internet tem muitos outros acontecimentos interessantes. Quer ver mais sobre isso? Então veja esse vídeo de 15 minutos que conta tudo com riqueza de detalhes, mas sem se tornar chato:

Canal TecMundo: <https://youtu.be/pKxWPo73pX0?t=27>

A Internet no Brasil

Em 1989, o **Ministério da Ciência e Tecnologia** lança um projeto pioneiro, a **Rede Nacional de Ensino e Pesquisa** (RNP). Existente ainda hoje, a RNP é uma organização de interesse público cuja principal missão é operar uma rede acadêmica de alcance nacional. Quando foi lançada, a organização tinha o objetivo de capacitar recursos humanos de alta tecnologia e difundir a tecnologia Internet através da implantação do primeiro backbone nacional.

O primeiro backbone brasileiro foi inaugurado em 1991, destinado exclusivamente à comunidade acadêmica. Mais tarde, em 1995, o governo resolveu abrir o backbone e fornecer conectividade a provedores de acesso comerciais. A partir dessa decisão,

Soluções digitais para negócios

hostnet



surgiu uma discussão sobre o papel da RNP como uma rede estritamente acadêmica com acesso livre para acadêmicos e taxada para todos dos outros consumidores. Com o crescimento da Internet comercial, a RNP voltou novamente a atenção para a comunidade científica.

A partir de 1997, iniciou-se uma nova fase na Internet brasileira. O aumento de acessos a rede e a necessidade de uma infraestrutura mais veloz e segura levou a investimentos em novas tecnologias. Entretanto, devido a carência de uma infraestrutura de fibra óptica que cobrisse todo o território nacional, primeiramente, optou-se pela criação de redes locais de alta velocidade, aproveitando a estrutura de algumas regiões metropolitanas. Como parte desses investimentos, em 2000, foi implantado o backbone RNP2 com o objetivo de interligar todo o país em uma rede de alta tecnologia. Atualmente, o RNP2 conecta os 27 estados brasileiros e interliga mais de 300 instituições de ensino superior e de pesquisa no país, como o INMETRO e suas sedes regionais.

Outro avanço alcançado pela RNP ocorreu em 2002. Nesse ano, o então presidente da república transformou a RNP em uma organização social. Com isso ela passa a ter maior autonomia administrativa para executar as tarefas e o poder público ganha meios de controle mais eficazes para avaliar e cobrar os resultados. Como objetivos dessa transformação estão o fornecimento de serviços de infraestrutura de redes IP avançadas, a implantação e a avaliação de novas tecnologias de rede, a disseminação dessas tecnologias e a capacitação de recursos humanos na área de segurança de redes, gerência e roteamento.

A partir de 2005, a comunicação entre os point of presence (PoPs) da rede começou a ser ampliada com o uso de tecnologia óptica, o que elevou a capacidade de operação a 11 Gbps. A base instalada de computadores no Brasil atinge 40 milhões, de acordo com pesquisa da Escola de Administração de Empresas de São Paulo da Fundação Getúlio Vargas. O número, que inclui computadores em empresas e residências, representa um crescimento de 25% sobre a base registrada no mesmo período do ano passado.

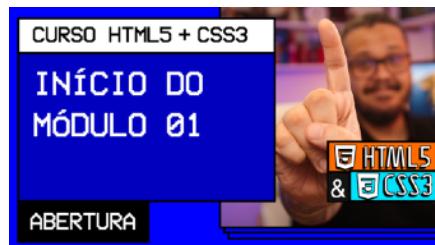


APRENDA MAIS: Quer ver mais informações sobre a chegada da Internet aqui no Brasil? Aqui vai mais um vídeo interessante que vai te contar todos os detalhes.

Canal TecMundo: https://youtu.be/k_inQhpKprg?t=43

Quer acompanhar tudo em vídeo?

Eu sei que às vezes as pessoas gostam mais de assistir vídeos do que ler livros, e é por isso que eu lanço há anos materiais no canal Curso em Vídeo no YouTube. O link que vou compartilhar contigo faz parte da playlist completa onde você encontra o **Módulo 1 do Curso de HTML5 e CSS3**, completamente gravado com base nesse material.



Além de acessar o link a seguir, você também pode ter acesso às aulas apontando a câmera do seu celular para o código QR ao lado. Todo dispositivo smartphone ou tablet atualizado já possui esse recurso de leitura de códigos habilitado por padrão.

Módulo 1 do curso: https://www.youtube.com/playlist?list=PLHz_AreHm4dkZ9-atkcmcBaMZdmLHft8n

Teste seus conhecimentos

Terminou de ler esse capítulo e já acompanhou todos os vídeos e referências externas que indicamos? Pois agora, responda a essas 10 perguntas objetivas e marque em cada uma delas a única opção verdadeira. Aí sim, você vai poder comprovar que realmente entendeu o conteúdo.



1. Em qual período Histórico da humanidade a Internet surgiu?

- A Durante a 2ª Guerra Mundial
- B Durante a Guerra dos 100 anos
- C Durante a Guerra Fria
- D Durante a Guerra do Golfo

2. A agência militar americana DARPA foi a responsável pelo início dos estudos que deram origem à Internet. Logo no início, a rede teve um nome. Você sabe qual foi?

- A DARPANET
- B ARPANET
- C MILNET
- D NSFNET

3. A primeira transmissão registrada na rede foi no ano de _____. A primeira palavra enviada de um computador para outro foi _____.
 A 1949 / HELLO
 B 1955 / FIRST
 C 1962 / WORKS
 D 1969 / LOGIN

4. A primeira versão da rede que deu origem à Internet interligava quantos computadores?

- A 4 computadores
- B 40 computadores
- C 4 mil computadores
- D 400 mil computadores

5. O primeiro protocolo usado na transmissão de dados entre os controladores tinha o nome de:

- A TCP
- B NetBEUI
- C NCP
- D IP

6. No dia 1 de Janeiro do ano de _____ o TCP/IP passou a ser o único protocolo aceito pela Internet. Esse dia ficou conhecido como _____.

- A 1973 / Turn Point
- B 1983 / Flag Day
- C 1985 / New Hope
- D 1988 / Brand Tech

7. O conjunto de protocolos TCP/IP foi criado por um funcionário da DARPA e um pesquisador da universidade de Stanford. São eles, respectivamente:

- A Vint Cerf e Robert Kahn
- B Robert Kahn e Tim Berners-Lee
- C Tim Berners-Lee e Vint Cerf
- D Robert Kahn e Vint Cerf

8. O cientista inglês Tim Berners-Lee foi o responsável pela criação de três coisas muito importantes para a Internet. Foram elas:

- A a linguagem HTML, o protocolo HTTP e o primeiro navegador Mosaic
- B o TCP/IP, o primeiro navegador Mosaic e o nome WWW
- C a linguagem HTML, o protocolo HTTP e o nome WWW
- D o nome WWW, o primeiro navegador Mosaic e a linguagem HTML

9. O primeiro backbone brasileiro surgiu em _____ para acesso acadêmico, mas só foi liberado para empresas em _____.

- A 1991 / 1995
- B 1990 / 1997
- C 1989 / 1991
- D 1985 / 1995

10. Atualmente, a Internet Brasileira atinge quantos estados?

- A 19 estados, além do Distrito Federal
- B 23 estados, além do Distrito Federal
- C 25 estados, além do Distrito Federal
- D 26 estados, além do Distrito Federal

**Cursos que vão te levar
ao próximo nível**

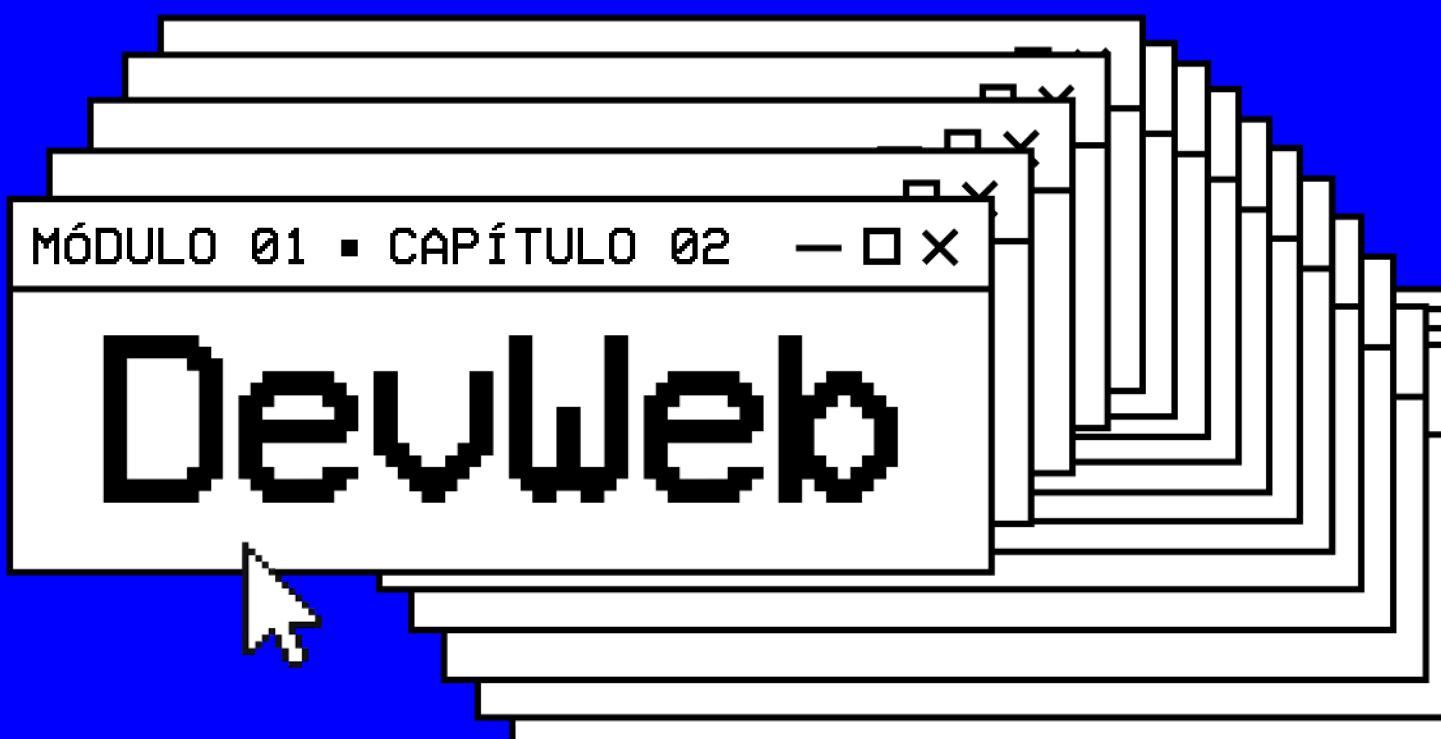


Suas anotações

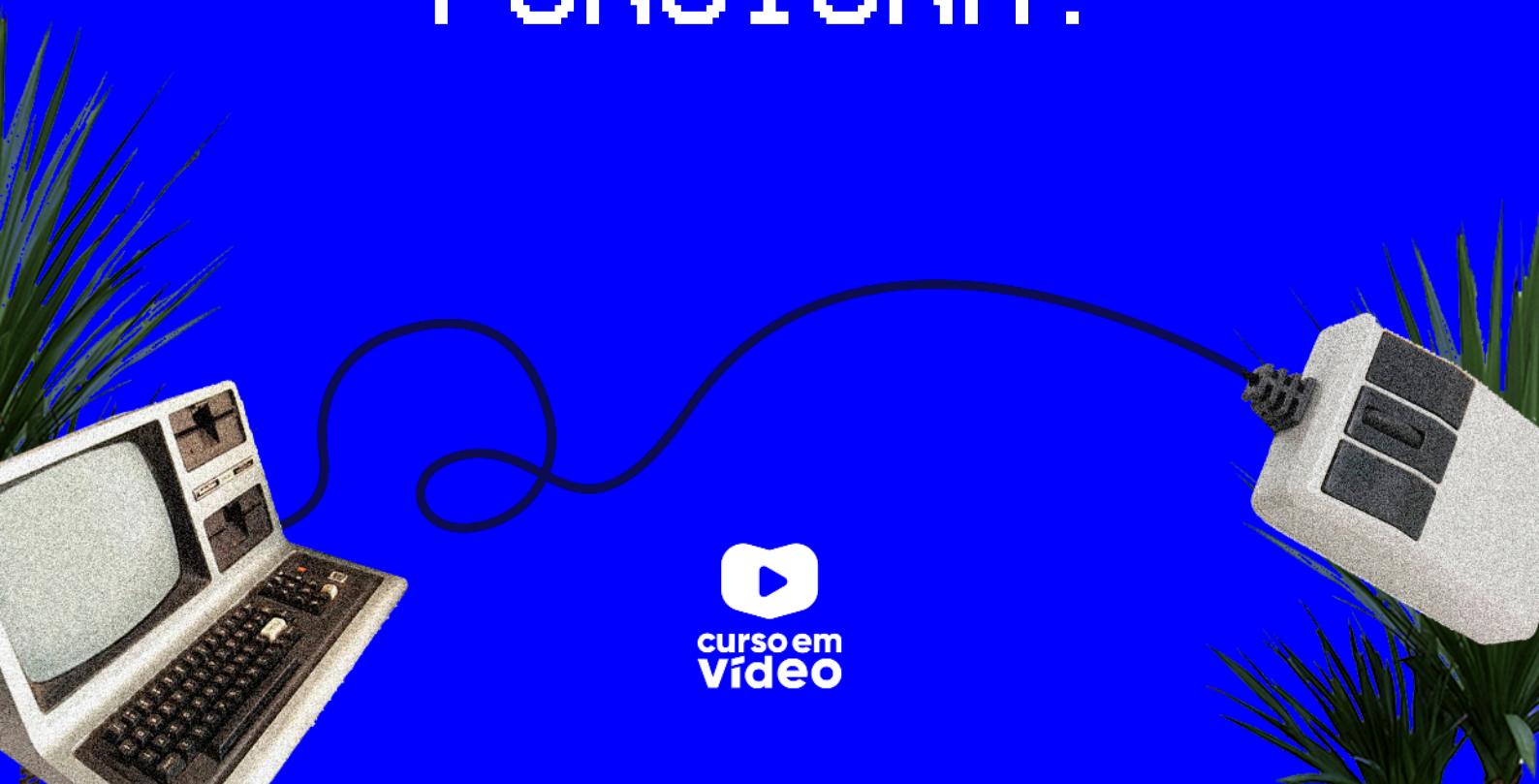
Não guarde conhecimento. Ele é livre. Compartilhe o seu e veja ele se espalhando pelo mundo 

CURSO DE INFORMÁTICA PARA INTERNET

Gustavo Guanabara



COMO A INTERNET FUNCIONA?



M01C02

veja em vídeo



COMO A INTERNET FUNCIONA?

Você é daquele tipo de pessoa curiosa, que não sossega enquanto não descobre como funcionam as coisas? Eu sou assim! Hoje em dia, a maioria das pessoas que conhecemos estão conectadas à Internet. Mas você sabe como funciona essa comunicação? Como é possível conversar ou mandar um arquivo para uma pessoa que está no Japão e tudo chegar em menos de 1 segundo? Esse é o nosso desafio nessa aula.



Você tem todo o direito de usar esse material para seu próprio aprendizado. Professores também podem ter acesso a todo o conteúdo e usá-lo com seus alunos. Porém todos que usarem esse material - seja para qual for a finalidade - deverão manter a referência ao material original, criado pelo **Prof. Gustavo Guanabara** e disponível no endereço do seu repositório público <https://github.com/gustavoguanabara/>. Este material não poderá ser utilizado em nenhuma hipótese para ser replicada - integral ou parcialmente - por autores/editoras para criar livros ou apostilas, com finalidade de obter ganho financeiro com ele.

Tudo começa com o sinal

Você sabe que um computador é um equipamento eletrônico capaz de entender apenas um tipo de sinal, não sabe?

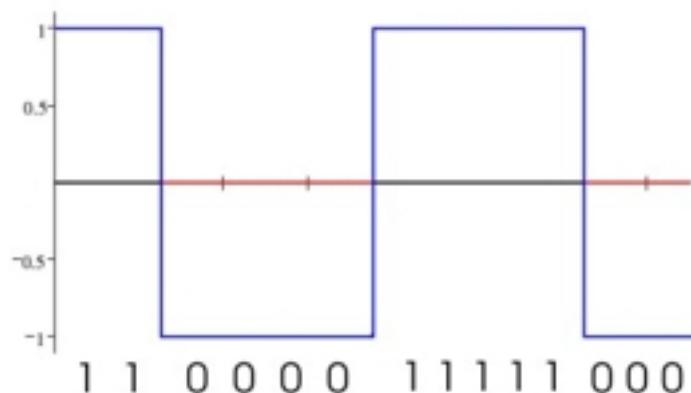
Tem gente que acha que um computador é um dispositivo super inteligente e cheio de capacidades de decidir nosso destino, mas na verdade ele é apenas uma máquina capaz de analisar sinais e fazer contas simples de uma maneira super rápida!

Eu sinto te informar, mas resumindo bastante aqui, o computador só é capaz de compreender duas coisas: 0 e 1.



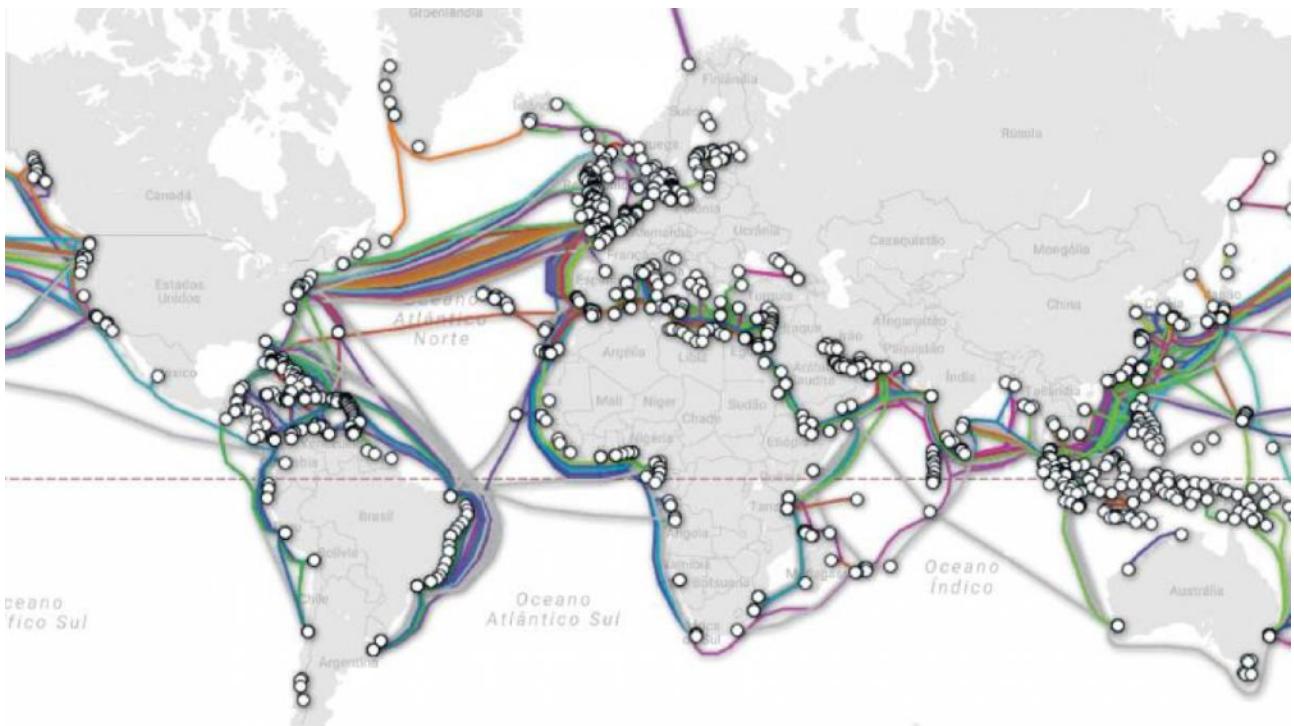
Bits, Bytes e seus múltiplos. Essa é a linguagem do seu computador

Mas é claro que o que circula dentro do seu computador não são pequenos números, são ondas (sinais). No caso de equipamentos eletrônicos processados, as ondas se parecem com as representadas a seguir.



Esta é uma onda que representa os bits, também chamada de onda digital

Como vimos na aula anterior, a Internet é uma rede gigantesca que interliga várias outras redes ao redor do mundo. E precisamos ter meios físicos para levar esses sinais de um lado para o outro.



Este aí é nosso planeta e suas interligações feitas pelos oceanos



APRENDA MAIS: Quer descobrir detalhes de cada um dos cabos que estão representados acima? O site Submarine Cable Map cria um mapa interativo e você pode clicar em cada um deles.

Acesse agora: <https://www.submarinemap.com>

E se nesse exato momento você está confuso(a) com esse conceito, saiba que a maioria das transmissões entre continentes não é realizada pelos **Satélites**, como a maioria das pessoas costuma pensar. Os satélites possuem uma limitação de tráfego e sofrem muito com interferências, e é por isso que os cabos de fibra ótica devem ser passados pelos oceanos. Um trabalho incalculável, mas necessário.

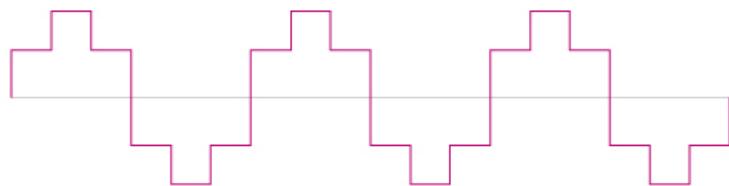


APRENDA MAIS: Quer aprender mais sobre os cabos submarinos? Assista esse vídeo de 9 minutos que explica mais detalhes.

Dicionário de Informática: <https://youtu.be/q-rBtDub3Hc>

Mas não dá para esses cabos submarinos saírem pela praia e seguirem caminho até a sua casa, e por isso precisam se interligar a outros sistemas de comunicação. Alguns dos sistemas utilizados sempre foram a telefonia tradicional, os sistemas de TV a cabo, os sinais via satélite e até as simples redes de radiofrequência.

sinal digital



sinal analógico



O problema é que os sistemas diferentes transmitem sinais em formatos diferentes, como você pode ver na imagem acima. Isso dificultaria a comunicação entre pontos, se não fosse um processo de "conversão", mais conhecido como **MODULAÇÃO**.

De uma maneira bem resumida, modular é conseguir ler uma onda no formato A compatível com um tipo de sistema de comunicação e convertê-la para um formato B, compatível com outro tipo de sistema.

E é exatamente para isso que servem aqueles aparelhos que você instala em sua casa para começar a receber Internet doméstica



Aparelho muito comum em casas com Internet

The banner features a purple background with white text and graphics. At the top, there's a standard window control bar with minimize, maximize, and close buttons. Below that, the text 'Soluções digitais para negócios' is displayed in large, bold, white font. To the right of the text is a QR code. At the bottom, the Hostnet logo is shown, consisting of a stylized cloud icon followed by the word 'hostnet' in lowercase.

Uma das funções desse aparelho é **MODULAR** os sinais que saem e **DEMODULAR** os sinais que chegam. E é por isso que chamamos esse aparelho de **MODEM**.

Já que falamos de Roteadores

As rotas são outro assunto muito importante para o funcionamento da Internet. Pense na rede como se fosse como se fosse um mapa com várias ruas, como no desenho a seguir. De quantas maneiras diferentes podemos chegar do ponto A até o ponto B? Eu imaginei três rotas diferentes e representei usando as linhas coloridas. Quem vai decidir a melhor rota é o Aplicativo de GPS, já que podem existir engarrafamentos e ruas fechadas.

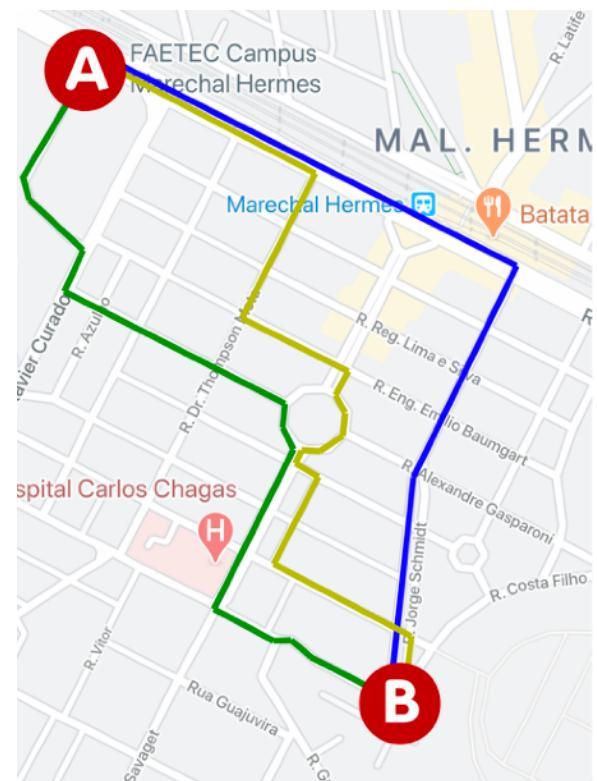
Na Internet também é assim. Para enviar um sinal de um dispositivo A para um dispositivo B, podemos ter várias rotas. Quem vai definir a melhor rota são os **ROTEADORES** que compõem a rede. Os pacotes de dados podem chegar em seu computador por diversas rotas diferentes, tudo vai depender do tráfego no momento da transmissão.

Agora que você já sabe como funcionam as rotas, vamos falar sobre os pontos.

Cliente e Servidor

Volte uma página e olhe aquele mapa de novo. Imagine que o ponto A é você na escola pedindo uma pizza. O ponto B é a pizzaria, que vai te fornecer o pedido que vai matar sua fome. Quando o pedido for confirmado, o motoboy que vai fazer a entrega é o pacote. Ele vai levar seu pedido até você por uma rota.

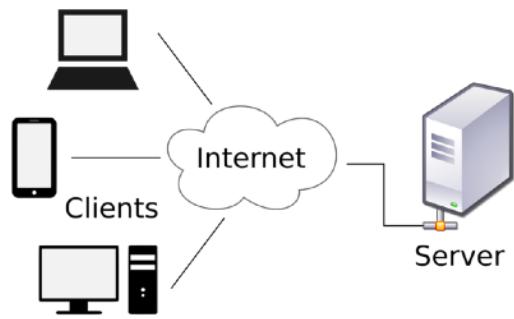
Nessa situação que descrevi acima, você no ponto A é o **CLIENTE**. A pizzaria no ponto B é o **SERVIDOR**. O motoboy é o **PACOTE** e sua pizza é o **DADO**. A Internet também vai funcionar dessa maneira.



IMPORTANTE!

Quando você estudar um pouco mais sobre redes de comunicação, vai descobrir que a função **MODEM** desses aparelhos é apenas uma das características do produto. Na verdade, esse dispositivo é um **GATEWAY**, que vai se ligar aos **ROTEADORES** do seu provedor de acesso, mas resolvi simplificar a explicação para não aprofundar tanto assim no início de tudo.

Olhe agora o desenho ao lado. Ele é bem parecido com a história do mapa. Imagine que você está no seu celular tentando acessar um site. O seu celular é o **CLIENTE** e está pedindo algo pela Internet. Ao descobrir onde está o site, a máquina que está hospedando ele será o **SERVIDOR**, que vai fornecer os arquivos que compõem o site. O caminho que vai criar uma ligação entre o servidor e você (cliente) vai ser decidido pelos roteadores da Internet.



Um servidor pode estar no seu bairro, na sua cidade, no seu país ou até mesmo do outro lado do mundo. Os pacotes podem girar o mundo todo em poucos segundos e o resultado será exibido na tela do seu celular/computador como se fosse magia, mas é pura TECNOLOGIA!

Na Internet existem vários servidores:

- Servidor de site (também chamado de WebHost)
- Servidor de streaming
- Servidor de arquivos
- Servidor de e-mail
- E muito mais...

Mas como será que o mecanismo da Internet consegue descobrir o local exato de um site? Como ele descobre em que servidor ele está? E como ele consegue encontrar a posição exata do servidor no Globo? Aí entramos no próximo assunto.

Identificando os nós

Como vimos anteriormente, a Internet funciona baseada em um conjunto de protocolos chamado **TCP/IP**. Um protocolo garante que todas as comunicações seguirão um mesmo padrão, permitindo que dispositivos que são diferentes, com tecnologias completamente distintas, possam se trocar mensagens.

Uma das funções do TCP/IP, mais especificamente do IP, é identificar os nós. Mas o que seria esse nó?

A resposta é simples: um nó é cada ponto que está conectado à rede. Quando você "se conecta" à Internet, recebe uma identificação única. Essa identificação é um **ENDEREÇO IP**.

Cursos grátis de tecnologia
que te preparam para o
mercado de trabalho

RECODE

QR code

Os IPs mais antigos (IPv4) usam 4 octetos, que são conjuntos de 8 bits separados por pontos, totalizando 32 bits por identificador.

Ex: **123.45.67.89** = 01111011.00101101.01000011.01011001

Os IPs mais modernos (IPv6), usam 128 bits ao todo (o que é 4x mais bits que o IPv4).

Ex: **2001:0db8:85a3:08d3:1319:8a2e:0370:7344**

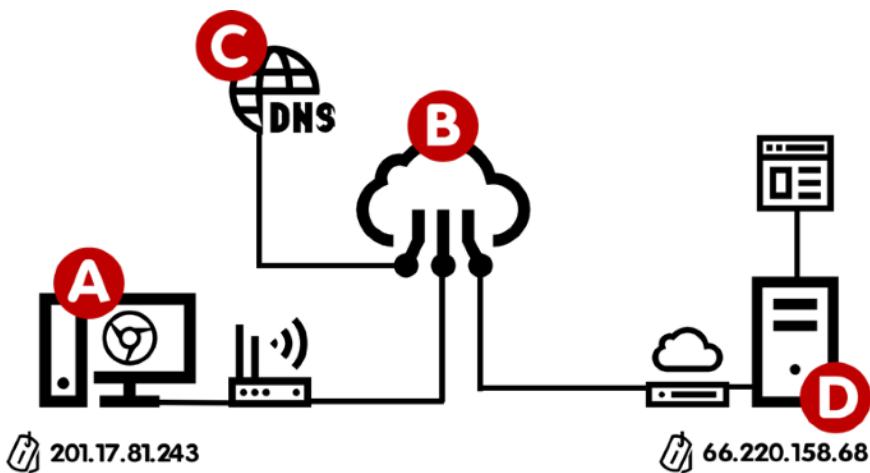


APRENDA MAIS: Entenda melhor qual é a necessidade de migrar do IPv4 para o IPv6. E o que acontecerá no futuro, quando a versão antiga parar de funcionar.

NICbr: https://youtu.be/_JbLr_C-HLK

Acessando um servidor

Agora que você já sabe como os pontos são identificados, vamos criar um simples cenário aqui. Analise a imagem abaixo e veja que você estaria no ponto A, tentando acessar o site que está guardado no servidor que é o ponto D.



Você também deve ter notado que o ponto A tem um IP (201.17.81.243) e o ponto D também tem o seu (66.220.158.68). Agora imagine que você deva ter que decorar o IP do seu site favorito. Isso dificultaria todo o processo, não é?

E é para isso que existe o **DOMAIN NAME SYSTEM**, ou sistema de nomes de domínio. Eles são como grandes “listas telefônicas”, criando uma ligação entre o nome do site e o número de IP relacionado a ele.

Importante deixar bem claro: os números de IP mudam constantemente! Sempre que você desconecta o gateway da sua operadora (aquele aparelho que tem instalado na sua casa), o seu número de IP vai mudar.



APRENDA MAIS: Saiba mais sobre o DNS assistindo esse vídeo bem simples e ilustrado.

NICbr: <https://youtu.be/ACGuo26MswI>

É possível fazer um exercício simples para descobrir o seu próprio IP ou até mesmo descobrir o IP atual de um site que você esteja acessando. Descubra como, lendo o quadro informativo abaixo.

Agora vamos voltar ao desenho da página anterior e entender o passo-a-passo desse acesso.

1. Você está no **ponto A** (conectado à Internet) e digita o endereço do site que está querendo acessar (ex: www.facebook.com).
2. A arquitetura da Internet (**ponto B**) vai encontrar o **Servidor DNS** que terá o registro do IP referente ao nome que você digitou.
3. O Servidor DNS, que é o **ponto C** do nosso desenho, vai resolver o endereço e retornar o IP atual do site que você pediu (ex: no dia em que eu escrevi esse material, o IP atual do facebook.com era 66.220.158.68).
4. Uma vez que seu navegador já sabe o IP que deseja acessar, vai poder se direcionar diretamente ao servidor correto.
5. Assim que chega uma solicitação ao **ponto D**, ele vai procurar a página solicitada e te enviar uma cópia do documento para o seu computador.



APRENDA MAIS: Tente descobrir o seu IP e também tente descobrir o IP atual do site do Facebook usando o site a seguir:

IP2Location: <https://www.iplocation.net>

EXTRA: Depois que você descobrir o IP do Facebook, tente digitar o número dele no lugar do endereço do site no seu Google Chrome e aperte o enter. Surpresa!

OBS: Caso você não solicite um documento específico, o servidor vai te enviar o arquivo de índice (geralmente chamado `index.html`).

6. Agora que o seu computador no **ponto A** tem o arquivo HTML, vai poder analisá-lo para descobrir que arquivos extras ele vai precisar (fotos, vídeos, estilos, etc). A partir daí ele vai fazer outras solicitações ao **ponto D**, para que ele possa enviar esses recursos extras. É por conta disso que os sites que você visita vão aparecendo aos poucos.

Domínio e Hospedagem

Tá aí um assunto que muita gente iniciante fica na dúvida. Até mesmo seus clientes vão perguntar sobre isso, pois serão duas contas pra pagar.

Quando você criar um site, vai querer que o maior número de pessoas tenha acesso a ele, não é? O problema é que enquanto você está desenvolvendo HTML em casa, na maioria das vezes esses arquivos estão guardados no seu próprio PC. Dessa maneira, não existe um endereço para outras pessoas poderem acessar o seu trabalho de fora.



Vamos começar com o Domínio

Quando você acessa um site, precisa saber uma **URL** (*Uniform Resource Locator*) para poder acessá-lo. A parte principal dessa URL é o **domínio**.

O **domínio** é um nome único que vai conseguir identificar o seu servidor ou as suas páginas. Para conseguir um domínio, você deve pagar pelo direito de usá-lo por um período mínimo de 1 ano.

Vejamos alguns exemplos de domínio:

cursoemvideo.com
faetec.rj.gov.br
github.io
universidadebrasil.edu.br



Analisando os endereços acima, temos domínios com várias terminações, como .com, .gov.br, .io, .edu.br. Essas terminações indicam tecnicamente que o site é de uma instituição comercial, governamental, educacional, ONGs, artistas, etc.

Além disso, alguns desses ainda indicam o país (.br). Esses são os **TLD** (*Top Level Domain*).

GTLD: São TLDs genéricos, sem indicação de país. Alguns dos domínios genéricos são .com, .net, .gov, .org, .io, .info, .online, .store, etc.

ccTLD: São TLDs com designação do país (*country code*). Alguns dos domínios desse tipo são .com.br, .edu.us, .co.fr, .jp, .es, etc.

Da junção entre o nome e a terminação, temos um domínio. Esse domínio pode ser usado para ter acesso ao seu servidor, seja ele web, servidor de ftp, servidor de e-mail (as contas de e-mail são no formato josesilva@cursoemvideo.com, onde o nome do usuário vem antes da @ e o domínio vem depois)

E já que eu falei de **sub-domínio**, podemos ter outros alem do www, ok? Um mesmo site pode ter vários sub-domínios. Vamos usar o Google como exemplo:

www.google.com.br - Dá acesso ao site principal do Google
images.google.com.br - Você vai para o Google Imagens
maps.google.com.br - Acessa o Google Maps
mail.google.com.br - Entra no Gmail



CUIDADO! Tem muita gente que pensa que o **www** que vem antes da URL faz parte do domínio. Não faz! Esse www é um sub-domínio que aponta diretamente para o seu servidor web padrão.

Nesse caso o www, images, maps e mail são sub-domínios do domínio google.com.br. Sacou?

Além do sub-domínio e domínio, uma URL também é composta pelo protocolo utilizado, que pode ser `http://` ou `https://`, dependendo se o seu servidor tem ou não segurança por **SSL** (um serviço de criptografia de dados).

E por fim, uma URL também pode ter um caminho extra, que indica pastas a percorrer para achar um arquivo específico. Por exemplo:

`https://www.github.com/gustavoguanabara/html-css/tree/master/aulas-pdf`

Identifique na URL acima os componentes da URL. Faça esse exercício.

Como vimos, o domínio é uma maneira mais fácil para que outras pessoas possam chegar aos arquivos que vão compor o seu site. Mas onde estarão esses arquivos?

Agora chegou a vez da Hospedagem

No momento em que você começa a desenvolver seu site, seus arquivos estão no PC da sua casa.



Mas seu computador fica ligado 24h por dia, 7 dias por semana, 365 dias ao ano? Se faltar luz, você tem uma sub-estação de energia pra manter tudo funcionando em caso de panes? Você tem uma banda de internet suficiente para suportar milhares ou até milhões de acessos simultâneos?

Pois para manter seus arquivos acessíveis, você vai precisar contratar um plano de hospedagem justamente para que tenha tudo aquilo que foi enumerado acima. Para isso, existem grandes **data centers** com toda a infraestrutura para fornecer disponibilidade, segurança e performance para guardar seus arquivos. Algumas **empresas de hospedagem** contratam esses data centers para fornecer espaços para colocar servidores e alugam esses servidores para nós.



VEJA POR DENTRO: Dá só uma olhada em como é um data center por dentro. E você aí achando que seu PC com no-break já dava conta do recado 😊

Data Center Ascenty: https://youtu.be/qV1Mg-H_fBI

Data Center UOL: <https://youtu.be/flxzjA4tM1E?t=74>

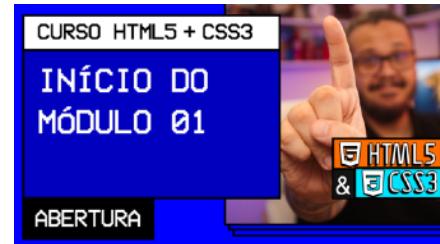
Data Center Locaweb: <https://youtu.be/a2vfVALjv18>

A **Hostnet**, que sempre foi nossa patrocinadora oficial do **Curso em Vídeo**, tem seus servidores na **Ascenty**, um dos maiores data centers do Brasil. Ao contratar uma **hospedagem** para o seu site, o plano normalmente inclui recursos como espaço em disco, memória, banco de dados, contas de e-mail, webmail, sub-domínios, certificados de segurança, etc. Nem todas as hospedagens oferecem tudo isso, por isso é muito importante saber escolher. Nós, é claro, recomendamos sempre a **Hostnet**.



Quer acompanhar tudo em vídeo?

Eu sei que às vezes as pessoas gostam mais de assistir vídeos do que ler livros, e é por isso que eu lanço há anos materiais no canal Curso em Vídeo no YouTube. O link que vou compartilhar contigo faz parte da playlist completa onde você encontra o **Módulo 1 do Curso de HTML5 e CSS3**, completamente gravado com base nesse material.



Além de acessar o link a seguir, você também pode ter acesso às aulas apontando a câmera do seu celular para o código QR ao lado. Todo dispositivo smartphone ou tablet atualizado já possui esse recurso de leitura de códigos habilitado por padrão.

Módulo 1 do curso: https://www.youtube.com/playlist?list=PLHz_AreHm4dkZ9-atkcmcBaMZdmLHft8n

Teste seus conhecimentos

Terminou de ler esse capítulo e já acompanhou todos os vídeos e referências externas que indicamos? Pois agora, responda a essas 10 perguntas objetivas e marque em cada uma delas a única opção verdadeira. Aí sim, você vai poder comprovar que realmente entendeu o conteúdo.



1. Atualmente, na maioria das transmissões feitas mundialmente pela Internet, por onde a maior quantidade de sinal vai passar?

- A Por baixo d'água, em cabos submarinos
- B Por sinal de satélite, partindo de antenas
- C Por antenas de comunicação via celular
- D Por cabos enterrados nos continentes

2. Os dados que vão trafegar pelas redes e chegar aos computadores vão estar codificados em:

- A letras e símbolos
- B conjuntos de bits e bytes (0s e 1s)
- C ondas codificadas sem um padrão definido
- D arquivos de imagem / textos

3. Quando um sinal está representado em um determinado tipo de onda (codificação) e precisamos transformá-lo em outro tipo, dizemos que estamos realizando uma:

- A extração
- B composição
- C descompressão
- D modulação

4. Na maioria das vezes que estamos realizando algum tipo de acesso via rede, um lado será o _____ e vai solicitar o uso de um determinado serviço. Quem vai atender à essa solicitação é considerado um _____.

- A servidor / cliente
- B cliente / servidor
- C solicitante / provedor
- D provedor / servidor

5. Todo e qualquer ponto conectado à Internet recebe um identificador, que se chama “endereço IP”. Na versão IPv4, um endereço possui _____ bits, já a versão IPv6 necessita de _____ bits para identificar um ponto.

- A 32 bits / 64 bits
- B 64 bits / 128 bits
- C 32 bits / 128 bits
- D 16 bits / 64 bits

6. Qual é o serviço (e seu respectivo significado) responsável por resolver os endereços IP dos servidores a partir de um nome? Ele faz com que os usuários não precisem decorar números IP (que inclusive, mudam constantemente).

- A DNS - Domain Name System
- B DNS - Domain Name Service
- C DNS - Domain Name Server
- D DNS - Data Name Server

Cursos que vão te levar
ao próximo nível



7. Uma URL é composta por vários componentes. Na URL `https://www.github.com/gustavoguanabara`, por exemplo, são respectivamente o domínio, o protocolo e o caminho os itens apontados na opção:

- A `github.com / https / gustavoguanabara`
- B `github.com / www / gustavoguanabara`
- C `github / https / gustavoguanabara`
- D `www.github.com / www / gustavoguanabara`

8. Quando chegar a hora de construir nossos sites, devemos informar aos nossos clientes que geralmente o domínio tem uma taxa de pagamento _____, enquanto as hospedagens são de pagamento _____ de forma geral.

- A mensal / anual
- B anual / diário
- C mensal / diário
- D anual / mensal

9. Os domínios possuem seus Top Level Domains. Vejamos como exemplo os domínios `estudonauta.com.br` e `cursoemvideo.com`. No primeiro caso, temos um _____ e no segundo caso, temos um _____.

- A GTLD / ccTLD
- B ccTLD / GTLD
- C DTLD / CTLD
- D CTLD / DTLD

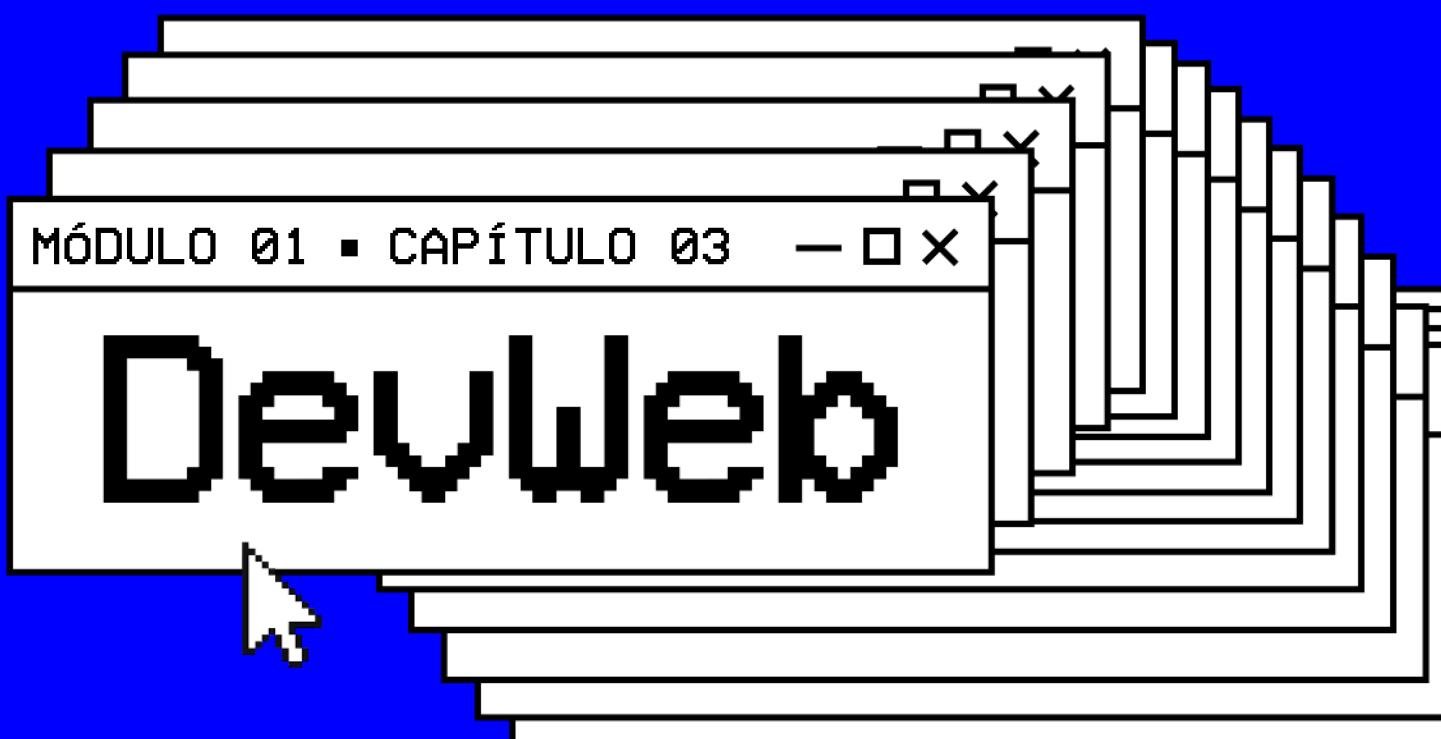
10. Todo bom serviço de hospedagem deve fornecer recursos e serviços valiosos para seus clientes. Entre os itens a seguir, qual é o único que NÃO É oferecido por uma empresa de hospedagem?

- A atendimento de suporte
- B espaço em disco para armazenamento
- C manutenção periódica no PC do cliente
- D backup constante dos arquivos e bancos de dados

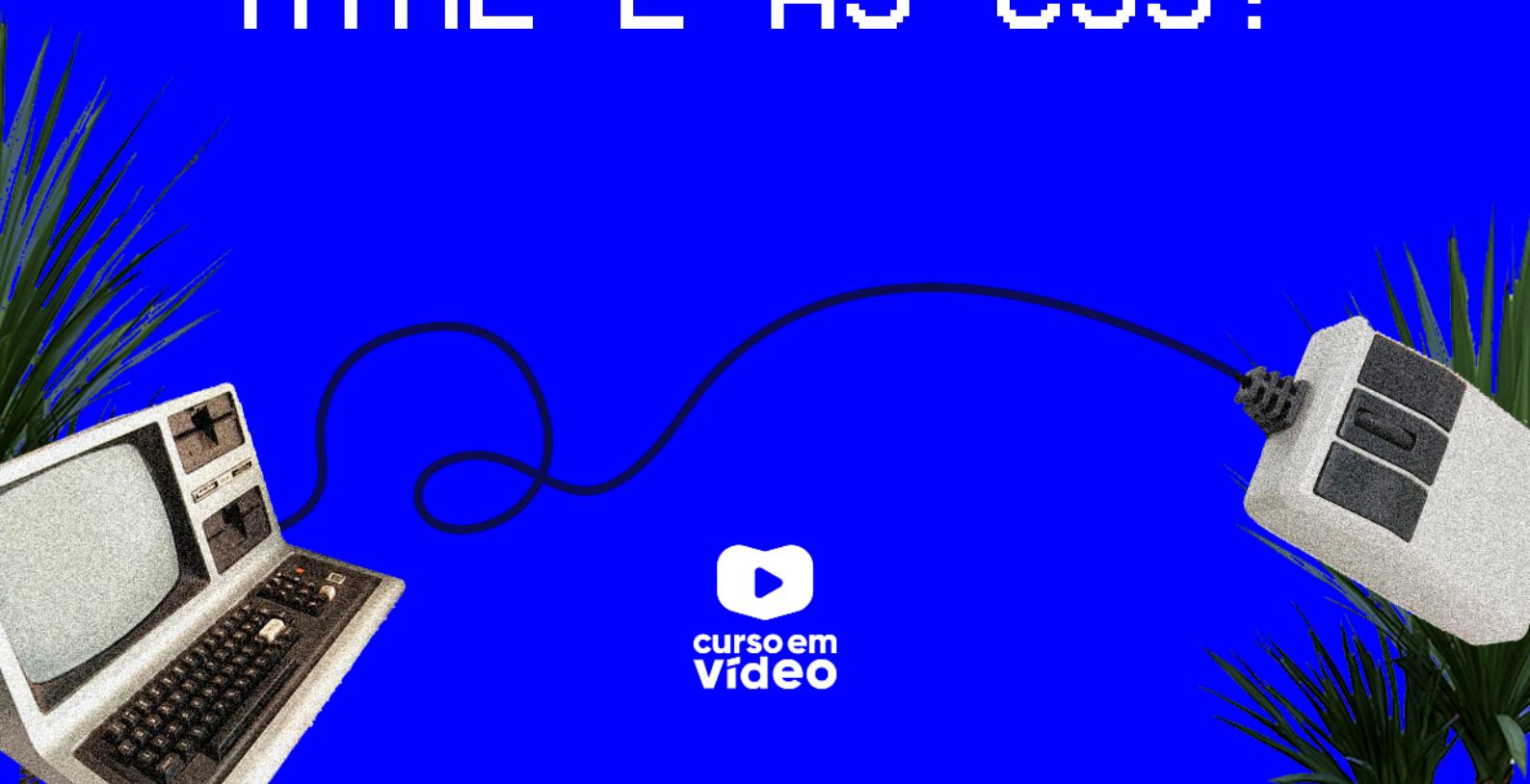
Suas anotações

Não guarde conhecimento. Ele é livre. Compartilhe o seu e veja ele se espalhando pelo mundo 

Gustavo Guanabara



COMO FUNCIONA A HTML E AS CSS?





COMO FUNCIONA A HTML E AS CSS?

Pode ser estranho pra você ler “a HTML” e também “as CSS”. Muita gente trata as duas tecnologias com o artigo masculino na maioria das vezes. Na verdade, isso nem está totalmente errado, mas eu sempre vou considerar defini-las pela tradução dos seus termos. HTML é uma Linguagem de Marcação Hipertexto. CSS são as Folhas de Estilo em Cascata. Notou que são termos em feminino? Pois as polêmicas não param por aqui não...



Você tem todo o direito de usar esse material para seu próprio aprendizado. Professores também podem ter acesso a todo o conteúdo e usá-lo com seus alunos. Porém todos o que usarem esse material - seja para qual for a finalidade - deverão manter a referência ao material original, criado pelo **Prof. Gustavo Guanabara** e disponível no endereço do seu repositório público <https://github.com/gustavoguanabara/>. Este material não poderá ser utilizado em nenhuma hipótese para ser replicada - integral ou parcialmente - por autores/editoras para criar livros ou apostilas, com finalidade de obter ganho financeiro com ele.

“Eu programo em HTML”

Talvez você já tenha ouvido alguém falar a frase acima. Geralmente ela é dita por pessoas iniciantes ou então aqueles que se baseiam apenas na capa de um único livro da série **Head First** (use a cabeça), que estampa na capa “*programação em HTML5*”.

Acontece que a própria sigla já entrega seu objetivo: **Hypertext Markup Language** traduzido para o bom e velho Português significa **Linguagem de Marcação Hipertexto**. Você vai notar daqui a pouco que ela não funciona com instruções, como toda linguagem de programação. A HTML é baseada em **marcações** chamadas **tags**, e elas comandam tudo.



Além disso, o termo “**programação**” envolve estruturas especializadas que dependem do uso de **variáveis simples** e **compostas**, **condições**, **laços** e até coisas mais complexas como **objetos**. Nada disso existe na HTML nem nas CSS. Por outro lado, todas essas características estão presentes na linguagem **JavaScript**. Essa sim é uma **Linguagem de Programação**.

HTML trabalha fundamentada apenas nas **marcas** ou **etiquetas** (do Inglês **tag**) e as CSS funcionam baseadas nos **seletores**, **propriedades** e **valores**. Você vai entender mais sobre isso mais pra frente.

Sendo assim, em breve você terá a base suficiente para conseguir dizer com certeza que HTML e CSS não são linguagens de programação.



IMPORTANTE! Dizer que HTML e CSS não são linguagens de programação não as torna **RUINS** de maneira nenhuma! O que estamos fazendo aqui é uma simples classificação. Não há nada de errado com elas e nenhuma outra linguagem é melhor ou pior que elas por conta disso, ok?

Para que serve HTML, CSS e JS?

Uma das coisas mais importantes para quem está começando o desenvolvimento de sites é compreender para que serve esse trio de tecnologias, que geralmente são estudados em conjunto. Basicamente, de forma resumida, temos um panorama simples:

HTML	Conteúdo
CSS	Estilo
JavaScript	Interatividade

Guarde bem a tabela anterior sempre que você precisar decidir qual linguagem vai utilizar em cada situação.

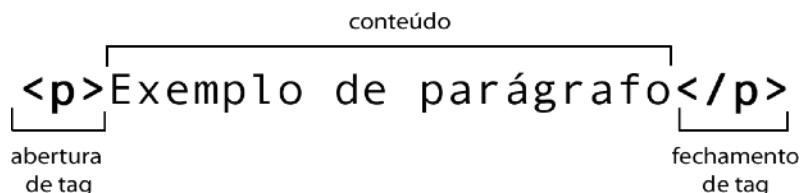
Abra aí o seu site de notícias favorito. Ao abrir uma determinada notícia, você vai ver o texto, as imagens, os vídeos e todo aquele conteúdo que compõe a notícia em si. Isso tudo foi criado em **HTML**. Ela é focada em **conteúdo**.

Agora preste atenção nas cores, na posição dos componentes e organização visual do conteúdo em colunas, blocos visuais e tudo mais. Tudo foi definido em **CSS**. Ela é focada no **design/estilo**.

Finalmente, provavelmente existe o menu do site. Quando você clica nele, acontece uma animação. Ao mover o mouse sobre as sessões, é possível que aconteçam algumas interações interessantes. Isso foi desenvolvido com ajuda de **JavaScript**. Ela é uma linguagem focada nas **interações**.

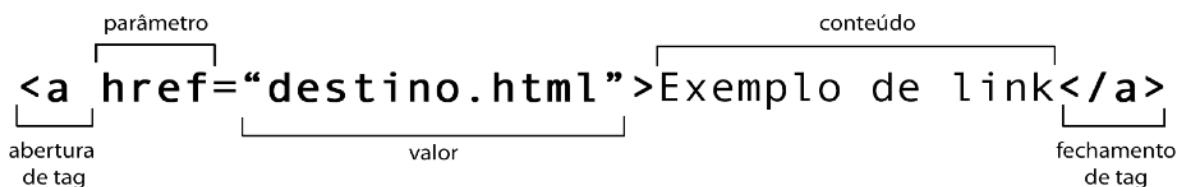
Tags HTML, aí vamos nós

Como eu já disse anteriormente, a HTML funciona baseada em marcações específicas chamadas **tags**. Uma tag é um conjunto de palavras entre sinais de **colchete angular**, conforme representado a seguir.



Na imagem anterior, você consegue perceber o uso da tag <p> para a criação de um parágrafo simples. A maioria das tags possuem uma **abertura** e um **fechamento**, e você identifica isso pela presença da barra no fechamento da tag.

Além disso, as tags também podem ter atributos e valores, que vão configurar seu comportamento:



O banner tem uma barra superior branca com ícones de minimização, maximização e fechar. O fundo é escuro com formas geométricas. No topo, a frase "Cursos que vão te levar ao próximo nível" é exibida em branco. No lado direito, há o logo do Estudonauta (uma figura com óculos) e uma扫二维码图标. Abaixo, o nome "estudonauta" é escrito.

Uma mesma tag pode ter vários parâmetros, cada um com seu valor. Entretanto, algumas tags não possuem a necessidade de **conteúdo** interno e por isso não possuem fechamento. É o caso, por exemplo, das tags `
` e ``. Isso é algo natural, não se preocupe com isso agora.

Eu ainda uso ``, `<center>`, `<s>`, `<u>`, ...

Com o surgimento da versão 5 da HTML, algumas tags simplesmente deixaram de existir ou tornaram-se **obsoletas**. Uma tag obsoleta pode até estar funcionando no seu navegador hoje em dia, mas a própria **W3C** - consórcio responsável por manter as especificações da linguagem - recomenda que elas **não sejam mais usadas** pelos profissionais e aos poucos não serão mais suportadas pelos navegadores nas suas futuras versões.



APRENDA MAIS: Sempre que você quiser saber quais são as tags que estão sendo consideradas obsoletas pelo **W3C**, basta consultar a referência oficial da linguagem, disponível no site abaixo e acessar o **item 15: Obsolete Features**.

HTML Standard: <https://html.spec.whatwg.org/multipage/>

De forma simples e direta (vou até escrever “gritando” aqui, pra dar ênfase): **NÃO USE TAGS OBSOLETAS NO SEU SITE!** Ufa! Que alívio colocar isso pra fora e desabafar 😊

Chegou a vez dos seletores CSS

Como já vimos anteriormente, as **CSS** são as **Cascading Style Sheets** (Folhas de Estilo em Cascata). Elas são usadas para configurar um **resultado visual** dos elementos HTML.

As configurações das CSS são realizadas através dos **seletores**. Vamos ver a anatomia de um seletor.

```
seletor
[ p {           declaração
    font-family: Arial;
    font-size: 12pt;
    color: blue;
}
propriedade   valor
```

O seletor apresentado anteriormente vai configurar o visual dos elementos de parágrafo do site corrente. O uso das chaves delimita todas as declarações relativas ao seletor atual. No seletor que eu te mostrei, serão feitas três configurações:

- A fonte escolhida foi *Arial*.
- O tamanho da letra será *12pt* (pontos).
- A cor da letra será *azul*.

Note que, ao final de cada **declaração**, temos que colocar ponto-e-vírgula para indicar que ela se encerrou.

Todas as **propriedades** devem ter seu **valor**, e eles devem ser separados por dois pontos. Você não é obrigado(a) a usar nenhuma declaração específica. Só utilize a propriedade que você realmente deseja alterar.

Estrutura básica de um documento HTML

Ao criar um novo documento HTML, devemos sempre escrever a estrutura básica de um documento desse formato. Vamos analisar cada uma das 11 linhas que compõem esse documento base.

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4  |   <meta charset="UTF-8">
5  |   <meta name="viewport" content="width=device-width,
6  |   initial-scale=1.0">
7  |   <title>Document</title>
8  </head>
9
10 <body>
11 </html>
```

Cursos grátis de tecnologia
que te preparam para o
mercado de trabalho

RECODE



- **Linha 1:** Indica que o documento atual será escrito na versão mais atualizada da linguagem (no caso, HTML5)
- **Linhas 2 e 11:** Delimitam o documento HTML, que é sempre dividido em duas partes: a cabeça e o corpo. Na linha 2, também estamos indicando que o conteúdo desse site será no idioma Português do Brasil.
- **Linhas 3 e 7:** Delimitam a **cabeça** da página, local onde são realizadas algumas configurações iniciais como formatos, estilos, ícone de favoritos, etc.
- **Linha 4:** adiciona ao documento atual o suporte a caracteres acentuados. Remover essa linha pode causar erros de renderização de algumas letras na tela.
- **Linha 5:** Indica que o conteúdo aparecerá, por padrão, ocupando todo o espaço disponível da tela e com uma escala de 1:1.
- **Linha 6:** Configura o título da página, que aparecerá como identificação da aba do navegador, ao lado do *favicon*.
- **Linhas 8 e 10:** Delimitam o **corpo da página**, a maior porção do site, que vai aparecer na tela. É aqui onde colocaremos todo o nosso conteúdo.

Por enquanto é isso!

Agora você já conhece a base suficiente para começar a criar seus próprios sites básicos. No próximo material vamos instalar os softwares necessários para a criação de documentos e começar a aprender as principais tags. Complemente sempre o nosso conteúdo com os vídeos que eu indico no final de cada material.

Quer acompanhar tudo em vídeo?

Eu sei que às vezes as pessoas gostam mais de assistir vídeos do que ler livros, e é por isso que eu lanço há anos materiais no canal *Curso em Vídeo* no YouTube. O link que vou compartilhar contigo faz parte da playlist completa onde você encontra o **Módulo 1 do Curso de HTML5 e CSS3**, completamente gravado com base nesse material.



Além de acessar o link a seguir, você também pode ter acesso às aulas apontando a câmera do seu celular para o código QR ao lado. Todo dispositivo smartphone ou tablet atualizado já possui esse recurso de leitura de códigos habilitado por padrão.

Módulo 1 do curso: https://www.youtube.com/playlist?list=PLHz_AreHm4dkZ9-atkcmcBaMZdmLHft8n

Teste seus conhecimentos

Terminou de ler esse capítulo e já acompanhou todos os vídeos e referências externas que indicamos? Pois agora, responda a essas 10 perguntas objetivas e marque em cada uma delas a única opção verdadeira. Aí sim, você vai poder comprovar que realmente entendeu o conteúdo.



1. Qual das frases a seguir é a única tecnicamente CORRETA de se falar?

- A "Eu programo em linguagem HTML"
- B "Eu programo em linguagem CSS"
- C "Eu programo em linguagem JavaScript"
- D "Eu programo em linguagem VSCode"

2. A sigla HTML significa:

- A Host Text Makeup Language
- B HyperText Markup Language
- C Hyper Tree Makeup Language
- D Host Tree Markup Language

3. A sigla CSS significa:

- A Cascading Style Sheets
- B Cell Safety Science
- C Characteristic Score Style
- D Chief Scale Sheets

4. Correlacione a coluna da esquerda com a da direita, de acordo com o foco de cada uma das tecnologias:

(1) HTML	() interatividade
(2) CSS	() conteúdo
(3) JS	() estilo

- A 1 - 2 - 3
- B 3 - 2 - 1
- C 1 - 3 - 2
- D 3 - 1 - 2

5. Qual tag abaixo não tem fechamento?

- A <title>
- B <meta>
- C
- D <head>

6. Na tag <a>, o href é um(a):

- A conteúdo
- B parâmetro
- C característica
- D valor

7. Todas as configurações visuais dos elementos HTML são realizadas pela linguagem CSS. Agrupamos todas as declarações CSS de um mesmo elemento dentro de um(a):

- A seletor
- B parâmetro
- C valor
- D selecionador

8. Para mudar a cor de um texto em CSS, configuramos qual propriedade?

- A text
- B text-color
- C color
- D font-color

Soluções digitais para negócios

 hostnet



9. Para indicar que um determinado documento HTML está escrito na versão mais recente da linguagem, devemos adicionar a seguinte instrução:

- [A] <html lang="version5">
- [B] <title>HTML5</title>
- [C] <meta name="lang" type="html5">
- [D] <!DOCTYPE html>

10. Qual é a tag de um documento HTML adicionada pra manter a compatibilidade com os caracteres acentuados, muito comuns na língua Portuguesa?

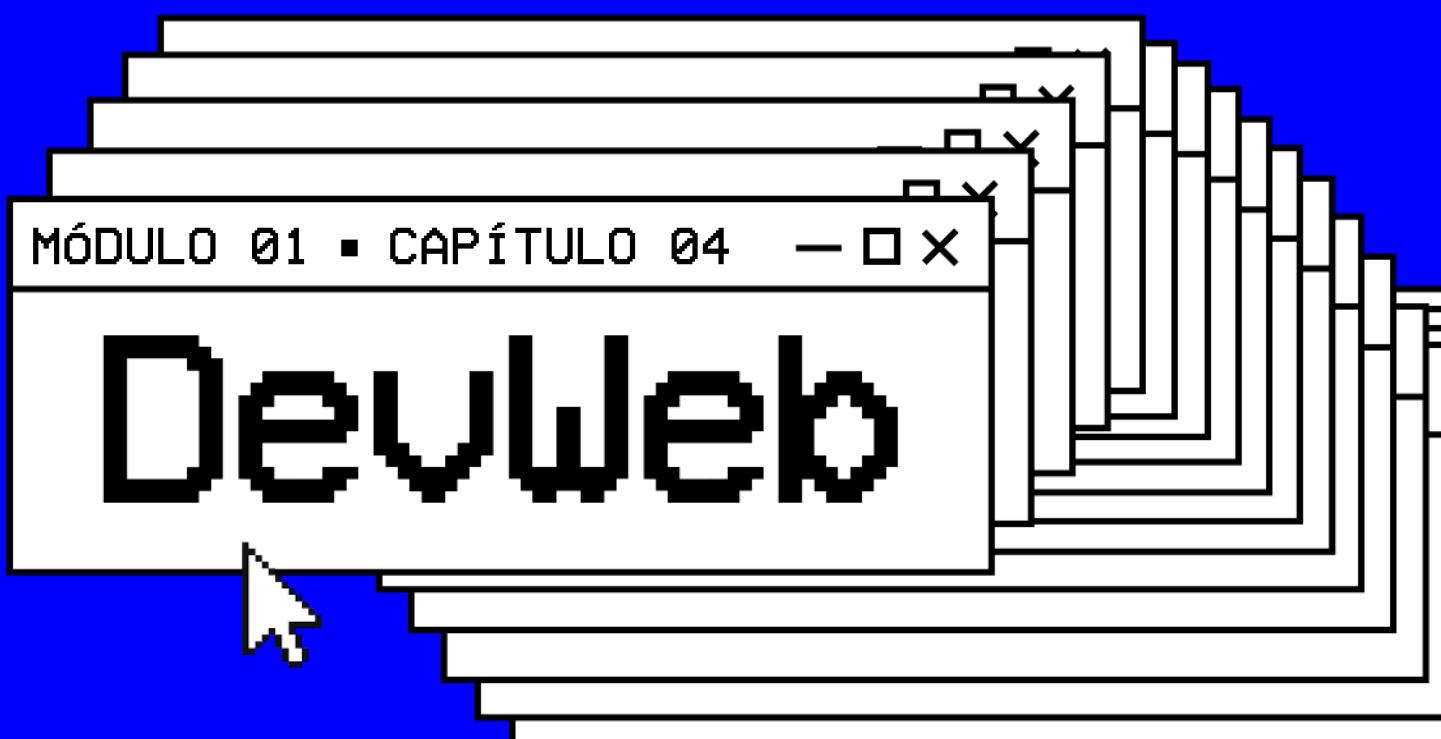
- [A] <html lang="pt-br">
- [B] <meta charset="UTF-8">
- [C] <body lang="br">
- [D] <head charset="UTF-8">

Suas anotações

Não guarde conhecimento. Ele é livre. Compartilhe o seu e veja ele se espalhando pelo mundo 

CURSO DE INFORMÁTICA PARA INTERNET

Gustavo Guanabara



PRIMEIROS PASSOS COM HTML5

M01C04

veja em vídeo



PRIMEIROS PASSOS COM HTML5

Pronto! Já vai começar a codificação que você está esperando tanto! Chegou a hora de colocar a mão na massa e criar os primeiros documentos HTML. E faremos do jeito tradicional: devagar e sempre. Ao utilizar esse método que eu prefiro aplicar às minhas aulas, veremos exercícios bem simples, com a aplicação de cada tag nova isoladamente. Tem gente que acha melhor começar com um projeto grande, mas em sala eu opto sempre por esse método. Vamos lá?



Você tem todo o direito de usar esse material para seu próprio aprendizado. Professores também podem ter acesso a todo o conteúdo e usá-lo com seus alunos. Porém todos que usarem esse material - seja para qual for a finalidade - deverão manter a referência ao material original, criado pelo **Prof. Gustavo Guanabara** e disponível no endereço do seu repositório público <https://github.com/gustavoguanabara/>. Este material não poderá ser utilizado em nenhuma hipótese para ser replicada - integral ou parcialmente - por autores/editoras para criar livros ou apostilas, com finalidade de obter ganho financeiro com ele.

Escolhendo um editor

Se você começar a frequentar grupos e fóruns de desenvolvedores, vai ver que tem muita gente perdendo um tempo precioso com discussões inúteis. Uma delas é: "qual é o melhor editor de códigos?" .

Na minha humilde opinião, o melhor editor de código é aquele que te atende! Geralmente as pessoas gostam mais do editor no qual elas começaram a desenvolver. Tem até gente que defende o uso de editores extremamente simples, como o Notepad (Windows) e o VIM (Linux). Se é o editor que você gosta e aprendeu a usar, continue nele!



Na imagem acima, coloquei alguns exemplos de editores de código bem famosos: na ordem, temos o **Visual Studio Code**, o **Atom**, o **Sublime**, o **Brackets** e o **Notepad++**. Todos são **ÓTIMOS** editores, cada um com suas vantagens e desvantagens. Nenhum deles é perfeito e todos podem travar.

Para construir o **curso de HTML5** que você encontra no YouTube, utilizei os editores **Notepad++** e o **WebStorm** da JetBrains. Naquela época eu usava muito esses dois. Hoje em dia, me adaptei 100% ao **Visual Studio Code**. Por conta disso, usaremos esse editor, que foi criado pela Microsoft, é distribuído gratuitamente e funciona no **Windows, Linux e Mac**.



INSTALE O VS CODE: Não adianta ficar apenas lendo esse material ou assistindo os vídeos. A instalação das ferramentas que usaremos no curso é **ESSENCIAL** para o seu aprendizado. Instale e use todos os dias! Crie muito!

Baixe o editor em <https://code.visualstudio.com/#alt-downloads>



**Cursos grátis de tecnologia
que te preparam para o
mercado de trabalho**

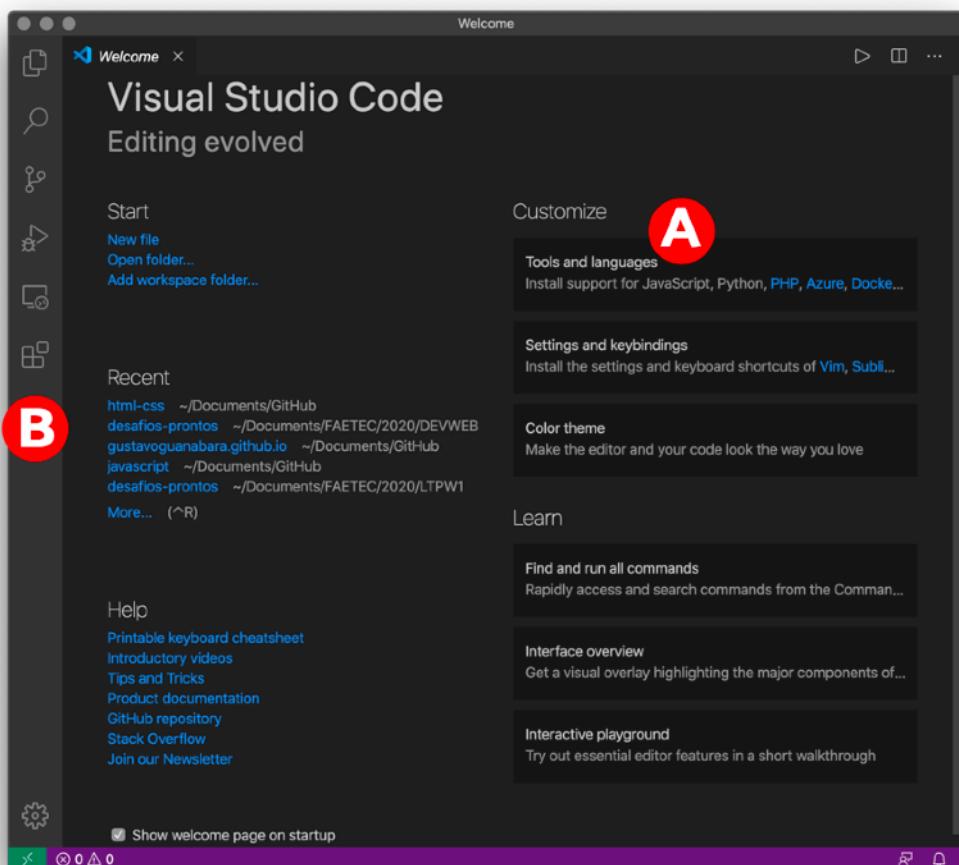
RECODE



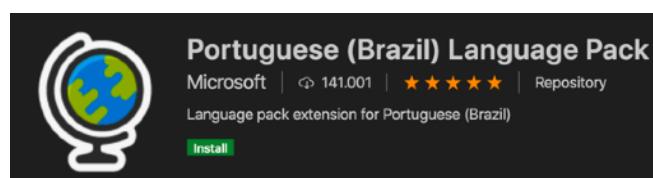
Dicas básicas para configurar seu VS Code

Depois de instalar o VS code, ele vai abrir no seu modo escuro (padrão) e recomendamos que você faça algumas configurações básicas para facilitar o seu dia-a-dia de desenvolvimento de sites.

Em primeiro lugar, vamos habilitar o suporte estendido para a linguagem JavaScript, que usaremos no nosso curso de JS. Na imagem que está a seguir, na tela de **Welcome** do seu editor, clique em “Install support for JavaScript” (**A** na imagem) na seção “Tools and languages”. Em seguida, vamos acessar a **Toolbar** (**B** na imagem) e clicar no último botão (o de baixo), que permite instalar **Extensões** (Extensions).



Na tela de extensões, procure por “Portuguese” e escolha a extensão que está representada ao lado. Clique na opção **Install** e, ao final da instalação, o VS code vai solicitar que você reinicie o editor. Pronto! Seu code estará em Português!





IMPORTANTE: Eu sei que tem gente que vai preferir manter o VS code em Inglês. Isso também é mais uma questão de opção pessoal. Nesse curso, utilizarei a versão em Português para eliminar possíveis barreiras de idioma, mas você é livre para optar pelo idioma original.

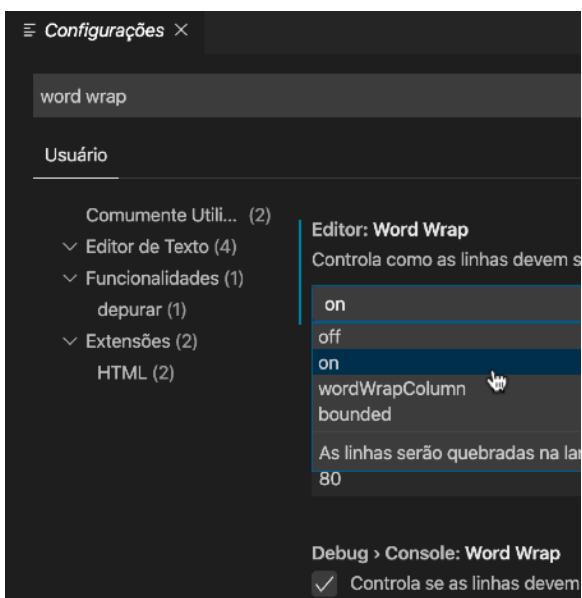
Em seguida, vamos habilitar o recurso de **salvar automaticamente** tudo aquilo que digitamos. Isso é muito importante para ganhar tempo enquanto desenvolvemos nossos códigos.

Além disso, manter essa opção sempre marcada evita problemas quando queremos ver os resultados no navegador, mas eles simplesmente não aparecem porque esquecemos de salvar.

Deixe sempre essa opção com um check marcado, exatamente como está aparecendo ao lado.



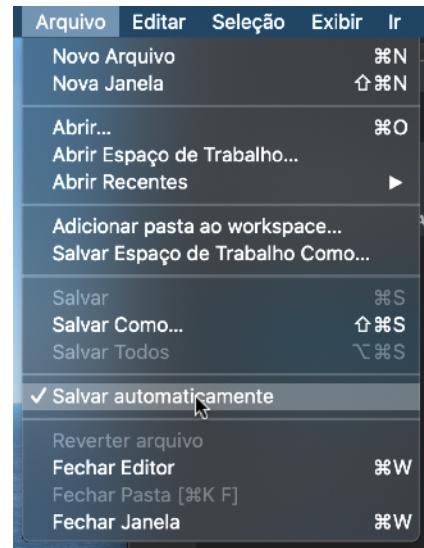
Outra coisa muito importante para deixar sempre ligado é a opção de **quebra automática de linhas** (Word Wrap). Para fazer isso, clique sobre o botão de opções que fica no canto inferior esquerdo do VS code (veja o ícone à esquerda) e escolha a opção **Configurações**.



Quando abrir a tela de configurações, digite “*word wrap*” (sem as aspas) e modifique a opção **Editor: Word Wrap**, colocando o valor **on** na caixa suspensa de seleção (veja a imagem ao lado).

Habilitar essa opção, vai melhorar a visualização do seu código, já que você não vai precisar fazer a rolagem lateral da tela em situações onde a linha vai ficar muito comprida. Pode acreditar, escreveremos algumas linhas bem longas mesmo.

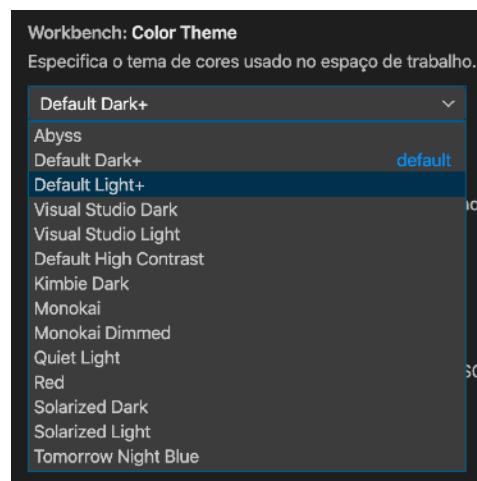
Outra configuração que você pode alterar caso ache necessário é o **Editor: Font Size**, que vai deixar aumentar ou diminuir o tamanho do código do editor.



Modo Escuro ou Claro?

Tá aí mais um motivo para discussão inútil em comunidades em redes sociais. Particularmente, prefiro o modo escuro para trabalhar no dia-a-dia. O contraste entre as letras e o fundo escuro facilitam a minha leitura, já que sou meio idoso e preciso cuidar da minha saúde visual 😊

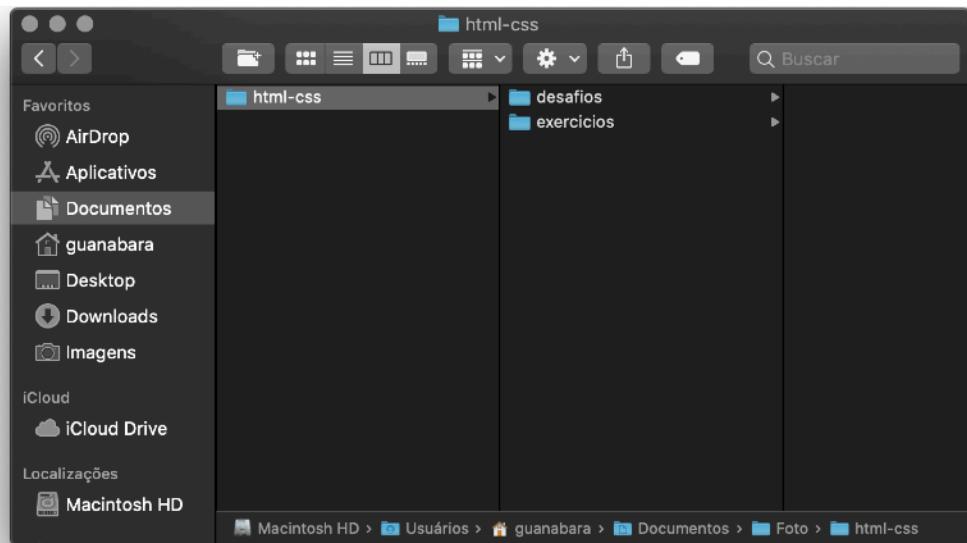
Mas como tudo aqui no curso, você tem total liberdade para modificar as configurações e deixar seu VS code da cor que quiser (até vermelho, mas não recomendo). Para isso, procure pela opção **Workbench: Color Theme** na sua área de configurações.



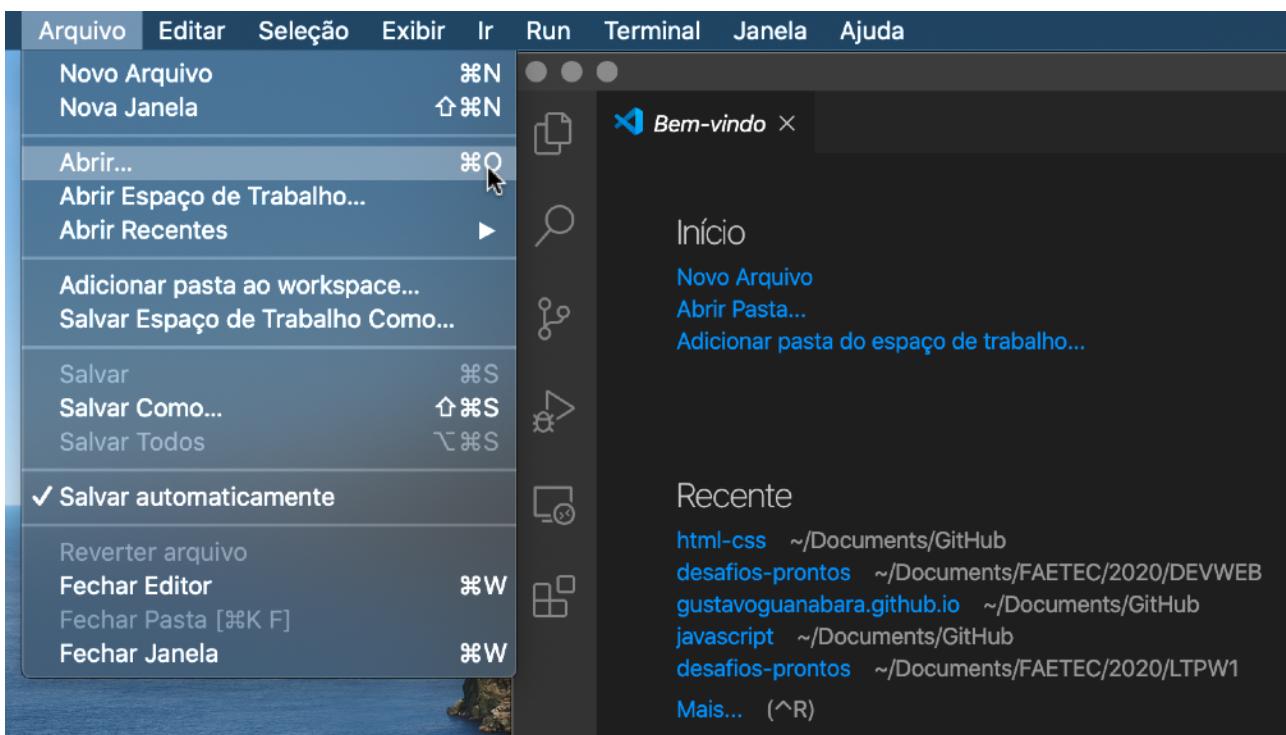
Projetos em pastas

Se eu puder te dar um conselho - e eu me sinto nesse direito - ele conselho seria: **ORGANIZE-SE!** Quem começa a desenvolver coisas e joga tudo na **Área de Trabalho** do Windows ou dentro da pasta **Meus Documentos** do seu computador, em pouco tempo fica simplesmente perdido(a).

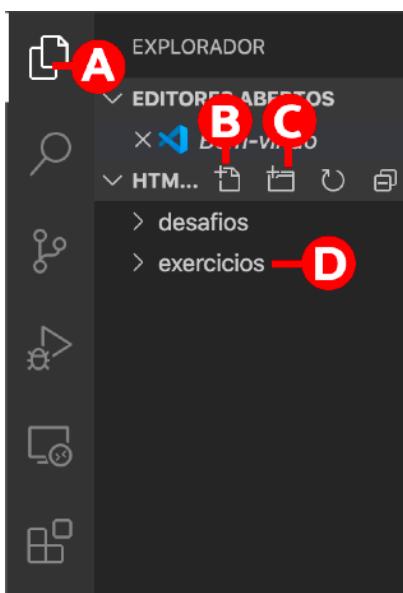
Para isso, vou te dar uma dica valiosa. Vá até a sua pasta de Documentos e crie uma pasta chamada **html-css**. Essa será a sua pasta principal do seu curso. Dentro dessa pasta, crie duas pastas: **desafios** e **exercícios** (note que eu não usei acentuações, nem espaços, nem letras maiúsculas).



Agora vamos abrir o VS code e abrir essa pasta que acabamos de criar. Se você ainda está com a tela de Bem-vindo (Welcome) aberta, basta clicar no link **Abrir Pasta...** que está na seção **Início** ao lado esquerdo (veja na foto) e escolher a pasta **html-css** que criamos anteriormente.



Além disso, você também pode clicar no menu do editor e escolher a opção **Abrir** (tecla de atalho Ctrl+O no teclado) e escolher a pasta.



Depois que você abrir a pasta, ela vai aparecer no lado esquerdo da tela, no **Explorador** (A). Note que as duas pastas que criamos, **desafios** e **exercícios** também está aparecendo.

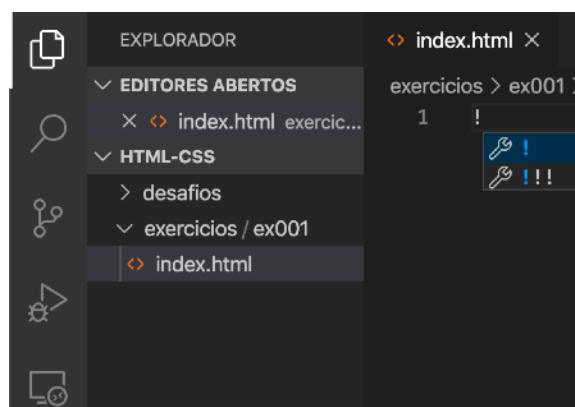
Vamos criar nosso primeiro exercício de maneira organizada. Clique primeiro no nome da pasta (D) para abri-la. Em seguida clique no botão para criar uma nova pasta (C) e dê o nome **ex001** para a pasta criada.

Depois clique na pasta que você acabou de criar e clique no botão para criar um novo arquivo (B) e dê o nome de **index.html** para o arquivo criado.

IMPORTANTE:

Não se esqueça de usar apenas

letras minúsculas, não usar acentos e não colocar espaços.



Feitos todos esses passos, a sua tela deve se parecer com a imagem à direita.

Note que o arquivo que foi criado já aparece aberto na área principal do editor, e o arquivo

com extensão **.html** vai aparecer com um pequeno ícone <>, indicando que foi criado no formato correto.

Para criar o código base HTML que vimos no capítulo anterior, vá até as linhas do arquivo **index.html** e digite apenas um ponto de exclamação ! e em seguida pressione a tecla **Enter** no seu teclado.

Viu só o resultado? O código foi criado automaticamente e precisa apenas de algumas pequenas atualizações. Complete seu código, adicionando as linhas para deixar o código igual ao que está representado abaixo.

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport"
6          content="width=device-width, initial-scale=1.0">
7      <title>Primeiro site</title>
8  </head>
9  <body>
10     <h1>Olá, Mundo!</h1>
11     <hr>
12     <p>Este foi o primeiro site que eu criei, com um
13         título, uma linha e um parágrafo.</p>
14 </body>
15 </html>
```

A **linha 2** foi alterada, indicando que nossa página terá seu conteúdo em Português Brasileiro. A **linha 6** também, para que nosso título seja *Primeiro site*.

Na **linha 9**, criamos um **título**. Nesse momento, vou te ensinar uma dica. No lugar de digitar `<h1></h1>` manualmente, digite apenas `h1`, sem os colchetes angulares e pressione Enter (veja a imagem). Automaticamente a tag será aberta e fechada. Isso facilita muito o nosso dia-a-dia também.

`<body>`

`h1`

`</body>` ↵ `h1`

`</html>`

Na **linha 10** criamos uma **linha horizontal** (horizontal row). Perceba que não existe fechamento para essa tag. Isso é normal e nós já tínhamos previsto isso no material anterior.

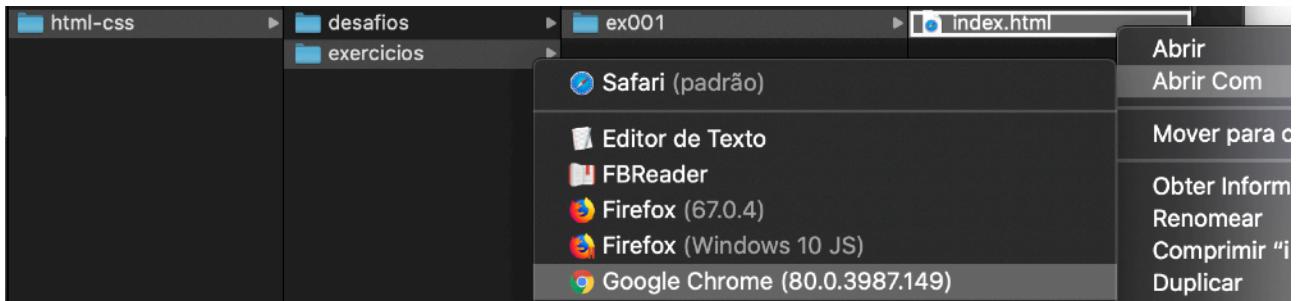


Na **linha 11** estamos criando um **parágrafo** com um texto de conteúdo. Note que na foto, a linha ocupa duas linhas, mas a contagem se mantém no 11. Isso acontece porque eu habilitei o **Word Wrap** na página 4 desse material.

Se você também habilitou a opção **Salvar automaticamente** (página 4) no seu VS code, tudo aquilo que você digitou já está gravado na sua pasta.

O próximo passo é ver se está tudo funcionando corretamente.

Chegou a hora de testar tudo



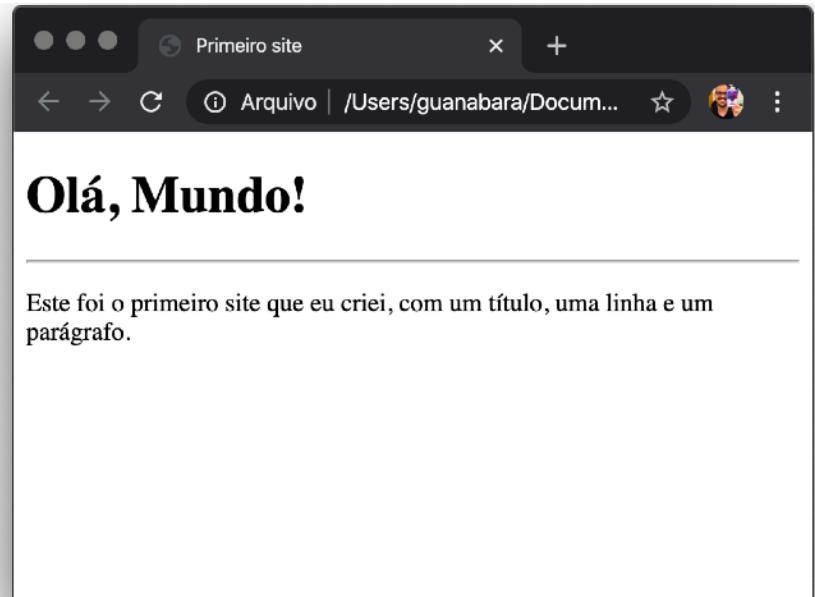
Se você se organizou direitinho e salvou o arquivo no local certo, vai poder abrir no seu explorador de arquivos a pasta Documentos/html-css/exercicios/ex001 e lá dentro vai encontrar o arquivo index.html. Clique com o botão direito sobre esse arquivo html e escolher a opção para **Abrir no Google Chrome**.

O resultado visual deverá ser semelhante ao apresentado ao lado.

Perceba na parte de cima do navegador, lá na aba do documento, aparecerá o termo **Primeiro site**. Isso foi o efeito da configuração do <title>.

Note que o uso da tag <h1> fez o seu conteúdo virar um título.

Já a tag <hr> fez aparecer uma linha horizontal, como já tínhamos previsto. Logo abaixo está o parágrafo.



Quer acompanhar tudo em vídeo?

Eu sei que às vezes as pessoas gostam mais de assistir vídeos do que ler livros, e é por isso que eu lanço há anos materiais no canal Curso em Vídeo no YouTube. O link que vou compartilhar contigo faz parte da playlist completa onde você encontra o **Módulo 1 do Curso de HTML5 e CSS3**, completamente gravado com base nesse material.



Além de acessar o link a seguir, você também pode ter acesso às aulas apontando a câmera do seu celular para o código QR ao lado. Todo dispositivo smartphone ou tablet atualizado já possui esse recurso de leitura de códigos habilitado por padrão.

Módulo 1 do curso: https://www.youtube.com/playlist?list=PLHz_AreHm4dkZ9-atkcmcBaMZdmLHft8n

Teste seus conhecimentos

Terminou de ler esse capítulo e já acompanhou todos os vídeos e referências externas que indicamos? Pois agora, responda a essas 10 perguntas objetivas e marque em cada uma delas a única opção verdadeira. Aí sim, você vai poder comprovar que realmente entendeu o conteúdo.



1. Qual dos programas a seguir é o único que não pode ser utilizado para escrever códigos em texto sem formatação para HTML + CSS + JS?

- A Atom
- B VIM
- C Brackets
- D Word

2. Qual é a empresa que fabrica o VSCode, o editor que escolhemos para escrever códigos para nosso curso?

- A Microsoft
- B Apple
- C Adobe
- D Google

Cursos que vão te levar ao próximo nível

 estudonauta



3. É possível usar o VSCode em idioma Português do Brasil?

- [A] Sim. O editor já é instalado em PT-BR como padrão
- [B] Sim. Porém, precisamos escolher o idioma durante a instalação
- [C] Sim. Mas não existe suporte nativo para PT-BR, mas existe uma extensão para isso.
- [D] Não. O VSCode ainda não funciona em PT-BR.

4. Qual é a opção que devemos habilitar nas configurações para que as instruções muito longas sejam quebradas em várias linhas automaticamente?

- [A] Wrap Line
- [B] Multi Line
- [C] Auto Break
- [D] Word Wrap

5. Qual é o caminho correto que devemos seguir para ligar o salvamento automático de arquivos no VSCode?

- [A] Arquivo > Salvar Automaticamente
- [B] Terminal > Salvar Automaticamente
- [C] Configurações > Salvar Automaticamente
- [D] Editar > Salvar Automaticamente

6. Qual é o nome do arquivo principal de um site feito em HTML, que será aberto assim que o visitante solicitar a abertura do site?

- [A] first.html
- [B] www.html
- [C] index.html
- [D] 001.html

7. Ao criar um arquivo HTML, o VSCode tem um atalho simples para criar o código base para um arquivo desse tipo. Que atalho é esse?

- [A] ! + Enter
- [B] # + Enter
- [C] ^ + Enter
- [D] ? + Enter

8. Qual é a tag HTML para criar uma linha horizontal no corpo do navegador?

- [A] <line>
- [B] <row>
- [C] <!-->
- [D] <hr>

9. Para criar um parágrafo em HTML, usamos a tag:

- [A] <p>
- [B] <para>
- [C] <paragraph>
- [D] <h1>

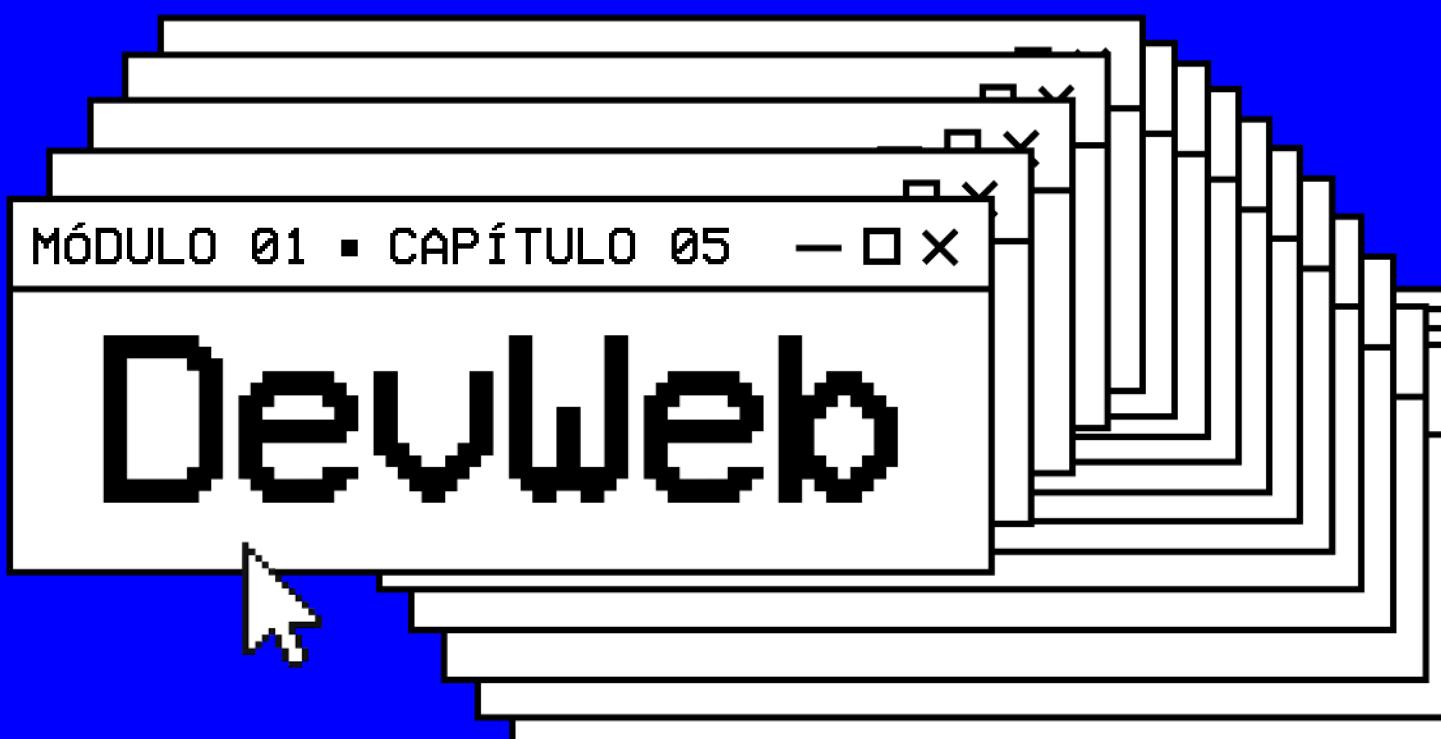
10. A tag <title> é adicionada dentro de qual área de um documento HTML?

- [A] <h1>
- [B] <p>
- [C] <head>
- [D] <body>

Suas anotações

Não guarde conhecimento. Ele é livre. Compartilhe o seu e veja ele se espalhando pelo mundo 

Gustavo Guanabara



CARACTERES, PARÁGRAFOS E QUEBRAS DE LINHA

M01C05

veja em vídeo



CARACTERES, PARÁGRAFOS E QUEBRAS DE LINHA

Eu sei que sites podem ter imagens, vídeos, animações e tudo mais. Mas o foco principal sempre é no texto em si. Palavras são valiosas. Sem palavras você não estaria nem lendo essa página. Então vamos dar o devido valor ao conteúdo escrito e ver as marcações importantes para os textos. Me acompanha?



Você tem todo o direito de usar esse material para seu próprio aprendizado. Professores também podem ter acesso a todo o conteúdo e usá-lo com seus alunos. Porém todos que usarem esse material - seja para qual for a finalidade - deverão manter a referência ao material original, criado pelo **Prof. Gustavo Guanabara** e disponível no endereço do seu repositório público <https://github.com/gustavoguanabara/>. Este material não poderá ser utilizado em nenhuma hipótese para ser replicada - integral ou parcialmente - por autores/editoras para criar livros ou apostilas, com finalidade de obter ganho financeiro com ele.

Vamos começar com os parágrafos

No capítulo anterior já vimos como criar um parágrafo de forma simples, foi só colocar todo o conteúdo dentro da tag `<p></p>`. Mas o que eu não comentei na ocasião foi o comportamento do navegador em relação à posição das palavras.

Crie aí na sua casa uma pasta **ex002** dentro da sua pasta que criamos em Documentos/html-css/exercícios para deixar tudo organizando. Lá dentro, crie um documento **index.html** e crie o código base do HTML5 (já ensinei isso anteriormente).

Dentro da tag `<body></body>` crie um parágrafo com a seguinte característica:

```
<p>Você pode  
    escrever um  
    parágrafo  
    de qualquer jeito.  
    Basta colocar tudo  
    no meio do par de tags &lt;p&gt; e &lt;/p&gt;  
</p>  
  
<p>Se precisar quebrar  
    o texto em algum lugar  
    específico <br> como esse,  
    você pode usar a tag &lt;br&gt;  
</p>
```

O que nós criamos no código aí acima foram dois parágrafos. Note que eu criei várias quebras de linha dentro de cada parágrafo. Salve o seu projeto e veja o resultado.

Na verdade, todas aquelas quebras que causamos não servem de **ABSOLUTAMENTE NADA!** Isso acontece porque quem comanda quebras de linhas/parágrafos não é o fato de apertar **Enter**. Quem manda são as tags HTML. A única quebra que **REALMENTE** aconteceu foi após a palavra "específico" no segundo parágrafo. Isso porque colocamos a tag `
` nesse local.

A tag `
` significa literalmente "quebre a linha" (*break row*) e agora que você sabe o significado, não precisa nem de explicação né?



IMPORTANTE: Uma das **GAMBIARRAS** que muitos iniciantes em HTML fazem constantemente é criar espaços verticais maiores usando `

` várias vezes. Ai vai um grande conselho: **NÃO FAÇA ISSO!** Mais tarde vamos aprender como configurar o espaçamento vertical através das folhas de estilo (CSS). Quem cria espaço com vários `
` é NOOB! Não faça!

Outra coisa que você deve ter notado é que tivemos problemas para fazer uma tag aparecer no seu navegador. Isso acontece porque, se você colocar `
` no seu código, o navegador vai entender como “quebre a linha” e não pra fazer ela aparecer.

Para isso, no lugar de usar os colchetes angulares `<>`, usamos caracteres especiais referentes ao código desses símbolos. No código anterior, quando usamos:

- Um `<` estamos pedindo pra colocar o símbolo menor que (less than).
- Um `>` estamos pedindo para coloca o símbolo maior que (greater than).

Existem alguns outros símbolos que podem ser exibidos usando códigos. Esses códigos são chamados de **HTML Entities**, basta uma breve consulta para descobrir quais são as opções, mas vou te dar uma moral e colocar uma tabela com outras opções.

Símbolo	Descrição	Entitity
	<i>Espaço em branco</i>	&nbsp
®	<i>Marca registrada</i>	®
©	<i>Copyright</i>	©
™	<i>Trade Mark</i>	™
€	<i>Euro</i>	€
£	<i>Libra (pound)</i>	£
¥	<i>Yen</i>	¥
¢	<i>Cent</i>	¢
Ø	<i>Vazio</i>	∅
Σ	<i>Soma</i>	∑
Δ	<i>Delta</i>	Δ
←	<i>Seta esquerda</i>	←
↑	<i>Seta acima</i>	↑
→	<i>Seta direita</i>	→
↓	<i>Seta baixo</i>	↓

The screenshot shows a web browser window with a purple advertisement. The ad has the following text:
Soluções digitais
para negócios
hostnet

On the right side of the ad is a QR code.



DICA: Você quer uma referência com vários símbolos e seus códigos, pois acesse o W3Schools, na área de Misc Symbols.

https://www.w3schools.com/charsets/ref_utf_symbols.asp

Inserindo Emojis

Outra coisa que você pode adicionar às suas páginas são os emojis, onde cada símbolo possui um código Unicode. Para uma pesquisa mais precisa e atualizada, acesse o site da emojipedia.org.

The screenshot shows a web browser displaying the Emojipedia.org website. The page title is "Smileys & People". Below the title, it says "Emojis for smileys, people, families, hand gestures, clothing and accessories." A list of emoji names and their corresponding code points is provided:

- Grinning Face
- Grinning Face with Big Eyes
- Grinning Face with Smiling Eyes
- Beaming Face with Smiling Eyes
- Grinning Squinting Face
- Grinning Face with Sweat
- Rolling on the Floor Laughing
- Face with Tears of Joy
- Slightly Smiling Face
- Upside-Down Face

The URL in the address bar is <https://emojipedia.org/grinning-face-with-smiling-eyes/>.

The screenshot shows a web browser displaying the Emojipedia.org website. The page title is "Smiling Face with Open Mouth". Below the title, it lists "Codepoints" and "Shortcodes" for the emoji, followed by a "See also" section. The "Codepoints" section shows the emoji with its code point: 😊 U+1F604. The "Shortcodes" section shows the emoji followed by the shortcode ":smile:". The "See also" section lists other related emojis:

- Cowboy Hat Face
- Grinning Cat with Smiling Eyes
- Grinning Face
- Beaming Face with Smiling Eyes
- Star-Struck
- Hugging Face

The URL in the address bar is <https://emojipedia.org/emoji/😊/>.

A purple rectangular advertisement. On the left, white text reads: "Cursos grátis de tecnologia que te preparam para o mercado de trabalho". To the right is the word "RECODE" in large white letters. Further right is a QR code.

Escolha uma categoria, clique no emoji escolhido e procure o seu **Codepoints**. No caso do exemplo acima, escolhemos o emoji com o código **U+1F604**. Para inserir esse emoji no seu site, use o código adaptado **😄** em qualquer lugar que aceite palavras. É só substituir o **U+** por **&#x** (com a letra x minúscula).

E começam os desafios!

Lá no repositório, além do material em PDF e dos códigos dos exercícios 100% disponíveis, também disponibilizamos alguns **desafios** que devem ser resolvidos. Esses desafios não incluem o código original e você deve tentar chegar à resposta sem copiar nenhum código.

Com todo o conteúdo que vimos até essa aula, você já pode resolver o **desafio d001**. Acesse o repositório público, abra a área do curso de HTML+CSS e clique no link de acesso aos desafios. Manda ver! Só não fica pedindo a resposta! Você consegue resolver isso sozinho(a)!

Repositório em: <https://gustavoguanabara.github.io>



Quer acompanhar tudo em vídeo?

Eu sei que às vezes as pessoas gostam mais de assistir vídeos do que ler livros, e é por isso que eu lanço há anos materiais no canal Curso em Vídeo no YouTube. O link que vou compartilhar contigo faz parte da playlist completa onde você encontra o **Módulo 1 do Curso de HTML5 e CSS3**, completamente gravado com base nesse material.



Além de acessar o link a seguir, você também pode ter acesso às aulas apontando a câmera do seu celular para o código QR ao lado. Todo dispositivo smartphone ou tablet atualizado já possui esse recurso de leitura de códigos habilitado por padrão.

Módulo 1 do curso: https://www.youtube.com/playlist?list=PLHz_AreHm4dkZ9-atkcmcBaMZdmLHft8n

Teste seus conhecimentos

Terminou de ler esse capítulo e já acompanhou todos os vídeos e referências externas que indicamos? Pois agora, responda a essas 10 perguntas objetivas e marque em cada uma delas a única opção verdadeira. Aí sim, você vai poder comprovar que realmente entendeu o conteúdo.



1. Os símbolos de < e > são usados em HTML para delimitar tags. Mas se quisermos mostrá-los no navegador, precisamos representá-los como:

- A < e >
- B #lt; e #gt;
- C &menor; e &maior;
- D #<; e #>;

2. Dentro de um parágrafo no código HTML, se escrevermos uma palavra em cada linha, elas serão apresentadas no navegador de que maneira?

- A uma em cima da outra
- B uma ao lado da outra
- C exatamente da maneira que digitamos
- D nenhuma palavra será exibida

3. A qualquer momento do código HTML, se quisermos criar uma quebra de linha, devemos usar qual tag?

- A <row>
- B <new>
- C

- D Não precisa usar tag, é só apertar ENTER

4. A melhor estratégia para criar um espaço relativo a dez quebras de linha em nosso site é:

- A usar a tag de quebra por 10 vezes consecutivas
- B usar a tag de parágrafo por 5 vezes consecutivas
- C usar configurações de estilo em CSS e especificar o espaçamento todo de uma vez
- D não existe a possibilidade de criar um espaçamento desse tamanho

5. Os códigos como € para mostrar o símbolo € nas páginas são conhecidos pelo nome de:

- A HTML entities
- B Symbol codes
- C HTML symbols
- D Symbol entities

6. Que código utilizaremos para mostrar uma seta apontando para a direita (→) na página?

- A &right;
- B &rarrow;
- C &arrow;
- D →

Cursos que vão te levar
ao próximo nível



7. Qual site é recomendado nas aulas para buscar os códigos usados para representar os emojis?

- A WikiMoji
- B EmojiBook
- C Emojipedia
- D Symbolpedia

8. Se o código de um emoji é U+1F913, como devemos representá-lo em nosso código HTML?

- A U+1F913;
- B &#X+1F913;
- C &#xU1F913;
- D 🤓

9. A marca br significa:

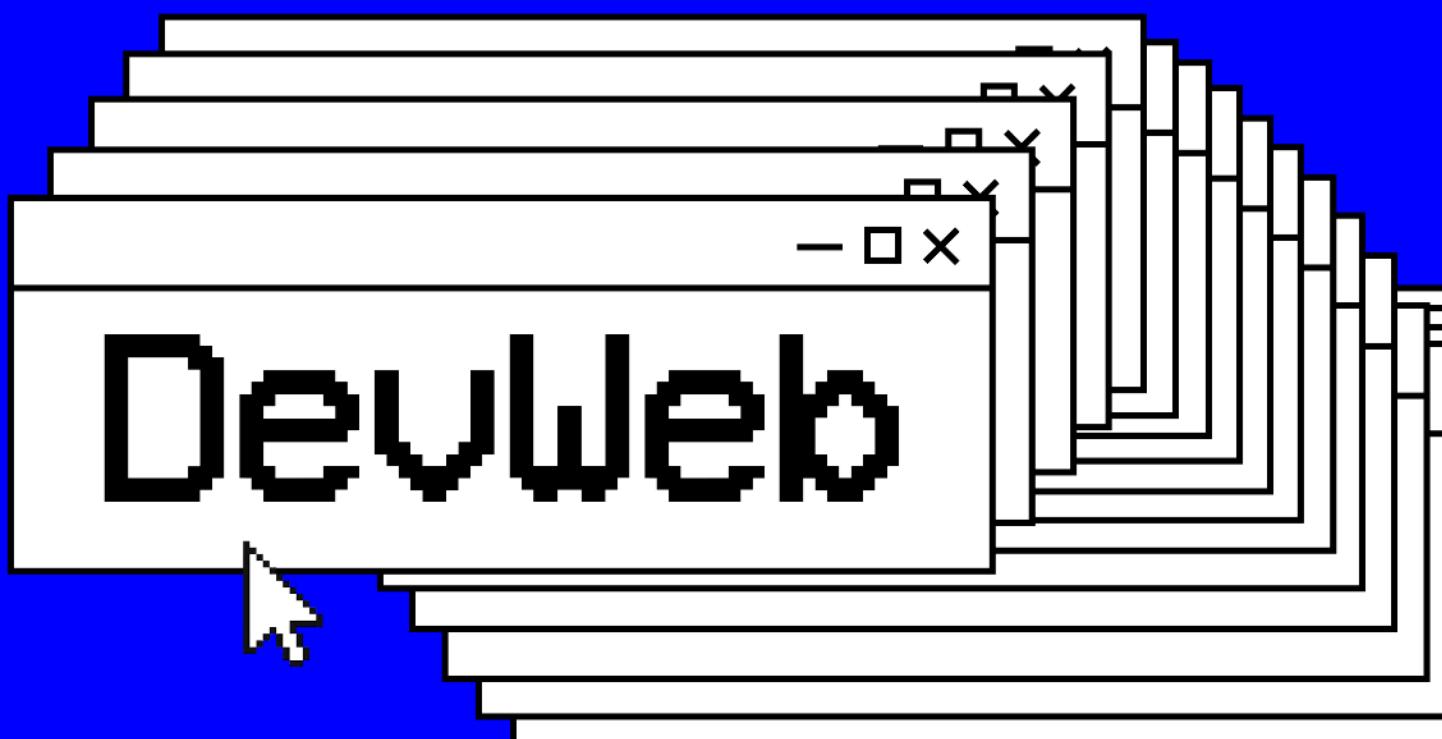
- A Que o idioma está em Português do Brasil
- B Break row, ou quebra de linha
- C Break right, ou quebra à direita
- D Broked row, ou linha quebrada

10. Qual é o TLD do domínio do site que indicamos para buscar emojis?

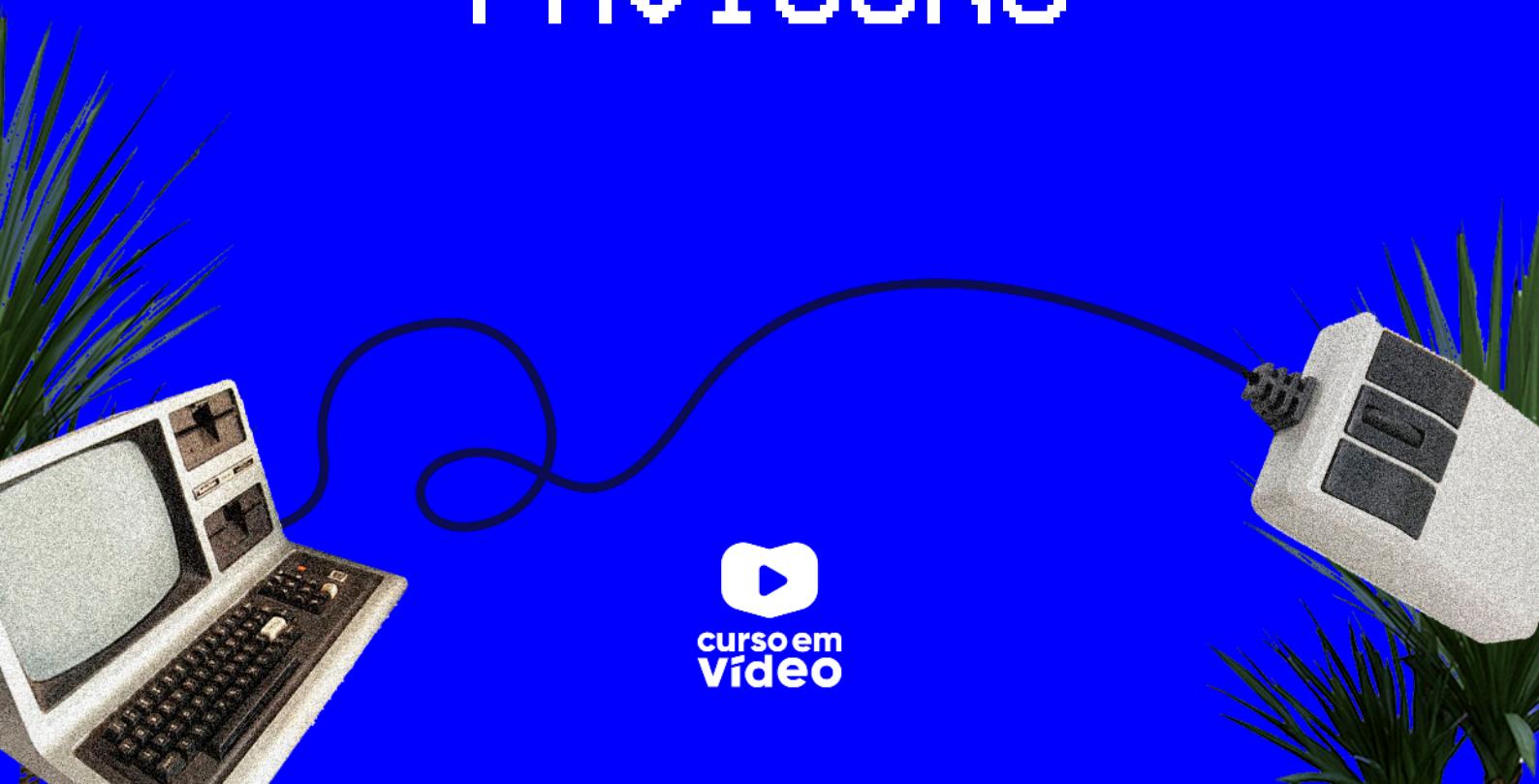
- A .org
- B .gov
- C .com
- D .edu

Suas anotações

Não guarde conhecimento. Ele é livre. Compartilhe o seu e veja ele se espalhando pelo mundo 



IMAGENS E FAVICONS



M01C06

IMAGENS E FAVICONS

veja em vídeo



Sabe aquela frase clássica que diz que “uma imagem vale mais do que mil palavras”? Pois é exatamente assim com a maioria dos sites. Tirando algumas pequenas exceções onde o público acessa um site apenas para “ler” seu conteúdo, as imagens são essenciais para a maioria dos casos. São elas que chamam a atenção e devem ser escolhidas com muita dedicação. Vamos nessa?

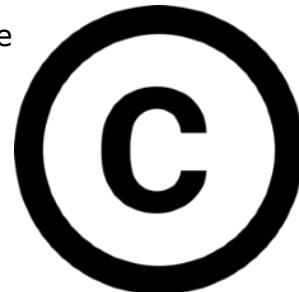


Você tem todo o direito de usar esse material para seu próprio aprendizado. Professores também podem ter acesso a todo o conteúdo e usá-lo com seus alunos. Porém todos que usarem esse material - seja para qual for a finalidade - deverão manter a referência ao material original, criado pelo **Prof. Gustavo Guanabara** e disponível no endereço do seu repositório público <https://github.com/gustavoguanabara/>. Este material não poderá ser utilizado em nenhuma hipótese para ser replicada - integral ou parcialmente - por autores/editoras para criar livros ou apostilas, com finalidade de obter ganho financeiro com ele.

Para começar, vamos falar de direitos autorais

Sabe aquela foto que você pegou lá no **Google Imagens** e se encaixou perfeitamente no seu site? Pois é bastante provável que você simplesmente **NÃO POSSA USÁ-LA** e pode até mesmo ser processado(a) por isso.

" - *Mas como assim, Guanabara!?*" - você pode estar gritando, mesmo que apenas mentalmente, nesse exato momento.



É exatamente o que eu falei, pequeno Gafanhoto! Na verdade não só as imagens, todo conteúdo disponível na Internet pode estar protegido por direitos autorais. Livros em PDF, textos em blogs, matérias em jornais online, fotos, animações, vídeos. Tudo pode ser protegido e você nem fazia ideia disso.



ANTES CONTINUAR: Eu não sou profissional do Direito, então faço uma pausa aqui no material para te indicar um vídeo onde temos o Especialista em Direito Digital **Alan Moreira Lopes** falando sobre esse assunto com muito mais propriedade. A partir de 29min ele responde umas perguntas bem comuns.

Canal Revolution: <https://youtu.be/Bkym20Gq0oQ?t=171>

E se nesse momento você está pensando: "*mas quem vai processar um jovem aluno que está aprendendo a criar sites usando imagens do Google?*".

Minha maior preocupação aqui não é te **PROIBIR** de pegar imagens no Google Imagens e usar nos seus sites enquanto você está aprendendo HTML. O meu medo é você achar que esse ato é **100% LEGAL** e comece a criar sites profissionalmente para empresas usando imagens e conteúdos que você não pode.



OLHA A TRETA: Um caso que ficou muito famoso na Internet foi com o canal **Nostalgia**. O Felipe Castanhari, dono do canal, vive recebendo avisos de direitos autorais por usar pedaços de vídeos de documentários, filmes, clipes e séries com direito autoral.

Canal Nostalgia: <https://youtu.be/SJacdAbjdZI>

Concordando ou não com isso, o fato é que a lei existe. E ir contra a lei é ficar totalmente vulnerável aos efeitos causados por ela. Eu até sugiro que façamos uma breve discussão sobre isso em sala de aula. Se eu não for seu professor, bata um papo com ele sobre o assunto. Também sugiro que você converse sobre isso com algum advogado especializado nesse ramo digital caso fique alguma dúvida (toda família tem um tio ou vizinho advogado).

Sendo assim, para que você seja profissionalmente capaz de criar sites com imagens que não vão gerar processos, existem quatro maneiras simples:

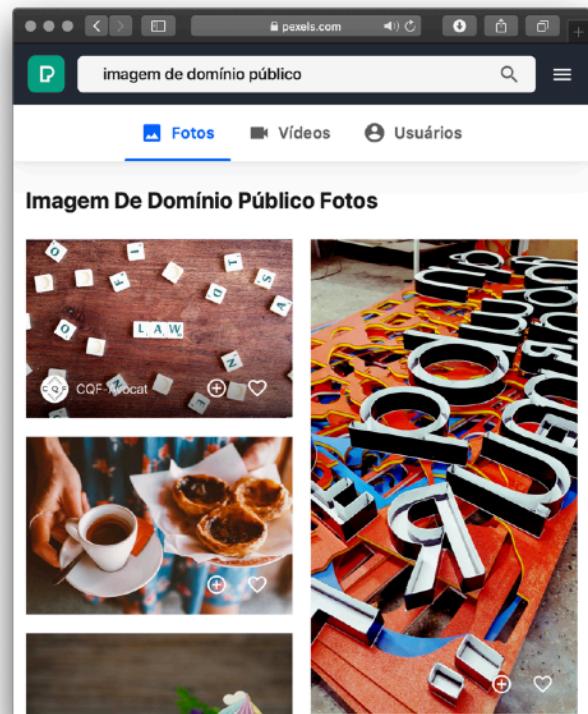
1. Seja o criador da própria arte. Use fotos que seu próprio cliente tirou ou que você mesmo criou.
2. Consiga uma autorização legal de quem é o detentor dos direitos autorais. Isso precisa de documentos específicos para tal.
3. Compre o direito de usar a imagem. Existem vários sites como o **iStockPhoto**, **DepositPhotos**, **ShutterStock**, etc. Só fica de olho aberto e leia as regras, pois na maioria dos casos essa imagem que você comprou só pode ser usada uma única vez ou em um único projeto.
4. Use imagens de domínio público, pois essas você tem a liberdade de colocar no seu site, seguindo as regras estabelecidas pelo autor.

E onde encontro imagens de domínio público?

Existem vários sites que oferecem imagens de domínio público para serem usados em seus projetos. Mas a dica que te dei antes continua valendo: leia atentamente as instruções para uso dessas imagens. Algumas delas deixam você usar a arte apenas se o artista for creditado em um texto localizado no seu site.

Alguns dos sites que eu mais gosto de visitar quando quero usar imagens com domínio público são:

- [UnSplash](#)
- [Pexels](#)
- [FreePik](#)
- [Rawpixel](#)
- [Pixabay](#)
- [Libreshot](#)
- [Wikimedia Commons](#)



An advertisement for Estudonauta. The background is dark blue with white text. The main text reads "Cursos que vão te levar ao próximo nível". Below this, the Estudonauta logo is shown, which consists of a stylized orange and white icon followed by the word "estudonauta". To the right of the text is a large QR code.

Mas muito **CUIDADO!** Alguns desses sites que disponibilizam imagens de domínio público possuem propagandas de serviços de venda de imagens e acabam te confundindo um pouco.

Vamos mudar de assunto, já que possivelmente você não está se preocupando muito com esse papo de direitos autorais, mesmo ele sendo muito importante para quando você virar um profissional de verdade.

Vamos falar sobre formatos de Imagem

Existem vários formatos de imagem, cada um com suas características, vantagens e desvantagens. Porém, vamos nos focar aqui nos dois formatos compactados mais usados para a criação de sites: **JPEG** e **PNG**.

O algoritmo de compactação **JPEG** é usado para gerar imagens fotográficas com um tamanho extremamente reduzido. Ele foi criado em 1983 por **Eric Hamilton** e hoje é gerenciado pelo **Joint Photographic Experts Group**. Ele é amplamente utilizado por câmeras digitais modernas e programas de tratamento de imagens.

A grande vantagem do uso de arquivos **JPG** (em formato JPEG) é gerar arquivos muito pequenos e que ocupam pouco espaço em disco. Isso é muito importante, pois quando colocarmos nosso site no ar, ele tem que ser leve e carregar as imagens muito rapidamente. Só toma cuidado para não exagerar na hora de configurar o nível de compactação. Isso pode fazer com que sua imagem fique horrível e toda borrada (dá só uma olhada na imagem abaixo).

O formato **Portable Network Graphics** (PNG) surgiu em 1996, desenvolvido pelo **W3C** (o mesmo órgão que gerencia a linguagem HTML) com o objetivo de substituir o formato **GIF** (que hoje voltou a ser popular graças ao WhatsApp e Instagram). Ele também é um formato compactado, mas não tanto quanto o JPEG.

A principal característica do **PNG** - e que o diferencia do JPEG - é a capacidade de configurar a opacidade de cada pixel (deixá-lo transparente ou com transparência limitada).

Mas para entendermos melhor a explicação acima, vamos a um exemplo visual (já que estamos falando de imagens né?)



As duas primeiras imagens estão compactadas no formato JPEG, mas na primeira eu coloquei o nível de compactação em qualidade 5% e ficou com 20KB e na segunda uma qualidade de 30% e ficou com 120KB. A última imagem está comprimida no formato PNG, ficou com 300KB e teve o contorno preservado por causa do fundo transparente.

De forma resumida, na hora de escolher o formato de imagem para o seu site, opte sempre pelo formato JPEG com uma compactação entre 30% e 50%. O formato PNG só deverá ser usado quando precisarmos de transparência na foto. Combinado?

Faltou falar sobre tamanho das imagens

Quando você está começando a desenvolver sites, acaba pensando que para obter o melhor resultado, sua imagem deve ter um tamanho **GIGANTE** para poder ter a maior qualidade.

Realmente, uma imagem com resolução grande (1920x1080) tem mais pontos que uma imagem com resolução pequena (500x280). Só que uma imagem 1920x1080 pode gerar um arquivo de até 3MB, enquanto uma 500x280 dificilmente vai passar de 500KB.



A regra de outro nesses casos é: **use imagens do tamanho certo!** Vai precisar de uma imagem que vai ter 200 pixels de largura? Gere um arquivo exatamente com esse tamanho! Nada de ficar salvando arquivos gigantes e diminuindo o tamanho da imagem com códigos.

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 tempão pra carregar o arquivo do seu servidor e vai exibir a imagem minúscula na tela.

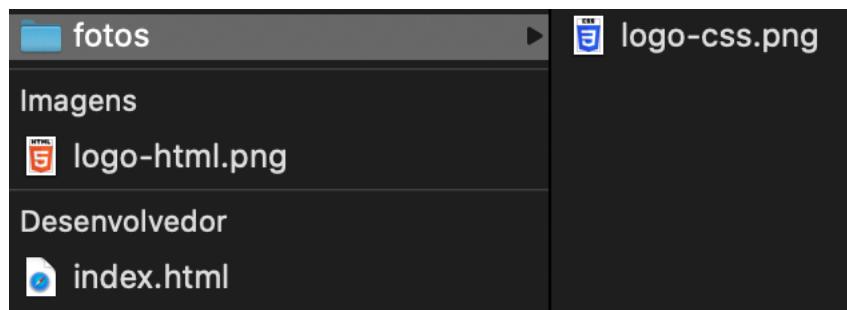
E no que isso te prejudica? Os mecanismos de busca como o Google **PENALIZAM** sites lentos e pesados, retirando-os da primeira página de buscas. Quem aqui quer criar um site que não aparece nos resultados do Google?

A purple rectangular banner with white text and a QR code. At the top, there are window control icons (-, □, X). The main text on the left reads "Cursos grátis de tecnologia que te preparam para o mercado de trabalho". Below that, the word "RECODE" is written in a bold, sans-serif font. To the right of the text is a large QR code.

Como carregar uma imagem em HTML

Agora que você está especialista em formatos e tamanhos de imagens, vamos gravar algum arquivo em uma pasta chamada **ex003** que você vai criar no seu repositório local. Se você visitar meu repositório em <http://gustavoguanabara.github.io/> e for até a parte de exercícios de HTML, vai ver que na pasta ex003 existem os arquivos logo-html.png e logo-css.png. Faça o download deles pra você ou salve outra imagem qualquer na sua pasta. Só fica de olho no formato dela: JPG ou PNG. Use o VScode para criar também o arquivo index.html e coloque dentro da pasta **ex003**, conforme vou apresentar a seguir.

Eu só te faço um alerta: analise a foto a seguir, onde eu mostro que o arquivo logo-html.png estará na pasta raiz do projeto (mesmo local onde está o arquivo index.html), enquanto o arquivo logo-css.png está localizado dentro da pasta fotos que está na mesma pasta do projeto.



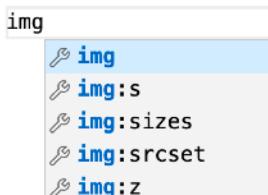
Dentro da pasta ex003, crie os arquivos acima

A posição dos seus arquivos vai fazer diferença e você deverá adaptar seu código para poder carregar as imagens sem problemas. Agora vá até o Visual Studio Code e edite o arquivo index.html, criando o código base de sempre e atualizando conforme o que apresento a seguir:

```
8 <body>
9     <h1>Testando carga de Imagens</h1>
10    <p>Abaixo você vai ver uma imagem que está na mesma pasta.</p>
11    
12    <p>Podemos também carregar imagens que estão em outra pasta, contanto
13       que ela esteja dentro da pasta atual.</p>
14    
15    <p>Também podemos carregar imagens externas, e para isso devemos ter a
16       sua URL completa.</p>
17    
19 </body>
```

O código acima só está começando na **linha 8** por questões de praticidade, pois as **linhas de 1 até 7** são aquelas criadas automaticamente pelo VScode. Não apague essas linhas, apenas vamos analisar o código do <body>.

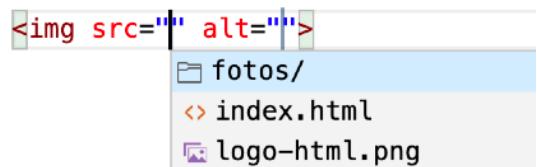
Na **linha 11**, adicionamos a tag , responsável por carregar imagens que estão na pasta do projeto atual ou em links externos. Essa tag tem como parâmetros básicos o src (que vem de *source*, origem) e alt (que vem de *alternative* ou texto alternativo).



Aqui vai uma **DICA IMPORTANTE**: quando for carregar uma imagem no Visual Studio Code, não digite a tag com os colchetes angulares. No lugar disso, digite apenas img e o próprio editor vai sugerir a inserção da imagem (veja foto ao lado).

Agora pressione a tecla **Enter**. Automaticamente, uma tag padrão será preenchida no seu código, incluindo as propriedades básicas src e alt. Mas a dica não para por aí. Sigamos.

Depois do código aparecer magicamente, o cursor do teclado vai aparecer dentro das aspas do parâmetro src. Pressione **Ctrl+Espaço** e mais uma mágica vai acontecer. Como a imagem ao lado mostra, o editor vai te Mostar uma listagem com todos os arquivo disponíveis na pasta do seu projeto. Escolha o arquivo clicando em logo-html.png para que você nem precise digitar o nome dele. Isso evita muitas falhas causadas por erros de digitação!



Agora volte para a página anterior e analise as linhas 13 e 15. A **linha 13** vai carregar a imagem logo-css.png que está dentro da pasta fotos, como vimos anteriormente. Note que foi necessário colocar o nome da pasta seguido de uma barra antes de colocar o nome do arquivo. Se isso não for obedecido, simplesmente a imagem não vai carregar!

Por fim, na **linha 15** fizemos a carga de um arquivo que não estava na pasta do projeto. A imagem js-small.gif está localizada em outro domínio, o jsdotgit.com. Nesse caso, precisamos indicar o caminho completo (**URL**) da imagem.

Você pode obter a URL completa de qualquer imagem, abra um site no Google Chrome, clique com o botão direito do mouse sobre a imagem e escolha a opção "*Copiar endereço da imagem*". Aí é só voltar no seu código HTML, colocar o cursor do teclado dentro das aspas da propriedade src da sua imagem e apertar **Ctrl+V**. Faça uma experiência em casa agora mesmo.

Para que servem os textos alternativos?

Toda imagem deve ter um texto alternativo, mesmo você achando que isso é muito chato de fazer. Textos alternativos ajudam muito na indexação do seu site em mecanismos de busca e também ajudam muito na **Acessibilidade**, pois se um visitante for deficiente visual, seu navegador vai ser capaz de descrever que tipo de foto está sendo mostrada ali.

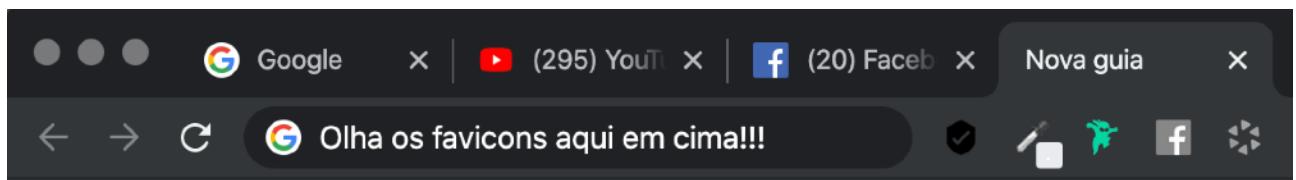
Tente ser objetivo na descrição da sua imagem, mas crie textos que fazem sentido. Tem um monte de gente que coloca o texto "foto" como descrição de uma imagem. Aí não dá!

Os textos alternativos também auxiliam o Google a saber o que tem dentro da sua foto e exibi-la nos resultados de busca do **Google Imagens**.

Usando Ícone de Favoritos (favicon)

Você já ouviu falar em **favicon**? Talvez você não os identifique pelo nome esquisito, mas com certeza já viu aqueles pequenos ícones que aparecem ao lado dos sites que visitamos, na parte superior do seu navegador. É o mesmo local onde aparecem o texto que você colocou na tag <title> do seu site.

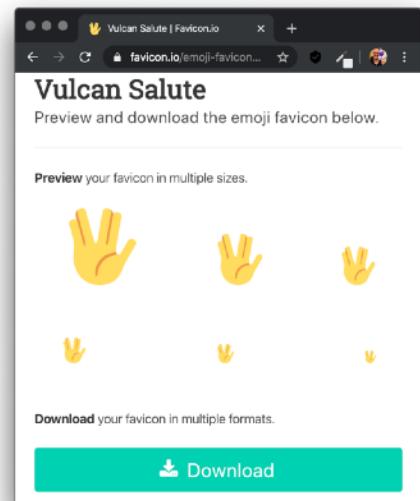
Na imagem abaixo, você consegue ver os favicons do Google, YouTube e Facebook. A última guia não tem favicon, porque não tem site nenhum aberto nela.



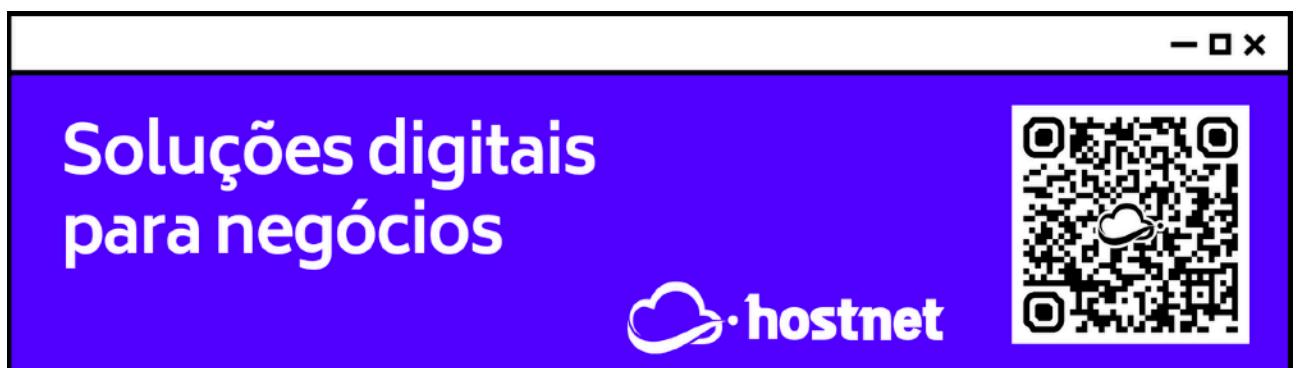
Para usar um favicon no seu site, você precisa ter o arquivo do ícone, que geralmente está no formato **ICO**. Se quiser baixar alguns ícones prontos, recomendo o site [IconArchive](#).

Se você tiver o dom e muita paciência, pode desenhar seu favicon no site [favicon.cc](#).

Agora, se seu objetivo é criar um ícone personalizado para seu site baseado em imagens que você já tem, recomendo o site [favicon.io](#), onde você pode submeter qualquer imagem e baixar um pacote com vários formatos de favicon. Lá você também vai poder criar um ícone de favoritos a partir de um texto ou então baseado em um emoji (veja foto ao lado).



Ao baixar o pacote com os ícones de favoritos, descompacte-os (geralmente é um arquivo ZIP), pegue o arquivo com extensão .ICO e coloque na pasta principal do seu projeto. Normalmente, colocamos o nome do arquivo como favicon.ico.



Agora que você já tem seu favicon salvo na pasta do seu projeto (recomendo criar o **ex004** no seu repositório local), vamos nos focar no código HTML para carregá-lo.

Crie seu arquivo `index.html` dentro da pasta **ex004**, incluindo seu código base automaticamente. Depois faça uma simples alteração na sua área `<head>`, incluindo uma tag `<link>`, digitando apenas a palavra `link` em uma linha em branco e escolhendo a opção `link:favicon`.

```
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <link rel="shortcut icon" href="favicon.ico" type="image/x-icon">
7    <title>Testando o ícone de Favoritos</title>
8  </head>
```

Pronto! É só dar uma olhada no nosso código acima e prestar atenção na **linha 6**. Se você usou a dica que dei acima, não precisou digitar mais nada! Todo o código foi feito automaticamente!

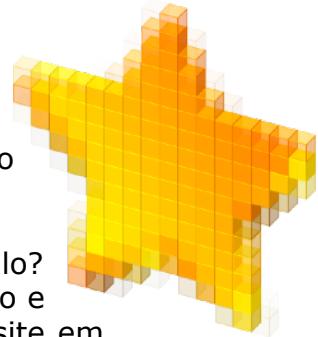
OBS: se seu arquivo não se chamar `favicon.ico`, você deve adaptar a linha 6, colocando o nome do seu ícone.

Quais são os formatos aceitos para Favicon?

Apesar da minha indicação para usar o arquivo no formato **ICO**, existem outros formatos suportados para o seu favicon, como o próprio **PNG** que estudamos anteriormente e até o formato **SVG**, que é vetorizado e seria a melhor das opções.

Mas se o **SVG** é a melhor das opções, por que não usá-lo? Infelizmente nem todo navegador é compatível com esse formato e acaba causando algumas inconsistências na exibição do nosso site em outras plataformas.

Sendo assim, pelo menos por enquanto, continuo dando o meu conselho: use o formato **ICO**.



Vamos aos desafios!

Lá no repositório, além do material em PDF e dos códigos dos exercícios 100% disponíveis, também disponibilizamos alguns **desafios** que devem ser resolvidos. Esses desafios não incluem o código original e você deve tentar chegar à resposta sem copiar nenhum código.

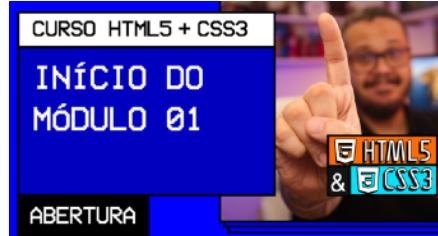
Com todo o conteúdo que vimos até essa aula, você já pode resolver do **desafio d001** até o **desafio003**. Acesse o repositório público, abra a área do curso de HTML+CSS e clique no link de acesso aos desafios. Manda ver! Só não fica pedindo a resposta! Você consegue resolver isso sozinho(a)!



Repositório em: <https://gustavoguanabara.github.io>

Quer acompanhar tudo em vídeo?

Eu sei que às vezes as pessoas gostam mais de assistir vídeos do que ler livros, e é por isso que eu lanço há anos materiais no canal Curso em Vídeo no YouTube. O link que vou compartilhar contigo faz parte da playlist completa onde você encontra o **Módulo 1 do Curso de HTML5 e CSS3**, completamente gravado com base nesse material.



Além de acessar o link a seguir, você também pode ter acesso às aulas apontando a câmera do seu celular para o código QR ao lado. Todo dispositivo smartphone ou tablet atualizado já possui esse recurso de leitura de códigos habilitado por padrão.

Módulo 1 do curso: https://www.youtube.com/playlist?list=PLHz_AreHm4dkZ9-atkcmcBaMZdmLHft8n

Teste seus conhecimentos

Terminou de ler esse capítulo e já acompanhou todos os vídeos e referências externas que indicamos? Pois agora, responda a essas 10 perguntas objetivas e marque em cada uma delas a única opção verdadeira. Aí sim, você vai poder comprovar que realmente entendeu o conteúdo.



1. Quando você está criando um site ou qualquer outro tipo de conteúdo, deve se preocupar bastante com os direitos de uso de determinada mídia (imagem, som, vídeo, etc). Qual dos itens a seguir melhor define esse assunto?

- A Se a mídia estiver em um site público como Google e YouTube, podemos usar sem problemas.
- B Não podemos usar nenhuma mídia sem a prévia autorização do seu criador.
- C Enquanto estamos estudando e aprendendo, temos total permissão para usar mídias de terceiros.
- D Podemos usar qualquer mídia e pagar uma multa quando formos descobertos.

2. Qual dos itens a seguir é incorreto quando o assunto é não passar por problemas relacionados a direitos autorais?

- A seja o criador das suas próprias artes
- B consiga uma autorização legal do detentor dos direitos autorais
- C compre os direitos de uso em sites especializados e use quantas vezes quiser
- D use imagens com licenças públicas, como Creative Commons

3. Qual dos sites a seguir é especializado em disponibilizar apenas imagens para uso liberado, sem direitos autorais?

- A Google Imagens
- B Getty Imagens
- C Shutterstock
- D Pexels

4. Qual foi o formato de imagens criado na década de 80 com o objetivo de reduzir o tamanho dos arquivos das imagens, sem reduzir sua dimensão?

- A JPEG
- B PNG
- C GIF
- D ZIP

5. Qual é o único dos formatos a seguir que possui suporte para imagens com partes transparentes e compactadas?

- A JPEG
- B BMP
- C TIFF
- D PNG

6. O que significa a sigla JPEG?

- A Joint Photographic Experts Group
- B JPG Photo Enhancement Graphic
- C Join Picture Enlarge Group
- D JPG Photography Enhancement Group

7. Muita gente fica na dúvida sobre o tamanho da imagem

- A use sempre a melhor qualidade possível para que seu site fique bonito
- B use sempre a pior qualidade possível, pois os sites precisam ser rápidos
- C tente balancear sempre qualidade e tamanho, para que seu site não fique pesado mas não se esqueça da experiência proporcionada ao seu visitante.
- D deixe o visitante escolher se quer imagens grandes ou sites rápidos

8. Qual é a tag usada na HTML para inserir imagens como parte do conteúdo do site?

- [A] <image>
- [B] <photo>
- [C] <picture>
- [D]

9. Dentro da tag para inserir a imagem, usamos um parâmetro especial para indicar a localização da imagem a ser carregada. Que parâmetro é esse?

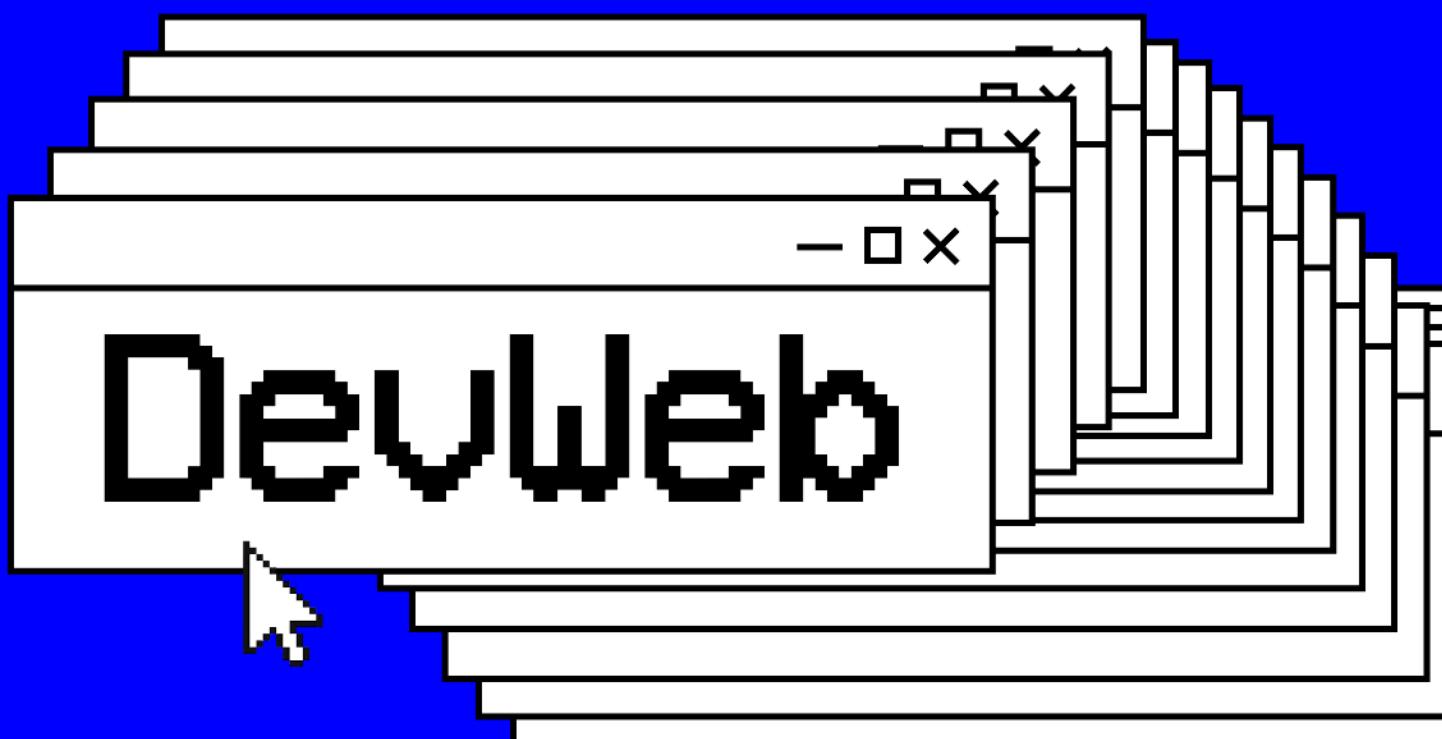
- [A] src
- [B] source
- [C] photo
- [D] jpeg

10. Para configurar uma imagem como ícone de favoritos de uma página (*favicon*), usamos a tag _____ dentro da área _____ do seu site. Qual dos itens a seguir, é o único que preenche corretamente as lacunas da frase anterior?

- [A] / <body>
- [B] <favicon> / <head>
- [C] <link> / <head>
- [D] <meta> / <body>

Suas anotações

Não guarde conhecimento. Ele é livre. Compartilhe o seu e veja ele se espalhando pelo mundo 



HIERARQUIA DE TÍTULOS

M01C07

veja em vídeo



HIERARQUIA DE TÍTULOS

Vamos agora começar a falar sobre organização conteúdo. Ainda teremos muito mais capítulos falando sobre isso daqui pra frente, mas tudo começa nos títulos. São eles que dizem quando existe um assunto principal ou um sub-assunto, dentro de um desses principais. Fazer isso em HTML5 é muito fácil e você está prestes a aprender isso. Vamos começar?



Você tem todo o direito de usar esse material para seu próprio aprendizado. Professores também podem ter acesso a todo o conteúdo e usá-lo com seus alunos. Porém todos que usarem esse material - seja para qual for a finalidade - deverão manter a referência ao material original, criado pelo **Prof. Gustavo Guanabara** e disponível no endereço do seu repositório público <https://github.com/gustavoguanabara/>. Este material não poderá ser utilizado em nenhuma hipótese para ser replicada - integral ou parcialmente - por autores/editoras para criar livros ou apostilas, com finalidade de obter ganho financeiro com ele.

Entendendo a hierarquia de títulos

Vamos começar com um exercício prático. Te convido a analisar uma matéria da revista **VOCÊ S/A**, de Fevereiro de 2015. Não precisa se focar em ler o conteúdo, apenas olhe para a imagem a seguir e tente identificar seus componentes visuais.

CARREIRA → LIDERANÇA

FALTA DE SINTONIA

Um chefe nem sempre simpatiza com seus subordinados. Mas há estratégias para o gestor contornar as diferenças sem prejudicar ninguém ou comprometer o resultado do time *Por Marcia Di Domenico*

Passamos tantas horas no escritório que é comum enfrentar sentimentos de amor, ódio e ansiedade. Também é inevitável criar laços de amizade e, claro, de inimizade. Quando o conflito se dá entre chefe e subordinado, a situação pede cautela para não colocar em risco o clima e a produtividade — fatores tão importantes em um período em que a palavra de ordem nas empresas é eficiência. “Não se espera que o líder goste, da mesma forma, de todos na equipe nem que seja igualmente querido por cada integrante”, diz Liane Davey, vice-presidente da Knightsbridge Human Capital Solutions, consultoria em gestão de pessoas em Ontário, no Canadá. “Mas manter uma relação cordial com cada um é obrigatório.” O relacionamento entre chefe e equipe vai bem não quando todo mundo se adora, mas quando o convívio não compromete o trabalho. Gestores que precisam motivar (e gerenciar) profissionais com os quais não se dão bem devem adotar algumas atitudes para que o desafeto não prejudique o relacionamento profissional.

Pense antes de agir

O fato de se sentir incomodado com alguém diz mais sobre si mesmo do que sobre o outro. Antes de tomar qualquer atitude, o líder deve refletir sobre se o comportamento do funcionário pode demonstrar, na verdade, que há algum problema com a própria gestão. “O chefe deve se perguntar se há alguma coisa que possa mudar em si mesmo para contornar a situação”, diz Jair Moggi, consultor da Adigo DEF, consultoria em gestão, de São Paulo. Isso evita agir por simples impaciência, poupando sua imagem de um desgaste desnecessário.

Passe mais tempo com seu desafeto

Evitar o contato ou ignorar o subordinado com quem você não se dá bem só piora as coisas. “Ele se sente desestimulado e desmoralizado dentro do time”, diz a consultora Liane Davey. Crie maneiras de passar mais tempo juntos — pode ser um almoço ou um café. Se achar que a iniciativa soa artificial, converse com outras pessoas, o que diminui o desconforto. Essa atitude ajuda a enxergar traços de personalidade às vezes invisíveis e, consequentemente, a entender determinados hábitos e atitudes.

Saiba com quem está lidando

Tão importante quanto entender seus pontos fortes e fracos como chefe é investir tempo e disposição para aprender sobre a biografia, a personalidade e as limitações de cada integrante do time. Num momento de conflito, esse conhecimento vai auxiliar o gestor a entender o que pode estar por trás de determinadas ações e a escolher o tratamento adequado. “O funcionário sente que pertence ao grupo, o que é um dos princípios fundamentais da motivação”, diz Jair. Deixar claro o que a empresa espera do profissional e se certificar de que o subordinado comprehende o que precisa fazer também é importante para estimular a equipe.

Tenha objetividade

Independentemente do motivo de não gostar de alguém, nunca leve a questão para o lado pessoal. A razão do mal-estar deve ser objetiva, assim o gestor conseguirá abordar o problema com racionalidade. A melhor maneira de falar sobre o assunto é em uma conversa a sós com o subordinado para evitar o desgaste da pessoa com o grupo. “Todo feedback precisa estar embasado na observação de alguma atitude ou episódio real e na reflexão do que isso representa para o trabalho”, diz Maria Cândida Baumer, diretora da People & Results, consultoria de São Paulo. Caso contrário, pode ser interpretado simplesmente como crítica e não repercutir como desejado. Outro erro é mandar recado por um colega da equipe, usar ironia ou jogar indiretas. “A agressividade

34 FEVEREIRO DE 2015 | VOCÊ S/A

Agora que você já deu uma olhada na estrutura, vamos procurar identificar a sua hierarquia de títulos. Com toda certeza, se eu te perguntar qual é o título principal da matéria, a resposta vem imediatamente: “*Falta de sintonia*”. Isso acontece porque esse é o texto que está em destaque principal no texto.

Mas aí, te faço outra pergunta: existem outros títulos nessa página? Uma breve análise visual nos leva a um resultado:

- *Pense antes de agir*
- *Passe mais tempo com seu desafeto*
- *Saiba com quem está lidando*
- *Tenha objetividade*

Outra coisa que você também consegue identificar é um parágrafo destacado logo após o título principal, onde a fonte do texto é até maior e mais destacada, mas não chega a ser um título. Em seguida, existe um parágrafo introdutório, onde encontramos um texto que aponta a essência da matéria.

Depois disso, começamos com os sub-títulos, que com certeza são partes que falam sobre o assunto "Falta de sintonia", mas separa o tema principal em sub-temas para melhorar a organização dos "pensamentos" de quem escreveu a matéria. Isso facilita muito na organização dos textos.

Dentro de cada um desses sub-títulos temos um parágrafo que desenvolve mais sobre cada assunto.



COMECE A PRESTAR ATENÇÃO NISSO: Algumas matérias em jornais e revistas simplesmente não separam o conteúdo em sub-títulos. Geralmente isso gera textos longos e cansativos. Separar seu texto em assuntos secundários prendem bastante o leitor.

Quantas vezes você lê um livro e fica constantemente virando as páginas para saber quanto falta para acabar o capítulo atual? Geralmente livros técnicos possuem capítulos separados por sub-títulos, exatamente como estou fazendo aqui. Isso não foi uma escolha arbitrária, prefiro ler coisas assim.

Acho que deu pra entender meu ponto aqui, não é? Organizar o nosso conteúdo quando criamos páginas é essencial para ter um bom site.

Como criar títulos em HTML?

Títulos em HTML são conhecidos como *headings* (que, traduzindo do Inglês, significa *título* mesmo 😅). Os títulos possuem **SEIS NÍVEIS** de hierarquia, e esses níveis servem para organizar nosso conteúdo. Para isso, usamos as tags `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>` e `<h6>` para demarcar nossos títulos.

Mas cuidado! Tamanho não é documento!

Tem muita gente por aí que acha que `<h1>` significa "*texto com fonte grande*" e `<h6>` significa "*texto com fonte pequena*". Isso não tem fundamento algum! Por favor, se alguém te disse isso algum dia, essa pessoa estava completamente equivocada.

Um `<h1>` significa que esse é um assunto principal e é sobre ele que vamos escrever. Já o `<h2>` significa que esse é um sub-assunto do `<h1>` que está imediatamente acima

A screenshot of a website with a purple header. The header contains the text "Soluções digitais para negócios" and the Hostnet logo. To the right of the text is a QR code.

dele. O `<h3>` significa que esse é um sub-assunto do `<h2>` que está acima dele. E assim sucessivamente até o `<h6>`.

Faça um outro exercício aí. Estamos agora na página 3 (é só olhar aqui embaixo). Vá para a página 1 desse capítulo e olhe o nosso `<h1>`: "Hierarquia de Títulos". Nessa página 2, temos o sub-título `<h2>`: "Entendendo a hierarquia de títulos". Ele é um sub-assunto do nosso `<h1>`.

Nessa página 3 também temos outro `<h2>`: "Como criar títulos..." e logo abaixo temos um `<h3>`: "Mas cuidado...". Note que esse é um sub-assunto de "como criar títulos...". É claro que o nosso H3 tem letra menor que nosso H2, mas não se trata de tamanho. É questão de hierarquia!

Só um H1 em uma página: fato ou lenda?

Rola por aí muita gente falando que um documento HTML só deve ter um título H1 e que todos os demais devem ser sub-títulos deles. Isso surgiu com a galera que falava sobre otimização de páginas para mecanismos de busca (SEO - *Search Engine Optimization*).

Hoje em dia, o próprio Google (maior mecanismo de busca do mundo) já diz que isso é uma lenda. Para eles, o que mais importa é dar significado ao seu H1. Ele é um tema principal, e se a sua página tem vários temas principais, use vários H1 sem medo.



OUÇA O ESPECIALISTA: Aqui embaixo você vai encontrar um link para o vídeo do canal do **Google para Webmasters** onde o especialista **John Mueller** fala sobre esse assunto. O conteúdo está em inglês, mas você pode clicar no botão da engrenagem no canto inferior direito do vídeo e mandar **traduzir automaticamente** as legendas para o Português. Fica ótimo!

Acesse: <https://youtu.be/WsgrSxCmMbM>

Vamos criar a hierarquia da matéria da revista

A missão agora é transformar o conteúdo da matéria que te mostrei no início do capítulo em uma página HTML. Eu não vou me focar em digitar o texto dos parágrafos de cada seção e no lugar deles vou usar os famosos *Loem Ipsum* genéricos. Vamos começar abrindo o **Visual Studio Code** e criando um documento teste.html em uma pasta qualquer. Dentro desse documento, crie o código base HTML usando a

exclamação, como já ensinei algumas vezes durante esse material (consulte o Capítulo 4, página 6). Dentro da tag <body>, criaremos a estrutura do conteúdo.



CRIANDO LOREM AUTOMATICAMENTE: Até na hora de criar um *Lore Ipsum*, o VS Code te ajuda. Dentro de um parágrafo recém criado, digite apenas a palavra lorem e pressione **Enter**. Pronto!

Você viu que a imagem acima começa na **linha 8**. As anteriores são o código HTML automático criado pelo VS Code.

```
« teste.html »
exercicios > ex000 > « teste.html > html
8  <body>
9    <h1>Falta de sintonia</h1>
10   <p class="destaque">Lorem ipsum dolor sit amet consectetur
11     adipisicing elit. Dicta ab assumenda labore, magni commodi
12     nesciunt praesentium non!</p>
13   <p>Lorem ipsum dolor, sit amet consectetur adipisicing elit.
14     Cumque iusto exercitationem aut a temporibus, suscipit sit,
15     quas repudiandae odit molestiae numquam officia illum sed
16     quisquam possimus quibusdam distinctio pariatur id!</p>
17   <h2>Pense antes de agir</h2>
18   <p>Lorem ipsum dolor sit amet consectetur adipisicing elit.
19     Sequi quas earum natus obcaecati esse repudiandae officiis
20     dolorum eaque qui autem corrupti atque cupiditate blanditiis
21     laboriosam maiores, aliquam aspernatur! Animi, debitis.</p>
22   <h2>Passem mais tempo com seu desafeto</h2>
23   <p>Lorem, ipsum dolor sit amet consectetur adipisicing elit.
24     Modi sed quisquam sequi perspiciatis mollitia optio ut,
25     ratione eius. Magni dolores repudiandae deserunt, molestias
26     consequuntur aliquam eius dolorum vel saepe recusandae?</p>
27   <h2>Saiba quem tá lidando</h2>
28   <p>Lorem ipsum dolor sit amet consectetur adipisicing elit.
29     Laudantium provident, iusto maiores corrupti fugit dolor
30     consequatur ad officiis sunt, adipisci distinctio quia unde
31     natus eum? Nobis vel deleniti id blanditiis.</p>
32   <h2>Tenha objetividade</h2>
33   <p>Lorem ipsum, dolor sit amet consectetur adipisicing elit.
34     Tempore repudiandae odit ratione magnam nihil libero
35     temporibus dolore incidunt, error iure harum, laborum
36     voluptates quasi? Dolorum consequuntur in doloribus natus
37     labore.</p>
20  </body>
```

Na **linha 9** temos nosso título principal. Nas **linhas 12, 14, 16 e 18** temos os sub-títulos do principal, como analisamos anteriormente.

OBS: Na **linha 10**, tivemos uma novidade que foi a atribuição de uma classe ao parágrafo após o <h1>. Mais para frente falaremos mais sobre isso, mas basicamente é para conseguirmos criar uma configuração visual diferente para esse parágrafo especial usando folhas de estilo CSS.

Hora de exercitar

Agora chegou a hora de praticar. Acesse agora mesmo o endereço <https://gustavoguanabara.github.io/html-css/exercicios/> e execute o **exercício 006** no seu computador e tente atingir esse mesmo resultado em casa, sem copiar o código que eu criei. Nesse momento, a prática é algo que você mais precisa. Se por acaso ficar difícil, pode acessar o repositório público de HTML e CSS e dar uma olhada nos comandos, mas **EVITE COPIAR**.



Quer acompanhar tudo em vídeo?

Eu sei que às vezes as pessoas gostam mais de assistir vídeos do que ler livros, e é por isso que eu lanço há anos materiais no canal Curso em Vídeo no YouTube. O link que vou compartilhar contigo faz parte da playlist completa onde você encontra o **Módulo 1 do Curso de HTML5 e CSS3**, completamente gravado com base nesse material.



Além de acessar o link a seguir, você também pode ter acesso às aulas apontando a câmera do seu celular para o código QR ao lado. Todo dispositivo smartphone ou tablet atualizado já possui esse recurso de leitura de códigos habilitado por padrão.

Módulo 1 do curso: https://www.youtube.com/playlist?list=PLHz_AreHm4dkZ9-atkcmcBaMZdmLHft8n

Cursos grátis de tecnologia
que te preparam para o
mercado de trabalho

RECODE

QR code linking to RECODE courses.

Teste seus conhecimentos

Terminou de ler esse capítulo e já acompanhou todos os vídeos e referências externas que indicamos? Pois agora, responda a seguintes perguntas objetivas e marque em cada uma delas a única opção verdadeira. Aí sim, você vai poder comprovar que realmente entendeu o conteúdo.



1. Normalmente nossos textos são organizados em sessões, e cada uma pode possuir um título. Em HTML5 existe alguma limitação em relação a quantidade de títulos que um documento pode ter?

- A sim. Só podemos ter no máximo seis títulos
- B não. Um texto pode ter quantos títulos e quantos níveis for necessário
- C não. Um texto pode ter vários títulos, mas apenas um máximo de seis níveis
- D sim. Podemos ter até 12 títulos, dois para cada nível

2. Os títulos podem ser criados em seis níveis diferentes, usando as tags:

- A <head1> até <head6>
- B <header1> até <header6>
- C <title1> até <title6>
- D <h1> até <h6>

3. Se estamos em um ponto do texto com título nível 3 e queremos aprofundar esse assunto com um subtítulo, devemos usar qual nível? Lembre-se de seguir a recomendação da W3C.

- A nível 4
- B nível 2
- C nível 3
- D qualquer nível, de 3 até 6

4. O título principal de um determinado documento ou página deverá sempre ser definido com um título de qual nível?

- A nível 1
- B nível 2
- C nível 3
- D tanto faz, o que importa é ser um título

5. Muitos dizem que um documento só deve ter um título de nível 1 (principal). Essa crença é uma...

- A verdade. Somente um título deve ser o principal e todos os demais deverão ter outros níveis.
- B lenda. A recomendação é que não exista nenhum título de nível 1.
- C verdade. Se colocarmos mais de um título nível 1, seu site não será indexado pelo Google.
- D lenda. É possível adicionar mais de um título de nível 1, só devemos nos dedicar a descrevê-lo muito bem para facilitar a indexação por mecanismos de busca.

6. O estudo da indexação do conteúdo de um site por meio de mecanismos de busca é conhecido popularmente como:

- A SEO - Seek Energy Organization
- B SEO - Search Engine Optimization
- C SEO - Seek Engine Order
- D SEO - Search Elements Order

7. Todo documento HTML deverá ter obrigatoriamente algum título de nível 6?

- A não. Os títulos de menor nível serão usados apenas se for necessário
- B sim. Pelo menos um título de nível 6
- C não. Os títulos de nível 6 devem ser evitados para não confundir o visitante
- D sim. É necessário ter dois ou mais títulos de nível 6

8. Quando estamos desenvolvendo um site e o conteúdo ainda não foi definido, uma técnica bastante utilizada é usar blocos de texto genéricos com conteúdo composto por palavras em *latim*. Esses blocos são mais conhecidos pelo termo:

- A Lorem Ipsum
- B Ipsum Lorem
- C Carpe Diem
- D Memento Vivere

9. Para criar um bloco de texto com termos genéricos em latim usando o VS Code, basta digitar a palavra _____ e depois apertar a tecla _____.

- A lorem / Esc
- B ipsum / Enter
- C lorem / Enter
- D ipsum / Esc

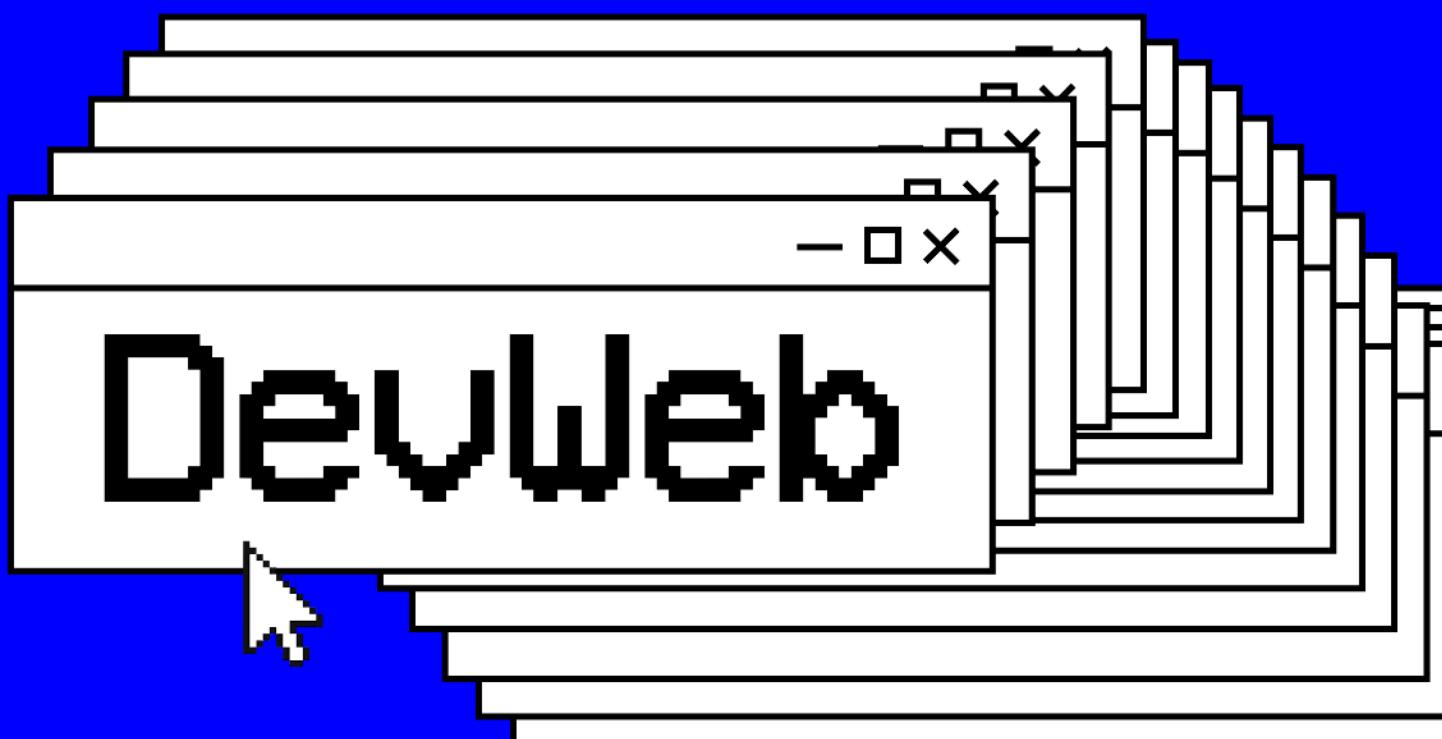
10. Se não configurarmos o estilo de cada título, os textos dentro da tag de nível 1 terão letras _____, enquanto aqueles dentro da tag de nível 6 terão letras _____. Qual dos itens abaixo é o único que preenche corretamente as lacunas desta frase na ordem correta?

- A menor / maior
- B em negrito / em itálico
- C maior / menor
- D azuis / vermelhas

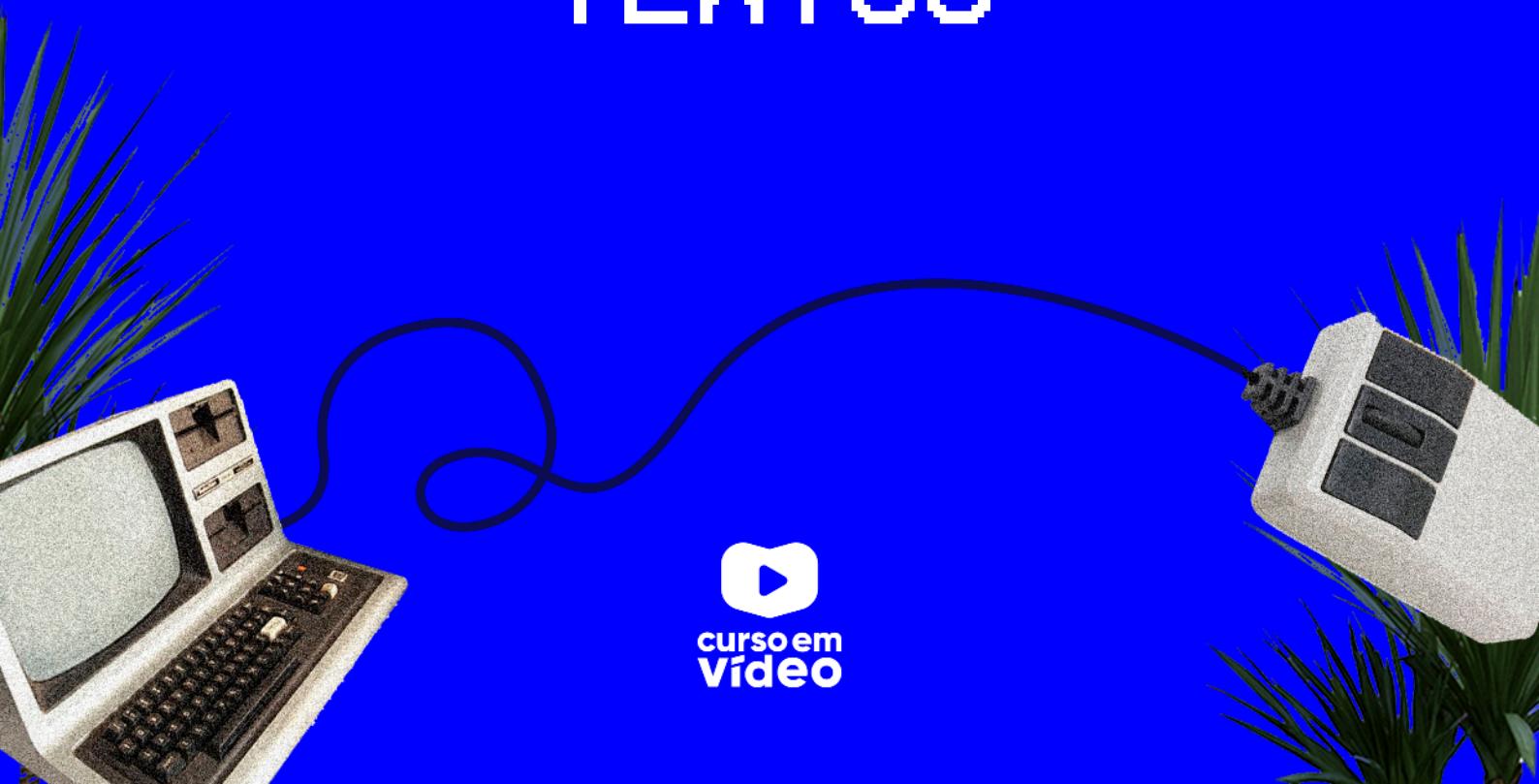
The advertisement features a dark blue background with white text. At the top, there's a window-like header with minimize, maximize, and close buttons. Below this, the text reads: "Cursos que vão te levar ao próximo nível". In the bottom right corner, there's the Estudonauta logo, which includes a small icon of a person with a brain-like shape above it, followed by the word "estudonauta". To the right of the text is a large QR code.

Suas anotações

Não guarde conhecimento. Ele é livre. Compartilhe o seu e veja ele se espalhando pelo mundo 



FORMATAÇÃO DE TEXTOS



M01C08

veja em vídeo



FORMATAÇÃO DE TEXTOS

Eu já consigo sentir que estamos evoluindo na HTML, você não? Agora nossos documentos começam a tomar uma organização maior com os títulos e parágrafos, mas às vezes precisamos de algo a mais. Vamos ver algumas formatações importantes e que usaremos quando começarmos a produzir nossos conteúdos. Não se esqueça de manter tudo anotado, pois veremos muitas tags novas.



Você tem todo o direito de usar esse material para seu próprio aprendizado. Professores também podem ter acesso a todo o conteúdo e usá-lo com seus alunos. Porém todos que usarem esse material - seja para qual for a finalidade - deverão manter a referência ao material original, criado pelo **Prof. Gustavo Guanabara** e disponível no endereço do seu repositório público <https://github.com/gustavoguanabara/>. Este material não poderá ser utilizado em nenhuma hipótese para ser replicada - integral ou parcialmente - por autores/editoras para criar livros ou apostilas, com finalidade de obter ganho financeiro com ele.

Você já ouviu falar de semântica?

Tá aí mais uma palavra bonita pra gente aprender: “semântica”. Se você nunca ouviu falar nela, provavelmente não está entendendo direito, mas saiba que a maior mudança da HTML4 para a HTML5 é o acréscimo da semântica aos elementos.

Vamos procurar no dicionário, e vou fazer isso no **Michaellis**, que tem uma versão online e gratuita. Segundo o pai-dos-burros:

“Semântica é o significado dos vocábulos, por oposição à sua forma.”



Analizando a frase acima percebemos que uma palavra pode ter **forma** e **significado**, e que a **semântica** dá mais valor ao significado.

Na HTML4, tínhamos tags como **** que colocava um texto em negrito, **<u>** que colocava o termo sublinhado e **<blink>** que fazia o texto piscar. Essas eram tags que representavam apenas uma **forma**. Você dizia que queria um texto sublinhado, mas qual era o motivo? Qual era o sentido de sublinhar alguma coisa? Vamos ver um exemplo:

Juvenal era um sujeito de muita sorte. E já começou de pequeno, onde morou na **<u>Rua Marquês de Lira Filho</u>**, um local de fácil acesso ao Centro da cidade.

Em HTML4, colocamos o par de tags **<u>** e **</u>** para delimitar o termo Rua Marquês de Lira Filho. Isso seria uma maneira de determinar somente um formato visual para chamar atenção para o endereço onde o cara nasceu. O sublinhado é apenas uma **forma**, sem **significado** explícito. Sublinhamos só pra chamar atenção visualmente.

Já a HTML5 chegou com o conceito de valorizar a **semântica**, logo suas tags tentam levar um **significado** embutido muito forte. Logo, a frase acima ficaria assim:

Juvenal era um sujeito de muita sorte. E já começou de pequeno onde morou na **<address>Rua Marquês de Lira Filho</address>**, um local de fácil acesso ao Centro da cidade.

A purple rectangular advertisement. At the top, there's a window-like header with close and minimize buttons. Below it, the text reads: "Cursos grátis de tecnologia que te preparam para o mercado de trabalho" and "RECODE". To the right is a large QR code.

Note que agora, usamos a tag <address> para dar um significado ao destaque que fizemos. Estamos chamando atenção para Rua Marquês de Lira Filho por se tratar do endereço da pessoa. No caso, um navegador de celular pode até sugerir que você veja o mapa do local e trace a rota para chegar lá. Viu? **SIGNIFICADO!**

Sendo assim, em HTML5, vemos de forma bastante evidente a presença do chamado **HTML semântico** ou **tags semânticas** ou ainda o **conteúdo semântico**. Você quer que um endereço apareça na forma de um texto sublinhado? Use CSS para configurar isso, a HTML serve para dar sentido ao conteúdo. É assim que tudo vai funcionar.

Desde a mudança de versões, a **W3C** - consórcio responsável por normatizar a HTML - tem dado muito valor por adicionar novas tags que tenham mais significado e a tirar algumas tags que só se focam no efeito visual (forma) de apresentação. O intuito é deixar a apresentação gráfica por conta das CSS.

Tags morrem, você sabia?

Existem tags que ainda funcionam hoje em dia, mas estão prestes a serem consideradas obsoletas, como as tags e <center>. E existem outras que simplesmente foram excluídas das versões mais atuais da HTML5, como é o caso do <applet> e <blink>.



Isso acontece porque a linguagem evolui, e nesse processo algumas tags param de fazer sentido pois existem outras bem melhores e que fazem mais sentido.

Sendo assim, a própria W3C sugere que no lugar de , que significa **bold** ou **negrito** e que seria simplesmente uma forma de apresentar um dado, passemos a usar a tag , que tem um significado de **força** ou **potência** dentro da frase.



FIQUE SEMPRE DE OLHO: Existe um documento oficial do Consórcio da World Wide Web (W3C) que é atualizado constantemente com as tags que estão ficando obsoletas e algumas substituição desejáveis que devemos fazer.

Diferenças entre HTML4 e 5: <https://www.w3.org/TR/html5-diff/#absent-attributes>

Elementos obsoletos na HTML5: <https://dev.w3.org/html5/pf-summary/obsolete.html>

Sendo assim, se você está aprendendo HTML com o uso das tags , <big>, <center>, <srtike> e muitas outras, pode mandar esses links **OFICIAIS** que estão aí em cima para quem está te ensinando desse jeito. Talvez essa pessoa nem saiba ainda que deverá atualizar seus materiais o quanto antes.

Antes de começar, fique de olho nos exercícios

Como você já deve saber, esse material é para ser consumido juntamente com os demais conteúdos disponibilizados no nosso repositório público do **Curso de HTML+CSS**, disponível em <https://gustavoguanabara.github.io>. Pois abra o código disponível dos exercícios resolvidos e analise os códigos dos exercícios **ex007** e **ex008**. Abra também o link onde você pode executar os exercícios e rode esses mesmos exemplos 007 e 008. As coisas vão ficar muito claras pra você daqui pra frente, eu garanto!



Negrito e Itálico

Vamos ver agora algumas formatações bem usadas das últimas versões da linguagem, começando pelos famosos **negrito** e *itálico*.

Como vimos anteriormente, existem as tags **** e **<i>** para essa tarefa, mas elas não possuem significado e focam apenas na forma, sendo assim, são pouco semânticos. Sendo assim, recomendamos que você passe a usar as tags **** e **** para realizar essas mesmas formatações visuais, só que agora com sentido.

A tag **** significa que o termo delimitado possui força dentro da frase. Logo, ele aparecerá em **negrito**.

Já a tag **** significa que queremos dar **ênfase** (do Inglês *emphasis*) ao termo. Logo, ele aparecerá em *itálico*.

Note que, ao usar **** e **** no lugar de **** e **<i>**, damos mais significado aos nossos termos e conteúdos. Como eles vão ser representados visualmente (forma), vai depender das nossas folhas de estilo CSS.

Você sabe usar marca texto?

Provavelmente você já viu ou usou uma dessas canetas marca texto em seu dia-a-dia. Elas servem para você marcar uma parte do texto na qual você quer dar uma ênfase descomunal, já que se trata de um **trecho muito valioso** para você, assim como acabamos de fazer.



Para fazer essas marcas em HTML5, usamos a tag **<mark>...</mark>** para delimitar o texto que queremos demarcar, como se estivéssemos usando uma caneta marcador.



COMO FAÇO PARA MUDAR A COR DA CANETA?

Com certeza você já deve estar se perguntando como usar outras cores. Pois saiba que isso é totalmente possível, contanto que as configurações sejam especificadas nas folhas de estilo CSS.

<big> morreu, mas o <small> ainda sobrevive

Tem certas coisas que simplesmente não são fáceis de entender. A tag `<big>` (que deixava o texto maior) está depreciada pela HTML5, mas a tag `<small>` (que deixa o texto menor) segue firme e forte na vida da linguagem.



E não adianta ficar me olhando com essa cara de reprovação. Eu não tenho culpa nenhuma por conta dessa decisão, sou apenas um mero professor que se foca em mostrar os fatos, não em fazê-los ter um sentido claro. Tem certas coisas que a gente não questiona, só aceita que dói menos.

Texto deletado

Antigamente existia a tag `<strike>`, que hoje também está depreciada pela última versão da HTML. No lugar dela entrou a tag ``, que significa que o texto está ali, pode até ser lido, mas deve ser desconsiderado pelo leitor. Exatamente como eu acabei de fazer.

Texto inserido

O texto inserido é o exato oposto do texto deletado que vimos anteriormente. Nesse caso, se colocarmos um texto qualquer dentro de `<ins>` e `</ins>`, estamos dizendo que o texto está ali, deve ser lido e você deve prestar atenção nele. É o que fazíamos anteriormente ao utilizar a tag `<u>` para sublinhar um texto.

Soluções digitais
para negócios

 hostnet



Textos sobreescrito e subscripto

Quantas vezes você tentou escrever um x^2 ou um H_2O em um documento e ficou na dúvida de como fazer pra esses números ficarem pequeninos ali em cima ou embaixo. Pois a HTML tem as tags `<sub>` e `<sup>` justamente para essa finalidade. Ao analisar os exemplos de código que apontei no início desse documento, você vai perceber melhor a diferença entre eles.

X_2 X^2

Trechos de código

Nós somos (ou seremos) programadores, e por isso compartilhamos muito código-fonte em diversas linguagens. Para isso, existe a tag `<code>` da HTML onde você pode delimitar seu código. A principal vantagem no uso dessa tag é o valor semântico que ela representa, indicando ao navegador que se trata de um código de computador. Porém, existe também um efeito visual, pois as letras ficam no modo mono-espacadas (monospace), o que facilita bastante a leitura do código. Analise, por exemplo, o trecho de código a seguir:

```
<h1>Exemplo de código em Python</h1>
<code>
num = int(input('Digite um número'))
if num % 2 == 0:
    print(f'O número {num} é par')
else:
    print(f'O número {num} é ímpar')
print('FIM DO PROGRAMA')
</code>
```

Você pode achar que o código vai aparecer lindamente no seu site, porém o que acontece é simplesmente uma mudança das letras, para que elas fiquem mono-espacadas (veja a imagem).



Para resolver esse problema, vamos usar uma outra tag HTML chamada `<pre>`, que mantém o texto pré-formatado, exatamente da mesma maneira na qual ele foi digitado, incluindo quebras de linhas, espaços e tabulações.

```
<h1>Exemplo de código em Python</h1>
<pre><code>
num = int(input('Digite um número'))
if num % 2 == 0:
    print(f'O número {num} é par')
else:
    print(f'O número {num} é ímpar')
print('FIM DO PROGRAMA')
</code></pre>
```

Notou agora a diferença? A junção das tags `<pre>` e `<code>` em conjunto nos trouxe um resultado visual bem mais interessante.



AS TAGS, UNIDAS JAMAIS SERÃO VENCIDAS! No exemplo acima você pode perceber que adicionamos uma tag dentro da outra. Quando isso acontece, é como se elas juntassem suas forças para gerar um resultado ainda mais poderoso. É possível então juntar `` e `` e gerar um resultado como esse: ***em negrito e em itálico ao mesmo tempo!***

“Citações”

Se você já escreveu um texto sequer na sua vida, com certeza já teve que fazer citações. Uma citação é um trecho de texto, escrito ou dito por outra pessoa, que vai ilustrar perfeitamente algo que você quer explicar.

“ ”

“ ”

Normalmente, uma citação aparece entre aspas ou com as margens deslocadas em relação ao texto. Confuso pra você? Pois volte na página 2 desse capítulo e lá você vai ver três citações escritas. Com certeza você vai descobrir!

Para criar uma citação em HTML, podemos usar a tag `<q>` (do Inglês *quote*, que significa citar). O texto que estiver entre `<q>` e `</q>` já vai receber automaticamente as aspas, mas não terá nenhum deslocamento. Essa técnica é mais usada quando queremos uma citação no meio de um parágrafo.

Também podemos criar citações mais longas (em bloco) e que tenham um parágrafo só para si. Nesse caso, colocaremos tudo dentro de `<blockquote>` e `</blockquote>`

o texto ganha um recuo automaticamente. Podemos também colocar um link para o texto original, usando o parâmetro `cite` dentro da tag.

```
<blockquote cite="https://www.martinsfontespaulista.com.br/
php-a-biblia-53304.aspx/p">
```

O PHP é uma linguagem para a criação de scripts para a Web do lado servidor embutidos em HTML, cujo código-fonte é aberto, e que é compatível com os mais importantes servidores Web (especialmente Apache).

```
</blockquote>
```

Abreviações

Essa é uma novidade da HTML5 e que ajuda muito em áreas como a de Tecnologia, que usa muitas siglas e abreviações. Sempre que você quiser escrever uma sigla, mas deixar claro ao usuário (e aos mecanismos de busca) o significado dela, use a tag `<abbr>`.

```
<p>Eu estou estudando <abbr title="Hyper Text Markup
Language">HTML</abbr> para criar sites.</p>
```

Note no código acima que usamos o `title` para indicar o significado da sigla. Ao lado, mostro o funcionamento desse código no navegador.

Eu estou estudando HTML para criar sites.

Hyper Text Markup Language

Quando passamos o mouse sobre a sigla abreviada, um pequeno texto aparece com o seu significado.

ODITREVNI OTXET

Não entendeu nada desse título? Pois volte ali em cima e leia da última letra até a primeira. Achou inútil? Eu também. Pois é exatamente essa a função da tag `<bdo>`.

Para começo de conversa, BDO significa *bi-directional override*. Ao usar essa tag, coloque também o parâmetro `dir` para indicar uma das duas direções possíveis:



rtl = da direita para a esquerda (*right-to-left*)

ltr = da esquerda para a direita (*left-to-right*)

```
<h2>Texto Invertido</h2>
```

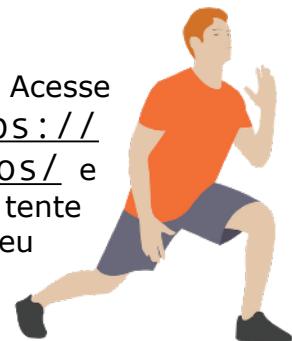
```
<bdo dir="rtl">Este texto todo está invertido no formato  
RIGHT-T0-LEFT.</bdo>
```

E aí, já acabou?

Nesse capítulo, aprendemos vários tipos de formatação de textos. Mas gostaria de deixar bem claro que existem muitos outros. O que fiz aqui foi uma seleção dos mais usados (e também o `<bdo>`, que é o mais inútil) e vamos dar prosseguimento na matéria. Se por acaso, mais pra frente, precisarmos usar alguma outra tag de formatação de textos, eu explico pontualmente. Combinado?

Hora de exercitar

Eu já dei esse conselho lá em cima, mas não custa nada repetir. Acesse agora mesmo o endereço do nosso repositório público em <https://gustavoguanabara.github.io/html-css/exercicios/> e execute o **exercício 007** e o **exercício 008** no seu computador e tente atingir esse mesmo resultado em casa, sem copiar o código que eu criei. Nesse momento, a prática é algo que você mais precisa. Se por acaso ficar difícil, pode acessar o repositório público de HTML e CSS e dar uma olhada nos comandos, mas **EVITE COPIAR**.



Quer acompanhar tudo em vídeo?

Eu sei que às vezes as pessoas gostam mais de assistir vídeos do que ler livros, e é por isso que eu lanço há anos materiais no canal Curso em Vídeo no YouTube. O link que vou compartilhar contigo faz parte da playlist completa onde você encontra o **Módulo 1 do Curso de HTML5 e CSS3**, completamente gravado com base nesse material.



Além de acessar o link a seguir, você também pode ter acesso às aulas apontando a câmera do seu celular para o código QR ao lado. Todo dispositivo smartphone ou tablet atualizado já possui esse recurso de leitura de códigos habilitado por padrão.

Módulo 1 do curso: https://www.youtube.com/playlist?list=PLHz_AreHm4dkZ9-atkcmcBaMZdmLHft8n

Teste seus conhecimentos

Terminou de ler esse capítulo e já acompanhou todos os vídeos e referências externas que indicamos? Pois agora, responda às seguintes perguntas objetivas e marque em cada uma delas a única opção verdadeira. Aí sim, você vai poder comprovar que realmente entendeu o conteúdo.



1. A versão anterior da linguagem, a HTML4 era bastante focada no(a) _____ do conteúdo. Já a versão HTML5 já fica mais voltada para o(a) _____ da estrutura desse conteúdo.

- [A] ordenação/ significado
- [B] forma / significado
- [C] ordenação / forma
- [D] forma / ordenação

2. A HTML5 chegou com a proposta de definir as chamadas _____, onde cada instrução passa a ter um significado, não apenas uma forma.

- [A] tags obsoletas
- [B] tags mecânicas
- [C] tags objetivas
- [D] tags semânticas

3. Na lista a seguir, qual é a única tag que é focada em significado e não em forma?

- [A]
- [B] <blink>
- [C]
- [D] <applet>

4. Qual das tags abaixo é a única que ainda permanece ativa? (todas as demais já estão na lista oficial de elementos obsoletos da HTML, disponível no link oficial da W3C)

- [A] <big>
- [B] <center>
- [C] <aside>
- [D] <tt>

5. No lugar da tag para negrito e <i> para itálico, é recomendável usar em HTML5 as novas tags:

- [A] e
- [B] <bold> e <italic>
- [C] e <italic>
- [D] <bold> e

6. Em relação ao tamanho das letras de um texto e o uso das tags <big> e <small> em documentos HTML5, assinale a única afirmativa correta:

- [A] a tag <big> ainda pode ser usada, mas a <small> não pode mais
- [B] todas as duas tags deixaram de existir
- [C] todas as duas tags seguem funcionando corretamente
- [D] a tag <big> deixou de existir e a tag <small> segue funcionando

7. Quando for para sublinhar um texto no sentido de chamar a atenção para ele, deixamos de usar o antigo <u> e passamos a usar o novo:

- [A] <underline>
- [B] <ins>
- [C] <sub>
- [D] <strike>

8. Para criar um texto sobreescrito (como em x^2) usamos a tag _____ e para criar um texto subscrito (como em H_2O), usamos a tag _____.

- [A] <sub> / <sup>
- [B] <up> / <down>
- [C] <down> / <up>
- [D] <sup> / <sub>

9. Em HTML5, podemos criar citações utilizando a tag _____, incluindo o parâmetro _____ para indicar o link para o conteúdo original da citação.

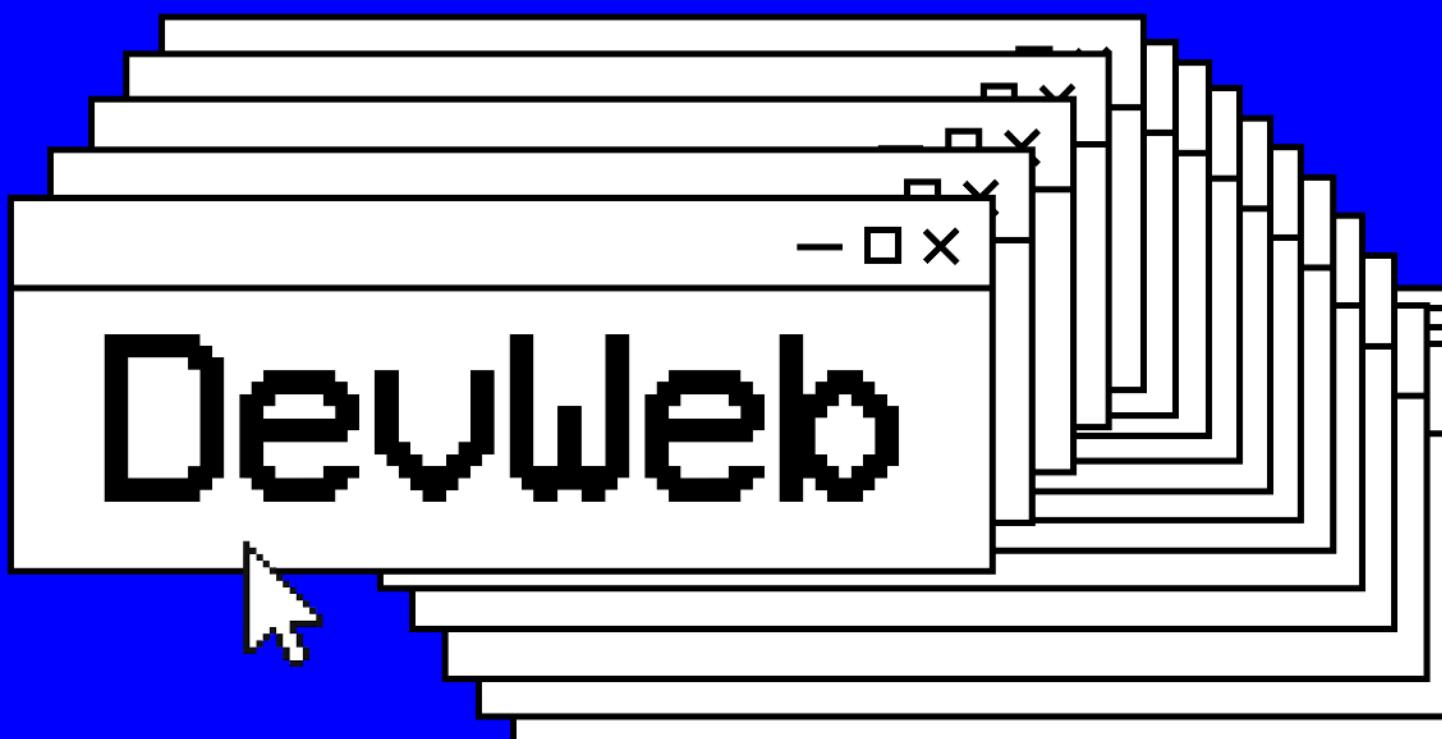
- [A] <cite> / src
- [B] <blockquote> / cite
- [C] <quote> / src
- [D] <cite> / quote

10. Para inverter uma palavra ou frase, devemos usar a tag _____ com o parâmetro dir configurado para o valor _____.

- [A] <bdo> / rtl
- [B] <bdo> / ltr
- [C] <direction> / rtl
- [D] <direction> / ltr

Suas anotações

Não guarde conhecimento. Ele é livre. Compartilhe o seu e veja ele se espalhando pelo mundo 



LISTAS COM HTML5

M01C09

LISTAS COM HTML5

veja em vídeo



- ★ Listas são importantes
- ★ Facilitam a leitura
- ★ Organizam o conteúdo
- ★ Simplificam a apresentação dos itens
- ★ Viu como é simples?
- ★ Quer aprender como se faz?
- ★ Vem comigo!



Você tem todo o direito de usar esse material para seu próprio aprendizado. Professores também podem ter acesso a todo o conteúdo e usá-lo com seus alunos. Porém todos que usarem esse material - seja para qual for a finalidade - deverão manter a referência ao material original, criado pelo **Prof. Gustavo Guanabara** e disponível no endereço do seu repositório público <https://github.com/gustavoguanabara/>. Este material não poderá ser utilizado em nenhuma hipótese para ser replicada - integral ou parcialmente - por autores/editoras para criar livros ou apostilas, com finalidade de obter ganho financeiro com ele.

O poder das listas

Pare para pensar por um só minuto: quantas listas você faz na sua vida? Eu, sinceramente não vivo sem listas: a lista do supermercado, lista de tarefas, a lista de chamada dos meus alunos, listas dos conteúdos a trabalhar cada semestre, lista de cursos que quero fazer, lista de cursos que quero produzir. Acho que já deu pra entender, não é?

Para quem produz conteúdo, listas também são muito úteis. Elas simplificam e sintetizam o conteúdo a ser passado e faz a demarcação de cada item para facilitar a visualização.



Por conta disso, a linguagem HTML disponibiliza vários tipos de lista para produzirmos nosso conteúdo e deixá-lo mais claro e eficiente.

Listas Ordenadas

A HTML chama de **ordered lists** todas aquelas listas onde a ordem dos itens é algo muito importante. Um passo-a-passo para criar um bolo, uma lista de aprovados no vestibular e uma lista com os carros mais caros do mundo são exemplos de listas ordenadas.

Para criar uma *ordered list*, vamos usar a tag `` para delimitar a lista e `` (*list item*) para identificar cada item da lista.

```
<ol>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
  <li>Item 4</li>
  <li>Item 5</li>
</ol>
```



IMPORTANTE! Segundo a **W3C**, a tag `` é de fechamento obrigatório (ou seja, devemos sempre usar ``). Já a tag `` tem seu fechamento **opcional** a partir da HTML5.

A horizontal banner with a purple background. On the left, there's a cartoon character holding an apple. In the center, the text "Soluções digitais para negócios" is written in large, white, sans-serif font. At the bottom left is the Hostnet logo, which consists of a blue cloud icon followed by the word "hostnet". On the right side, there's a QR code.

A tag `` possui um parâmetro `type`, onde configuramos o tipo de marcador da lista atual. As opções de valores para esse parâmetro são:

- ▶ 1 - Valor padrão. Cria listas numeradas. Ex: 1, 2, 3, 4, ...
- ▶ A - Cria listas alfabéticas em maiúsculas. Ex: A, B, C, D, ...
- ▶ a - Cria listas alfabéticas em minúsculas. Ex: a, b, c, d, ...
- ▶ I - Cria listas com algarismos romanos em maiúsculas. Ex: I, II, III, IV, ...
- ▶ i - Cria listas com algarismos romanos em minúsculas. Ex: i, ii, iii, iv, ...

Você também pode indicar o início da contagem usando o parâmetro `start`.

Por exemplo, a tag `<ol type="I" start = "5">` vai gerar itens numerados como V, VI, VII, VIII, IX, ...

Listas não Ordenadas

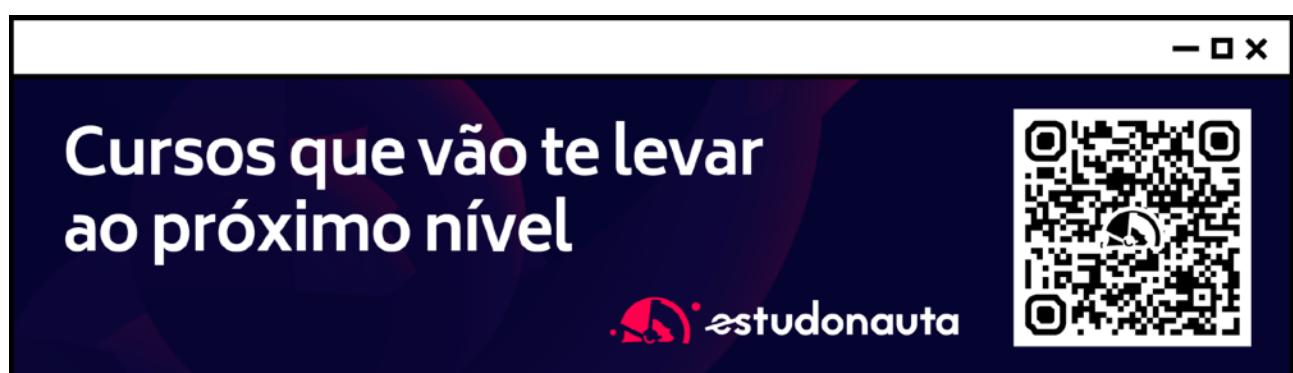
Se você compreendeu a criação de listas ordenadas, com certeza vai entender as ***unordered lists***, também chamadas de listas com marcadores, que são aquelas onde a ordem dos itens não influenciará no significado da lista. Ela é apenas uma ótima maneira para organizar os itens que não apresentam uma classificação necessariamente.

Para criar uma *unordered list*, vamos usar a tag `` para delimitar a lista e a tag `` para criar cada um dos seus itens internos.

```
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
  <li>Item 4</li>
  <li>Item 5</li>
</ul>
```

O marcador padrão é a bolinha preta totalmente preenchida (circle), mas existe a opção de configurar a propriedade `type` da tag `` com os seguintes valores:

- ▶ disc - padrão. Uma bola preta totalmente pintada
- ▶ circle - Uma bola com uma borda preta e sem preenchimento
- ▶ square - Um pequeno quadrado preto totalmente pintado



The screenshot shows a dark-themed web page with a QR code in the bottom right corner. The main text on the page reads: "Cursos que vão te levar ao próximo nível". Below this text is the logo for "estudonauta", which features a stylized figure and the word "estudonauta" in lowercase.

Misturando as coisas

Podemos também criar listas mistas, configurando listas dentro de outras listas. Veja o exemplo a seguir:

```
<ol>
  <li>Item 1</li>
  <li>Item 2</li>
  <ol type="a">
    <li>Item 2.1</li>
    <li>Item 2.2</li>
    <li>Item 2.3</li>
  </ol>
  <li>Item 3</li>
  <li>Item 4</li>
  <li>Item 5</li>
</ol>
```

O resultado visual do código que vimos anteriormente será semelhante ao que temos a seguir:

1. Item 1
2. Item 2
 - a. Item 2.1
 - b. Item 2.2
 - c. Item 2.3
3. Item 3
4. Item 4
5. Item 5

Note que os itens da lista interna (a, b, c, ...) está deslocado para a direita em relação ao item 2, do qual essa sub-lista é filha.



DICA: Além de aninhar listas ordenadas, podemos juntar listas `` com `` e vice-versa. As listas internas sempre terão deslocamento interno para a direita.

Exercício proposto

Crie seu próprio código e faça a seguinte lista aninhada com seus jogos favoritos de acordo com a plataforma.

Lista dos meus jogos favoritos

- Nintendinho
 - 1. Mario Bros
 - Super Mario Bros 3
 - Mario: Lost Levels
 - 2. Ninja Gaiden
 - Ninja Gaiden III
- Super Nintendo
 - 1. Super Mario World
 - 2. Donkey Kong
 - Donkey Kong Country III
 - Diddy's Kong Quest
- Playstation 1
 - 1. Final Fantasy
 - Final Fantasy VII
 - Final Fantasy IX
 - 2. Castlevania
 - Symphony of the Night

Tenha paciência e dedique-se, com certeza você vai conseguir fazer uma lista como essa aí em cima e vai aprender muito sobre HTML.

Lista de Definições

É como se fosse um dicionário, temos os termos e as suas descrições. É uma lista sem demarcadores, mas bem útil em alguns casos.

Toda lista de definições está dentro de uma tag `<dl> </dl>` (*definition list*). Cada termo é um `<dt>` (*definition term*) e cada descrição é um `<dd>` (*definition description*). Assim como os itens da lista, essas duas últimas tags possuem fechamento opcional, segundo a referência oficial da HTML5.

Vamos ver um exemplo simples que cria uma lista com três definições que já conhecemos bem aqui pelo curso.

```
<dl>
  <dt>HTML</dt>
  <dd>Linguagem de marcação utilizada para criar o conteúdo de sites.</dd>

  <dt>CSS</dt>
  <dd>Linguagem de marcação para a especificação de estilos em sites.</dd>

  <dt>JavaScript</dt>
  <dd>Linguagem de programação para criar interatividades em sites.</dd>
</dl>
```

O resultado visual dessa lista aplicado pelo navegador é:

HTML

Linguagem de marcação utilizada para criar o conteúdo de sites.

CSS

Linguagem de marcação para a especificação de estilos em sites.

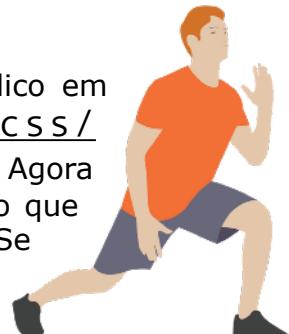
JavaScript

Linguagem de programação para criar interatividades em sites.

Cada navegador pode mostrar um resultado ligeiramente diferente para listas de definição, mas o que mais importa é o significado desse tipo de listagem, pois ele relaciona diretamente os pares Termo + Descrição e isso nos ajuda bastante com mecanismos de busca.

Hora de exercitar

Chegou a hora de acessar o endereço do nosso repositório público em <https://gustavoguanabara.github.io/html-css/exercicios/> e executar o **exercício 009** no seu computador. Agora tente atingir esse mesmo resultado em casa, sem copiar o código que eu criei. Nesse momento, a prática é algo que você mais precisa. Se por acaso ficar difícil, pode acessar o repositório público de HTML e CSS e dar uma olhada nos comandos, mas **EVITE COPIAR**.



Quer acompanhar tudo em vídeo?

Eu sei que às vezes as pessoas gostam mais de assistir vídeos do que ler livros, e é por isso que eu lanço há anos materiais no canal Curso em Vídeo no YouTube. O link que vou compartilhar contigo faz parte da playlist completa onde você encontra o **Módulo 1 do Curso de HTML5 e CSS3**, completamente gravado com base nesse material.



veja em vídeo



Além de acessar o link a seguir, você também pode ter acesso às aulas apontando a câmera do seu celular para o código QR ao lado. Todo dispositivo smartphone ou tablet atualizado já possui esse recurso de leitura de códigos habilitado por padrão.

Módulo 1 do curso: https://www.youtube.com/playlist?list=PLHz_AreHm4dkZ9-atkcmcBaMZdmLHft8n

Teste seus conhecimentos

Terminou de ler esse capítulo e já acompanhou todos os vídeos e referências externas que indicamos? Pois agora, responda a essas 10 perguntas objetivas e marque em cada uma delas a única opção verdadeira. Aí sim, você vai poder comprovar que realmente entendeu o conteúdo.



1. Em HTML5, temos vários tipos de lista. Qual dos itens a seguir não é um tipo de lista suportado pela linguagem?

- A lista ordenada
- B lista não ordenada
- C lista semi ordenada
- D lista de definição

2. Na construção das listas ordenadas e nas não ordenadas, o uso de uma das tags se repete nos dois casos. Que tag é essa?

- A
- B
- C
- D <dl>

3. Para criar uma lista ordenada, devemos limitar todos os seus itens dentro de um par único de tags:

- A e
- B e
- C <dl> e </dl>
- D <dt> e </dt>

4. Segundo as novas normas da W3C, qual das tags de lista a seguir é a única que possui fechamento opcional?

- A <dt>
- B <dd>
- C
- D

5. Em listas ordenadas, podemos usar o parâmetro _____ para modificar o formato da contagem e o parâmetro _____ para mudar o número inicial da contagem.

- A type / init
- B style / start
- C type / start
- D style / init

6. Ao personalizar a numeração de uma lista ordenada, podemos indicar vários tipos de contagem. Qual das opções a seguir é a única que não pode ser usada para essa personalização?

- A 1
- B A
- C I
- D X

Cursos grátis de tecnologia
que te preparam para o
mercado de trabalho

RECODE



7. Podemos personalizar as listas não ordenadas, configurando o parâmetro `type` da tag `` com alguns valores especiais. Qual dos elementos a seguir é o único aceito para modificar o marcador padrão dos itens?

- A star
- B square
- C rectangle
- D triangle

8. Todos os elementos de uma lista de definições devem estar limitados entre o par de tags:

- A `<dt> e </dt>`
- B `<dd> e </dd>`
- C `<dl> e </dl>`
- D `<dc> e </dc>`

9. A tag `<dt>` em uma lista de definição tem o significado de:

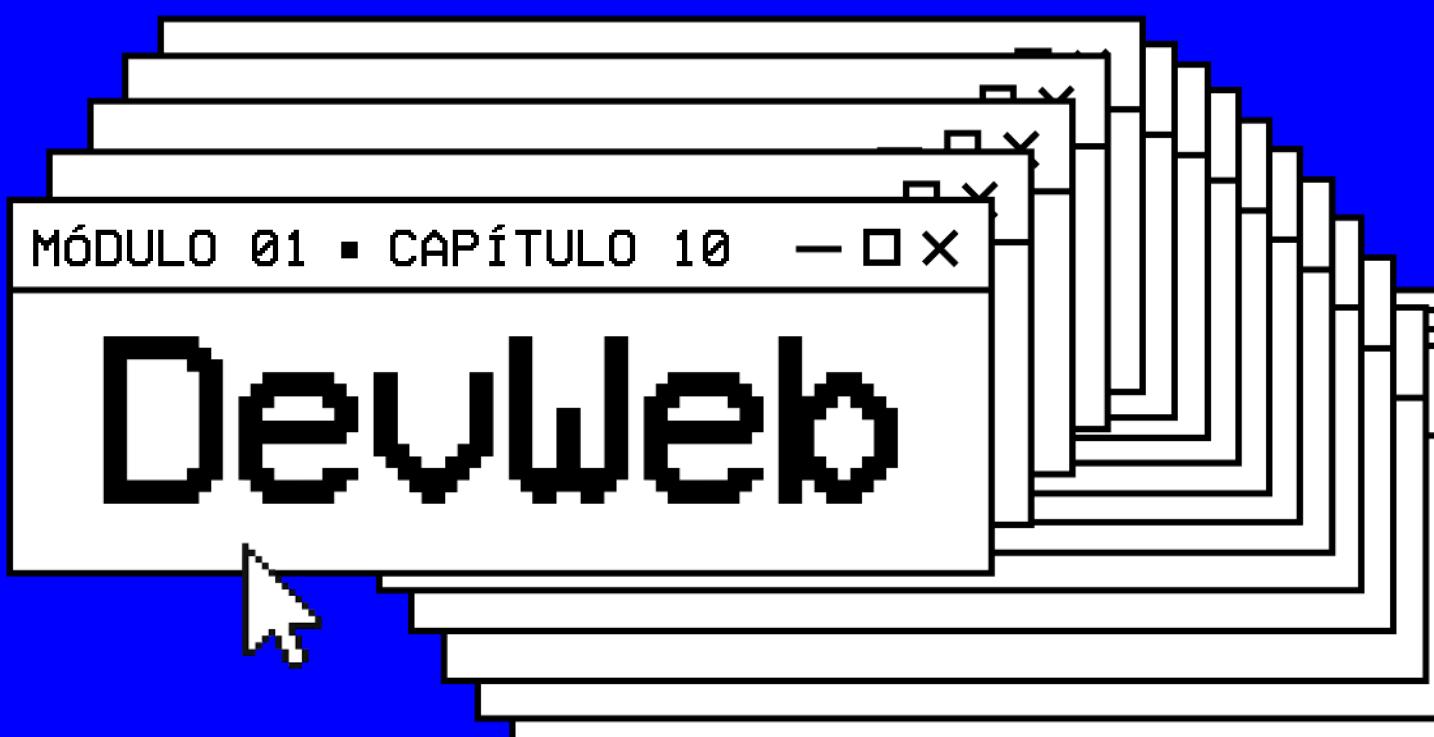
- A definition term
- B description text
- C definition text
- D description term

10. A tag `<dd>` em uma lista de definição tem o significado de:

- A description definition
- B definition description
- C definition detail
- D detail definition

Suas anotações

Não guarde conhecimento. Ele é livre. Compartilhe o seu e veja ele se espalhando pelo mundo 



LIGAÇÕES EM TODA PARTE



M01C10

LIGAÇÕES EM TODA PARTE

veja em vídeo



O link ou ligação é a essência fundamental do hipertexto. Ela nos permite ligar documentos entre si e permitir a navegação entre essas páginas. Nesse capítulo vamos aprender a configurar as âncoras para vários comportamentos, desde o mais simples que é criar um link interno até ligações que habilitam downloads de arquivos.



Você tem todo o direito de usar esse material para seu próprio aprendizado. Professores também podem ter acesso a todo o conteúdo e usá-lo com seus alunos. Porém todos que usarem esse material - seja para qual for a finalidade - deverão manter a referência ao material original, criado pelo **Prof. Gustavo Guanabara** e disponível no endereço do seu repositório público <https://github.com/gustavoguanabara/>. Este material não poderá ser utilizado em nenhuma hipótese para ser replicada - integral ou parcialmente - por autores/editoras para criar livros ou apostilas, com finalidade de obter ganho financeiro com ele.

Joga a âncora, marujo

Os *hyperlinks* são um dos conceitos mais antigos da história da linguagem HTML. Eles permitem que você ligue um ponto a outro na *World Wide Web*. Toda vez que você está acessando um site e clica em um local para ir para outra página, outro site ou até para baixar um arquivo, você está interagindo com um *hyperlink*.



Até os mecanismos de busca se utilizam dos hyperlinks de um site. O **Google**, por exemplo, para achar um determinado site, fica vasculhando constantemente todos os outros sites da Internet procurando por links para descobrir novos conteúdos. Por isso é tão importante conseguir links válidos de outros sites para o nosso próprio site.



APRENDA MAIS: Veja com mais detalhes como funcionam os algoritmos fundamentais de busca assistindo esse vídeo do próprio **Google**, onde o engenheiro **Matt Cutts** explica o mecanismo básico da ferramenta mais valiosa do mundo. Habilite as legendas em PT-BR.



Google: <https://youtu.be/BNHR6IQJGZs>

Para criar um hyperlink, devemos criar **âncoras** através da tag `<a>`. O principal atributo dessa tag é o `href`, que cria uma referência hipertexto. Vamos ver um exemplo simples:

```
<h1>Vamos criar um link</h1>
<a href="https://gustavoguanabara.github.io">Acesse meu perfil GitHub</a>
```

Note que dentro do atributo `href`, o que colocamos foi uma **URL** completa para outro site.



NÃO ENTENDEU? Você sabe o que é uma URL? Pois volte para o nosso **capítulo 02** e vá até a parte onde falamos de domínios e hospedagem. Lá explicamos melhor o que são *Uniform Resource Locators* e quais são os seus principais componentes.

Cursos grátis de tecnologia
que te preparam para o
mercado de trabalho

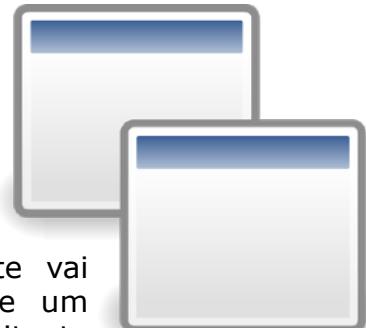
RECODE



Outro atributo bem útil da tag de âncora é o hreflang, que permite indicar qual é o idioma principal do site para onde o link está desviando o fluxo de navegação. Isso vai permitir avisar ao navegador e a softwares de tradução como lidar caso o visitante opte por traduzir automaticamente os conteúdos.

```
<a href="https://www.w3schools.com/html/" hreflang="en">  
| Site da W3Schools (em Inglês)  
</a>
```

Mira no Alvo



Por padrão, sempre que um visitante clique em um *hyperlink*, o site de destino abre na mesma janela do site que continha esse link. Ou seja, o conteúdo anterior vai deixar de ser exibido para mostrar o novo conteúdo.

Esse é um comportamento desejado quando o visitante vai continuar a visitar o nosso site, apenas mudando de um documento para outro. Mas e quando um clique leva o visitante para outro site e provavelmente ele nunca mais voltará ao nosso?

Para poder controlar onde o site de destino vai abrir, podemos usar o atributo target, que suporta os seguintes valores:

- _blank vai abrir o link em uma nova janela em branco
- _self vai abrir o link na janela ou frame atual (padrão)
- _top vai desfazer todos os frames e abrir o destino no navegador completo
- _parent similar ao uso do _top em uma referência à janela mãe
- nome-do-frame caso esteja usando frames, indicar o nome da janela a abrir

Como o uso de frames é uma técnica quase em desuso, vamos nos basear apenas nas duas primeiras opções _blank e _self.

```
<a href="pagina2.html" target="_self">  
| Continuar navegando no site  
</a>  
  
<a href="https://gustavoguanabara.github.com" target="_blank">  
| Abrir perfil GitHub em nova janela  
</a>
```

Esse link é seu ou dos outros?



Existe um recurso bem interessante para links que é indicar qual é a natureza do destino usando o atributo rel. Esse atributo aceita vários valores, entre eles vou citar:

- next indica que o link é para a próxima parte do documento atual
- prev indica que o link é para a parte anterior do documento atual
- author indica que é um link para o site do autor do artigo atual
- external indica que é um link para outro site que não faz parte do site

atual

- ▶ nofollow indica que é um link para um site não endossado, como um link pago

```
<a href="pagina2.html" target="_self" rel="next">  
| Continuar navegando no site  
</a>
```

```
<a href="https://gustavoguanabara.github.com" target="_blank" rel="external">  
| Abrir perfil GitHub em nova janela  
</a>
```

No código anterior, o primeiro link é o que chamamos de **link local** ou **link interno**, já que ele leva o visitante a outra página dentro do nosso próprio site. Note que não é necessário nem indicar a URL completa nesses casos.

Já o segundo link vai nos levar para um outro site, o que chamamos de **link externo**. Nestes casos, devemos indicar a URL completa, incluindo o protocolo http:// ou https:// e o caminho que leve à uma página específica, se for necessário.



NAVEGANDO POR PASTAS LOCAIS: Não sei se você conhece um pouco do mundo **Linux**, mas servidores Web normalmente rodam esse sistema operacional. Se estiver se referindo à pasta atual do servidor, pode usar ./ antes do nome do arquivo. Se quiser se referir à pasta imediatamente superior na hierarquia, use ../ para voltar um nível para a pasta mais externa. E por favor, **aprenda Linux** !

E para fazer Downloads?

Outra coisa que aparece bastante em sites são os links para efetuar download de algum material em PDF, ou de um arquivo ZIP qualquer. A partir da versão HTML5, as âncoras receberam atributos especiais para isso. Basta fazer o link diretamente para o arquivo que se deseja efetuar o download e adicionar o atributo download com o valor configurado para o nome do arquivo a ser baixado e o atributo type para indicar ao navegador que tipo de arquivo está sendo baixado. Vamos ver um exemplo:



A screenshot of a website with a dark purple background. On the left, white text reads "Cursos que vão te levar ao próximo nível". At the bottom, there is a logo for "estudonauta" featuring a stylized figure and the word "estudonauta" in white. To the right of the logo is a large QR code.

```
<a href="arquivos/meulivro.pdf" download="meulivro.pdf" type="application/pdf">  
| Baixe aqui o PDF do meu livro  
</a>
```

Aqui vão alguns *media types* bem usados no nosso dia-a-dia:

- ▶ application/zip
- ▶ video/mp4
- ▶ audio/mpeg
- ▶ text/html
- ▶ video/H264
- ▶ font/ttf
- ▶ text/css
- ▶ video/JPEG
- ▶ image/jpeg
- ▶ text/javascript
- ▶ audio/aac
- ▶ image/png

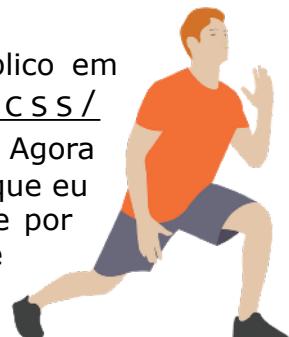


LISTA DE MEDIA TYPES: Se você quer saber o que escrever dentro do atributo type de uma âncora de hypertext, consulte a lista oficial da IANA.org disponível no link abaixo.

<https://www.iana.org/assignments/media-types/media-types.xhtml>

Hora de exercitar

Chegou a hora de acessar o endereço do nosso repositório público em <https://gustavoguanabara.github.io/html-css/exercicios/> e executar o **exercício 010** no seu computador. Agora tente atingir esse mesmo resultado em casa, sem copiar o código que eu criei. Nesse momento, a prática é algo que você mais precisa. Se por acaso ficar difícil, pode acessar o repositório público de HTML e CSS e dar uma olhada nos comandos, mas **EVITE COPIAR**.



Tenho desafios pra você!

Lá no repositório, além do material em PDF e dos códigos dos exercícios 100% disponíveis, também disponibilizamos alguns **desafios** que devem ser resolvidos. Esses desafios não incluem o código original e você deve tentar chegar à resposta sem copiar nenhum código.

Com todo o conteúdo que vimos até essa aula, você já pode resolver o **desafio d005** e o **desafio006**. Acesse o repositório público, abra a área do curso de HTML+CSS e clique no link de acesso aos desafios. Manda ver! Só não fica pedindo a resposta! Você consegue resolver isso sozinho(a)!



Repositório em: <https://gustavoguanabara.github.io>

Quer acompanhar tudo em vídeo?

Eu sei que às vezes as pessoas gostam mais de assistir vídeos do que ler livros, e é por isso que eu lanço há anos materiais no canal Curso em Vídeo no YouTube. O link que vou compartilhar contigo faz parte da playlist completa onde você encontra o **Módulo 1 do Curso de HTML5 e CSS3**, completamente gravado com base nesse material.



Além de acessar o link a seguir, você também pode ter acesso às aulas apontando a câmera do seu celular para o código QR ao lado. Todo dispositivo smartphone ou tablet atualizado já possui esse recurso de leitura de códigos habilitado por padrão.

Módulo 1 do curso: https://www.youtube.com/playlist?list=PLHz_AreHm4dkZ9-atkcmcBaMZdmLHft8n

Teste seus conhecimentos

Terminou de ler esse capítulo e já acompanhou todos os vídeos e referências externas que indicamos? Pois agora, responda a essas 10 perguntas objetivas e marque em cada uma delas a única opção verdadeira. Aí sim, você vai poder comprovar que realmente entendeu o conteúdo.



1. Toda vez que estamos em um site e clicamos em uma área sensível que nos leva a outro ponto ou a outro documento, estamos interagindo com um:

- A hyperlink
- B metadado
- C ancoramento
- D subdomínio

2. Para criarmos links em nosso site, utilizaremos a tag:

- A <link>
- B <l>
- C <lnk>
- D <a>

3. Dentro da tag de âncora, o parâmetro _____ serve para indicar a URL completa para onde o fluxo será desviado.

- A destiny
- B ref
- C href
- D link

4. Também é possível indicar o idioma principal de um documento configurado em uma âncora usando o parâmetro:

- A lang
- B reflang
- C rellang
- D hreflang

5. A tag de âncora também suporta o parâmetro target, onde podemos colocar vários valores, exceto:

- A _parent
- B _self
- C _external
- D _top

6. A maneira mais recomendada de indicar um link externo é configurando o seguinte par de parâmetro e valor:

- A external = "true"
- B href = "external"
- C target = "_external"
- D rel = "external"

7. Ao criar links que levam a documentos que estão em pastas imediatamente superiores na hierarquia do site, devemos adicionar ao link os símbolos:

- A ../
- B < /
- C # /
- D - /

8. Ao criar um link para download, recomenda-se configurar os parâmetros adicionais:

- A link e download
- B download e type
- C link e type
- D download e link

9. O parâmetro type de um link que vai servir para baixar um arquivo PDF deve estar configurado com o valor:

- A ebook/pdf
- B file/pdf
- C application/pdf
- D download/pdf

10. Os valores application/zip, audio/aac e font/ttf são exemplos de:

- A type files
- B media types
- C multimedia patterns
- D format files

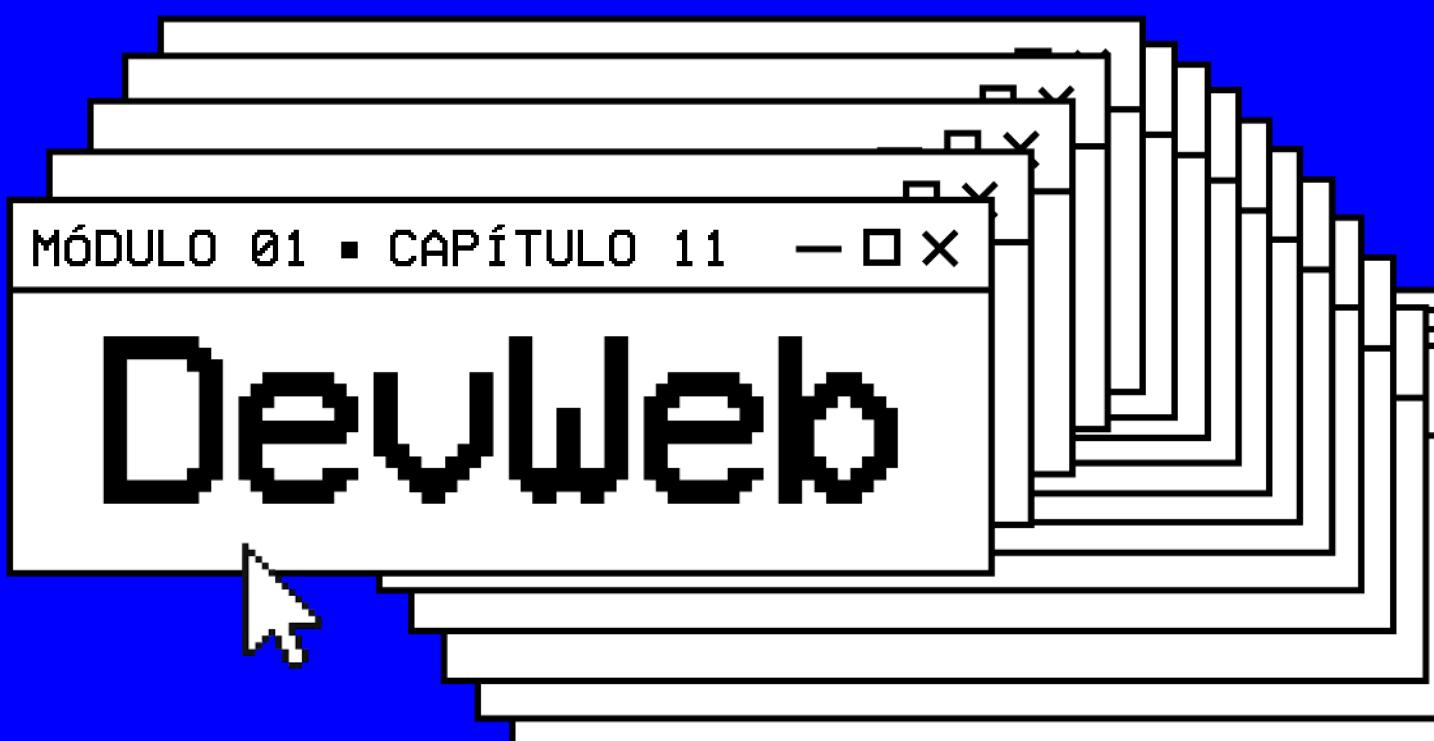


Soluções digitais
para negócios

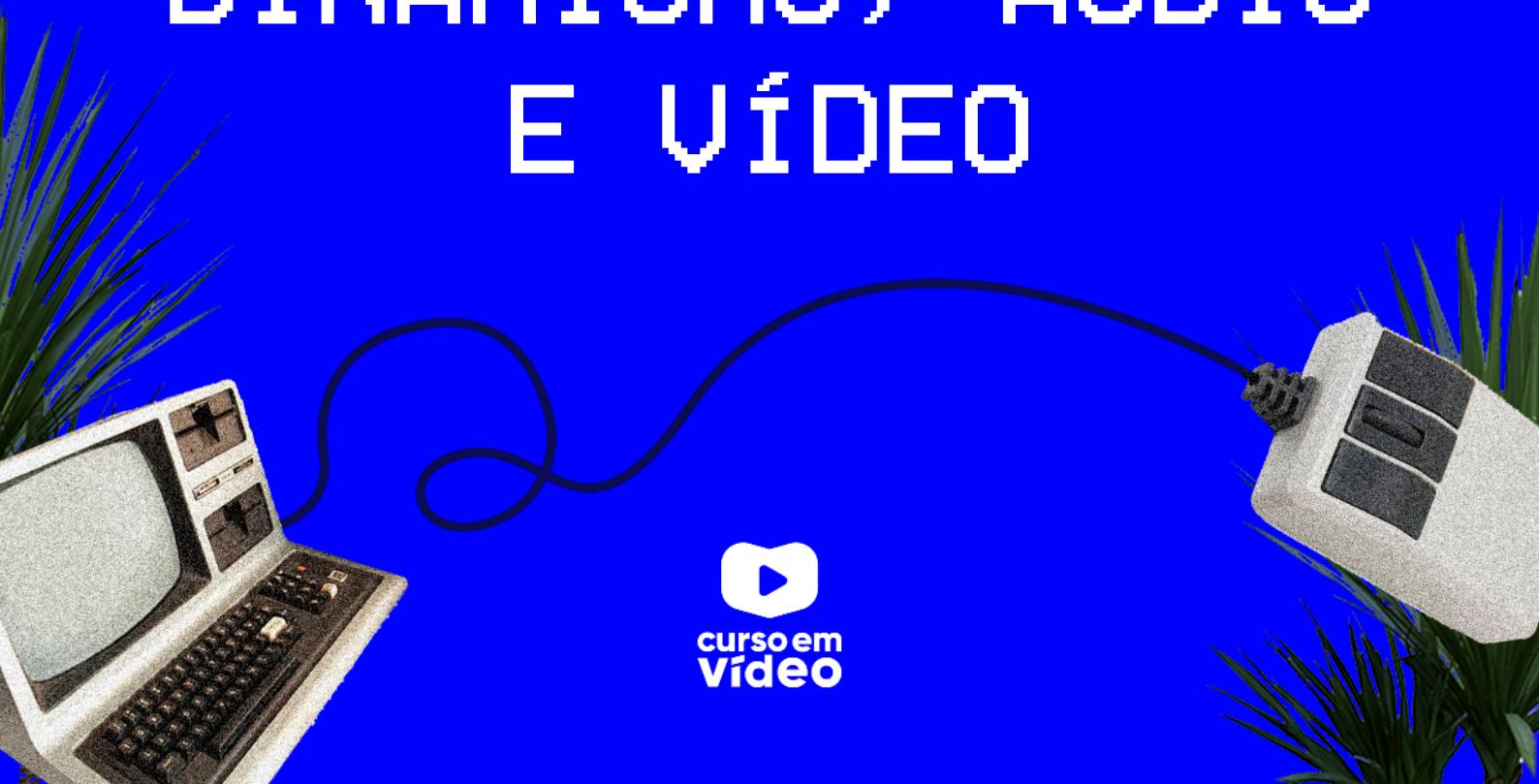


Suas anotações

Não guarde conhecimento. Ele é livre. Compartilhe o seu e veja ele se espalhando pelo mundo



IMAGENS DINÂMICAS, ÁUDIO E VÍDEO



M01C11

veja em vídeo



IMAGENS DINÂMICAS, ÁUDIO E VÍDEO

Fotos, áudios e vídeos são essenciais para a construção dos sites hoje em dia. Ilustrar bem o seu conteúdo é imprescindível para deixar clara a mensagem que queremos passar. Mas é preciso tomar cuidado com formatos e tamanhos, pois muita gente abrirá os sites em seus celulares e dispositivos móveis. Vamos falar sobre esse assunto no capítulo que iniciamos agora.



Você tem todo o direito de usar esse material para seu próprio aprendizado. Professores também podem ter acesso a todo o conteúdo e usá-lo com seus alunos. Porém todos que usarem esse material - seja para qual for a finalidade - deverão manter a referência ao material original, criado pelo **Prof. Gustavo Guanabara** e disponível no endereço do seu repositório público <https://github.com/gustavoguanabara/>. Este material não poderá ser utilizado em nenhuma hipótese para ser replicada - integral ou parcialmente - por autores/editoras para criar livros ou apostilas, com finalidade de obter ganho financeiro com ele.

Seu site deve se adaptar ao tamanho da tela

Como eu disse ali na introdução do capítulo, atualmente podemos acessar sites em todo tipo de dispositivo: smartphones, tablets, notebooks, computadores desktop, aparelhos de smart TV e muito mais. Pensando nisso, você que está começando a desenvolver sites vai precisar adaptar seu conteúdo a tamanhos de tela diferentes.

A mídia que mais sofre com esse problema de tamanho de exibição são as imagens. Uma TV pode exibir imagens gigantes, mas se usarmos essa mesma foto para ser apresentada em um celular, teremos que redimensionar forçadamente a imagem com CSS.



Porém, essa prática não vai fazer com que o tamanho (em bytes) diminua também. Isso acaba aumentando o consumo de dados em dispositivos móveis e deixando seu site muito pesado, e ninguém tem paciência para acessar site lento.



CUIDADO! Sites lentos diminuem a **taxa de retenção** dos usuários, que ficam menos tempo acessando e podem prejudicar a indexação da sua página em mecanismos de busca como o **Google**. Veja no vídeo a seguir, um especialista em SEO (otimização para mecanismos de busca) falando sobre lentidão de sites, principalmente via 3G e 4G do celular.

Portal SEO: <https://youtu.be/jWnMfvSdo1E>

The advertisement features a dark purple background with a large white QR code on the right. To the left of the QR code, the text "Cursos que vão te levar ao próximo nível" is displayed in a large, bold, white font. Below this text is the logo for "estudonauta", which includes a small icon of a person with a backpack and the word "estudonauta" in a white, sans-serif font.

Imagens Flexíveis



Nosso primeiro passo no caminho de adaptar nosso conteúdo ao tamanho da tela vai ser aprender a gerar imagens de tamanho diferentes e a fazer o navegador carregar a imagem certa para cada situação. Para isso, devemos conhecer as tags `<picture>` e `<source>`.



Para esse exemplo, criamos as três imagens ao lado: a menor tem 300x300px, a média tem 700x700px e a maior tem 1000x1000px. Usamos o programa **GIMP**, que é um editor de imagens 100% gratuito. Essas imagens serão carregadas pelo navegador de acordo com o tamanho da janela atual. Para isso, criamos o seguinte código base:

```
<picture>
  
</picture>
```



Note que colocamos a tag `` exatamente como aprendemos no **capítulo 06** do nosso material. A novidade aqui é que inserimos essa imagem dentro da tag `<picture>`, que vai concentrar as outras fontes de imagem. Por padrão, a imagem **foto-g.png** (1000x1000px) será carregada.

O problema vai começar a surgir quando a janela do navegador chegar perto dos 1000 pixels de largura, pois a foto não vai mais caber lá. Vamos agora adicionar uma linha para resolver esse problema:

```
<picture>
  <source media="(max-width: 1050px)" srcset="foto-m.png" type="image/png">
  
</picture>
```

Note que a tag `<source>` possui três atributos:

- `type` vai indicar o *media type* da imagem que usamos (veja mais informações sobre *media types* no capítulo 10)
- `srcset` vai configurar o nome da imagem que será carregada quando o tamanho indicado for atingido
- `media` indica o tamanho máximo a ser considerado para carregar a imagem indicada no atributo `srcset`.



ATENÇÃO! Você pode até colocar o valor exato de 1000px na propriedade `max-width`, mas vai perceber que um valor ligeiramente acima vai gerar resultados mais interessantes.

Agora, recarregue seu código e mude o tamanho da janela do navegador. Você vai perceber que a imagem muda automaticamente conforme aumentamos ou diminuímos o tamanho da tela.

Vamos continuar e acrescentar mais um `<source>` à nossa imagem:

```
<picture>
  <source media="(max-width: 750px)" srcset="foto-p.png" type="image/png">
  <source media="(max-width: 1050px)" srcset="foto-m.png" type="image/png">
  
</picture>
```

É importante que existe uma ordem entre os `<source>`, e nessa nossa configuração, os itens mais acima sejam os menores tamanhos para `max-width` e que os seguintes sejam maiores, de forma crescente. O último item dentro de `<picture>` deve ser a imagem padrão.

Faça seus testes, modifique as ordens, entenda os resultados!

Vamos falar sobre áudio

Quando comecei a produzir conteúdo para Internet, os vídeos ainda não eram uma realidade palpável, pois a Internet brasileira era muito lenta e cara. Iniciativas como Videolog e YouTube estavam começando a surgir, mas ainda mostravam vídeos muito curtos e com qualidade sofrível.

E foi em 2006 que comecei a produzir um podcast sobre tecnologia que chegou a ser bem conhecido e ter um número significativo de downloads por cada episódio.



Me lembro bem de que na época era bem complicado colocar conteúdo em áudio em um site. Precisava da adição de bastante JavaScript, manipulação de plugins específicos e um conhecimento bem consistente em RSS. A HTML não me ajudava em nada. Mais tarde, com a evolução dos vídeos e a melhoria da banda larga nacional, acabei optando por focar meus esforços no formato vídeo, mas tenho um carinho especial pelo áudio até hoje.



QUER ME OUVIR? Eu também participei de dois episódios de um dos maiores podcasts do Brasil: o **Nerdcast**. Dei meus pitacos sobre as profissões de Professor e de Programador. Você pode acessar o site abaixo e conferir.

https://jovemnerd.com.br/?podcast_guest=gustavo-guanabara

Com a HTML5, veio também a facilidade em compartilhar áudio nos nossos sites e sem depender necessariamente de JavaScript ou plugins extras. A partir de agora, basta uma tag `<audio>` e alguns `<source>` para fazer seu site ser capaz de tocar qualquer áudio.

```

<audio preload="metadata" controls autoplay loop>
  <source src="midia/guanacast-33.mp3" type="audio/mpeg">
  <source src="midia/guanacast-33.ogg" type="audio/ogg">
  <source src="midia/guanacast-33.wav" type="image/wav">
  <p>Seu navegador não suporta áudio <a href="midia/guanacast-33.mp3"
  download="guanacast-33.mp3" type="audio/mpeg"></a></p>
</audio>

```

Vamos analisar os principais atributos da tag `<audio>` antes de mais nada:

- ▶ O atributo `preload` indica se o áudio será pré-carregado ou não e aceita três valores:
 - `metadata` vai carregar apenas as informações sobre o arquivo (tamanho, tempo, informações de direitos, etc)
 - `none` não vai carregar absolutamente nada até que o usuário clique no botão play ou um script inicie a reprodução
 - `auto` (padrão) vai carregar o arquivo de áudio inteiro assim que a página for carregada, mesmo que o usuário nunca aperte o play
- ▶ O atributo `controls` vai apresentar o player na tela. Caso não seja colocado na tag `<audio>`, o controle será transparente e o usuário não poderá interagir com ele.
- ▶ O atributo `autoplay`, quando inserido, vai iniciar a reprodução do áudio assim que a página for carregada.
- ▶ O atributo `loop` vai fazer com que o áudio seja repetido eternamente assim que terminar a sua reprodução.

Dentro da tag `<audio>`, adicionamos vários `<source>` com formatos diferentes do mesmo áudio. Coloque na parte de cima o seu formato favorito. Os demais só serão carregados caso o de cima falhe. Caso todos falhem, criamos um parágrafo que permite o download do arquivo MP3 para ouvir no player padrão do dispositivo.



GUANACAST: Eu deixei no GitHub um episódio do meu antigo podcast. Basta acessar a área de exercícios de HTML, ir até a pasta ex011 e baixar o arquivo MP3 que está dentro da pasta midia.



— □ ×

Cursos grátis de tecnologia que te preparam para o mercado de trabalho

RECODE



E os vídeos?

Com a expansão da banda larga no Brasil (sim, eu sei que ainda estamos longe de algo ideal na maioria do território nacional) tornou-se possível até assistir um vídeo em 4K em nosso plano 4G do celular (é caro, mas possível).



Para inserir um vídeo em nosso site, podemos utilizar a nova tag `<video>` da HTML5, caso o arquivo esteja hospedado no nosso próprio servidor.

```
<video width="600" poster="thumb.jpg" controls autoplay>
  <source src="meu-video.webm" type="video/webm">
  <source src="meu-video.mp4" type="video/mp4">
  <source src="meu-video.ogv" type="video/ogg">
  <p>Infelizmente seu navegador não conseguiu carregar o vídeo.</p>
</video>
```

Antes de mais nada, vamos criar a tag `<video>` e configurar alguns atributos importantes:

- `width` vai indicar a largura que o vídeo vai ter na tela. Nesse exemplo, 600px.
- `poster` configura uma imagem que vai aparecer como uma capa, enquanto o visitante não aperta o play para reproduzir o vídeo
- `controls` vai configurar se os controles do vídeo vão aparecer na parte inferior da mídia. Por padrão, os controles não aparecerão, mas basta colocar a palavra `controls` na tag `<video>`.
- `autoplay` diz para o navegador se o vídeo vai começar a tocar automaticamente, assim que a página for carregada.

Que formatos são esses?

Arquivos de vídeo não são tão simples quanto imagens e áudios, onde o formato indica o padrão para abrir e reproduzir a mídia. Vídeos possuem formatos e codecs e isso pode tornar o vídeo inviável de ser reproduzido pela maioria dos navegadores na maioria dos dispositivos. É preciso prestar muita atenção nesse pequeno detalhe.



Os formatos suportados são MPEG, WEBM e OGG, mas os dois primeiros são os que possuem maior compatibilidade com os navegadores atualmente.



UM ÓTIMO CONVERSOR: Para gerar arquivos em vários formatos e usando codecs padronizados, recomendo usar o programa open source chamado **Handbrake**, disponível para várias plataformas.

<https://handbrake.fr/downloads.php>

Navegador	Arquivos compatíveis
Microsoft Edge	.mp4 .m4v
Apple Safari	.mp4 .m4v
Google Chrome	.mp4 .m4v .webm .ogv
Mozilla Firefox	.webm .ogv
Opera	.webm .ogv

Hospedar seus próprios vídeos pode ser caro

Quando colocamos vídeos no nosso próprio servidor, podemos passar por problemas com alto consumo de banda, site lento e incompatibilidades com alguns navegadores por conta dos codecs. E geralmente só percebemos esses problemas quando colocamos nosso projeto no ar e lançamos oficialmente.

Vamos fazer uma conta simples: um vídeo simples, com poucos minutos, em formato mp4 com codec padrão deve ocupar uns 150MB com facilidade. Agora imagine que você lance seu site e 200 visitantes (um número super possível) acessem seu site e reproduzam o vídeo. Pronto, você acabou de utilizar 29GB de tráfego! Imagine o quanto isso pode deixar seu site lento ou consumir sua franquia de hospedagem, caso não seja ilimitada.



E é claro que seu site possivelmente não vai ter apenas um vídeo, não vai ter apenas 200 visitantes em um dia e aí a conta só aumenta. É preciso tomar muito cuidado quando decidimos guardar nossos próprios vídeos.

E agora, quem poderá nos defender?

Euuuuuu! Ou melhor, o **YouTube** ou o **Vimeo**! Esses são serviços para a hospedagem de vídeos que vai evitar consumir nossos próprios recursos de host contratado. Cada um tem suas vantagens e desvantagens:



O **YouTube** é o serviço de hospedagem de vídeos mais popular do mundo e é gerenciado pelo Google. Sua principal vantagem é que seus servidores são ultra rápidos. Por outro lado, a ideia do Google é deixar todos os vídeos públicos e disponíveis, o que pode ser uma dor de cabeça caso você queira limitar quem vai ter acesso a determinado vídeo (uma escola online, por exemplo, onde queremos que apenas os alunos matriculados possam assistir).

O **Vimeo** resolve o problema que apontamos anteriormente. Ele permite limitar quem vai poder ver o vídeo, o que é especialmente vantajoso para quem quer criar produtos em forma de vídeo, entregues por demanda dentro de um site personalizado. Como desvantagens desse serviço, ele é pago por uma taxa anual e seus algoritmos não são tão eficientes quanto os do YouTube, logo os vídeos apresentam pequenos travamentos às vezes.



E dá pra incorporar com esses aí em HTML?

Para incorporar vídeos que você subiu no **YouTube** ou **Vimeo**, existem recursos que te dão o código pronto em HTML5.

No **YouTube**, abra o vídeo que você quer incorporar e clique no link **COMPARTILHAR** que fica abaixo do título (veja imagem a seguir).



Ao clicar no link indicado anteriormente, uma janela vai aparecer, te dando as opções de compartilhamento. Escolha o item INCORPORAR.



O código HTML personalizado vai aparecer em uma nova janela de contexto, incluindo um botão que permite COPIAR o código com a tag do <iframe> que vai aparecer diretamente na sua página. Volte ao seu editor de código e cole a tag no seu arquivo HTML que vai apresentar o vídeo.

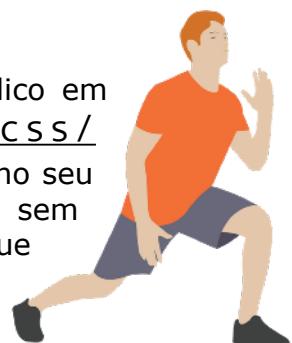
Quando você está usando o Vimeo, o procedimento é bastante semelhante. Abra seu painel de controle do serviço e vá para a sua lista de vídeos hospedados. Na lista de mídias armazenadas, clique no botão com reticências e escolha a opção **Incorporação** (veja imagem abaixo). Na janela, você vai poder clicar no botão Código de incorporação e a tag <iframe> personalizada também vai aparecer.

The screenshot shows the Vimeo dashboard with three video clips listed. The first clip is titled "M02A03 - W3TotalCache: configurações e teste de perfo..." and has a duration of 13:34. The second clip is titled "M02A03 - W3TotalCache" and has a duration of 0 16:10. The third clip is titled "M02A02 - Seu WordPress muito mais rápido" and has a duration of 0 16:10. Below the clips is a toolbar with buttons for "Ocultar", "Copiar link" (which is circled in red), and other options. Above the toolbar, a button labeled "</> Incorporação" is also highlighted with a red box.

The screenshot shows a video player displaying a video of two men. Above the video, the URL "https://vimeo.com/386829741" is shown. To its right is a button labeled "Código de incorporação". Below the video player is a large image of the video frame itself.

Hora de exercitar

Chegou a hora de acessar o endereço do nosso repositório público em <https://gustavoguanabara.github.io/html-css/exercicios/> e executar o **exercício 011** e o **exercício 012** no seu computador. Agora tente atingir esse mesmo resultado em casa, sem copiar o código que eu criei. Nesse momento, a prática é algo que você mais precisa. Se por acaso ficar difícil, pode acessar o repositório público de HTML e CSS e dar uma olhada nos comandos, mas **EVITE COPIAR**.



Tenho desafios pra você!

Lá no repositório, além do material em PDF e dos códigos dos exercícios 100% disponíveis, também disponibilizamos alguns **desafios** que devem ser resolvidos. Esses desafios não incluem o código original e você deve tentar chegar à resposta sem copiar nenhum código.

Com todo o conteúdo que vimos até essa aula, você já pode resolver o **desafio d007**, o **desafio008** e o **desafio009**. Acesse o repositório público, abra a área do curso de HTML+CSS e clique no link de acesso aos desafios. Manda ver! Só não fica pedindo a resposta! Você consegue resolver isso sozinho(a)!



Repositório em: <https://gustavoguanabara.github.io>

Quer acompanhar tudo em vídeo?

Eu sei que às vezes as pessoas gostam mais de assistir vídeos do que ler livros, e é por isso que eu lanço há anos materiais no canal Curso em Vídeo no YouTube. O link que vou compartilhar contigo faz parte da playlist completa onde você encontra o **Módulo 1 do Curso de HTML5 e CSS3**, completamente gravado com base nesse material.



Além de acessar o link a seguir, você também pode ter acesso às aulas apontando a câmera do seu celular para o código QR ao lado. Todo dispositivo smartphone ou tablet atualizado já possui esse recurso de leitura de códigos habilitado por padrão.

Módulo 1 do curso: https://www.youtube.com/playlist?list=PLHz_AreHm4dkZ9-atkcmcBaMZdmLHft8n

Teste seus conhecimentos

Terminou de ler esse capítulo e já acompanhou todos os vídeos e referências externas que indicamos? Pois agora, responda a essas 10 perguntas objetivas e marque em cada uma delas a única opção verdadeira. Aí sim, você vai poder comprovar que realmente entendeu o conteúdo.



1. Qual das afirmações a seguir é a mais correta em relação ao cuidado que devemos ter em relação ao uso de imagens em sites?

- A Use sempre a imagem com maior qualidade e resolução possível
- B Use sempre a imagem com menor qualidade e resolução possível
- C Use sempre a imagem com qualidade e resolução balanceadas
- D Use sempre a menor quantidade de imagens para não deixar o site pesado

2. Deixar um site “pesado” com imagens grandes pode gerar:

- A Aumento nas visitas por conta da qualidade visual
- B Diminuição na taxa de retenção e possível penalização em mecanismos de busca
- C Aumento na taxa de retenção, pois o usuário passa mais tempo no site
- D Diminuição no tempo das visitas, mas isso acaba melhorando a indexação

3. Para usar imagens dinâmicas em nossos sites, devemos aprender a usar três tags, que são:

- A <picture>, <source> e
- B <image>, <picture> e <source>
- C <picture>, <image> e
- D , <source> e <image>

4. A tag <source> possui o parâmetro _____ para configurar as dimensões da mídia, o _____ para indicar o formato da mídia e o parâmetro _____ para indicar o arquivo de destino. A única opção que mostra os parâmetros que preenchem as colunas na ordem correta é:

- A media, type e srcset
- B type, media e srcset
- C size, format e src
- D size, type e src

5. Podemos adicionar arquivos de áudio reproduzíveis às nossas páginas usando a tag:

- A <sound>
- B <music>
- C <podcast>
- D <audio>

6. Qual dos valores a seguir é o único que não é um parâmetro da tag de áudio?

- A controls
- B autoplay
- C player
- D loop

7. Quando inserimos um áudio no site, qual é o comportamento padrão do navegador em relação à carga do conteúdo?

- A o arquivo de áudio não será carregado até que o visitante aperte o Play
- B o arquivo de áudio será completamente carregado, mesmo que nunca seja apertado o Play
- C os primeiros minutos de áudio serão carregados para facilitar a reprodução
- D apenas os metadados do arquivo (informações básicas sobre o arquivo) serão carregados automaticamente

8. Para carregar vídeos hospedados localmente, podemos usar a tag HTML:

- A <video>
- B <iframe>
- C <media>
- D <youtube>

9. Quais são os formatos de vídeo que devemos considerar para manter a compatibilidade com a maioria dos navegadores?

- A .mov .mp4 .avi .webm
- B .m4v .ogv .avi .mov
- C .mp4 .m4v .webm .ogv
- D .mp4 .mov .webm .avi

10. Para inserir vídeos do YouTube no nosso site, usamos a tag:

- A <video>
- B <iframe>
- C <link>
- D <media>

Suas anotações

Não guarde conhecimento. Ele é livre. Compartilhe o seu e veja ele se espalhando pelo mundo 