**CURSO DE HTML5 E CSS3**

PROFESSOR: Gustavo Guanabara

LOCAL: site - <https://www.cursoemvideo.com>

**MÓDULO: 2**

REPOSITÓRIOS PÚBLICOS: github.com/gustavoguanabara E gustavoguanabara.github.io

NÚMERO DE VÍDEOS:

DATA DE INÍCIO: 04/02/2024

DATA DE TÉRMINO: 18/03/2024

1. Vídeo 1 – O que vamos aprender no módulo 2?

Principais assuntos: manipulação das cores (psicologia, harmonização), estudo das fontes (tipos e formatação em geral), uso do ID, classes, variáveis, modelo de caixas, iframe.

1. Vídeo 2 – Psicologia das cores

As cores influenciam as pessoas de um modo inconsciente. A harmonização das cores em um produto, por exemplo, ou em uma vitrine de uma loja, ajudam a destaca-lo diante dos clientes e em muitos casos até determinam a venda. Na criação de sites acontece o mesmo: sites melhor harmonizados com cores bem distribuídas aumentam suas chances de visitação, permanência e resultados, como por exemplo, gerar vendas.

\*É interessante estudar um pouco o significado das cores. Isso é importante no momento em que terminamos o site e vamos coloca-lo no ar.

1. Representando cores em CSS3

\*Técnica para representar maior número de cores.

* 1. Modo de representação por nomes (já aprendido anteriormente). Exemplo de código:

<body>

     <!--Representação por nomes-->

     <h1 style="background-color: darkgreen; color: white;">Exemplos de Cores</h1>

     <p> As cores influenciam as pessoas de um modo inconsciente. A harmonização das cores em um produto, por exemplo, ou em uma vitrine de uma loja, ajudam a destaca-lo diante dos clientes e em muitos casos até determinam a venda. Na criação de sites acontece o mesmo: sites melhor harmonizados com

cores bem distribuídas aumentam suas chances de visitação, permanência e resultados, como por exemplo, gerar vendas.</p>

* 1. Modo de representação por códigos hexadecimais. Exemplo de código:

<!--Representação por códigos hexadecimais-->

     <!--Decimal:0,1,2,3,4,5,6,7,8,9-->

     <!--Hexadecimal:0,1,2,3,4,5,6,7,8,9 + A,B,C,D,E,F (total de 16)-->

     <h1 style="background-color: #0000ff; color: #ffffff;">Exemplos de Cores</h1>

     <p> As cores influenciam as pessoas de um modo inconsciente. A harmonização das cores em um produto, por exemplo, ou em uma vitrine de uma loja, ajudam a destaca-lo diante dos clientes e em muitos casos até determinam a venda. Na criação de sites acontece o mesmo: sites melhor harmonizados com cores bem distribuídas aumentam suas chances de visitação, permanência e resultados, como por exemplo, gerar vendas.</p>

OBS: No Visual Studio Code, depois de “background-color” aparece um quadrado pintado com a cor correspondente, o que não apareceu aqui no word. Também é importante saber como os códigos representam as cores. É mais ou menos assim. Toda cor é representada por porcentagens de três cores básicas: vermelho, verde e azul. Então, nos códigos, os dois primeiros símbolos representam vermelho, os dois do meio representam verde e os dois últimos representam azul. Observe. O código do verde que estamos usando é #0000ff, sendo 00=vermelho, 00=verde e ff=azul. Cada valor vai de 0 a 255. Assim, nesse código, não tenho nada de vermelho(por isso é 00), nada de verde(por isso também é 00) e tudo de azul(por isso é 255). Dá para ver isso claramente em um teste usando o app “Gimp” em sua paleta de cores. Lá tudo que está escrito aqui fica visualizável.

* 1. Modo de representação por “rgb”(parâmetro)

<!--Representação por rgb-->

     <h1 style="background-color: rgb(0, 0, 255); color: rgb(255,255, 255);">Exemplos de Cores</h1>

     <p> As cores influenciam as pessoas de um modo inconsciente. A harmonização das cores em um produto, por exemplo, ou em uma vitrine de uma loja, ajudam a destaca-lo diante dos clientes e em muitos casos até determinam a venda. Na criação de sites acontece o mesmo: sites melhor harmonizados com cores bem distribuídas aumentam suas chances de visitação, permanência e resultados, como por exemplo, gerar vendas.</p>

* 1. Modo de representação por função “hsl”

\*Essa função representa três características das cores: matiz(h), saturação(s) e luminosidade(l). O código ficou assim:

<!--Representação por função hsl(características de cores)-->

     <h1 style="background-color: hsl(240, 100%, 50%); color: hsl(0, 0%, 100%);">Exemplos de Cores</h1>

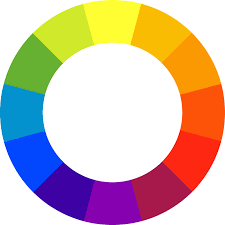
     <p> As cores influenciam as pessoas de um modo inconsciente. A harmonização das cores em um produto, por exemplo, ou em uma vitrine de uma loja, ajudam a destaca-lo diante dos clientes e em muitos casos até determinam a venda. Na criação de sites acontece o mesmo: sites melhor harmonizados com cores bem distribuídas aumentam suas chances de visitação, permanência e resultados, como por exemplo, gerar vendas.</p><!--Esses dois últimos exemplos de representaçaõ de cores, não funcionaram. Provavelmente o problema está relacionado com os parênteses que ficaram amarelos. É preciso resolver esses problemas para que as funções funcionem.Aqui no word os parênteses não ficaram amarelos, mas no Visual Studio Code sim-->

1. Harmonia das cores

\*A harmonização das cores é uma disciplina da área de design, por isso aqui serão dadas apenas algumas considerações a respeito, para que possamos saber que o profissional desenvolvedor de web também precisa entender um pouco de design. E a harmonização das cores é parte importante do design.

\*Círculo Cromático – Aqui temos dois exemplos básicos de círculos cromáticos. O objetivo principal de usá-los é aprender a harmonia das cores. Junto com a simetria, a harmonização de cores traz beleza e beleza é um elemento importante de comodidade visual. No desenvolvimento web não é diferente.

Completo com milhões de cores Simplificado com 12 cores principais



\*As divisões das cores dentro do círculo cromático são: primárias, secundárias e terciárias. Primárias: amarelo, vermelho e azul. Secundárias: laranja, violeta e verde. Terciárias: são todas as outras que restaram e são mistura entre todas. Os nomes das cores terciárias são compostos. Exemplos: amarelo-esverdeado, vermelho-alaranjado, azul-esverdeado. Como podemos ver, o primeiro nome sempre é o da cor primário. Também é possível fazer a divisão do círculo cromático por “temperatura”. Nesse caso, temos dois grupos: cores frias e cores quentes. As cores frias são as da esquerda e as quentes as da direita.

\*Na prática, precisamos gerar uma “paleta” de cores, que é um conjunto de cores que se relacionam de algum modo(tonalidade, subtonalidade, contraste,etc) e quem deve ter entre 3 a 5 cores no máximo, lembrando que preto e branco sempre estão presentes. Lembrando que as cores devem conectar o cliente com a ideia de seu site ou produto.

\*Cores complementares: são aquelas que mais contrastam umas com as outras. Para encontrar a cor que mais contrasta com a outra em nosso círculo cromático simples de 12 cores, basta traçar uma reta na direção oposta de determinada cor.

\*Cores análogas: são aquelas parecidas em tonalidade e estão geralmente ao lado uma da outra no círculo cromático. Dá para misturar as cores, por exemplo, pegando duas cores análogas e uma complementar. Com essa mistura, já posso fazer uma paleta de cores simples, com quatro cores: uma cor primária, duas análogas(vizinhas) e uma complementar(contrastante posicionada na direção oposta do círculo cromático).

\*Cores intercaladas: escolher uma cor primária e as outras a partir de pulos para o lado no círculo cromático.

\*Cores triádicas: começar de uma cor, pular três, pegar a próxima e fazer isso de novo. No final, você terá três cores escolhidas.

\*Cores em quadrado: parecida com as triádicas, mas o pulo é de duas em duas.

\*Cores tetrádicas: Escolha duas cores e suas complementares. Isso formará um “triângulo” no seu círculo cromático.

\*Monocromia: trabalha apenas com uma cor, trabalhando a mudança de duas características: saturação e brilho. Cria um efeito “degradê”.

1. Paleta de cores

\*Sugestão da ferramenta “adobe color” (site). Na opção criar, da para criar uma paleta de cores baseadas nos vários princípios acima citados e também nas opções “extrair tema” e “extrair degradê” dá para trazer um arquivo de fora para configurá-lo (por exemplo, uma logo). Se você quiser uma sugestão de paletas, pode ir na opção “explorar/exibir/temas de cor/mais usadas.

\*Outra sugestão de ferramenta é o “Paletton” (site). Essa ferramenta é interessante porque depois de escolher uma paleta, você pode clickar em uma sugestão de modelo de site com as cores que você escolheu, tipo um “template”. Depois de montar sua paleta, clique em “exemplos”. Automaticamente será gerado um modelo de site com as cores das sua paleta aplicadas.

\*A terceira sugestão é o site <https://coolors.co/>. Clique no “Criar paleta” e depois é só apertar a barra de espaços para receber sugestões de paletas. Se você gostar de alguma das 5 cores que aparecerem, pode clicar no “cadeado” e fixar aquela cor. Dali em diante, ela ficará imóvel e o algoritmo vai continuar mudando as outras agora relacionando-as com a escolhida. No final vc pode copiar o código da cor e jogar no seu código html.

1. Como capturar cores da tela
2. Como criar degradê com CSS?

Essa é uma codificação simples, apenas usando a tag <style> dentro da seção <head>. O código fica assim (a parte verde é comentário):

<style>

            body {

                background-image:  linear-gradient(to bottom, white, green);

           }

    </style><!--O degradê pode ser lateral, indo da direita para a esquerda(to left) ou o contrário(to right). Também pode ser vertical, indo de cima para baixo(to bottom) ou o contrário(to top). Dá também para fazer degradê na diagonal usando graus, por exemplo "45deg" ou "90deg", no lugar do "to bottom". É possivel também trazer uma paleta de cores e usar. Também é possível transformar o degradê em um círculo usando duas mudanças. A primeira é trocar "linear-gradient" por "radial-gradient" e "to bottom" para "circle"-->

</head>

1. Criando um exemplo real

\*Foi criado um site com dois documentos, um base de HTML e outro de configurações de estilo de CSS. As configurações de CSS foram aplicadas no site através de um link colocado no documento de HTML. Abaixo estão os dois documentos para visualização e melhor compreensão:

* 1. Documento HTML

<!DOCTYPE html>

<html lang="pt-br">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Site de Exemplo</title>

    <link rel="stylesheet" href="style.css">

</head>

<body>

     <main>

           <h1>Site de Exemplo</h1>

           <h2>Título 1</h2>

           <p>\*A harmonização das cores é uma disciplina da área de design, por isso aqui serão dadas apenas algumas considerações a respeito, para que possamos saber que o profissional desenvolvedor de web também precisa entender um pouco de design. E a harmonização das cores é parte importante do design. </p>

           <p>\*A harmonização das cores é uma disciplina da área de design, por isso aqui serão dadas apenas algumas considerações a respeito, para que possamos saber que o profissional desenvolvedor de web também precisa entender um pouco de design. E a harmonização das cores é parte importante do design. </p>

           <h2>Título 2</h2>

           <p>\*A harmonização das cores é uma disciplina da área de design, por isso aqui serão dadas apenas algumas considerações a respeito, para que possamos saber que o profissional desenvolvedor de web também precisa entender um pouco de design. E a harmonização das cores é parte importante do design. </p>

           <p>\*A harmonização das cores é uma disciplina da área de design, por isso aqui serão dadas apenas algumas considerações a respeito, para que possamos saber que o profissional desenvolvedor de web também precisa entender um pouco de design. E a harmonização das cores é parte importante do design. </p>

           <h2>Título 3</h2>

           <p>\*A harmonização das cores é uma disciplina da área de design, por isso aqui serão dadas apenas algumas considerações a respeito, para que possamos saber que o profissional desenvolvedor de web também precisa entender um pouco de design. E a harmonização das cores é parte importante do design. </p>

           <p>\*A harmonização das cores é uma disciplina da área de design, por isso aqui serão dadas apenas algumas considerações a respeito, para que possamos saber que o profissional desenvolvedor de web também precisa entender um pouco de design. E a harmonização das cores é parte importante do design. </p>

     </main>

</body>

</html>

* 1. Documento CSS

@charset "UTF-8"

\*{

    font-family: Arial, Helvetica, sans-serif;

    height: 100%;

}

body{

    background-image: linear-gradient(to right, #CCEA8D, #A6BC09);

}

main{

    background-color: white;

    border-radius: 20px;

    box-shadow: 10px 10px 10px black;

    width: 600px;

    padding: 10px;

    margin: auto;

}

h1{

    color: #01415B;

    text-align: center;

    box-shadow: 1px 1px 2px #005148;

}

h2{

    color: #019587;

}

p{

  text-align: justify;

}

1. Primeiros passos em Tipografia

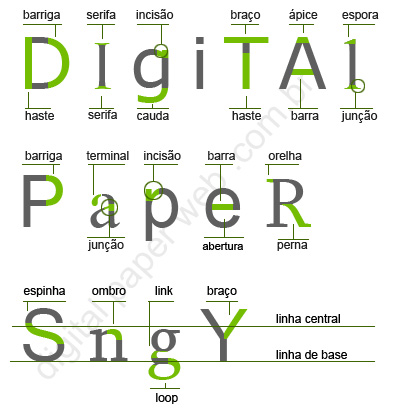
\*Tipografa é o estudo da palavra escrita, em qualquer lugar, seja em papéis, telas, etc. Um dos objetivos principais da tipografia é tornar os textos mais legíveis, melhor visualizáveis. A Tipografia é importante no web desenvolvimento porque nesse trabalho precisamos juntar boas imagens, boas cores e boas letras, para tornar o ambiente visual do nosso site mais agradável e eficiente. Como desenvolvedores, precisamos lembrar que além do conteúdo de um site, também é importante a apresentação visual desse conteúdo.

1. Anatomia do tipo

\*É importante saber a anatomia do tipo porque se trata de tamanhos das fontes. O elemento principal desse tamanho para nós desenvolvedores é o “corpo”, que é uma espécie de tamanho geral da fonte, a somatória de outros tamanhos como altura-x, altura das maiúsculas, altura ascendente(minúsculas para cima)e altura descendente(minúsculas para baixo).

\*A serifa é um detalhe que forma uma linha imaginária nos textos e que ajuda na leitura.

\*Partes geométricas das letras: haste, filete, arco, esporão, vértice, terminal, braço, perna, pé, espinha, barriga, olho, orelha, cauda. Abaixo temos uma imagem que nos ajuda a visualizar essas partes das letras:



\*glifos: são caracteres ou letras e o conjunto de glifos é o que chamamos de “fonte”.

\*Categorias das fontes: (1) Serifada, porque tem serifa. Exemplo: Times New Roman. (2) sans-serif, porque não tem serifa. Exemplo: Sans serif. (3) Monoespaçada, toda letra tem sempre a mesma largura e ocupam o mesmo espaço. (4) Handwriting, simula escrita á mão. Exemplo: Kunstler Script..(5)Display, comemorativas, criativs, que remetem a qualquer coisa. Exemplo: algeriam.

1. Famílias de fonte com CSS

\*Para escolher uma família de fontes é simples. Basta criar uma tag <style> dentro da tag <head> e embaixo da tag<title>. Dentro dessa tag <style> use o parâmetro “body” e dentro dele o parâmetro “font-family”. O código fica assim:

<title>Fontes em CSS</title>

    <style>

          body{

              font-family: Arial, Helvetica, sans-serif;

          }

    </style>

</head>

OBS: Essas três fontes acima(arial, helvética e sans-serif) fazem parte da mesma família. Esse é um comando de “segurança”, porque o computador tem alternativas para trabalhar. Ele tenta a primeira fonte(arial), se não der certo tenta a segunda(helvética) e em último caso a terceira(sans-serif). Provavelmente o dispositivo terá pelo menos uma dessas fontes. É possível também colocar um parâmetro “font-family” fora do campo “head”, no campo “body” por exemplo, para alterar um título e assim a fonte do título será diferente da fonte do parágrafo. Para maiores informações sobre combinação segura de fontes, pesquise no google por “css web safe fonts combinations” ou vá direto para o endereço <https://www.w3schools.com/cssref/css_websafe_fonts.php> .

1. Tamanho de fontes e suas medidas

\*Sobre os tamanhos das fontes, temos dois tipos de medidas: absolutas e relativas. As absolutas são centímetro(cm), milímetro(mm), polegada(in), pixel(px), ponto(pt) e paica(pc). Esses dois últimos, ponto e paica, são para material impresso, por isso não se recomenda usar em material que vai ser exibido apenas em tela. As relativas são tamanho atual da fonte(em), á altura “x” da fonte(ex), para fonte que está configurada no “body”(rem), largura da view corte(vw), altura da view corte(vh), porcentagem(%). Aqui trabalharemos com o parâmetro “font-size” e para esse parâmetro recomenda-se apenas a medida absoluta “px” e a relativa “em”. Geralmente o tamanho do “em” equivale a 16px. A linha de código pronta será essa: “font-size:1em;”. No código ficará assim:

<title>Fontes em CSS</title>

    <style>

          body{

              font-family: Arial, Helvetica, sans-serif;

              font-size:1em;

          }

    </style>

</head>

1. Peso, estilo e shorthand font

\*Peso é quando deixamos a fonte mais “magrinha” ou mais “gordinha”. Para aprender sobre peso das fontes, precisamos de fontes mais específicas, porque essa comuns que sempre usamos só tem dois pesos: magrinha(sem negrito) e gordinha(com negrito). O professor indica a fonte “work sans” e diz que na próxima aula vai dar maiores detalhes de como utilizá-la já que ela não está no computador. O peso da fonte em parâmetro é “font-weight”. Alguns pesos aceitos nesse parâemetro são: ligther, normal, bold, bolder. Assim, a linha de comando completa desse parâmetro será “font-weight:lighter”. Lembrando que no lugar de “ligther” podemos usar as outras três palavras: normal, bold ou bolder. Além desse tipo de variação, temos também a numérica que começa em 100 e termina em 900. Começando em 100 e indo até 900 a fonte vai “engrossando” á medida que os números vão crescendo. A linha de código ficará assim: font-weight:100. Obviamente, no lugar do 100 você pode colocar qualquer número entre 100 e 900 e ir fazendo seus testes. É possível também acrescentar o efeito itálico junto com o peso. A linha de código para itálico é: font-style:italic. Além disso temos também o sublinhado: text-decoration:underline. Todas essa configurações estão feitas dentro de “body” que está dentro da tag “style” que por sua vez está dentro da tag “head”. Também é possível aplicar todas essas configurações aos títulos e subtítulos(h1,h2,etc)

\*shorthand = é um parâmetro que abarca todas as configurações das fontes de uma só vez(font-family, font-weight, font-size, font-style). Assim você transformará 4 linhas em apenas 1. Observe a diferença:

* Sem shorthand: font-family:work-sans, sans-serif;

font-weight:bolder

font-size:3em

font-style:italic

* Com shorthand: **font:italic bolder 3em ‘work sans’, sans-serif;**

OBS: Não existe “shorthand” para tudo, mas existem vários tipos de shorthand que você pode facilitar para diminuir o tamanho do seu código. Posso fazer um shorthand com menos propriedades, mas nunca fora da ordem. A ordem é: font-style, font-weight, font-size, font-family.

1. Usando o Google fonts

\*As fontes embutidas nos navegadores são um pouco limitadas, mas existe uma alternativa externa para esse problema: o google fonts(<https://fonts.google.com/>).

\*Dentro do Google fonts, o caminho para escolher uma fonte é: classification (escolher uma família de fontes), na lista de fontes que aparece (escolher uma fonte), clicar em “get font”(botão do lado direito superior), clicar em “get embed code”(botão do mesmo lado direito superior), marcar a opção “@import”(ao invés de “link”), marcar o código todo para copiar(sem a tag “style”), copiar, voltar para o seu documento html e colar dentro da tag “style”. É possível também copiar uma família inteira de fontes. Além disso, você tem a opção “baixar” ou “download”, ao invés dessa “embed” que é apenas “incorporar”. Depois, para usá-la no meu código, é necessário apenas incluí-la no parâmetro “font-family”, que fica dentro do parâmetro “body” que por sua vez está dentro da tag “style”. Lembrando que o parâmetro “body” será aplicado apenas nos parágrafos. Para mudar uma fonte de títulos precisa-se usar o parâmetro “h1” ou “h2”, etc.

\*É preciso tomar cuidado para não exagerar na quantidade de fontes. Geralmente um site precisa de apenas 3 ou 4 tipos de fonte no máximo: uma para títulos, outra para destaque e outra normal para textos em geral.

1. Usando fontes externas baixadas

\*Existem muitos sites para pegar fontes externas na internet. Um deles é o “dafont.com”. Nesse site, primeiro você vai escolher a fonte na página principal e depois você vai clicar em “baixar” (botão do lado direito).

\*Ao escolher uma fonte, é preciso ficar atento a alguns detalhes, como por exemplo, acentuação. Algumas fontes externas não possuem a configuração de acentuação e, portanto, para nós brasileiros que usamos a língua portuguesa com várias acentuações, esse tipo de fonte não é útil. Precisamos, obviamente, de uma fonte com acentuações.

\*Para incluir essa fonte baixada no código, é preciso usar um parâmetro chamado “@font-face”, dentro da sua tag “style”. Os arquivos baixados serão de extensão “OTF” ou “TTF”. Dependendo do navegador, a melhor escolha será um ou outro, ou seja, alguns navegadores aceitarão os dois, mas outros apenas um dos dois; é preciso testar para ver a aceitação. Na “url” de “src” dentro de “font-face”, coloque o nome do arquivo.

OBS: Não consegui encontrar arquivos “otf” ou “ott”, mas apenas arquivos “zip”. E os arquivos “zip” não consegui usar porque o parâmetro “format” não aceita. Portanto, preciso resolver esse problema e pensei em apenas duas formas no momento: ou encontrando as extensões “otf/ott” em outros sites diferentes dos pesquisados(“dafont.com” e “fontsquirrel.com”) ou tentar incluir a extensão “zip” no parâmetro “format”.

1. Capturando as fontes usadas em um site

\*Extensão do Google Chrome que permite capturar fontes de sites que você esteja visitando. Lembrando que esse “plug-in” só funciona para textos normais, não textos dentro de imagens. Entre no Google Web store. Procure por uma extensão na caixa de pesquisa. Aqui no exemplo, a extensão escolhida foi “Fonts Ninja”. Clique em “usar no chrome”, depois “adicionar a extensão”. O ícone vai aparecer na parte superior direita do navegador. Para desligar uma extensão, clique nos três pontinhos(canto superior direito)/mais ferramentas/extensões e desative qualquer extensão.

\*Essa extensão vai proporcionar a você capturar o nome da fonte usada no site em que está e com o nome você pode procura-la em qualquer site de fontes como o Google fonts, por exemplo.

17. Detectando fontes dentro de imagens

\*3 sites úteis: (1) <https://www.whatfontis.com> , (2) <https://www.fontsquirrel.com> e (3) <https://www.myfonts.com/es> .

\*whatfontis = na primeira página, coloque uma imagem no local indicado; depois, selecione a fonte que quer e clique em “next step”(botão abaixo); configure contraste e brilho; clique em “next step” novamente e vai aparecer o nome da fonte na próxima página.

\*fontsquirrel = clique em “fontidentier”, coloque uma imagem no local indicado,

\*myfonts = clique em “what the font”,

OBS: O que teve melhor desempenho foi o primeiro “whatfontis”.

1. Alinhamento de textos com CSS

\*O alinhamento de textos possui 4 características básicas: á direita, á esquerda, centralizado e justificado. Observe como fica no código:

<title>Alinhamento de textos</title>

    <style>

          body{

               font: normal 16px Arial Verdana serif

               text-align: left;

          }

          h1{

            text-align: center;

          }

          h2{

            text-align: right;

          }

          p{

            text-align: justify;

            text-indent: 30px;

          }

    </style>

</head>

OBS: A recomendação é para alinhar o texto usando CSS, não HTML.

1. Usando o id com CSS

\*O “id” é um identificador que destaca algo para ser tratado de modo especial e ser configurado de modo mais específico. Por exemplo, se eu coloco o “id” dentro de um “h1”, estou dizendo que aquele “h1” vai receber uma configuração mais específica, diferente das configurações padrões dos outros “h1” por exemplo.

\*O seletor personalizado é uma configuração especial feita no arquivo CSS. Observe o resultado:

@charset "UTF-8"

body{

    background-color: darkcyan;

    font-family: Arial, Helvetica, sans-serif;

}

h1{

    color: darkgreen;

}

h1#principal{

            text-align: center;

            background-color: cadetblue;

            color: white;

}

h2{

    color: rgb(104, 155, 53);

}

OBS: O seletor personalizado é essa configuração destacada em amarelo.

1. As diferenças entre id e class

\*O elemento “id” pode ser usado mais vezes, mas não é recomendável. Recomenda-se ter apenas um “id” por documento HTML. Depois do primeiro “id”, vamos usar “class”. O comando “class” também faz a configuração personalizada e é indicada no documento CSS pelo ponto “.”. Veja como ficam as configurações, primeiro no documento HTML, depois no CSS:

* HTML

<body>

     <h1 id="principal">Criando sites com HTML e CSS</h1><!--Tudo que em HTMl é "id", em css É "#". Tudo que em HTMl é "class", em CSS é "."-->

     <h1>Aprendendo HTML</h1>

     <h2 class="basico"> HTML básico</h2>

     <p>\*Existem muitos sites para pegar fontes externas na internet. Um deles é o “dafont.com”. Nesse site, primeiro você vai escolher a fonte na página principal e depois você vai clicar em “baixar” (botão do lado direito).</p>

     <h2 class="intermediario" > HTML intermediário</h2>

     <p>\*Existem muitos sites para pegar fontes externas na internet. Um deles é o “dafont.com”. Nesse site, primeiro você vai escolher a fonte na página principal e depois você vai clicar em “baixar” (botão do lado direito).</p>

     <h2 class="avancado" > HTML avançado</h2>

     <p>\*Existem muitos sites para pegar fontes externas na internet. Um deles é o “dafont.com”. Nesse site, primeiro você vai escolher a fonte na página principal e depois você vai clicar em “baixar” (botão do lado direito).</p>

OBS: Essa mesma configuração foi feita também no restante do documento, na parte de CSS.

* CSS

h2{

    color: rgb(104, 155, 53);

}

.basico{

        color: rgb(107, 155, 53);

}

.intermediario{

              color: rgb(107, 155, 53);

}

.avancado{

          color: rgb(107, 155, 53);

}

\*É possível aplicar mais de um elemento em uma classe. Por exemplo: “avançado” e “destaque”. É possível também colocar um elemento de “class” em um “id”, mas o elemento do “id” vai se sobrepor sempre.

21. Pseudo-classes em CSS

\*Pseudo-classes são relacionadas ao estado de um determinado elemento e seu símbolo é o “:”. Existem várias pseudo-classes. Aqui no exemplo foi usada a pseudo-classe “hover”, que provoca mudanças no texto quando você passa o mouse em cima. O código ficou assim:

<title>Exemplo de hover</title>

    <style>

          body{

               font-family: Arial, Helvetica, sans-serif;

          }

          div > p{

               display: none;

          }

          div:hover >p{

                  display: block;

                  color: white;

                  background-color: red;

                  width: 300px;

          }

        div:hover{

                       color: darkred;

              }

    </style>

</head>

<body>

      <h1>Exemplo de hover</h1>

      <p>Passe o mouse sobre o texto abaixo</p>

      <div>Passe o mouse aqui

            <p>Texto escondido</p>

      </div>

</body>

</html>

1. Pseudo-elementos em CSS

\*Os pseudo-elementos são parecidos com as pseudo-classes. A única diferença é que seu símbolo de código é “::” ao invés de “:” das pseudo-classes. Veja o exemplo:

* a:active = pseudo-classe
* a::after = pseudo-elemento

\*Agora observe a diferença no código. Os dois primeiros são pseudo-classes e os dois últimos pseudo-elementos:

  a:hover{

                  color: black;

                  text-decoration: underline;

           }

           a:active{

                   color: yellow;

           }

           b::before{

                    content: '[link]';

           }

           a::after{

                   content: '[link]';

                   text-decoration: none;

                   font-weight: normal;

                   color: darkgray;

1. Modelo de Caixas: primeiros passos

\*”Caixa” é o espaço de qualquer conteúdo que será apresentado em um site. Por exemplo, se você tem a seguinte linha de código: <h1>Título</h1). Sabemos que no site aparecerá apenas a palavra “Título”, pois os “h1” aparecerão apenas no código. Essa palavra que aparece no site tem um “tamanho”, marcado por uma espécie de linha imaginária ao redor dela. Ficaria mais ou menos assim Título

Essa borda ao redor do título é como se fosse uma linha imaginária que mede o tamanho da palavra. Essa medição tem altura (height)e largura(width). Temos também “border” que é a linha imaginária citada acima e “padding” que é um espaço entre a borda e o início de cada letra. É possível também criar um espaço externo na borda, como se fosse a “borda da borda”. Esse espaço chama “margin”. Temos também “outline” que é uma espécie de “contorno” entre “margin” e “border”.

\*Existem dois tipos de caixa: “box level” e “inline level”. O “box-level” sempre ocupa o espaço total da tela do navegador. O “box-level” também pula para a linha debaixo. Já o “inline-level” não ocupa o espaço total porque ele quebra linha pela metade se tornando apenas um salto dentro da mesma. Um exemplo de caixa “box-level” é a tag “div” e um exemplo de “inline-level” é a tag “span”. Abaixo veja listas de exemplos de tags dos dois:

**BOX-LEVEL**

* <div>
* <h1>-<h6>
* <p>
* <main>
* <header>
* <nav>
* <article>
* <aside>
* <footer>
* <form>
* <video>

**INLINE-LEVEL**

* <span>
* <a>
* <code>
* <small>
* <strong>
* <em>
* <sup>-<sub>
* <label>
* <button>
* <input>
* <select>

1. Modelo de Caixas na prática(pt.1)

\*Vamos precisar usar o “DevTools” do Google Chrome. Para abrí-lo, basta clicar com o botão direito em qualquer lugar da página aberta no Google Chrome e depois selecionar a opção “inspecionar”. Outro caminho para abrir o “DevTools” é clicar nos três pontinhos no canto superior direito do navegador, opção “mais ferramentas” e opção “ferramentas do desenvolvedor”. Ir para a área de “elementos”. Clique nos três pontinhos no canto superior direito dentro do “DevTools” e depois na opção “undock” para abrí-lo em uma janela separada.

\*Esse vídeo precisa ser assistido de novo para melhor entendimento.

\*O código das configurações das caixas ficou assim:

<title>Modelo de Caixas</title>

    <style>

          h1{

             background-color: lightgray;

             height: 300px;

             width: 300px;

             border: width 10px;

             border-style: solid;

             border-color: darkslategray;

             padding-top: 10px;

             padding-left: 10px;

             padding-right: 10px;

             padding-bottom: 10px;

          }

          a{

            border-width: 10px;

            border-style: solid;

            border-color: red;

          }

    </style>

</head

1. Modelo de Caixas na prática(pt.2)

\*Para centralizar sua caixa use o parâmetro “margin”. Ficará assim: margin:auto;

\*Também existe o parâmetro outline que configura uma linha pontilhada fora da borda. Suas configurações ficam abaixo de “margin” no código.

\*Esse vídeo também precisa ser assistindo novamente para melhor entendimento.

1. Grouping tags em HTML5

\*Primeiramente temos as duas tags de agrupamento consideradas principais: <div> e <span>. A tag <div> é do tipo “box-level” e a <span> do tipo “inline-level”. Existem outras tags que podem substituir as “divs”. Elas são por exemplo: <header>(cabeçalho), <main>(corpo do site) e <footer>(rodapé). Essas são tags semânticas de agrupamento (grouping tags) que ajudam a configurar melhor as caixas “box-level” e “inline-level”.

\*Essas tags de agrupamento ajudam a organizar e configurar o site para melhor visualização através das “box-level” e “inline-level”. Observe como ficou o código:

<!DOCTYPE html>

<html lang="pt-br">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Grouping tags</title>

    <style>

           body{

               font-family: 'Times New Roman', Times, serif;

               background-color: blueviolet;

               margin: 0px;

           }

           header{

                 background-color: aqua;

                 padding: 10px;

                 margin: 10px;

           }

           nav{

               background-color: aquamarine;

               padding: 3px;

           }

           nav > a{

                   text-decoration: none;

                   color: white;

                   font-weight: bold;

                   margin-right: 30px;

           }

           nav a:hover{

                     text-decoration: underline;

           }

           main{

               background-color: brown;

               padding: 10px;

               margin: 10px;

           }

           article{

                  background-color: blue;

                  padding: 5px;

           }

           article > aside{

                           background-color: chartreuse;

           }

           footer{

                 background-color: black;

                 color: white;

                 text-align: center;

                 padding: 1px;

                 margin: 0px;

           }

    </style>

</head>

<body>

     <header>

             <h1>Meu site</h1>

             <nav>

                 <a href="#">Link</a>

                 <a href="#">Link</a>

                 <a href="#">Link</a>

                 <a href="#">Link</a>

                 <a href="#">Link</a>

             </nav>

     </header>

     <main>

           <section id="Assuntos">

            <p>Esportes, Política, Tecnologia</p>

           </section>

           <section id="Notícias">

                   <article>

                    <h1>Meu Artigo</h1>

                    <p>\*Primeiramente temos as duas tags de agrupamento consideradas principais: "div" e "span". A tag "div" é do tipo “box-level” e a "span" do tipo “inline-level”. Existem outras tags que podem substituir as “divs”. Elas são por exemplo: "header"(cabeçalho), "main"(corpo do site) e "footer"(rodapé). Essas são tags semânticas de agrupamento (grouping tags) que ajudam a configurar melhor as caixas “box-level” e “inline-level”. </p>

                            <aside>

                                  <p>Artigo escrito por THiago Nascimento</p>

                            </aside>

                   </article>

           </section>

     </main>

    <footer>

           <p>Desenvolvido por Thiago Nascimento</p>

    </footer>

    <div></div>

      <span></span>

</body>

</html>

1. Sombra nas Caixas

\*Na tag “nav” será usado um “box-shadow”.

**\*Esse conteúdo de “Caixas” terá que ser revisto e reestudado porque estou com dificuldade de acompanhar o professor na execução dos exercícios, principalmente no uso da extensão “DevTools”. No vídeo, o professor ajusta essa ferramenta ao código e consegue configurar com facilidade, mas eu não estou conseguindo fazer isso e não sei o motivo. Além disso, meu navegador Google Chrome, onde abro os projetos e exercícios, está muito lento, dificultando minha configuração.**

1. Caixas com vérticesarredondados

\*Aqui é apresentada a configuração “border-radius” que “arredonda” os quatro cantos da borda da caixa. Essa configuração tem 4 níveis: superior esquerdo, superior direito, inferior esquerdo e inferior direito. Você pode configurar esses níveis de 3 maneiras. Primeiro, colocando apenas um valor que representará por igual as quatro partes. Exemplo: border-radius:10px; . Segundo, colocando dois valores que representarão as diagonais. Por exemplo: border-radius:10px 10px;(a primeira diagonal é canto superior esquerdo com canto inferior direito; a segunda é canto superior direito com canto inferior esquerdo). Terceiro, colocando quatro valores referentes aos quatro cantos: Por exemplo: border-radius:10px 10px 10px 10px; . Obviamente, no primeiro caso, o único valor colocado valerá para os quatro cantos.

1. Bordas decoradas

\*Primeiro foi criado um arquivo de imagem no “Gimp” e depois foram feitas configurações no código usando esse arquivo. Foi usada essa imagem criada no Gimp como borda da caixa e essa borda ficou configurada da seguinte maneira:

<!DOCTYPE html>

<html lang="pt-br">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Borda Personalizada</title>

    <style>

          border: 20px solid black;

          padding: 10px;

          border-image-source: url ('borda.png')

          border-image-slice: 27px;

          border-image-repeat: repeat;

    </style>

OBS: Essas 5 configurações da borda podem ser feitas através de um “shorthand”.

1. Desafio do Módulo 2

\*O desafio é criar um site baseado no modelo apresentado no vídeo.

\*Ainda não fiz esse desafio por dois motivos: primeiro porque meu navegador está muito lento, demorando demais para abrir e atualizar; segundo porque parece que esse desafio foi colocado no lugar errado, visto que o modelo exige conhecimentos que ainda não aprendemos e serão ensinados nos próximos vídeos.

1. Criando um projeto a partir do zero

\*Para ver a imagem e explicação do projeto desse desafio, basta clicar nesse link a seguir: <https://github.com/gustavoguanabara/html-css/blob/master/desafios/modulo-02/d010/desafio-android.pdf> (na verdade não dá para clicar aqui nesse link, você precisa copiar e colar no seu navegador).

1. Navegando pelo projeto pronto

Foi apresentado o projeto e também dadas dicas.

1. Planejando a estrutura do seu site

\*O planejamento de um site é uma das partes muito importantes. Para isso existem ferramentas que nos ajudam nessa etapa inicial. As três partes básicas para o início do planejamento são: layout, cores e fontes. O layout pode ser feito no “MockFlow”( ou qualquer outro site semelhante). A paleta de cores pode ser montada no “AdobeColor” (ou outros sites sugeridos pelo professor). As fontes podem ser encontradas no “GoogleFonts” (ou outros sites sugeridos pelo professor).

1. Transformando o layout em código

\*Aqui teremos organização de layout através de tags semânticas. Criação das principais partes de um site: cabeçalho, menus, conteúdo principal, conteúdo periférico, rodapés.

\*Atalho para colocar texto para a direita no código: selecionar texto inteiro e apertar a tecla “tab” 3 vezes. Para voltar um “tab”, tecle “shift-tab”.

\*Foi feita a criação do layout através das tags semânticas e de acordo com as principais partes de um site.

1. Organizando o conteúdo do site

\*O conteúdo será organizado usando tags para envelopar partes específicas como parágrafos, títulos, imagens, vídeos. Utilização do recurso “wrap” do Visual Studio Code. Estruturação do conteúdo para boa apresentação.

\*Envelopamento: Ctrl+Shift+p/Emmet:Envolver com abreviação ou wrap with abreviation/ clica e depois digita o nome da tag.

1. Variáveis em CSS

\*As variáveis servem para guardar informações e aqui nas CSS não é diferente.

\*Essa variável guarda tanto cores como fontes. Aqui ela é declarada. Observe:

:root {

      --cor0: #c5ebd6;

      --cor1: #83e1ad;

      --cor2: #3ddc84;

      --cor3: #2fa866;

      --cor4: #1a5c37;

      --cor5: #063d1e;

      --fonte-padrao: Arial, Verdana, Helvetica, sans-serif;

      --fonte-destaque: 'Bebas Neue', cursive;

      --fonte-android: 'Android', cursive;

\*Mais á frente no código, a variável é invocada. Veja:

body{

     background-color: var(--cor0);

     font-family: var(--fonte-padrao);

}

header{

      background-color: var(--cor4);

}

header > h1{

           color: white;

           font-family: var(--fonte-destaque);

1. Responsividade para sites

\*Basicamente, um site responsivo é adaptável ao tamanho de telas. Adaptar tanto o conteúdo quanto o tamanho aos variados estilos de telas diferentes.

\*É preciso definir a largura máxima e mínima da tela. Há pessoas que criam um site para cada modelo de tela, por exemplo, um site para tela de celular e outro para tela de computador, mas isso não é bom nem recomendável, pois é possível você ter responsividade apenas com um mesmo código.

\*É preciso instalar a extensão “Window Resizer” do Google Chrome.

\*Como foi dito acima, é preciso definir a largura máxima e mínima da tela, para que a responsividade seja boa. Isso é simples de fazer. As configurações são “min-width” para largura mínima e “max-width” para largura máxima. Observe como fica no código:

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Criando modelo de responsividade</title>

    <style>

          body{

               background-color: beige;

          }

          main{

               background-color: white;

               padding: 10px;

               border-radius: 10px;

               min-width: 320px;

               max-width: 800px;

               margin: auto;

          }

          img{

              width: 100%;

          }

    </style>

</head>

OBS: Observe que as duas primeiras configurações “min-width” e “max-width” só resolveram o problema no texto, não na imagem. Para resolver o problema na imagem foi preciso criar essa tag “img” com configuração de “width” específica para a imagem.

\*Existe também uma segunda maneira de resolver esse problema na imagem. É envelopando a tag “img”. Para envelopar, você seleciona a linha de código da tag usando Ctrl+Shift+p, dá enter, escreve Picture. NO código vai aparecer a tag “Picture” em volta da linha de código onde está “img”. Em cima da linha “img” e dentro da tag “Picture”, você acrescenta o parâmetro “source” e faz as configurações. Veja o exemplo:

<body>

     <main>

          <picture>

                   <source media="(max-width:600px)" srcset="Imagens/irina-blok-pq">

                   <img src="imagens/irina-blok.jpg" alt="Irina Blok">

          </picture>

     </main>

</body>

1. Responsividade na prática

\*Foram feitas várias configurações que geram responsividade, principalmente as de largura mínima e máxima, tanto no código HTML quanto no arquivo CSS.

1. Configurando o “header” e o menu do site

\*Foram feitas algumas configurações no cabeçalho(header) e no menu. NO computador do professor tudo funcionou, mas no meu projeto apenas algumas configurações funcionaram. Essa aula precisa ser revista.

1. Melhorando o formato do conteúdo

\*Foram feitas as configurações que melhoraram o formato do conteúdo do site.

1. Rodapé, conteúdo periférico e links

\*Foram feitas configurações de estilo nesses itens acima citados.

1. Tornando um vídeo responsivo

\*As configurações do vídeo foram as seguintes: primeiro, envelopamento da tag “iframe”, criando uma tag “div” (no documento html). Segundo, configurações de estilo no documento CSS usando os parâmetros “div.video” e “div.video > iframe”. Veja como ficou no código:

\* “div” no código HTML:

**<div** class="video">

**<iframe** width="560" height="315" src="https://www.youtube.com/embed/l2UDgpLz20M?si=wHqdkOU9GojufdJ4" title="YouTube video player" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share" allowfullscreen>**</iframe>**

**</div>**

\* “div.video” no código CSS:

div.video{

          background-color: #1a5c37;

          margin-bottom: 30px;

          margin: 0px -20px 30px -20px;

          padding: 20px;

          padding-bottom: 58%;

          position: relative;

}

div.video > iframe{

                   position: absolute;

                   top: 5%;

                   left: 5%;

                   width: 90%;

                   height: 90%;