Nesta aula, aprendemos:

* A mexer na apresentação dos textos
* No alinhamento deles (text-align)
* No tamanho da fonte (font-size)
* Na cor de fundo (background)
* Na cor do texto (color)
* CSS inline
* Na linha onde temos a nossa tag, adicionamos a propriedade do CSS
* A tag <style>
* Dentro da tag, podemos colocar marcações de CSS referentes aos elementos que temos no nosso HTML
* A apresentação do CSS com um arquivo externo
* Como funciona o estilo em cascata do CSS
* Como importar um arquivo externo de CSS dentro da nossa página HTML
* Como representar cores no CSS
* Através do nome da cor
* Através do seu hexadecimal
* Através do seu RGB

Nesta aula, começamos a mexer na apresentação dos textos, no alinhamento deles e no tamanho da fonte. Na próxima, vamos ver como fazer isso em grandes quantidades de texto e de forma muito mais organizada. Te espero lá.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Identificador de elemento e tag de imagem

Vamos começar nossa aula organizando um pouco nosso código, porque ele está bem bagunçado. Tem algumas coisas que não fazem mais sentido. Por exemplo, a tag style, que está vazia. Podemos removê-la.

[00:14] Outra coisa que não está legal é termos CSS no nosso código enquanto temos um arquivo de CSS. O ideal seria movermos todo esse conteúdo para lá.

[00:27] O h1 é muito fácil. É só recortar o style, voltar para o nosso arquivo de CSS, e, como eu disse anteriormente, sempre tentando deixar a estrutura do CSS parecida com a estrutura do HTML. Temos o body, o h1, e aí o parágrafo. Vamos colocar o h1 e depois o style que tínhamos recortado.

[01:02] Vou apenas colar o que estava lá. Reparem no editor de código funcionando. Ele marcou que tem algo muito errado nessa linha, porque colocamos as aspas e o style, já que eu só recortei e colei. Nós não precisamos deles.

[01:14] Se eu salvar, voltar para o navegador, recarregar, o código está perfeito. Não tem nada errado. Funcionando do mesmo jeito. Só movemos o CSS.

[01:28] Mas temos um problema com esse CSS, porque ele é específico para um parágrafo. Para resolver isso, precisamos ter uma marcação para o HTML específica para esse elemento, e no CSS referenciar isso.

[01:49] Usamos para isso o marcador de identificação. É o id. Vamos colocar id = “”. Entre as aspas você coloca um nome qualquer, que é o mesmo que vai ser usado no nosso CSS.

[02:07] Vamos chamar nosso id de “missao". Vamos recortar o código CSS e colar no arquivo CSS. O “missao" é um dos parágrafos, ele vem logo depois do anterior. E como fazemos para marcar? Toda vez que estamos marcando uma tag e queremos referenciar ela no CSS, usamos o nome da tag. Toda vez que temos um identificador no HTML e queremos referenciar ele no CSS, usamos a #. Logo: #missao.

[02:50] Reparem que o editor de códigos já marcou com outra cor, para marcar que está certo.

[02:57] Vamos colar aquele código que tínhamos e salvar. Se recarregarmos o navegador, o código está perfeito. Não tem nada errado.

[03:11] Vamos evoluir esse código então. Vamos adicionar mais conteúdo à página, uma imagem.

[03:21] Essa imagem já está na pasta do projeto de vocês, podem baixar. É uma imagem com cadeiras de barbearia e um balcão atrás. Ela está com o nome “banner.jpg”.

Você pode baixar um ZIP com a imagem banner.jpg [aqui](https://caelum-online-public.s3.amazonaws.com/1179-html5-css3/04/banner.zip).

[03:41] Assim como o CSS, a imagem também é um arquivo externo. Nós colocamos a chamada para o CSS no head, porque é uma informação que estamos passando para o navegador. A imagem é uma informação que queremos exibir na nossa página, então vamos colocá-la dentro do body.

[04:01] Na estrutura visual da nossa página, a imagem vai ser a primeira coisa que vai aparecer. Então, em cima de tudo, vou chamá-la. Usamos a tag img para adição da imagem ao conteúdo.

[04:19] A tag img serve para chamarmos uma imagem, e como ela é um arquivo externo, precisamos dizer onde está, dizer qual a fonte dessa imagem. Para isso, usamos a propriedade source. Só que não escrevemos isso por inteiro. Abreviamos para src e colocamos o endereço: img src=”banner.jpg”.

[04:53] Como essa é uma tag de chamada de arquivo externo, não abrimos e fechamos. Ela é só a definição da tag.

[05:02] Se salvarmos o arquivo e abrirmos no navegador, recarregando a página, a nossa imagem já está ali.

[05:12] A imagem é essa tag, com uma fonte e um endereço.

[05:20] Na próxima aula vamos entender como fazer com que a imagem ocupe toda a largura da página, como fazer ela se posicionar melhor no texto, como brincar com espaçamento, etc. Mas isso fica para a próxima. Aqui, nós vimos como reestruturar nosso código, criar um identificador para marcar especificamente um elemento e adicionar uma imagem à nossa página.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CSS para imagens

Na aula passada, adicionamos uma imagem ao conteúdo da nossa página. Nesta aula, vamos aproveitar para mexer no CSS da nossa imagem.

[00:09] Se repararmos, podemos ver um erro. Nossa imagem não ocupa 100% da largura da página.

[00:18] Vamos aproveitar essa aula também para falar de todas as dimensões de um elemento no HTML.

[00:27] Se olharmos nosso HTML, vamos reparar que a imagem está adicionada antes do h1. Para criarmos um CSS que funcione para ela, vamos adicionar um identificador. Ele vai ter o mesmo nome da imagem, “banner”.

[00:49] Já vimos no nosso CSS que a posição em que ele deve estar é logo acima do h1. Para criarmos um estilo para o identificador, colocamos uma # e no nome dele: #banner.

[01:05] Nele, vamos adicionar todo o estilo que queremos.

[01:09] Voltando ao navegador, é importante falarmos sobre uma ferramenta que todos os navegadores modernos entregam para os desenvolvedores. É chamada de developer tools, ferramentas para o desenvolvedor.

[01:25] Se vocês apertarem F12, irão reparar que a ferramenta do desenvolvedor está à direita. Ela tem, entre outras coisas, um layout mobile para sua página, te deixa mexer no console, saber que tipo de arquivos estão sendo transacionados. Mas isso é assunto para o futuro. O que é importante agora é vermos que no developer tools temos uma divisão muito clara entre o que é o HTML, o que é o CSS e o que está sendo exibido no nosso elemento.

[01:57] Como isso funciona? Digamos, por exemplo, que eu quero ver meu h1. Se eu clicar sobre ele, vou ver todo o estilo dele, todo o estilo que já foi criado, e esse estilo vem do nosso arquivo CSS, e verei também a marcação que ele tem na minha página. Ou seja, a largura, altura, qual o espaçamento que ele tem.

[02:25] Essa ferramenta é fundamental para entendermos e trabalharmos com desenvolvimento. Com certeza você vai usar muito.

[02:32] O que queremos agora é mudar a largura do nosso elemento. Falando sobre dimensões no CSS, temos quatro parâmetros muito importantes. A largura de um elemento, a altura do elemento, e os espaçamentos. Espaçamento interno e espaçamento externo.

[02:54] Vamos item por item.

[02:56] Vou usar nosso h1 como exemplo e vou mexer no developer tools. Reparem que se eu clicar logo depois de text align center ele vai criar uma nova linha para eu poder inserir uma modificação.

[03:10] Só por curiosidade, vamos colocar a cor azul, para vocês verem funcionando.

[03:17] Imediatamente ele vai alterar a exibição do elemento, deixando o título do h1 em azul. Eu não quero em azul, então vou simplesmente apagar.

[03:29] Posso também, voltando com o título azul, recarregar a página, e quando recarrego ele some com aquele estilo criado. É só uma ferramenta para o desenvolvedor, na máquina dele, enquanto ele está fazendo aquilo funcionar.

[03:50] No meu h1, vamos falar algumas coisas sobre a dimensão dos elementos.

[03:55] A dimensão de altura é muito fácil de identificar. Eu insiro com height e coloco em pixels, por exemplo, 100 pixels, ou 100px. Esse elemento passou a ter um espaço muito maior.

[04:17] Colocando o mouse em cima, consigo ver no espaço sombreado de azul a altura do elemento. Vamos alterar para 150px. Vemos que a área aumentou na altura. Só que esse elemento já está ocupando a largura inteira da página. Vamos agora na nossa imagem, onde já temos o identificador do banner, mexer na largura.

[04:43] A largura de um elemento geralmente é adicionada em pixels, ou podemos também colocar a largura em percentual. Se eu quero que ocupe a página inteira, vou colocar 100%. Imediatamente altero a largura da imagem. Ela passa a ocupar 100% da largura da página.

[05:06] Vamos ver o que acontece se eu mexer na altura da imagem e colocar um height com 120px. Reparem que ele respeitou a medida de largura, ou seja, 100%, e ajustou a imagem para que ela ficasse apertada com 120px de largura.

[05:27] A imagem mexe proporcionalmente. Então, se eu aumento a largura, a altura aumenta proporcionalmente. Se eu aumento a largura e a altura, preciso fazer um cálculo muito correto para essa imagem não ficar distorcida. Nós não queremos alterar a altura do elemento.

[05:48] Vamos voltar ao h1. Eu já falei sobre a largura e sobre a altura. Vamos falar sobre o espaçamento.

[05:55] Para fazer isso, vou criar uma borda no elemento, que tem 10px, com formato sólido e cor preta. Não precisa se preocupar agora com essas configurações. Vamos entender sobre bordas e como elas funcionam no futuro.

[06:22] Mexendo na borda, conseguimos entender a largura e a altura do elemento. 100% de largura e 150px de altura. Vamos alterar a altura em pixels para entendermos o que acontece.

[06:39] A borda vai se ajeitando. Ela é a finalização do elemento.

[06:49] O que acontece então se colocamos um espaçamento interno? Ele é chamado de padding, e é configurado de várias formas. Ou ele funciona em todos os lados, ou só para cima, ou só para as laterais, ou só para baixo.

[07:07] Vamos imaginar um espaçamento interno para todos os lados, para ficar fácil de entender. Se eu colocar um espaçamento de 20px internamente, ele aumenta o meu elemento, criando um respiro entre a borda e o conteúdo. Ou seja, ele criou uma margem interna no elemento.

[07:33] Vou clicar do lado para desmarcar o padding, ou seja, ele vai cancelar essa criação, para ficar clara a diferença.

[07:42] Se eu colocar um espaçamento interno, ele vai criar um quadro dentro, dando um respiro para todos os lados. Se eu coloco, por exemplo, um espaçamento só para cima de 20px, ele cria esse respiro só na parte de cima. Esse é o espaçamento interno. É para fazer seu elemento se comportar melhor no espaço que ele tem.

[08:10] O espaçamento interno serve, por exemplo, para eu criar um respiro maior da borda para fora. Então, se eu quiser que a borda não esteja colada na lateral, preciso de um espaçamento externo lateral. Para fazer isso, adiciono a margin, que é o espaçamento externo, ou para a direita ou para a esquerda.

[08:35] Vamos fazer só na esquerda uma margem de 40px. Nossa borda descolou da lateral, criamos um espaço para fora do elemento, entre o elemento e a margem do navegador.

[08:53] Nosso elemento já tem um espaço para cima e para baixo, que foram adicionados pelo navegador inicialmente. Se eu quiser colocar uma margem para todos os lados igual, coloco só o nome dela, só a declaração margin. E na hora que renderizo, ele vai criar uma moldura em volta do meu elemento.

[09:14] Com isso, conseguimos mexer na configuração e na apresentação de todos os itens. Conseguimos mexer na largura, na altura, na borda, no espaçamento interno e no espaçamento externo.

[09:28] Vou recarregar a página, voltando para o que já tínhamos. A única coisa que quero fazer é mudar a largura da imagem, então vamos ao nosso CSS. Agora que já aprendemos, vou mudar a largura com a propriedade width e vou colocar o valor 100%.

[09:46] Salvando e recarregando no navegador temos o site do jeito como queremos.

[09:55] Reparem que quando fechei o developer tools, a imagem se ajustou para sempre o tamanho da página. Isso vai acontecer com qualquer tamanho de tela. Ela vai sempre ocupar 100%.

[10:09] Nós ainda vamos praticar muito espaçamento, largura e altura dos elementos. Se isso não ficou extremamente claro, não se preocupe. Aos poucos, com a prática, vai ficando natural.

[10:22] Na próxima aula, vamos melhorar ainda mais esse conteúdo. Te vejo lá.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Trabalhando com listas

Já evoluímos nosso site até aqui. Agora, vamos adicionar mais conteúdo. Vamos adicionar uma nova seção no site, que serão os benefícios da nossa barbearia.

[00:10] Para isso, vamos voltar ao nosso HTML. Abaixo do último parágrafo, vamos começar uma nova seção, chamada “Benefícios”.

[00:23] Vimos na primeira aula que para criarmos um título com HTML, usamos a tag h. e usamos h1 para ser o título principal da página. Agora que estamos criando uma nova seção, outra parte do nosso site, vamos colocar outro título. Seguindo a sequência, vamos usar h2.

[00:46] O h2 começa nosso novo conteúdo. Se salvarmos isso e formos no navegador, veremos o “Benefícios” destacado em negrito, seguindo o mesmo padrão que vimos para o título principal da página.

[01:07] Agora, queremos listar os benefícios da nossa barbearia, que são quatro.

[01:13] Vamos ao HTML. A estrutura que eu desejo é: o primeiro item, atendimento aos clientes é o nosso maior benefício. O segundo item, espaço diferenciado. Terceiro item, localização. Quarto e último item, profissionais qualificados.

[01:51] Essa é a intenção que eu tenho. Quero que esses itens estejam listados.

[01:58] Se formos até nossa página e recarregarmos, veremos que estão todos um do lado do outro, sem uma forma. Falta uma tag para marcarmos esses itens.

[02:10] No nosso HTML, a solução mais fácil dadas as ferramentas que temos seria colocar uma tag de parágrafo, mas ela não é correta para essa demanda. Não é a semântica correta para uma lista. As listas têm uma tag própria.

[02:31] Temos as listas não ordenadas, onde não importa qual item vem primeiro ou depois. E temos as listas ordenadas, como por exemplo uma receita de bolo, onde primeiro tenho que ligar o forno e depois colocar a massa.

[02:49] Aqui queremos uma lista não ordenada. Para isso, usamos a tag ul, de unordered list. Uma lista sem ordem.

[03:01] Se quisermos uma lista ordenada, é só trocar para ol, de ordered list.

[03:08] A tag ul é uma tag de conteúdo. Ela inicia e fecha. Precisa fechar ao final da lista. Vamos consertar a endentação dos nossos itens para visualmente sabermos tudo que está acontecendo. Todos os itens da nossa lista vão dentro da ul.

[03:29] Se salvarmos isso e recarregarmos o navegador, veremos que nada aconteceu, só temos agora um pequeno espaço no início da primeira linha. Por quê? A lista, além da tag ul ou ol, no caso usamos ul para lista não ordenada, ainda precisa de marcação para cada item.

[03:50] Para isso, usamos a tag li, de list item. Para cada um dos itens, vou colocar a tag li. Ela inicia em cada uma das linhas e termina ao fim daquele conteúdo, daquele item.

[04:18] Agora que já temos os itens marcados podemos voltar ao navegador. Temos uma lista com os itens não ordenados. Podemos até remover o \*, que serviu só para explicarmos visualmente qual era o objetivo principal.

[04:40] Agora que temos a lista, que temos toda essa estrutura para nosso conteúdo, queremos criar um CSS para ele. Mas até agora só vimos duas formas de marcar o CSS. Primeiro usando as tags e o segundo usando os identificadores.

[05:02] O problema de usarmos as tags é que todas as listas do nosso site terão o mesmo estilo. E o problema de usar identificadores é que eles são únicos. Ou seja, temos uma lista com quatro itens e para marcar cada um deles, teríamos que criar quatro identificadores.

[05:19] Existe uma solução para isso. É o conceito das classes no CSS. As classes no CSS servem para marcarmos itens, para posteriormente colocarmos estilo em cada um deles, só que são repetíveis. Ou seja, podemos marcar todos os nossos itens com a mesma classe.

[05:38] Vamos fazer isso.

[05:40] Assim como no identificador, em que adiciono a propriedade id e dou um nome para ela, nas classes faço a mesma coisa. Adiciono a propriedade “class” e entre aspas o nome dela.

[06:14] Podemos colocar o mesmo nome em todos os itens. Essa vai ser a classe que usaremos para marcar todos esses nossos itens.

[06:20] Agora que temos os itens marcados, queremos criar um tamanho de fonte específico para eles e colocar todos em itálico.

[06:30] Vimos nas primeiras aulas como colocar um elemento em itálico visualmente simplesmente usando uma tag, que é ênfase, a tag em.

[06:43] Como fazemos para colocar vários elementos em itálico sem ser uma ênfase, sendo só um detalhe visual? No nosso CSS, vamos criar a referência para classe. Para fazer isso, quando usamos o identificador, usamos a #. Quando usamos uma classe, colocamos um “.”.

[07:02] Então .itens, e aí podemos criar todo o CSS para aqueles quatro itens da nossa lista. Quero dizer que o texto está em itálico, e para isso vamos colocar a propriedade font style, com a variável itálico: font-style: italic.

[07:22] Ou seja, estamos transformando todos aqueles itens em itálico.

[07:28] Se recarregarmos nossa página, veremos todos os itens marcados com itálico.

[07:35] Só que no nosso layout visual quero separar esse conteúdo. Quero que comecemos a ter um fundo branco nos itens, ao invés do fundo cinza, justamente para gerar esse destaque.

[07:48] Por isso, na próxima aula nós vamos falar em como criar divisões no nosso conteúdo.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Divisões de conteúdo

Agora, queremos criar mais uma divisão no nosso site, uma separação entre as duas seções.

[00:06] A primeira, onde teremos todo o texto de apresentação do nosso site, e a segunda, onde teremos os benefícios apresentados ao nosso cliente.

[00:14] Para isso, realmente precisamos ter o conceito de uma divisão. Criar dois blocos de conteúdo e separá-los com tags de HTML.

[00:25] Se formos ao nosso HTML, veremos que toda essa primeira parte, a partir do h1, e os quatro parágrafos, respondem a uma primeira seção, que poderíamos chamar de principal; e o h2 e nossa lista pertencem a uma segunda seção, que poderíamos chamar de Benefícios.

[00:52] Para fazer isso, criamos a tag de divisão, a tag div. A primeira vai englobar o h1 e os três parágrafos, e aí vamos indentar isso para ficar correto. Uma dica: se você selecionar tudo com o mouse ou com o teclado e apertar tab uma vez, ele vai indentar tudo como um bloco para você.

[01:25] Vamos fazer a mesma coisa nos benefícios. Criar outra divisão que fecha depois da nossa lista e indentar isso para ficar correto.

[01:42] Temos então no nosso HTML duas divisões. Vamos entender como isso ficou visualmente no nosso navegador. Observando, não notamos diferença alguma.

[01:52] As divisões têm por padrão não interferir na apresentação visual do nosso conteúdo. Elas servem para marcarmos nosso conteúdo, e, a partir do CSS fazermos os efeitos que quisermos. Mas como vamos marcar o conteúdo?

[02:12] Temos duas tags com o mesmo nome. Para isso, precisamos de especificidade, precisamos ou criar um identificador, ou criar uma classe.

[02:25] Já falamos a diferença dos dois. O identificador serve para itens únicos e a classe serve para itens que serão repetidos.

[02:33] Como só temos esse conteúdo, sabemos que eles são únicos, mas se quisermos criar um layout robusto, com todas as técnicas, e quisermos replicar esse conteúdo no futuro, o ideal é criarmos classes. Vamos seguir esse padrão a partir de agora. Sempre que quisermos marcar um conteúdo visualmente, vamos criar uma classe para ele.

[02:54] A classe que queremos criar é a dos benefícios, então vamos criar uma classe com o nome “benefícios”.

[03:02] Se formos no CSS, a marcação de uma classe é um ponto e o nome dela. O que queremos aqui é que ele tenha um fundo branco, e um fundo branco, como vimos, é com a propriedade background.

[03:27] Para fazer a cor branca, precisamos que todos os itens da nossa cor estejam no máximo, ou seja, todas as cores. Então, #FFFFFF.

[03:41] Se recarregarmos, veremos que essa nossa divisão está perfeita no nosso conteúdo. Mas temos um problema. Ela está com fundo branco, mas o fundo da minha página inteira está cinza. Por que isso aconteceu?

[04:00] Quando colocamos o fundo cinza, colocamos na tag do background, e não no nosso conteúdo de texto de apresentação. Portanto, vamos mudar isso a partir de agora.

[04:13] Na nossa tag do conteúdo principal, vamos criar uma classe com o nome “principal”. Para esta tag, quando voltarmos no nosso HTML, lembrando que a estrutura do CSS deve ser correspondente à estrutura do HTML, logo abaixo do banner vamos criar a classe principal. E nela vamos recortar o fundo cinza do background e vamos colar no principal.

[04:54] Agora, temos o site inteiro com o fundo branco e nosso espaço também com fundo branco.

[05:05] Um problema que temos que resolver no nosso site é que temos um espaço branco indesejado entre a imagem e o nosso conteúdo principal, e um espaço branco indesejado na lateral do nosso site.

[05:23] Isso foi o navegador que adicionou. Já vamos aprender a resolver.

[05:29] No momento, temos nosso título de benefícios e nossa lista de itens. Vamos continuar estilizando-os.

[05:38] Seguindo o padrão do bloco de cima, quero que o Benefícios seja centralizado. Para fazer isso, lembrando que no nosso HTML ele está com a tag h2, voltando no nosso CSS, vou criar um marcador para ele a partir da tag h2: text-align: center

[06:06] Voltando na nossa página, já temos um título centralizado.

[06:12] Por último, para completar esse bloco de conteúdo, precisamos adicionar mais uma imagem. Também já sabemos como fazer isso. Através da tag img.

[06:25] Vamos criar a img e vamos colocar o source dela, a fonte, onde essa imagem está. Ela é nossa imagem beneficios.jpg, que está na pasta do projeto.

[06:42] Quando voltarmos ao nosso navegador, a imagem está lá.

[06:48] Reparem que ela está muito grande, ocupando a largura inteira da página. O ideal é que ela fique ocupando 50% da página do lado direito, ao lado dos itens da lista.

[07:06] Na próxima aula, vamos entender como criar essa divisão e posicionar esses elementos do jeito que queremos.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**06Inline e Block**

Vamos criar o estilo necessário para fazer este bloco de conteúdo ficar perfeito.

[00:06] A primeira coisa que temos que fazer é ajustar a imagem. Ela está muito grande, ocupando 100% da largura do elemento, e não é isso que queremos. Queremos que ela fique posicionada ao lado da lista de benefícios.

[00:21] Para fazer isso, vamos mexer na nossa imagem, e já sabemos como. Precisamos criar uma classe para esse elemento.

[00:31] Eu vou criar a classe e chamá-la de “imagembeneficios”.

[00:45] No meu CSS, vou criar uma marcação para essa classe: .imagembeneficios

[00:56] E vou dizer que ela vai ter, a partir de agora, 50% da largura: width: 50%

[01:05] Quando voltamos ao navegador, essa imagem já passou a ocupar 50% da largura da página. Só que o que aconteceu para ela não estar ao lado dos itens?

[01:16] Se lembrarmos quando fizemos nossa lista sem a marcação dos itens, da tag li, estavam todos em uma linha. A mesma coisa aconteceu quando fizemos os parágrafos iniciais. Quando colocamos a tag li, ela passou a ocupar 100% da largura da página e quebrar a linha depois dela. Ou seja, o próximo item começava na linha de baixo.

[01:41] Esse comportamento de uma tag ocupar a largura inteira da página é chamado “block”. Ele bloqueia o conteúdo daquela linha. Todos os itens da nossa linha têm o comportamento block. Uma imagem não bloqueia o conteúdo, ela deixa que existam outros na lateral, e esse tipo de conteúdo é considerado inline.

[02:10] A diferença entre eles é que mesmo que eu diminua o tamanho, diminua a largura de um elemento block, ele vai sempre ocupar aquela linha, mesmo preenchendo só metade.

[02:25] Um elemento inline não me deixa alterar, por exemplo, o espaçamento externo e interno dele.

[02:34] Para isso, existe uma terceira característica, quando o elemento possui as duas condições. Ele é inline e block ao mesmo tempo. Ou seja, ele bloqueia uma largura, mas essa largura é fixa. Sou eu que dou o tamanho. E ele me deixa também mexer na largura e nos espaçamentos interno e externo.

[02:58] Vamos aplicar isso para ficar mais claro.

[03:01] A primeira coisa que temos que fazer é que a nossa lista precisa de uma marcação. Ou seja, nosso ul precisa de uma marcação. Eu falei do inline e do block. Isso são características do display. Ele pode ser block, inline ou juntar as duas coisas e ser do tipo inline block.

[03:33] O display inline block ocupa só o tamanho do conteúdo, mas me deixa mexer na largura e espaçamentos. Vamos ver na prática.

[03:43] No nosso navegador, quando recarregamos, vemos que o elemento está ocupando só o tamanho do seu conteúdo, e do lado dele já entrou nossa imagem. Mas aconteceu um erro. Na verdade, é uma características do CSS.

[04:01] Todos os elementos são alinhados pela linha de baixo e queremos que sejam alinhados pela linha de cima. Para isso, no nosso CSS, adicionaremos a propriedade “vertical-align: top”.

[04:18] Quando recarregamos a página, nosso título está centralizado, nossa lista de itens está ocupando só o tamanho dela e nossa imagem está ocupando 50% da página.

[04:33] Vamos agora mexer um pouco na nossa lista de itens. Dar um espaço e aumentar a largura. Como ele é um elemento do tipo inline block, eu posso dizer que ela tem de largura 20% da página e que ela ainda tem um espaçamento externo para a direita de mais 15%.

[05:00] Se recarregarmos a página, veremos os itens se dividirem melhor. Veremos a imagem ocupando metade do conteúdo, alinhada à nossa lista de itens e alinhada também a esse espaçamento que desejamos no meio.

[05:22] Para finalizar esse layout desses dois blocos, agora que estamos revisando largura e espaçamento, vamos mexer também no espaçamento inteiro dos dois blocos.

[05:35] No primeiro, no Benefícios, vamos adicionar um espaçamento interno para todos os lados de 20px, ou seja, padding: 20px

[05:48] Quando voltarmos ao navegador, veremos as coisas se organizando um pouco melhor. Temos uma margem de respiro em baixo e na lateral.

[05:59] No nosso bloco principal, quero adicionar um espaçamento interno de 30px, padding: 30px

[06:13] Agora nosso conteúdo está melhor distribuído e nosso site começando a ter uma cara mais definitiva.

[06:23] Para finalizar todo o nosso visual, precisamos apenas adicionar um cabeçalho à nossa página. Mas isso é matéria da próxima aula

.

**Nesta aula, aprendemos:**

* **A trabalhar com listas não-ordenadas e listas ordenadas**
* **Para cada um dos itens da lista, utilizamos a tag <li>**
* **O conceito das classes no CSS**
* **Elas servem para marcar itens, só que são repetíveis**
* **Como referenciar uma classe no CSS**
* **Divisões de conteúdo, utilizando a tag <div>**
* **Os comportamentos inline e block**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Cabeçalho

O que queremos agora é criar o cabeçalho da nossa página, onde estarão o nome da nossa barbearia e as informações mais importantes do nosso site.

[00:09] Para isso, no nosso HTML, antes da primeira imagem, dentro do body, vamos criar outra divisão. Essa divisão vai ter um conteúdo. Esse conteúdo vai ser o nome Barbearia Alura.

[00:33] Duas coisas sobre isso: essa divisão é a divisão do cabeçalho. Como isso é uma coisa muito comum em todas as páginas web, na última versão do HTML criaram uma tag específica para o cabeçalho. É uma tag mais semântica que diz para o navegador que aquilo ali é a primeira informação a ser apresentada.

[01:01] Ao invés de div, nós vamos usar a tag header. Ao invés de fechar div, vamos fechar o header.

[01:09] É importante destacar que header é o cabeçalho da página, e fica dentro do conteúdo, e o head é a tag padrão do HTML que serve para passarmos a informação para o navegador. Não vamos confundir as duas coisas.

[01:26] A segunda coisa sobre o cabeçalho é que queremos que o nome da nossa barbearia seja a coisa de mais destaque. Vamos entender como isso funciona no navegador.

[01:37] Salvando e recarregando a página, teremos o nome como a primeira coisa a ser lida pelo usuário. Se pensarmos em qualquer matéria de jornal ou texto, a primeira coisa que lemos é o que está no canto superior esquerdo. Ou seja, este título é o conteúdo principal da nossa página.

[02:02] Sempre que temos o conteúdo principal da página, o título principal, usamos a tag h1. Só que no nosso conteúdo usamos a tag h1 para o título Sobre a Barbearia Alura. Vamos trocar.

[02:17] Sobre a Barbearia Alura passa a ter a tag h2, o Benefícios passa a ter a tag h3, e o nosso principal título, nossa informação principal, passa a ter a tag h1.

[02:40] Se recarregarmos e voltarmos ao nosso site, veremos que ficou uma bagunça. Por quê? Usamos a tag h2 para marcar nosso CSS, usamos a tag h1 para marcar o CSS, e no cabeçalho ganhamos aquele estilo que tínhamos criado sem precisar.

[03:04] Essa é a maior dificuldade quando criamos estilos usando tags. Por isso é sempre recomendado usarmos classe para tudo.

[03:14] Vamos então modificar o conteúdo que já temos para que os estilos que criamos fiquem em tags.

[03:22] A primeira coisa no nosso CSS é que o nosso h1 tinha um alinhamento ao centro. Só que o nosso h2 também tinha a mesma informação. Ou seja, criamos dois estilos para a mesma coisa.

[03:39] Se viermos no nosso HTML, podemos transformar em uma classe, chamando essa classe de “titulo-centralizado”, copiar isso e botar essa mesma classe para nosso h3 e copiando o nome, indo no CSS, posso apagar a declaração do h2, e onde tenho a declaração para o h1, trocar ela para .titulo-centralizado

[04:20] Com isso, temos essa mesma característica sendo aplicada nos dois elementos. Se salvarmos nosso HTML CSS e recarregarmos no navegador, nosso título Barbearia Alura já está posicionado à esquerda, nosso título Sobre a Barbearia Alura, que agora é o h2 está posicionado ao centro, e nossos Benefícios também estão centralizados.

[04:48] Por fim, só para termos mais uma perspectiva visual e mexermos um pouco mais nos espaçamentos, vamos dar um espaço interno no nosso título principal para a esquerda. Ou seja, criaremos um espaçamento de 20px.

[05:09] No nosso HTML, vamos dar uma classe para ele, chamando de titulo-principal. Vamos copiar isso, salvar a página, no nosso CSS vamos criar a classe titulo-principal, que vai ter um espaçamento interno para a esquerda de 20px: padding-left: 20px

[05:38] Salvando esse conteúdo, voltando para o navegador e recarregando a página temos esse respiro do nosso título para nossa página.

[05:50] Com isso, marcamos e criamos todo o conteúdo necessário para a primeira página da Barbearia Alura.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Conclusão

Concluímos com sucesso a criação da página da Barbearia Alura.

[00:03] Tínhamos o desejo de criar essa página e passo a passo conseguimos.

[00:08] Obrigado por me acompanhar nessa jornada. Espero que você tenha aprendido bastante.

[00:14] Nossa página Barbearia Alura já tem agora todo o conteúdo que queríamos. Esse conteúdo não só está bem marcado com as tags do HTML, como está perfeitamente estilizado com o CSS.

[00:25] Vamos rapidamente revisar o que fizemos.

[00:28] No nosso HTML, falamos desde a estrutura do HTML, com as tags do DOCTYPE, HTML, head, body, como falamos também das informações que passamos para o navegador, como por exemplo o link do CSS.

[00:44] Falamos ainda de todas as tags que montam o conteúdo, desde a separação do header e das divisões, falamos de imagem, de títulos, dos textos e listas.

[00:56] Também falamos sobre como marcar esses elementos usando identificadores e classes. Pegamos essa marcação inteira e fomos para o CSS.

[01:06] No CSS, vimos como marcar com uma tag, como marcar com identificador, como marcar com uma classe. E abordamos os detalhes principais do CSS, todas as questões das dimensões, como a largura, altura, padding, a margem. Abordamos a criação de um fundo colorido ou de uma cor no elemento. Mexemos ainda no alinhamento dos elementos, no estilo da fonte.

[01:34] Passamos por isso tudo para chegar na página da Barbearia Alura.

[01:41] Espero que você tenha aprendido muito, mas tenho certeza que você está com várias dúvidas ainda. Por exemplo, como criamos outras páginas e fazemos um link entre elas? Como colocamos um formulário de contato? Um preço? Como vendemos algo da Barbearia Alura? Como podemos criar um rodapé e outros estilos básicos que vemos em todas as páginas da web? Por isso, quero que você me acompanhe na parte 2 desse curso. Espero que você aprenda ainda mais.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Códigos Html:

Texto em negrito

Você pode facilmente adicionar negrito em seus textos, adicionando a tag <strong>. Veja o exemplo abaixo:

<**strong**>Texto em negrito</**strong**>

Links

Links (ou texto âncora) são importantes para ligar conteúdos em documentos HTML. Para criar links em HTML, basta utilizar a tag **<a>**.

Note que é necessário utilizar o atributo href, que mostra para onde o navegador deve direcionar o usuário caso o link seja clicado.

<**a** href="https://portaldesenvolvedor.com">Portal Desenvolvedor</**a**>

Além disso, você tem alguns atributos que podem ser utilizados dentro a tag <a>:

* href=”endereço” Localização do endereço que será direcionado.
* name=”??” Nome do autor
* target=”?” para onde o link será aberto: \_self, \_blank, \_top, \_parent.
* href=”url#bookmark” Adiciona o site aos favoritos.
* href=”mailto:email” É utilizado para criar um link para o e-mail. Dependerá do cliente de e-mail instalado no dispositivo do usuário. No entanto, não é uma boa prática. O ideal neste caso é utilizar um [formulário de contato](https://portaldesenvolvedor.com/blog/como-criar-um-formulario-de-contato-em-html-e-php/).

Listas

Você pode criar listas em HTML facilmente utilizando as tags.

* <ol> … </ol> Lista ordenada
* <ul> … </ul> Lista sem ordem
* <li> … </li> Itens de uma lista. Podem ser utilizado tanto com as tags <ol> quanto <ul>
* <ol type=”?”> Define o tipo de ordenação: A, a, I, i, 1
* <ol start=”??”> Define o valor de início da lista.
* <ul type=”?”> Define o tipo de lista não-ordenada: disc, circle, square
* <li value=”??”> Define o valor do item
* <li type=”??”> Define o tipo do item
* <dl> … </dl> Lista de conceitos
* <dt> … </dt> Termo ou frase
* <dd> … </dd> Definição detalhada sobre o tiem

Imagens

Imagens são importantes para um conteúdo rico e ilustrado. Nós utilizamos a tag <img> para esta finalidade. A propriedade src são atributos que nós podemos utilizar para configurar qual imagem será utilizada.

<**img** src="portaldesenvolvedor.jpg" />

Note que a tag encerra com />. Isto foi uma inovação no XHTML, indica que a marcação é uma instrução completa e não precisa ser encerrada. Nós falaremos mais para frente nos próximos artigos.

Formatação de corpo da página

A tag body é responsável por controlar a aparência básica da página, como cor ou imagem de fundo, cor do texto, margem e muito mais.

Exemplo:

<**body** bgcolor="#FFFFFF">

Além disso, temos os seguintes atributos do comando Body:

* background=”url” – Endereço da imagem de fundo
* bgcolor=”#” – Cor de fundo
* text=”#” – Cor do texto
* link=”#” – Cor do Link
* vlink=”#” Cor do link que foi visitado pelo usuário no mesmo navegador
* alink=”#” Cor do link ativo

Esses atributos podem também ser controlados pelo CSS. Além disso, é uma boa prática de desenvolvimento Web.

Caracteres especiais no HTML

Você também pode utilizar caracteres especiais em seus sites, e desta forma, não serem confundidos com outras tags HTML. Os comandos HTML abaixo podem ser utilizados livremente, sem a necessidade dos sinais < e >.

* &lt; < – Menor que;
* &gt; > – Maior que;
* &amp; & – E comercial;
* &quot; ” – Sinal de citação;
* &copy; © – Copyright;
* &trade; ™ – Trademark;
* &nbsp; – Espaço;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

* A criar uma tabela HTML
* A *tag* table, que representa a tabela
* A *tag* tr, que representa a linha da tabela
* A *tag* td, que representa a célula da tabela
* A *tag* thead, que representa o cabeçalho da tabela
* A *tag* tbody, que representa o corpo da tabela
* A *tag* th, que representa a célula do cabeçalho da tabela
* A *tag* tfoot, que representa o rodapé da tabela
* A estilizar a tabela

Seletores avançados CSS

Seletor >, para acessar os filhos de determinado elemento. Por exemplo, para acessar todos os p dentro de main:

main > p {

Seletor +, para acessar o primeiro irmão de determinado elemento. Por exemplo, para acessar o primeiro p após um img:

img + p {

Seletor ~, para acessar todos os irmãos de determinado elemento. Por exemplo, para acessar todos os p após um img:

img ~ p {

Seletor not, para acessar os elementos, exceto algum. Por exemplo, para acessar todos os p dentro de main, exceto o p que tem id missao:

main p:not(#missao) {

}COPIAR CÓDIGO

Como fazer contas com CSS, com a propriedade calc

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

document.querySelector('span.bstn-hl-title').innerText = 'Virginia Sousa Nasciutti é considerada a melhor funcionária da Algar Tech em Uberlândia.'

DevTools

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Nesta aula, aprenderemos a gerar opacidade em nossos elementos. A opacidade é uma camada a mais posta sobre a imagem, como um insulfilme para a janela de carro, que oferece níveis diferentes de proteção contra o Sol.

Utilizaremos a imagem da sessão "Benefícios" para compreendermos o funcionamento da ferramenta de opacidade no HTML. Em index.css , utilizaremos a propriedade opacity que pode ter valores entre 0 a 1, sendo que 0 o elemento estará totalmente oculto. Se quisermos exibir a imagem com opacidade de 30%, ajustaremos o valor da propriedade para 0.3.

.imagem-beneficios {

width: 60%

opacity: 0.3;

Esse recurso é interessante quando montamos composições visuais com elementos sobrepostos, ou para sugerir interações com o mouse ao utilizarmos o hover. Quando o mouse estiver posicionado sobre a imagem teremos 1 de opacidade.

.imagem-beneficios:hover {

opacity: 1;

Temos ainda o recurso transition, em que podemos estabelecer uma transição da opacidade em determinados milissegundos.

.imagem-beneficios {

width: 60%

opacity: 0.3;

transition: 400ms;

Em nosso site, faremos com o que o comportamento padrão da imagem seja de opacidade 1, e no momento em que mouse estiver sobre ela a opacidade será de 0.3, o que irá gerar um belo efeito visual.

Atualmente, podemos utilizar opacidade em todos os elementos e em cores, vamos entender melhor como isso ocorre.

A cor de titulo-principal possui a cor padrão #00000, preto. Como podemos fazer com que essa cor que está em hexadecimal? se torne RGB? Isto é, a mesma cor escrita em outra linguagem? Do seguinte modo:

.titulo-principal {

text-align: center;

font-size: 2em;

margin: 0 0 1em;

clear: left;

color: rgb(0,0,0);

Com CSS 3, podemos utilizar mais uma camada - a de opacidade, chamada alfa - nas cores em RGB. Para tanto, utilizamos o rgbae então definir os valores que quisermos.

.titulo-principal {

text-align: center;

font-size: 2em;

margin: 0 0 1em;

clear: left;

color: rgb(0,0,0,0.3);

Esse recurso pode ser utilizado em todas as cores: de fundo, texto, borda e assim por diante.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Nesta aula, aprenderemos a gerar opacidade em nossos elementos. A opacidade é uma camada a mais posta sobre a imagem, como um insulfilme para a janela de carro, que oferece níveis diferentes de proteção contra o Sol.

Utilizaremos a imagem da sessão "Benefícios" para compreendermos o funcionamento da ferramenta de opacidade no HTML. Em index.css , utilizaremos a propriedade opacity que pode ter valores entre 0 a 1, sendo que 0 o elemento estará totalmente oculto. Se quisermos exibir a imagem com opacidade de 30%, ajustaremos o valor da propriedade para 0.3.

.imagem-beneficios {

width: 60%

opacity: 0.3;

Esse recurso é interessante quando montamos composições visuais com elementos sobrepostos, ou para sugerir interações com o mouse ao utilizarmos o hover. Quando o mouse estiver posicionado sobre a imagem teremos 1 de opacidade.

.imagem-beneficios:hover {

opacity: 1;

Temos ainda o recurso transition, em que podemos estabelecer uma transição da opacidade em determinados milissegundos.

.imagem-beneficios {

width: 60%

opacity: 0.3;

transition: 400ms;

Em nosso site, faremos com o que o comportamento padrão da imagem seja de opacidade 1, e no momento em que mouse estiver sobre ela a opacidade será de 0.3, o que irá gerar um belo efeito visual.

Atualmente, podemos utilizar opacidade em todos os elementos e em cores, vamos entender melhor como isso ocorre.

A cor de titulo-principal possui a cor padrão #00000, preto. Como podemos fazer com que essa cor que está em hexadecimal? se torne RGB? Isto é, a mesma cor escrita em outra linguagem? Do seguinte modo:

.titulo-principal {

text-align: center;

font-size: 2em;

margin: 0 0 1em;

clear: left;

color: rgb(0,0,0);

Com CSS 3, podemos utilizar mais uma camada - a de opacidade, chamada alfa - nas cores em RGB. Para tanto, utilizamos o rgbae então definir os valores que quisermos.

.titulo-principal {

text-align: center;

font-size: 2em;

margin: 0 0 1em;

clear: left;

color: rgb(0,0,0,0.3);

Esse recurso pode ser utilizado em todas as cores: de fundo, texto, borda e assim por diante.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Nesta aula, aprenderemos sobre as sombras, uma outra novidade que CCS 3 inseriu. Fazer sombras em elementos sempre foi muito difícil nas versões anteriores, hoje em dia, trata-se de algo trivial.

Continuaremos a usar a imagem da sessão de "Benefícios". A sombra é resultado de um efeito de "luz" que lançaremos sobre o elemento. O nome da propriedade que utilizaremos para gerar esse efeito é box-shadow, que possui a propriedade do eixo X, T e uma cor. Queremos deslocar 10px no eixo X e Y, a cor que utilizaremos será preto.

.imagem-beneficios {

width: 60%

opacity: 1;

transition: 400ms;

box-shadow: 10px 10px #000000;

Carregarmos a página, teremos uma sombra projetada, quadrada.

Podemos melhorar a qualidade estética dessa sombra ao adicionarmos uma terceira propriedade chamada blur, em que podemos aplicar um nível de desfoque específico, no caso, inseriremos um valor de 5px. Quanto maior a quantidade de pixels que inserirmos, mais claro sera o efeito de desfoque.

.imagem-beneficios {

width: 60%

opacity: 1;

transition: 400ms;

box-shadow: 10px 10px 5px #000000;

Temos ainda uma quarta propriedade que configura a intensidade da borda a partir do tamanho do elemento, isto é, o tamanho da sombra projetada. Neste caso, inseriremos 20px:

.imagem-beneficios {

width: 60%

opacity: 1;

transition: 400ms;

box-shadow: 10px 10px 5px 20px #000000;

No navegador, a sombra sugirá expandida.

sombragrande

Podemos adicionar várias sombras em um mesmo elemento, basta que o conteúdo esteja separado por uma vírgula. Essa nova sombra terá valores negativos e terá a cor amarela.

.imagem-beneficios {

width: 60%

opacity: 1;

transition: 400ms;

box-shadow: 10px 10px 5px 20px #00000, -10px -10px yellow;

Será projetada uma sombra por debaixo da anterior.

Podemos, ainda, inserir uma terceira sombra com a cor rgba() com a camada de opacidade e borda vermelha.

.imagem-beneficios {

width: 60%

opacity: 1;

transition: 400ms;

box-shadow: 10px 10px 5px 20px #00000, -10px -10px yellow, -20px 20px rgba(255,0,0,0.5);

Ao recarregarmos a página teremos três sombras.

Outra possibilidade no CSS 3 é criar sombras internas. Utilizaremos a própria sessão de benefícios para exempliciar esse efeito, que será utilizado em box-shadowe se chama inset. Seu posicionamento será a partir do centro do elemento e terá a cor vermelha.

.beneficios {

padding: 3em 0;

background: #888888;

box-shadow: inset 0 0 #FF0000;

Ao carregarmos a página, não notaremos qualquer mudança. Isso se deve pelo fato de que a sombra possui o mesmo tamanho do elemento. Para que ela se torne perceptível, criaremos um espaçamento de 30px.

.beneficios {

padding: 3em 0;

background: #888888;

box-shadow: inset 0 0 30px #FF0000;

Feito isso, a sombra será interna em toda a sessão de benefícios.

Apagaremos todas as sombras coloridas e manteremos apenas a sombra leve em imagem-beneficios.

.imagem-beneficios {

width: 60%

opacity: 1;

transition: 400ms;

box-shadow: 10px 10px 10px 0 #000000;

Para fechar o tópico de sombras, por último aprenderemos a inserir sombras em textos. Em titulo-principal, utilizaremos a propriedade text-shadow, que recebem os valores que já conhecemos.

text-shadow: 2px 2px #FF0000;

Será criada uma sombra para o título.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

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

<iframe width="100%" height="315" src="https://www.youtube.com/embed/58TWc1dN0X0" title="YouTube video player" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

criar uma tabela HTML

A tag table, que representa a tabela

A tag tr, que representa a linha da tabela

A tag td, que representa a célula da tabela

A tag thead, que representa o cabeçalho da tabela

A tag tbody, que representa o corpo da tabela

A tag th, que representa a célula do cabeçalho da tabela

A tag tfoot, que representa o rodapé da tabela

A estilizar a tabela

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Como reestruturar o nosso código, removendo os CSS inline e colocando-os no arquivo CSS externo

Como criar um identificador para marcar especificamente um elemento

Como fazer referência a esse identificador no CSS

Como adicionar uma imagem à nossa página

Como ajustar a altura do elemento, através da propriedade height

Como ajustar a largura do elemento, através da propriedade width

Como ajustar o espaçamento interno do elemento, através da propriedade padding

Como ajustar o espaçamento externo do elemento, através da propriedade margin