Por padrão, os elementos <h1> - <h6>, <p> e <header> ocupam 100% da largura de seus pais, e a largura do elemento <body> é igual a largura visível da janela do navegador. Você não precisa tatuar isso em seu braço, mas por favor, tente se lembrar dessa informação

Se você quiser dominar as imagens de fundo, as três propriedades principais que você precisa saber são background-size, background-repeat, e background-position.

Para adicionar opacidade, é necessário deixar com uma cor transparente. Quando for colocar rgb, acrescenta o a ficando rgba e nos valores, no final acrescenta um valor de 0 a 1, que significa quando maior for, mais opaco sera.

Podemos colocar “caixas dentro de caixas” no caso do exemplo dado no exercício, coloquei o cabeçalho h1 dentro do div fiz com que o titulo ficasse mais visível com a opacidade

Herança

Todo conteúdo que recebe uma modificação estando dentro de um elemento.

EX: ao colocar h1, h2 dentro do div e no div tiver configurações feitas na pagina de CSS, os dois elemento h irão receber essas configurações

No elemento body, não é bom ficar mudando muito ou colocar coisas desnecessárias, pois, toda mudança afetara todo conteúdo dentro do body, importante também que o body é colocado depois de alguma mudança padrão

1. font-family declara a família da fonte, juntamente com uma ou mais fontes de backup. Por exemplo, font-family: Inter, Arial, sans-serif. A fonte disponível mais à esquerda será renderizada pelo navegador.
2. font-size declara o tamanho da fonte. Frequentemente, isso é especificado em pixels. Por exemplo, font-size: 16px.
3. font-weight declara o peso, ou negrito, de uma fonte. Pode ser especificado usando números ou com palavras-chave. Exemplos:

Os elementos que você já conhece, como <div>, <section>, <header>, <h1> - <h6> e <p>, ocupam toda a largura de seu pai por padrão, quando colocados no fluxo. Eles são chamados de elementos de nível de bloco, ou block-level.

Este é o modelo básico para cada elemento de nível de bloco:

* O conteúdo do elemento, dimensionado com width e height
* O espaço entre o conteúdo e a borda do elemento: padding
* Bordas, que você ainda não viu no código, mas logo verá
* O espaço entre a borda do elemento e outros elementos: margin

Box shadow

Aqui está um detalhamento dos valores aplicados à propriedade na ordem:

* Horizontal offset (se o número for -2px, então a sombra lançará 2px para a esquerda da caixa. Se for 2px, lançará 2px para a direita)
* Vertical Offset (se o número for -2px, então a sombra lançará 2px acima da caixa. Se for 2px, a sombra lançará 2px abaixo)
* Blur radius (raio de desfoque, quanto maior o número, mais larga e pálida é a sombra)
* Color (cor, atribuída da mesma forma que o texto ou cor de fundo)

Às vezes, não há necessidade de mover a sombra, e embaçá-la é o suficiente. Nesse caso, basta definir os dois primeiros valores iguais a 0.

Footer

Parte de baixo de uma pagina, geralmente ficam informações de contato, menus etc.

Pode ser criado após fechamento da tag </section>

**Centralização vertical**

Você já conseguiu centralizar o título "QUATRO REGRAS DE DESIGN" verticalmente. No entanto, daquela vez, você definiu valores específicos para a propriedade padding para mover o texto para o centro.

Embora possa parecer a maneira mais fácil de fazer as coisas, nem sempre funciona. Se você definir o preenchimento como 100px em um dispositivo pequeno, pode ocupar um terço da tela. No entanto, em um monitor grande de 4K, você nem vai notar. Dito isso, tente evitar o ajuste de tamanhos em pixels, porque as pessoas visitarão seu site em todos os tipos de dispositivos diferentes com vários tamanhos de tela, então você precisa de uma abordagem que leve isso em consideração.

A humanidade está à beira de enviar pessoas para Marte, mas ainda não descobrimos uma forma de centralizar automaticamente um elemento?

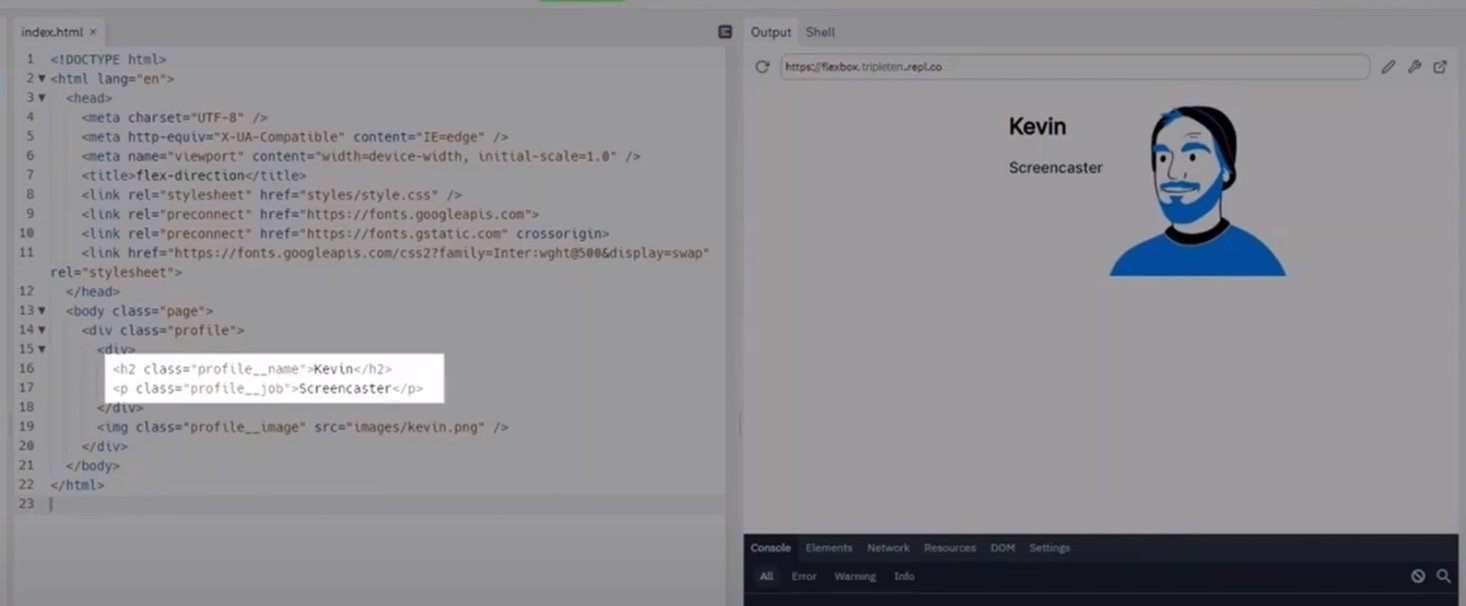
Notícias de última hora! Na verdade, existe uma maneira. Você já conhece a propriedade display. Em vez de block, inline ou inline-block, tente aplicar o valor flex, de flexível:

Copiar códigoCSS

display: flex;

Um elemento que possui essa propriedade torna-se imediatamente um elemento flexível e ganha superpoderes como um super-herói. Por exemplo, quando margin: auto; é colocado dentro desse super elemento, ele não funcionará apenas para centralização horizontal, mas também para centralização vertical! Pode parecer meio doido, mas é verdade. A seguir, tentaremos centralizar o texto em nosso rodapé para entender melhor esse conceito.

Existem também atalhos de teclado para acessar diferentes partes da interface do DevTools:  
Para abrir o DevTools e retornar à aba que você usou por último, use Command+Option+I ou Control+Shift+I no Windows/Linux.  
Para abrir a aba Inspector e Elements, use Command+Option+C ou Control+Shift+C no Windows/Linux

.No caso dessa imagem, foi alinha com combinando display:flex com tirando uma parte do código sem ser alinha com display:flex, foi adicionado uma nova tag, para isolar os textos e coloca-los como “1” elemento dentro do div, com isso o novo div que possue os elementos de texto virou um elemento e só do div recebeu o flex, dentro dele não foi recebido o fles, com isso o texto ficou um embaixo do outro.

**Note que quando a direção é em coluna, justify-content muda para a vertical e align-items para a horizontal.**