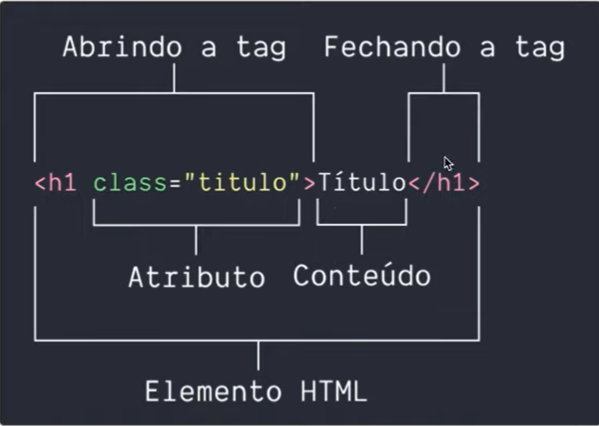
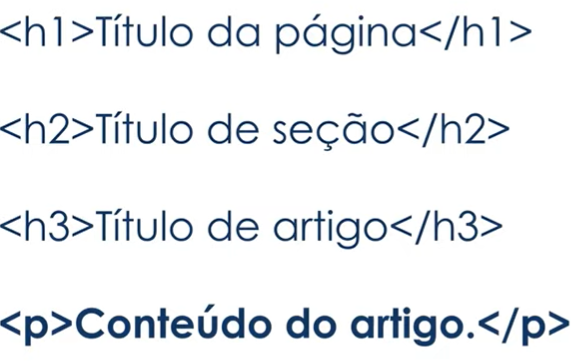
# HTML

* Definição
  + 5 versões
* Elemento
  + Base do html, tudo é elemento
    - Tag de abertura
    - atributo
    - conteúdo
    - tag de fechamento
    - 
* Estrutura
  + 

# HTML – SEMANTICA 1 e 2

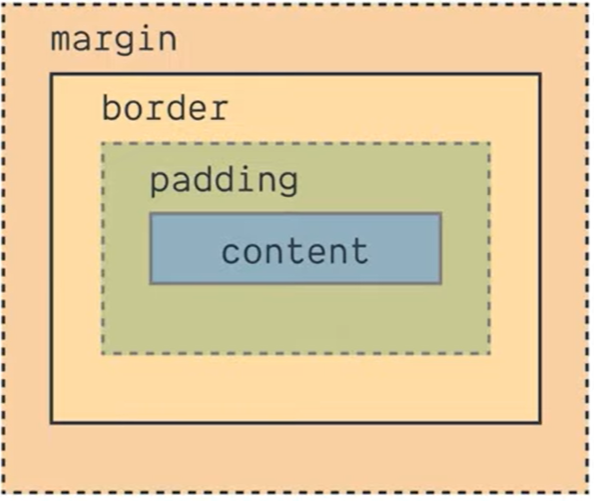
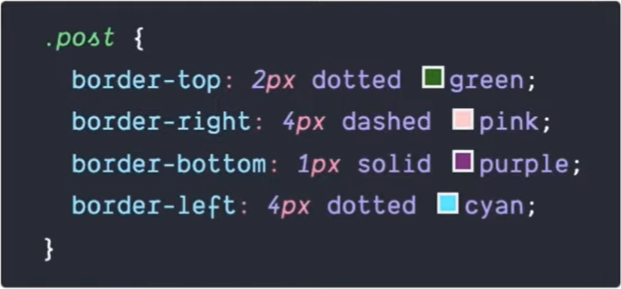
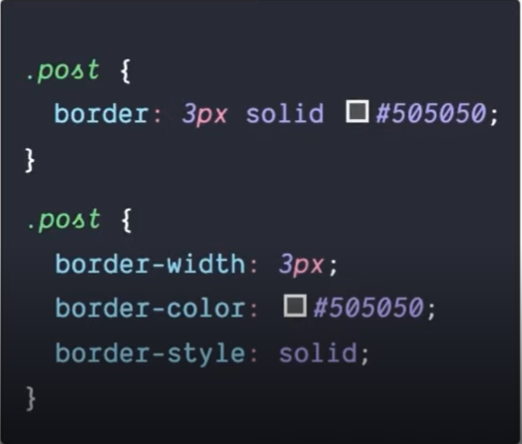
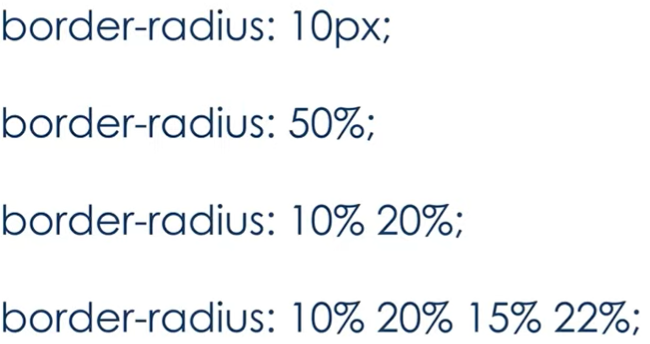
* ATÉ 2014 SO EXISTIA a tag div para criação de elementos, a partir de 2014 foram introduzidas outros elementos já estruturados:
* <section>
  + representa sesao genérica
  + lista de artigos
* <header>
  + cabeçalho de uma pagina ou de parte dela
* <article>
  + representa conteúdo relevante
  + artigo de um blog
* <aside>
  + conteúdo relacionado ao conteúdo principal
  + biografia
  + representado por barra lateral
* <footer>
  + rodapé da pagina ou de parte dela
* <h1>-h6>
  + importância de um titulo dentro da pagina
  + só 1 h1 por pagina
* Textos e links
  + 
  + para textos maiores e densos utilizamos o elemento p
    - p representa um paragrafo que pode ser texto, imagem, vídeo etc
  + elemento a
    - ancora, interliga vários conteudos
    - href – hiperlink para aonde meu site esta apontando, paginas e-mails e telefones
      * usado com prefixo mailto e tel
    - target – indica como o link vai ser aberto \_blank(abre link em uma nova aba
  + elemento img
    - img src=”img/avatar.jpg”>
      * obrigatório e guarda o caminho da foto
    - img alt=”Foto de Fabao”>
      * mostra descrição da foto quando não e carregada, leitores de tela podem usar essa descrição par mostrar aos usuários o que a imagem significa
  + Listas
    - Agrupar coleção de itens
    - <ul>
      * lista onde a ordem dos itens não é importante
    - <ol>
      * ordem dos itens é importante
    - <li>
      * item da lista

# CSS3

Linguagem de estilo

regras de estilo para elementos e grupos de elementos



* Id X Classe
  + no html
    - <header id=”header” class = “header></header>
    - <header class = “heaer”></header>
  + no CSS
    - id = .header{padding:10px}
    - class = #heaer{}
  + id so pode ser usado uma vez na pagina
* box model, navegador representando minha pagina
  + 
  + margin
    - espaçamento entre elementos
  + border
    - circundam o paddin e conteúdo
    - mudamos largura e cor
  + padding
    - entre a borda e o conteuso
    - padding:10px
      * borda para todos os lados
    - padding: 10px 5px;
      * 10px eixo y, bordas de cima e de baixo
      * 5px eixo x, bordas na esquerda e direira
    - paddingo: 15px 10px 5px 0
      * topo, direita, inferior esquerdo
    - pading-top: 15px;
    - padding-right: 10px;
    - padding-bottom: 10px;
    - padding-left: 10px;
  + content
    - conteúdo em si, texto, imagens, vídeo
  + background-color: “green”; nome da cor ou exadecimal
    - background-image: url(”caminho”);
    - background-position: top;
  + 
  + 
  + 
  + 
  + border raius
    - arredonda os cantos
  + 
* Estilizando textos
  + font-family
    - altera a fonte do texto
    - font-family: “nome fonte”, “nome da fonte”;
      * se dois nomes o segundo serve como backup
  + font-size
    - altera o tamanho do texto
    - font-size: 15px;
  + font-style
    - altera aparência do texto
    - font-style: normal;
      * normal – tamanho padrão
      * italic – itálico
  + font-weight
    - altera o peso do texto
    - font-weight: normal;
    - font-weight: bold; - negrito
  + text-transform
    - text-transform: uppercase;
    - lowercase
    - capitalize – primeira letras maiuculas
  + text-decoration: “nome”
    - coloca sublinhado
    - underline – abaixo
    - overline – acima
    - line-through – tachado
* list-style-type
  + estilizando listas
  + list-style-type: square;
    - altera marcador para quadrado na lista não ordenada
  + list-style-type: Upper-roman;
    - romano maiúsculo na lista ordenada
  + list-style-type: “\1F44D”;
    - coloca símbolos na lista não ordenada (joinha)
  + list-style-image: url(“caminho da imagem”);
    - adiciona imagem em lista não ordenadas
* DIMENSÃO E ALINHAMENTO
  + Width - largura
  + Heigh – altura
  + Max-width – largura máxima
  + max-heigh – altura máxima
  + margin – espaçamento entre elementos
  + text align – alinhar textos
    - left, righ, center, justify

# FLEXBOX EM CSS

* FLEXBOX
  + modelo de layout unidimensional, oferece distribuição de espaço entre os itens
  + estrutuba básica
  + flex container
    - tag que envolve os itens transformando-os em filhos – display: flex
    - display
      * inicializador do container
      * aplicável para qualquer tipo de tag no html
    - flex-direction
      * direcionamento dos itens seja em linha ou coluna
      * Entender e modificar a orientação horizontal
      * eixo linha – horizontal –
        + row – da esquerda para a direita
        + row-reverse – ordenação inversa
      * eixo coluna vertical
        + column

eixo vertical

ordenação de cima para baixo ordenado

* + - * + column reverse

eixo vertical

de cima pra baixo com a ordem reversa dos itens

* + - flex-wrap
      * define quebra de linha ou não do container – por padrão não quebram linha
      * itens vazam do container
      * wrap-reverse
    - flex-flow
      * abreviação para o direct ou wrap
    - justify-content
      * alinha itens do container de acordo com sua direção preenchendo os espaços
      * flex-start – inicio do container
      * flex-end – final do container
      * center – ao centro do container
      * space-between – cria espaçamento igual entre os elementos
      * space-around – espaçamento do meio duas vezes maior que o inicial e final
    - align-items
      * alinha itens de acordo com o eixo do container
      * alinhamento é diferente para linhas e colunas
      * permite alinhamento no eixo vertical, no meio da tela
      * center – tras itens para o centro
      * stretch – padrão – itens crescam igualmente
      * flax-start – alinhamento dos itens no inicio do containar
      * flax-end – final do container
      * baseline – alinhamento de acordo com a linha base da tipografia dos itens
    - align content
      * alinha as linhas do container – trata alinhamento vertical
      * necessário:
        + Container utilizar quebra de linha (wrap)
        + Altura do container seja maior que a soma das linhas dos itens
      * Center
      * stretch – representando cresimento padrão
      * flex-start
      * flex-end
      * space-between
      * space-around
  + flex item
    - elementos filhos diretos do container
    - flex-grow
      * define fator de crescimento, a proporcionalidade de crescimento dos itens, respeitando o tamanho de seus conteúdos – não funciona com justify-content
      * sempre uzamos números, 0 ou a proporcionalidade de crescimento
    - flex-basis
      * define tamanho inicial desse item antes da distribuição do espaço do container
      * tem relação com os flex-itens
      * estabelece tamanho inicial dos itens, antes da distribuição de espaço restante dentro dele, usando como base o conteúdo disposto
        + pega o tamanho mínimo que o item irá precisar e depois faz a distribuição da sobra de espaço entre o item e o conteúdo
      * valor 0 – terá relação com flex grow
    - flex-shrink
      * define a capacidade de redução ou compressão do tamanho de um item
    - flex
      * define a ação para grow, shrink e basis
    - order
      * ordem de distribuição e listagem dos itens
    - align-self
      * define o alinhamento de um item especifico do container de maneira individual
      * auto: valor padrão
      * flex-start – inicio do container
      * flex-end – final do container
      * center – centro de acordo com o eixo
      * strech – elemento ocupa todo o espaço
      * baseline – usa linha base da tipografia