CSS

**FUNDO**

background -> Define todos os parâmetros referentes ao fundo de um elemento.

* background-color: define a cor de fundo de um elemento
  + valores: cor| #rrggbb(hexadecimal)| #rgb| rgb(n,n,n|n%,n%,n%)| transparent
* background-image: define uma imagem como fundo de um elemento
  + valores: url()| none
* background-repeat: define se a imagem vai repetir ou não e a direção da repetição
  + valores: repeat| repeat-x| repeat-y| no-repeat
* background-attachment: define se a imagem ficará fixa ou vai rolar com a página
  + valores: scroll| fixed
* background-position: define o ponto onde inicia o desenho da imagem de fundo
  + valores: top(left|center|right)| center(left|center|right)|bottom(center|right) | x unid\_css y unid\_css
* background-size: define o tamanho de uma imagem
  + valores: px px (width height)

**BORDAS**

border-> Define o estilo, a espessura e a cor das 4 bordas (top| right| bottom| left).

* border-style: define o estilo da borda
  + valores: none| hidden| dotted| dashed| solid| double| groove| ridge| inset| unset
* border-width: define a espessura da borda
  + valores: thin| medium| thick| unid\_css
* border-color: define a cor da borda
  + valores: cor| #rrggbb(hexadecimal)| #rgb| rgb(n,n,n|n%,n%,n%)
* border-radius: define a curvatura da borda
  + valores: px| %
* border-image: define uma imagem de borda
  + valores: url round| url stretch| url repeat
    - border-image-source: define uma imagem como a borda em torno de um elemento
      * + valores: none| image| initial| inherit
    - border-image-width: define a largura da imagem de borda
      * + valores: length(px)| número| %| auto| initial| inherit
    - border-image-slice: define como cortar a imagem da borda
      * + valores: número| %| (%) fill| initial| inherit
    - border-image-outset: define o espaço para fora da borda do elemento
      * + valores: length| número| initial| inherit

LINHAS DE CONTORNO

outline -> Define uma linha que é desenhada em torno de elementos (fora das bordas) para tornar o elemento "destacado". **Nota**: *O contorno não é uma parte das dimensões do elemento, portanto as propriedades de largura e altura do elemento não contêm a largura do contorno, sempre joga para fora a largura se aumentada*.

* outline-color: define a cor de uma linha pontilhada
  + valores: invert| color| inital
* outline-width: define a espessura da linha
  + valores: medium| thick| thin| initial| px
* outline-style: define o estilo da linha
  + valores: none| hidde| dotted| dashed| solid| double| groove| ridge| inset| ouset| initial| inherit
* outline-offset: define um contorno 15px fora da borda
  + valores: length| initial| inherit

MARGENS

margin -> Define uma margem para um elemento (top| right| bottom| left).

* margin-top: define uma margem superior ao elemento
* margin-right: define uma margem direita ao elemento
* margin-bottom: define uma margem inferior ao elemento
* margin-left: define uma margem esquerda ao elemento
  + valores: %| px| pt| cm| auto|initial| inherit

ESPAÇOS EM BRANCO

padding -> Define os prenchimentos ou o espaço entre o conteúdo e os 4 limites de um elemento.

* padding-top: define uma margem superior ao elemento
* padding-right: define uma margem direita ao elemento
* padding-bottom: define uma margem inferior ao elemento
* padding-left: define uma margem esquerda ao elemento
  + valores: %| px| pt| cm| auto|initial| inherit

EXIBIÇÃO

position -> Define a posição de um elemento.

* static: valor padrão, elementos renderizados por ordem, conforme aparecem no fluxo de documentos
* relative: o elemento é posicionado em relação à sua posição normal, então "left: 20px" adiciona 20 pixels à posição LEFT do elemento
* absolute: o elemento é posicionado em relação ao seu primeiro elemento pai posicionado (não estático)
* fixed: o elemento é posicionado em relação à janela do navegador

overflow -> Define o que deve ocorrer quando o conteúdo de um elemento excede sua área.

* visible: define que a sobra do elemento ficará visivel
* hidden: define que a sobra do elemento ficará escondida
* scroll: define que aparecerá barras de rolagem para ver todo o conteúdo do elemento
* auto: define que aparecerá uma barra de rolagem na horizontal para ver o conteúdo
  + overflow-x: define o que fazer com as bordas esquerda / direita do conteúdo
  + overflow-y: define o que fazer com as bordas supeior / inferior do conteúdo

display -> Define se um elemento deve ser mostrado e a sua forma de exibição.

* valores: inline| block| inline-block| flex| inline-flex| compact| marker| run-in| table| inline-table| table-row| table-row-group| table-header-group| table-footer-group| table-column| tabble-column-group| table-cell| table-caption| none| initial| inherit

float -> Define o alinhamento de um elemento e controla o fluxo de texto ou de outros elementos ao seu redor.

* valores: left| right| none| initial| inherit

clear -> Define os lados de um elemento onde não são permitidos elementos flutuantes.

* valores: none| left| right| both| initial| inherit

cursor -> Define o tipo de cursor que será exibido.

* valores: alias| all-scroll| auto| cell| col-resize| copy| context-menu| crosshair| default| e-resize| grab| help| move| n-resize| ne-resize| nw-resize| pointer|   
  progress| s-resize| se-resize| sw-resize| text| w-resize| wait| not-allowed| no-drop| vertical-text| grabbing| zoom-in| zoom-out

visibility -> Define se um elemento deve ficar visível ou escondido.

* valores: visible| hidden| collapse(apenas tabelas)

backface-visibility -> Define se o fundo de um elemento deve ficar visível ou escondido quando animado em rotação.

* valores: visible| hidden| collapse(apenas tabelas)

opacity -> Define o nível de opacidade de um elemento.

* valores: número (0-1)| initial| inherit

clip -> Define a forma de um elemento.

* valores: auto| shape(rect (px, px, px, px))| unset

POSICIONAMENTO

top, right, bottom, left -> Define a distância do elemento em relação ao qual está contido.

* valores: %| px| pt| cm| auto|initial| inherit

vertical-align -> Define o alinhamento vertical do elemento.

* valores: px| %| baseline| sub| sup| top| text-top| middle| bottom| text-bottom| initial| inherit

z-index -> Define a ordem de apresentação dos elementos quanto a sobreposição entre outros elementos . A propriedadade z-index só funciona em elementos posicionados com posição absoluta, relativa ou fixa.

* valores: auto| número| initial

DIMENSIONAMENTO

width, height, max-width, max-height, min-width, min-height -> Define a largura e a altura de um elemento, tanto máximo quanto mínimo valores podem ser estabelecidos.

* valores: auto| none| px| %| cM

BOX

box-sizing -> Define qual o posicionamento de um elemento dentro de um box, é definido através de width, height e padding do elemento a ser posicionado, porém a soma do width deste elemento tem que ser igual ao do box.

* valores: border-box

box-shadow -> Define uma sombra para um box.

* valores: v-shadow(px) h-shadow(px) blur-radius(px) cor

FONTE

font -> Define os valores para cada fonte utilizada.

* font-family: define a família de fontes, o valor padrão depende do navegador
  + valores: nome da família| nome genérico
* font-size: define o tamanho da fonte, o valor padrão é "normal"
  + valores: medium| xx-small| x-small| small| large| x-large| xx large| smaller | larger| initial| inherit| %| px| cm| em
    - font-size-adjust: melhor controle do tamanho da fonte quando a primeira fonte selecionada não está disponível.
      * valores: none| número| initial| inherit
    - font-stretch: permite-lhe tornar o texto mais amplo ou mais estreito.
      * Valores: condensed| semi-condensed| ultra-condensed| extra-condensed| | normal| expanded| semi-expanded| extra-expanded| ultra-expanded| initial| inherit| normal | wider| narrower
* font-style: define o estilo da fonte, o valor padrão é "normal"
  + valores: normal| italic| oblique| initial| inherit
* font-weight: define o peso da fonte, o valor padrão é "normal"
  + valores: normal| bold| bolder| lighter| número(100-900)| initial| inherit
* font-variant: efine se um texto deve ou não ser exibido em uma fonte small-caps
  + valores: normal| small-caps| initial| inherit;
* caption-side: define a colocação de uma legenda de tabela.
* icon: usa a fonte que é usada por etiquetas de ícone
* message-box: usa as fontes que são usadas por caixas de diálogo
* small-caption: uma versão menor da fonte de legenda
* status-bar: usa as fontes que são usadas pela barra de status
* menu: Usa as fontes usadas pelos menus suspensos
* @font-face: Na nova regra @ font-face, você deve primeiro definir um nome para a fonte (por exemplo, myFirstFont) e, em seguida, apontar para o arquivo de fonte

TEXTO

color -> Define a cor do texto.

* valores: cor| #rrggbb(hexadecimal)| #rgb| rgb(n,n,n|n%,n%,n%)

line-height -> Define a distância entre as linha do texto.

* valores: normal| número| %| px| cm

text-align - > Define o alinhamento do texto.

* valores: left| center| right| justify
  + text-align-last: define o alinhamento do último paragráfo do texto
    - valores: auto| left| right| center| justify| start| end| initial| inherit;

text-indent - > Define a identação do primeiro paragráfo do texto.

* valores: %| px| cm| initial| inherit

text-decoration - > Define a decoração adicionada ao texto.

* valores: none| underline| overline| blink| line-through| initial| inherit
  + text-decoration-color: define a cor da decoração de texto
    - valores: color| initial| inherit
  + text-decoration-line: define que tipo de linha, se houver, a decoração terá.
    - valores: none| underline| overline| line-through| initial| inherit
  + text-decoration-style: definecomo a linha, se houver, será exibida.
    - valores: solid| double| dotted| dashed| wavy| initial| inherit;

text-transform - > Define se o texto deve ser capitalizado ou todo em maíusculo ou minúsclulo.

* valores: none| capitalize| uppercase| lowercase| initial| inherit

text-shadow - > Define um sombra ao texto. Esta propriedade aceita uma lista separada por vírgulas de sombras a serem aplicadas ao texto.

* valores: v-shadow(px) h-shadow(px) blur-radius(px) cor|none| initial|inherit

text -overflow- > Define como a sobra do elemento que não é exibida deverá ser mostrada ao usuário.

* valores: clip| ellipsis| string| initial| inherit

text-justify -> Define como o **texto justificado** deve ser alinhado e espaçado.

* valores: auto| inter-word| inter-ideograph| inter-cluster| distribute| kashida| trim| initial| inherit

direction -> A propriedade direção especifica a direção do texto / direção de escrita.

* valores: ltr(left to right)| rtl(right to left)| initial| inherit;

white-space -> Define como o espaço em branco dentro de um elemento é tratado.

* valores: normal| nowrap| pre| pre-line| pre-wrap| initial| inherit;

white-space -> Define como o espaço em branco dentro de um elemento é tratado.

* valores: normal| nowrap| pre| pre-line| pre-wrap| initial| inherit;

word-spacing, letter-spacing -> Define um aumento ou recuo de espaço entre as palavras.

* valores: normal| %| px| cm| initial| inherit;

word-wrap -> Define que palavras longas possam ser quebradas e se encaixe na próxima linha.

* valores: normal| break-word| initial| inherit;

word-break -> Define regras de quebra de linha para scripts não-CJK(chinês, japonês, coreano).

* valores: normal| break-all| keep-all| initial| inherit;

QUEBRA DE PÁGINA

page-break -> Define onde devem ocorrer as quebras de páginas dentro de um documento.

* page-break-before: Define se um quebra de página deve ocorrer ANTES de um elemento especificado. Esta propriedade não pode ser usada em um <div> vazio ou em elementos com posição absolute.
  + valores: auto|always|avoid|left|right|initial|inherit;
* page-break-after: Define se um quebra de página deve ocorrer DEPOIS de um elemento especificado. Esta propriedade não pode ser usada em um <div> vazio ou em elementos com posição absolute.
  + valores: auto|always|avoid|left|right|initial|inherit
* page-break-inside: Define se uma quebra de página é permitida dentro de um elemento especificado. Esta propriedade não pode ser usada em elementos com posição absolute.
  + valores: auto|avoid|initial|inherit

LISTAS E MARCADORES

list-style-type -> Define o tipo de marcador de item da lista.

* valores: none| disc| circle| square| decimal| decimal-leading-zero| lower-roman| upper-roman| lower-alpha| upper-alpha| lower-greek| lower-latin| upper-latin| hebrew| armenian| georgian| cjk-ideographic| hiragana| katakana| hiragana-iroha| katakana-iroha

list-style-position -> Define a posição do marcador de item da lista.

* valores: inside| outside

list-style-image -> Define uma imagem como marcador de item da lista.

* valores: none| url

TABELAS

border-collapse -> Define se os traços da borda serão fundidos ou mantidos separados.

* valores: collapse| separate

border-spacing - > Define a distância entre as bordas de células adjacentes. Só aplicavél quando border-collapse for separate.

* valores: %| px| cm

empty-cells -> Define se as células sem conteúdo visível devem ter bordas ou não. Só aplicavél quando border-collapse for separate.

* valores: show| hide

caption-side -> Define a posição da legenda na tabela.

* valores: top| bottom| left| right

table-layout -> Define como será desenhada a tabela.

* valores: auto| fixed

GERAÇÃO DE CONTEÚDO

content -> Define conteúdo para um elemento. Usada com pseudo-elementos (before| after).

* valores: string| url| counter(name)| counter(name| list-style-type)| counters(name| string)| counters(name| string| list-style-type)| atrr(X)| open-quote| close-quote| no-open-quote| no-close-quote

counter-inrement-> Define em quanto será incrementado o contador a cada ocorrência de um seletor.

* valores: none| número do identificador

counter-inrement-> Define o valor a ser atribuído ao contador a cada ocorrência de um seletor.

* valores: none| número do identificador

quotes-> Define o estilo das aspas.

* valores: none| string string

PSEUDO-CLASSES

* a:link – estado do link no carregamento da página
* a:hover – estado quando o mouse está sobre o link
* a:active – estiliza o link no ato do clique
* a:visited – estiliza o link já visitado
* input:checked – seleciona todos os inputs verificados
* input:disabled – seleciona todos os inputs desabilitados
* input:enabled - seleciona cada input habilitado
* input:focus – seleciona o input que tem foco
* input:in-rage – seleciona o input com um valor dentro de um intervalo especificado
* input:valid – seleciona todos os inputs com um atributo válido
* input:invalid – seleciona todos os inputs com um valor inválido
* input:optional – seleciona os input sem o atributo "required"
* input:out-of-range – seleciona os inputs com um valor fora de um intervalo especificado
* input:read-only – seleciona o input com um atributo "readonly" especificado
* input:read-write – seleciona o input sem o atributo "readonly"
* input:required – seleciona o input com um atributo "required" especificado
* p:empty – seleciona cada elemento p que não tem filhos
* p:first-child – seleciona todos os elementos p que é o primeiro filho de seu pai
* p:first-of-type – seleciona cada elemento p que é o primeiro elemento p de seu pai
* p:lang(br) – seleciona cada elemento p com um valor de atributo lang inicado por "br"
* p:last-child – seleciona todos os elementos p que é o último filho de seu pai
* p:last-of-type – seleciona cada elemento p que é o último elemento p de seu pai
* p:nth-child(2) – seleciona cada elemento p que é o segundo filho de seu pai
* p:nth-last-child(2) – seleciona cada elemento p que é o segundo filho de seu pai, contando com o último filho
* p: nth-last-of-type(2) – seleciona cada elemento p que é o segundo elemento p de seu pai, contando com o último filho
* p:nth-of-type(2) – seleciona cada elemento p que é o segundo elemento p de seu pai
* p:only-of-type – seleciona cada elemento p que é o único elemento p de seu pai
* p:only-child – seleciona cada elemento p que é o único filho de seu pai
* not:p – seleciona cada elemento que não é um elemento p
* #news: target – seleciona o elemento atual #news ativo (clicado em um URL contendo esse nome de âncora)
* root – seleciona o elemento raiz do documento

PSEUDO-ELEMENTOS

* p::after – insere algo após o conteúdo de cada elemento p
* p::before – insere algo antes do conteúdo de cada elemento p
* p::first-letter – seleciona a primeira letra de cada elemento p
* p::first-line – seleciona a primeira linha de cada elemento p
* p::selection – seleciona a parte de um elemento que é selecionado pelo usuário

CORES

nome -> Define uma cor para um elemento através do seu nome.

* valores: red| blue| green| black| white| orange| yellow

hexadecimal -> Define uma cor para um elemento através de uma combinação hexadecimal.

* valores: #FF0000| #0000FF| 00FF00| #000000| #FFFFFF| #FF6600| #FFFF00

rgb -> Define uma cor através de uma combinação de vermelho(red) azul(blue) e verde(green) aumentando ou diminuindo suas tonalidades respectivas que vai de 0-255.

* valores: rgb(X , X, X)

rgba -> Define uma cor através de uma combinação de rgb com opacidade que vai de 0-1.

* valores: rgba (X, X, X, N)

hsl -> Define uma cor através de uma combinação de matiz(hue) saturação(saturation) e lightness(luminosidade).

* valores: hsl(º, %, %)

hsla -> Define uma cor através de uma combinação hsl com opacidade que vai de 0-1.

* valores: hsl(º, %, %, N)

hwb -> Define uma cor através de uma combinação matiz(hue) palidez(whiteness) e escuridão(blackness).

* valores: hsl(º, %, %)

cmyk -> Define uma cor através de uma combinação de ciano(cyan),magenta, amarelo(yellow) e preto(black).

* valores: cmyk(%, %, %, %)

linear-gradient -> Define um gradiente de cores linear de um elemento.

* to top: para cima
* to bottom: para baixo
* to left: para a esquerda
* to right: para a direita
* to top left: para o topo e para a esquerda
* to top right: para o topo e para a direita
* to bottom left: para baixo e para a esquerda
* to bottom right: para baixo e para a direita
* graus: pode-se definir a direção do gradiente através de graus
  + valores: podem ser combinados rgb, rgba, hsl, hsla, opacidade, etc...

radial-gradient -> Define um gradiente de cores radial(centro para fora) de um elemento.

* valores: X %, X %, X %

shape -> Define uma forma para o gradiente.

* valores: circle| ellipse

size -> Define o tamanho do gradiente.

* valores: px| cm

fechamento -> Define a forma que deve ser fechado o gradiente.

* closest-side: a forma de fechamento do gradiente encontra o lado do elemento mais próximo ao seu centro (para círculos) ou encontra ambos os lados vertical e horizontal mais próximos ao centro (para elipses)
* closest-corner: a forma de fechamento do gradiente tem o tamanho exato para encontrar o canto da caixa mais próximo ao seu centro
* farthest-side: a forma de fechamento fica com o tamanho necessário para encontrar o lado do elemento mais distante do centro do gradiente (ou lados vertical e horizontal).
* farthest-corner: a forma de fechamento do gradiente tem o tamanho exato para encontrar o canto da caixa mais distante de seu centro.

FILTROS

filter -> Permite aplicar filtros, antes vistos apenas em programas como Photoshop, diretamente no browser com CSS. Podendo-se combinar valores entre os filtros.

* none: não aplica os filtro
* blur: aplica um efeito de borrão
  + valores: px
* brightness: aplica um efeito de brilho
  + valores: %
* contrast: aplica um efeito de contraste
  + valores: %
* drop-shadow: aplica um efeito de sombreamento
  + valores: v-shadow(px) h-shadow(px) blur-radius(px) cor
* grayscale: aplica um efeito de tons de cinza
  + valores: %
* sepia: aplica um efeito de sépia(amarronzado)
  + valores: %
* invert: aplica um efeito de inversão de tonalidades
  + valores: %
* opacity: aplica um efeito de opacidade
  + valores: %
* hue-rotate: aplica um efeito de acordo com o ângulo da propriedade hue da cor
  + valores: º
* saturate: aplica um efeito de saturação
  + valores: (%)

MODO DE MISTURA

mix-blend-mode -> Define como o conteúdo de um elemento deve se misturar com seu plano de fundo. Isso significa que qualquer imagem ou texto, bordas ou títulos serão influenciados por esta propriedade.

* initial: a configuração padrão da propriedade que não configura um modo de mistura.
* inherit: um elemento herdará o modo de mistura do seu elemento pai.
* unset: remove o modo de mistura atual de um elemento.
* normal: este atributo não aplica nenhuma mistura.
* multiply: o elemento é multiplicado pelo fundo e substitui a cor de fundo. A cor resultante é sempre tão escura como o plano de fundo.
* screen: multiplica o plano de fundo e o conteúdo então complementa o resultado. Isso resultará em conteúdo que é mais brilhante do que a cor de fundo.
* overlay: multiplica ou exibe o conteúdo dependendo da cor de fundo. Este é o inverso do modo de mistura de luz dura.
* darken: o fundo é substituído pelo conteúdo em que o conteúdo está mais escuro, caso contrário fica como foi.
* lighten: o fundo é substituído pelo conteúdo onde o conteúdo é mais leve.
* color-dodge: este atributo ilumina a cor de fundo para refletir a cor do conteúdo.
* color-burn: isso escurece o plano de fundo para refletir a cor natural do conteúdo.
* hard-light: dependendo da cor do conteúdo, esse atributo irá exibi-lo ou multiplicá-lo.
* soft-light: dependendo da cor do conteúdo, isso irá escurecer ou iluminá-lo.
* difference: isso subtrai a mais escura das duas cores da cor mais clara.
* exclusion: semelhante à diferença, mas com menor contraste.
* hue: cria uma cor com a tonalidade do conteúdo combinada com a saturação e a luminosidade do fundo.
* saturation: cria uma cor com saturação do conteúdo combinada com a tonalidade e a luminosidade do fundo.
* color: cria uma cor com a tonalidade e saturação do conteúdo e a luminosidade do fundo.
* luminosity: cria uma cor com a luminosidade do conteúdo e a tonalidade e saturação do plano de fundo. Este é o inverso do atributo de cor.

Vale ressaltar que definir um modo de mistura diferente do normal gerará um novo contexto de empilhamento que deve ser combinado com o contexto de empilhamento que contém o elemento.

FLEX-BOX

display -> Define que cada elemento filho se comportará como bloco.

* valores: flex

flex-direction - > Define a direção dos elementos.

* valores: row| row-reverse| column| column-reverse

flex-wrap -> Define se os itens flexíveis devem ser compactados ou não.

* valores: wrap| nowrap| wrap-reverse

justify-content ->Alinha os itens do container quando os itens não usam todo o espaço disponível no eixo principal (horizontalmente).

* flex-start: os itens são posicionados no início do container
* flex-end: os itens são posicionados no final do container
* center: os itens são posicionados no centro do container
* space-between: os itens são posicionados com espaço entre as linhas
* space-around: os itens são posicionados com espaço antes, entre, e depois das linhas

align-items -> Alinha verticalmente os itens flexíveis quando os itens não usam todo o espaço disponível no eixo transversal.

* stretch: os itens são esticados para caber no container
* center: os itens são posicionados no centro do container
* baseline: os itens são posicionados na linha de base do container
* flex-start: os itens são posicionados no início do container
* flex-end: os itens são posicionados no final do container

align-content -> Modifica o comportamento da propriedade flex-wrap. É semelhante aos itens de alinhamento, mas ao invés de alinhar itens flexíveis, alinha linhas flexíveis. Deve haver várias linhas de itens para que esta propriedade tenha algum efeito. Só funciona com a propriedade flex-wrap recebendo o valor wrap.

* stretch: as linhas se estendem para ocupar o espaço restante
* center: as linhas são posicionadas em direção ao centro do container
* flex-start: as linhas são alinhadas no início do container
* flex-end: as linhas são alinhadas no início do container
* space-between: as linhas são distribuídas uniformemente no container
* space-around: as linhas são distribuídas uniformemente no container, com espaços de tamanho médio em cada extremidade

align-self -> Especifica o alinhamento para o item selecionado dentro do container.

* auto: o elemento herda a propriedade de itens de alinhamento do container principal ou "stretch" se não tiver um recipiente pai
* stretch: o elemento está posicionado para caber no container
* center: o elemento está posicionado no centro do container
* baseline: o elemento está posicionado na linha de base do container
* flex-start: o elemento está posicionado no início do container
* flex-end: o elemento está posicionado no início do container

flex-grow -> Especifica o quanto o item crescerá em relação ao resto dos itens flexíveis dentro do mesmo container.

* valores: números

flex-shrink -> Especifica o quanto o item encolherá em relação ao resto dos itens flexíveis dentro do mesmo container.

* valores: números

flex-basis -> Especifica o comprimento inicial de um item flexível.

* valores: número| auto

order -> Especifica a ordem de um item flexível em relação ao resto dos itens flexíveis dentro do mesmo container. É alinhado por números.

* valores: item: nth-child(1){ order: 2;}

flex-flow -> É uma propriedade abreviada para as propriedades flex-direction e flex-wrap.

MÚLTIPLAS COLUNAS

multi-column -> Permite a fácil definição de múltiplas colunas de texto.

* column-count: especifica o número de colunas em que um elemento deve ser dividido.
* column-gap: especifica o espaço entre as colunas
* column-fill: especifica como preencher colunas
* column-rule-style: especifica o estilo da linha entre as colunas
* column-rule-width: especifica a largura da linha entre as colunas
* column-rule-color: especifica a cor da linha entre as colunas
* column-rule: propriedade abreviada para definir todas as propriedades de linha
* column-span: especifica quantas colunas um elemento deve abranger
* column-width: especifica uma largura sugerida e ideal para as colunas
* columns: propriedade abreviada para definir a largura da coluna e a contagem de colunas

TRANSIÇÕES

transition -> Propriedade abreviada para definir as quatro propriedades de transição em uma única propriedade. Podem ser combinadas conforme a necessidade. Para haver o efeito de transição é necessário que seja descrito pelo menos dois parâmetros.

* valores: propriedades/css duração timing-function delay|initial|inherit
  + exemplo – transition: width 2s -> propriedade e duração foram usados

transition-property -> Especifica o nome da propriedade CSS para o efeito de transição.

* valores: none| all| propriedades/css| initial| inherit
  + exemplo – transition: width 2s -> propriedade/css e duração foram usados

transition-duration -> Especifica quantos segundos ou milissegundos (ms) um efeito de transição leva para concluir sua ação.

* valores: tempo(s| ms)| initial| inherit
  + exemplo – transition: width 2s -> propriedade/css e duração foram usados

transition-delay -> Define quando o efeito de transição começará.

* valores: tempo(s| ms)| initial| inherit
  + exemplo – transition: width 1.5s -> propriedade/css e delay foram usados

transition-timing-function -> Especifica a curva de velocidade do efeito de transição.

* ease: especifica um efeito de transição com um início lento, depois rápido, e depois termina lentamente (equivalente ao bezier cúbico (0,25,0.1,0.25,1))
* linear: especifica um efeito de transição com a mesma velocidade do início ao fim (equivalente ao bezier cúbico (0,0,1,1))
* ease-in: especifica um efeito de transição com um início lento (equivalente ao bezier cúbico (0,42,0,1,1))
* ease-out: especifica um efeito de transição com uma extremidade lenta (equivalente a bezier cúbico (0,0,0,58,1))
* ease-in-out: especifica um efeito de transição com um início e fim lento (equivalente ao bezier cúbico (0,42,0,0,58,1))
* steps(int,start|end): especifica uma função de pisar, com dois parâmetros. O primeiro parâmetro especifica o número de intervalos na função. Deve ser um número inteiro positivo (maior que 0). O segundo parâmetro, que é opcional, é o valor "iniciar" ou "fim", e especifica o ponto em que a mudança de valores ocorre dentro do intervalo. Se o segundo parâmetro for omitido, é dado o valor "fim"
* cubic-bezier(n,n,n,n): define seus próprios valores na função bezier cúbico. Os valores possíveis são valores numéricos de 0 a 1

TRANSFORMAÇÕES

transform -> Aplica uma transformação 2D ou 3D a um elemento. Esta propriedade permite que você gire, dimensione, mova, incline, etc., elementos.

* origin: posição de origem do elemento transformado
  + valores: eixo-x eixo-y eixo-z(3D) | initial | inherit
    - eixo-x: left| center| right| length| %
    - eixo-y: top| center| bottom| length| %
    - eixo-z: length
* rotate: roaciona um elemento no sentido horário ou anti-horário de acordo com um determinado grau ou eixo
  + valores: (º)
* scale: aumenta ou diminui o tamanho(escala) de um elemento (de acordo com os parâmetros dados para a largura e a altura)
  + valores: (2, 3) -> altura e largura
* translate: move o elemento no seu eixo x(horizontal) e y(vertical)
  + valores: (x y)
* skew: desvia ângulos para inclinação do elemento
  + valores: (º)
* matrix: define uma transformação 2D, usando uma matriz de seis valores
  + valores: (scaleX(), translateX(), skewX(), scaleY(), translateY(), skewY())

Pode-se usar apenas um valor (X ou Y e Z (3D)) nas propriedades scale, translate e skew declarando a proprieadade scaleX() ou scaleY() por exemplo. Dentro dos parênteses é utilizado graus.

transform-style -> Especifica como os elementos aninhados são renderizados no espaço 3D.

* flat: especifica que os elementos filhos NÃO conservarão sua posição 3D. Isso é padrão
* preserve-3d: especifica que os elementos filho preservarão sua posição 3D

perspective -> Define quantos pixels um elemento 3D terá. Esta propriedade permite que você altere a perspectiva sobre como os elementos 3D são vistos. Ao definir a propriedade de perspectiva para um elemento, são os elementos FILHOS que obtêm a vista em perspectiva, NÃO o próprio elemento. Afeta apenas elementos transformados em 3D!

* valores: length| none

perspective-origin -> Define onde um elemento 3D se baseia no eixo x e y. Esta propriedade permite que você altere a posição inferior dos elementos 3D. Ao definir a propriedade de origem de perspectiva para um elemento, são os elementos FILHOS que estão posicionados, NÃO o elemento em si. Afeta apenas elementos transformados em 3D, os valores padrão são de 50%.

* valores: eixo-x eixo-y
  + eixo-x: left| center| right| length| %
  + eixo-y: top| center| bottom| length| %

KEYFRAMES

@keyframes -> Especifica o código de animação. A animação é criada mudando gradualmente de um conjunto de estilos CSS para outro. Durante a animação, você pode alterar o conjunto de estilos CSS muitas vezes. Especifique quando a mudança de estilo acontecerá em porcentagem, ou com as palavras-chave from(de) e to(para), que é igual a 0% e 100%. 0% é o início da animação, 100% é quando a animação é completa.

* animation-name: define um nome para a animação
* keyframes-selector: porcentagem de duração de efeitos da animação
* css-styles: requerido uma ou mais propriedades de estilo CSS

|  |
| --- |
| **@keyframes**  Exemplo{  from {top: 0px;}  to {top: 200px;}  } |

|  |
| --- |
| **@keyframes**  Exemplo{  0% {top: 0px;}  50% {top: 200px;}  100% {top: 0px;}  } |

**OU**

animation-delay -> Especifica um atraso para o início de uma animação. O valor de atraso de animação é definido em segundos (segundos) ou milissegundos (ms).

* valores: s| ms

animation-direction -> Especifica se uma animação deve ser executada na direção inversa ou alternar ciclos.

* normal: valor padrão. A animação deve ser renderizada como normal
* reverse: a animação deve ser executada na direção inversa
* alternate: A animação será renderizada como normal em todos os momentos ímpares (1,3,5, etc.) e na direção inversa a cada momento par (2,4,6, etc ...)
* alternate-reverse: a animação será renderizada na direção inversa a cada momento ímpar (1,3,5, etc.) e em uma direção normal a cada momento par (2,4,6, etc ...)

animation-duration -> Especifica quantos segundos ou milissegundos leva a uma animação para completar um ciclo.

* valores: s| ms

animation-fill-mode -> Especifica um estilo para o elemento quando a animação não está sendo reproduzida (quando ele é concluído ou quando tem um atraso).

* none: Valor padrão. a animação não aplicará nenhum estilo ao elemento alvo antes ou depois que ele estiver executando
* forwards: após a finalização da animação (determinada pela animation-iteration-count), a animação aplicará os valores da propriedade para o tempo que a animação terminou
* backwards: a animação aplicará os valores de propriedade definidos no quadro-chave que iniciará a primeira iteração da animação, durante o período definido pelo atraso de animação. Estes são os valores do quadro-chave (quando a direção da animação é "normal" ou "alternativo") ou a do quadro-chave (quando a direção da animação é "inversa" ou "alternada-reversa")
* both: a animação aplicará os valores de propriedade definidos no quadro-chave que iniciará a primeira iteração da animação, durante o período definido pelo atraso de animação. Estes são os valores do quadro-chave (quando a direção da animação é "normal" ou "alternativo") ou a do quadro-chave (quando a direção da animação é "inversa" ou "alternada-reversa")

animation-iteration-count ->Especifica o número de vezes que uma animação deve ser reproduzida.

* valores: números| infinite

animation-play-state -> Especifica se a animação está em execução ou em pausa.

* valores: paused| running

animation-timing-function -> Especifica a curva de velocidade da animação.

* linear: a animação tem a mesma velocidade do começo ao fim
* ease: a animação tem um início lento, depois rápido, terminando devagar
* ease-in: a animação tem um início lento
* ease-out: a animação tem um fim lento
* ease-in-out: a animação tem um início lento e um fim lento

animation -> Maneira curta de se configurar um animação.

* valores: name duration timing-function delay iteration-count direction fill-mode play-state

SELETORES CSS

elemento - > Seleciona um elemento <tag> pelo seu nome.

* exemplo – p -> seleciona todas as tags p do documento

elemento, elemento -> Seleciona vários elementos para definir regras css.

* exemplo – p, h1 -> aplica o mesmo estilo css na tag p e na tag h1

elemento elemento -> Seleciona elementos dentro de elementos anteriores

* exemplo – div p -> seleciona todos elementos p dentro da tag div

elemento > elemento -> Seleciona o elemento filho do elemento anterior.

* exemplo – div > p -> seleciona apenas o primeiro p dentro da tag div

elemento + elemento -> Seleciona imediatamente depois do elemento anterior.

* exemplo – h1 + p -> seleciona elementos p após elementos h1

\* -> Seleciona todos os elementos do documento.

* exemplo - \* -> seleciona todos elementos da página.

. -> Seleciona um elemento pelo valor do atributo class.

* exemplo – .titulo -> aplica estilo css aos elementos definidos com a classe titulo

# -> Seleciona um elemento pelo valor do atributo id.

* exemplo - #titulo -> aplica estilo css ao elemento definido com o id titulo (o id é único)

SELETORES POR ATRIBUTOS

[atributo] -> Seleciona o elemento que tem o atributo setado.

* exemplo – a[target] -> seleciona todos os elementos com o atributo target

[atributo = valor] -> Seleciona elementos que tenham o atributo setado com o valor igual ao valor setado.

* exemplo – a[target = \_self] -> seleciona todos os elementos com o atributo target com valor \_self

[atributo ~= valor] -> Seleciona o elementos que tenham o atributo setado e que tenham uma descrição setada.

* exemplo – a[title ~= exemplo] -> seleciona todos os elementos com o atributo title que tenham a palavra exemplo

[atributo |= valor] -> Seleciona elementos que tenham o atributo setado e que o valor comece com o valor setado.

* exemplo – a[title = exemplo] -> seleciona todos os elementos com o atributo title e que a primeira palavra comece com exemplo