

CSS

CASCADING STYLE SHEETS

AVISO

- Alunos e amigos.. esta apostila ainda está em desenvolvimento e será atualizada conforme andamentou ou necessidade dentro do curso.

FUNDAMENTOS

- CSS é a sigla para Cascading Style Sheets que significa Folhas de Estilos em Cascata. São formadas por regras, geralmente em arquivos separados e servem para aplicar estilos em documentos HTML puros como já pudemos ver anteriormente.

FUNDAMENTOS

- Existem três tipos de CSS:
 - INLINE: Utilizado através do atributo `STYLE` da própria TAG a ser estilizada.
 - INCORPORADO: As regras são escritas entre as TAGs `<STYLE>` `</STYLE>`
 - EXTERNO: Regras escritas em arquivo externo com a extensão `.css` e chamado através da TAG `<LINK>` - `<link rel="stylesheet" href="caminho_do_arquivo">`

FUNDAMENTOS - REGRAS

- Para aplicar um estilo em um determinado elemento do documento HTML temos as regras responsáveis por esse fim.
- **Regra:** seletor, declaração e propriedade no seguinte formato:
- seletor{ propriedade: valor; } Exemplo: p { color : red; }

SELETORES

- Temos os seguintes seletores:
- Seletor universal;
- Seletor de tipo;
- Seletor de atributo;
- Seletor de id;
- Seletor de class;

COMENTÁRIOS

- Os comentários servem para colocar lembretes, documentar trechos de regras ou simplesmente para esconder do navegador a interpretação de alguma regra ou trecho de regra específico:
- `H2 {color: blue;}`
- `/* P {font-size: 15px} */`
- No trecho acima somente o elemento h2 será estilizado. Já o parágrafo não será interpretado pelo navegador, pois está comentado.

DIMENSÃO – WIDTH HEIGHT

- WIDTH – propriedade para dimensionar a largura dos elementos estilizados. Por padrão os elementos recebem width com tamanho de 100%, dentro do elemento pai;
 - `div {width: 500px;}` ou `div {width: 70%;}`
- HEIGHT: propriedade para dimensionar a altura dos elementos estilizados. Por padrão segue-se o tamanho do conteúdo;
 - `div {height: 100px}`

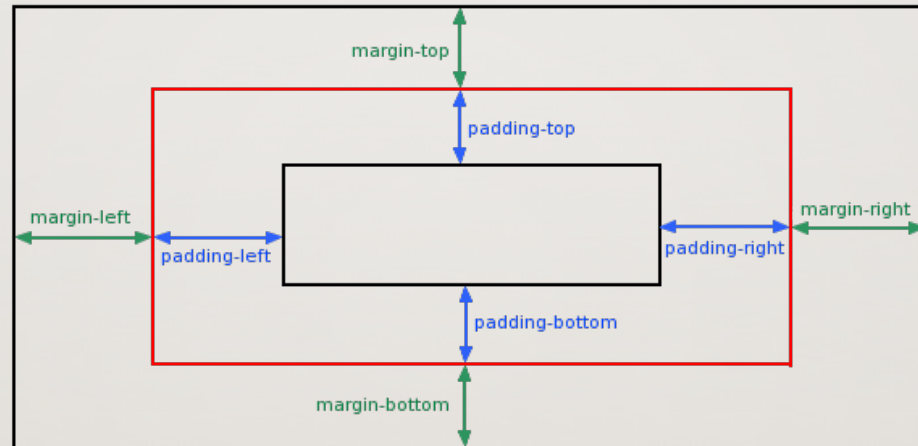
DIMENSÃO MIN E MAX

- MAX-WIDTH – define a largura máxima do elemento
- MIN-WIDTH – define a largura mínima do elemento
- MAX-HEIGHT – define a altura máxima do elemento
- MIN-HEIGHT – define a altura mínima do elemento
- Ex: `div {max-width : 500px; }`

MARGIN E PADDING

- Margins e paddings são utilizados para definir uma margem para o elemento em relação aos elementos externos (margin) ou elementos internos (padding), a partir da borda, veja:
- `div {margin-top: 20px; margin-right: 20px; margin-bottom: 20px; margin-left: 20px;}`
- `div {padding-top: 20px; padding-right: 20px; padding-bottom: 20px; padding-left: 20px;}`

MARGIN E PADDING



Margin e padding são somados ao tamanho (width) do elemento

BORDER

- Seguindo o mesmo conceito de de margin e padding, BORDER também está relacionado ao elemento e segue as 4 posições sentido horário;



- `div {margin-top: 20px; margin-right: 20px; margin-bottom: 20px; margin-left: 20px;}`

BORDER RADIUS

- BORDER-RADIUS :Altera o ângulo das bordas.

- border-top-left-radius
- border-top-right-radius
- border-bottom-right-radius
- border-bottom-left-radius

```
.cantos-arredondados {  
    border-top-left-radius:    2px;  
    border-top-right-radius:   2px;  
    border-bottom-right-radius: 2px;  
    border-bottom-left-radius: 2px;  
}
```

- Ou simplesmente border-radius;

```
.cantos-arredondados {  
    border-radius: 2px 2px 2px 2px;  
}
```

BOX MODEL

- Todos os elementos html podem ser considerados como caixas que contém margin, padding, borda e o conteúdo interno.
- O conceito box model está relacionado ao layout. Falamos de box model para pensar na estrutura, disposição dos elementos na página.

<H2>BEM VINDO</H2>



BOX-SIZING

- A propriedade de dimensionamento de caixa (box-sizing) é usado para informar ao navegador que as propriedades de dimensionamento (width e height) devem ser incluídas, de acordo com o valor da propriedade:
- VALORES:
- content-box (default) – width e height só são calculados considerando apenas o conteúdo da caixa;
- **border-box** – width e height incluem o padding e a borda para o seu cálculo,
- padding-box - width e height incluem o tamanho padding size, mas não incluem o tamanho da borda ou margem

UNIDADES DE MEDIDA

EM, PX, PT, CM, IN...

CSS oferece um número de unidades diferentes para a expressão de comprimento. Algumas têm sua história em tipografia, como o ponto (**pt**) e a paica (**pc**), outros são conhecidos devido a seu uso diário, como centímetro (**cm**) e polegada (**in**). E há, também, uma unidade "mágica" que foi inventada especificamente para CSS: **px**. Isso significa que diferentes propriedades precisam de diferentes unidades?

Não, as unidades não têm nada a ver com as propriedades, mas têm tudo a ver com a mídia de saída: tela ou papel.

Não há qualquer restrição sobre quais unidades podem ser usadas em qual lugar. Se uma propriedade aceita um valor em **px** ('margin: 5px') ela também aceita o valor em polegadas ou centímetros ('margin: 1.2in; margin: 0.5cm') e vice-versa.

Mas, em geral, você deve usar um conjunto diferente de unidades para exibição em tela do que para a impressão em papel. A tabela abaixo apresenta o uso recomendado:

	Recomendado	Uso ocasional	Não recomendado
Tela	em, px, %	ex	pt, cm, mm, in, pc
Impresso	em, cm, mm, in, pt, pc, %	px, ex	

A relação entre as unidades absolutas é: 1in = 2,54cm = 25,4mm = 72pt = 6pc

UNIDADE EM

- A unidade relativa EM tem seu padrão de 16px em todos os navegadores. Você pode fazer uma comparação da seguinte forma:
- Imagine que você gostou da fonte em tamanho 40px. Você pode calcular $40/16 = 2,5$. Ou seja pode usar 2.5em.
- `.meu-texto { font-size : 2.5em }` 😊

TEXT

- COLOR: define cor do texto: ex color: #ccc;
- TEXT-ALIGN: **left**, right, center, justify ex: text-align: center;
- TEXT-DECORATION: none, underline, overline, line-through;
- TEXT-TRANSFORM: uppercase, lowercase, capitalize;
- TEXT-INDENT: recuo na primeira linha, ex: p{**text-indent: 10px;**}
- LINE-HEIGHT: altura da linha, ex: p{**line-height: 20px;**}
- DIRECTION: Direção do texto – rtl(direita pra esquerda) e ltr(esquerda para direita), ex: p{**direction:ltr;**}
- LETTER-SPACING: espaçamento entre as letras, ex: p{**letter-spacing: -2px**}
- WORD-SPACING: espaçamento entre as palavras, ex: ex: p{**word-spacing: -2px**}
- TEXT-SHADOW (horizontal,vertical, **desfoque**, cor): adiciona sombra às letras, ex: .letra{**text-shadow: 3px 3px 4px red**}

OVERFLOW E OPACITY

- OVERFLOW: **visible**, hidden, scroll, auto.
- OPACITY: varia de 0 a 1 – 0 é ausência de opacidade e 1 é total opacidade.

DISPLAY POSITION FLOAT Z-INDEX

- DISPLAY: modo visualização do conteúdo.. Inline, block, etc.
- POSITION: esquema de posicionamento de elemento relativo ao seu elemento anterior (pai) ou ao browser.
- FLOAT: posicionamento flutuante do elemento.
- Z-INDEX: Profundidade do elemento – quem vem primeiro.. Quem fica atrás.

FLOAT

- A propriedade float deixa o elemento em modo flutuante. Você escolhe o lado: direito ou esquerdo.
 - LEFT: Flutua o elemento a esquerda.
 - RIGHT: esquema de posicionamento de elemento relativo ao seu elemento anterior (pai) ou ao browser.
 - CLEAR: anula o efeito do float

POSITION

- Especifica o tipo de método utilizado para o posicionamento de um elemento. Os elementos são posicionados através das propriedades TOP, BOTTOM, LEFT e RIGHT, após você inserir POSITION, com exceção do static que só aceita o posicionamento através das margins(margin-xtop).
- **STATIC:** valor **padrão** que acompanha o fluxo dos elementos. Por exemplo, uma div já tem o posicionamento estático ao ser criada
- **ABSOLUTE:** deixa posição absoluta, a partir do browser.
- **FIXED:** posiciona o elemento fixamente na tela do dispositivo. Ele não roda com o scroll da página.
- **RELATIVE:** posiciona os elementos relativamente a ele, ou seja, dentro de si. Os posicionamentos de seus filhos serão em relação a ele. Exemplo: um elemento filho com posicionamento absoluto, será absoluto ao seu pai e não mais ao browser. LOKO, não!!!?