

CURSO HTML 5

MÓDULO 10



GINEAD

www.ginead.com.br

AVISO DE RESPONSABILIDADE

As informações contidas neste material de treinamento são distribuídas “NO ESTADO EM QUE SE ENCONTRAM”, sem qualquer garantia, expressa ou implícita. Embora todas as precauções tenham sido tomadas na preparação deste material, a Alfamídia Prow não tem qualquer responsabilidade sobre qualquer pessoa ou entidade com respeito à responsabilidade, perda ou danos causados, ou alegadamente causados, direta ou indiretamente, pelas instruções contidas neste material ou pelo software de computador e produtos de hardware aqui descritos.

Módulo 1 – Visão geral do HTML5

Módulo 2 – Análise do suporte atual pelos navegadores

Módulo 3 – Estrutura básica, DOCTYPE e charsets

Módulo 4 – Modelos de conteúdo

Módulo 5 – Novos elementos e atributos

Módulo 6 – Elementos modificados e ausentes

Módulo 7 – Elementos áudio e vídeo, e codecs

Módulo 8 – Menus e toolbars

Módulo 9 – Tipos de links

Módulo 10 – Introdução ao CSS3

Módulo 10 – Introdução ao CSS3

Seletores complexos

A sintaxe do CSS é simples:

```
seletor {  
    propriedade: valor;  
}
```

A propriedade é a característica que você deseja modificar no elemento. O valor é o valor referente a esta característica. Se você quer modificar a cor do texto, o valor é um Hexadecimal, RGBA ou até mesmo o nome da cor por extenso. Até aqui, nada muito diferente. Muitas vezes você não precisa aprender do que se trata a propriedade, basta saber que existe e se quiser decorar, decore. Propriedades são criadas todos os dias e não é um ato de heroísmo você saber todas as propriedades do CSS e seus respectivos valores. Os seletores são a alma do CSS e você precisa dominá-los. É com os seletores que você irá escolher um determinado elemento dentro todos os outros elementos do DOM para formatá-lo. Boa parte da inteligência do CSS está em saber a utilizar os seletores de uma maneira eficaz, escalável e inteligente. Os seletores complexos selecionam elementos que talvez você precisasse fazer algum script em Javascript para poder marcá-lo com uma CLASS ou um ID para então você formatá-lo. Com os seletores complexos você consegue formatar elementos que antes eram inalcançáveis. exemplo de funcionamento Imagine que você tenha um título (h1) seguido de um parágrafo (p). Você precisa selecionar todos os parágrafos que vem depois de um título H1. Com os seletores complexos você fará assim:

```
h1 + p {  
    color:red;  
}
```

- Gradiente

O gradiente será feito no background. Você define a direção do gradiente, onde ele começará e terminará, e quais as cores que farão a transição.

Veja abaixo como fazemos um gradiente linear.

```
.classe {  
    /* Firefox */  
    background: -moz-linear-gradient(top, rgba(0,0,0,1) 0%, rgba(188,33,36,1) 100%);  
  
    /* Chrome e Safari */  
    background: -webkit-linear-gradient(top, rgba(0,0,0,1) 0%,rgba(188,33,36,1) 100%);
```

```
/* Opera */
background: -o-linear-gradient(top, rgba(0,0,0,1) 0%,rgba(188,33,36,1) 100%);

/* IE6-9 */
filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#000000',
endColorstr='#bc2124',GradientType=0 );
}
```

Veja abaixo como fazemos um gradiente radial, infelizmente o Internet Explorer não fornece suporte a essa funcionalidade.

```
.classe {

background: -webkit-radial-gradient(50% 50%, cover, rgb(0, 0, 0) 0%, rgb(255, 0, 0) 100%);

background: -moz-radial-gradient(50% 50%, cover, rgb(0, 0, 0) 0%, rgb(255, 0, 0) 100%);

background: -ms-radial-gradient(50% 50%, cover, rgb(0, 0, 0) 0%, rgb(255, 0, 0) 100%);

background: -o-radial-gradient(50% 50%, cover, rgb(0, 0, 0) 0%, rgb(255, 0, 0) 100%);

}
```

Exemplo completo no documento.

[exemplo_gradiente.html](#)

- Columns

Com o controle de colunas no CSS, podemos definir colunas de texto de forma automática. Até hoje não havia maneira de fazer isso de maneira inteligente com CSS e o grupo de propriedades `columns` pode fazer isso de maneira livre de gambiarras.

column-count

A propriedade `column-count` define a quantidade de colunas terá o bloco de textos.

```
/* Define a quantidade de colunas, a largura é definida uniformemente.*/
-moz-column-count:2;
-webkit-column-count:2;
```

column-width

Com a propriedade `column-width` definimos a largura destas colunas.

HTML5

/* Define qual a largura mínima para as colunas. Se as colunas forem espremidas, fazendo com que a largura delas fique menor que este valor, elas se transformam em 1 coluna automaticamente */

```
-moz-column-width: 400px;  
-webkit-column-width: 400px;
```

column-gap

A propriedade column-gap cria um espaço entre as colunas, um gap.

/* Define o espaço entre as colunas. */

```
-moz-column-gap: 50px;  
-webkit-column-gap: 50px;
```

Utilizamos aqui os prefixos -moz- e -webkit-, estas propriedades não funcionam oficialmente em alguns browsers. Mas já podem ser usados em browsers como Firefox, Safari e Opera.

Exemplo completo no documento.

[exemplo_colunas.html](#)

- Transform

Veja o código abaixo e seu respectivo resultado:

```
img {  
  -webkit-transform: skew(30deg); /* para webkit */  
  -moz-transform: skew(30deg); /* para gecko */  
  -o-transform: skew(30deg); /* para opera */  
  transform: skew(30deg); /* para browsers sem prefixo */  
}
```

O código acima determina que o ângulo da imagem seja de 30deg. Colocamos um exemplo para cada prefixo de browser. Ficando assim:

Com a propriedade aplicada

Imagem original



Várias transformações em um único elemento

Para utilizarmos vários valores ao mesmo tempo em um mesmo elemento, basta definir vários valores separando-os com espaços em uma mesma propriedade transform:

```
img {  
  -webkit-transform: scale(1.5) skew(30deg); /* para webkit */  
  -moz-transform: scale(1.5) skew(30deg); /* para gecko */  
  -o-transform: scale(1.5) skew(30deg); /* para opera */  
  transform: scale(1.5) skew(30deg); /* para browsers sem prefixo */  
}
```

transform-origin

A propriedade transform-origin define qual o ponto do elemento a transformação terá origem. A sintaxe é idêntica ao background-position. Observe o código abaixo

```
img {  
  -webkit-transform-origin: 10px 10px; /* para webkit */  
  -moz-transform-origin: 10px 10px; /* para webkit */  
  -o-transform-origin: 10px 10px; /* para webkit */  
  transform-origin: 10px 10px; /* para webkit */  
}
```

- Transições e animações

Com a versão anterior do CSS, formatavam-se fontes, backgrounds, cores, tamanhos e medidas de distâncias e posições, agora existe a possibilidade de animar elementos, não ao ponto de criar filmes, para isso tem as opções Canvas e SVG. Vale lembrar que existe a possibilidade de algumas funções não funcionar em alguns navegadores. Antes de utilizar alguma função como padrão em algum projeto, verifique nos principais browsers.

Abaixo exemplo de utilização:

HTML5

```
a {
  color: white;
  background: gray;
}

a {
  color:white;
  background:green;
  -moz-transition: 0.5s linear; /* Firefox */
  -webkit-transition: 0.5s linear; /* Safari and Chrome */
  -o-transition: 0.5s linear; /* Opera */
}

a:hover {
  color:black;
  background:yellow;
  -moz-transition: 0.5s linear; /* Firefox */
  -webkit-transition: 0.5s linear; /* Safari and Chrome */
  -o-transition: 0.5s linear; /* Opera */
}
```

Código para testar:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
<meta charset="utf-8">
<title>Transition</title>
<style>
a {
  font-size:36px;
  padding:10px;
  color:white;
  background:green;
  -moz-transition: 0.5s linear; /* Firefox */
  -webkit-transition: 0.5s linear; /* Safari and Chrome */
  -o-transition: 0.5s linear; /* Opera */
}
a:hover {
  color:black;
  background:yellow;
  -moz-transition: 0.5s linear; /* Firefox */
  -webkit-transition: 0.5s linear; /* Safari and Chrome */
  -o-transition: 0.5s linear; /* Opera */
}
</style>
</head>
<body>
  <a href="#">Alfamidia</a>
</body>
</html>
```


Exemplo completo no documento.
`exemplo_transition.html`

- Bordas

Utilizar uma imagem como background da borda que o desenvolvedor definiu, mais uma novidade interessante da versão do CSS3. Como de costume, a aplicação é simples:

```
.classe {  
    display: block;  
    background: #f0f0f0;  
    height: 300px;  
    width: 300px;  
    border-image: url(bg.png) 10 10 10 10 stretch;  
}
```

Abaixo alguns exemplos de utilização:

```
#exemplo_a {  
    height: 65px;  
    width: 160px;  
    -moz-border-radius-bottomright: 50px;  
    border-bottom-right-radius: 50px;  
}
```

```
#exemplo_b {  
    height: 65px;  
    width: 160px;  
    -moz-border-radius-bottomright: 50px 25px;  
    border-bottom-right-radius: 50px 25px;  
}
```

```
#exemplo_c {  
    height: 65px;  
    width: 160px;  
    -moz-border-radius-bottomright: 25px 50px;  
    border-bottom-right-radius: 25px 50px;  
}
```

```
#exemplo_d {  
    height: 5em;  
    width: 12em;  
    -moz-border-radius: 20px 80px 20px 80px;  
    border-radius: 20px 80px 20px 80px;  
}
```

```
#exemplo_e {  
    height: 65px;
```

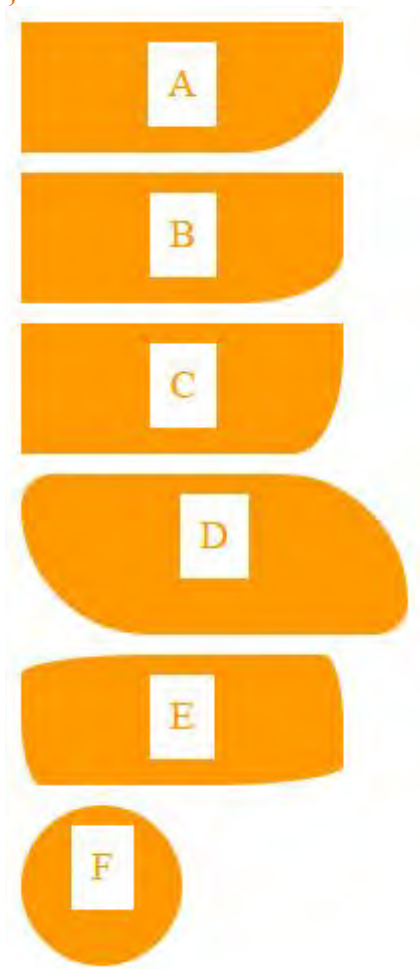
HTML5

```

width:160px;
-moz-border-radius: 35px 10px / 10px 35px;
border-radius: 65px 10px / 10px 35px;
}

#exemplo_f {
height: 80px;
width: 80px;
-moz-border-radius: 40px;
border-radius: 40px;
}

```



Exemplo completo no documento.
[exemplo_bordas.html](#)

- Múltiplos backgrounds

```

body {
text-align:center;
background:
url(bg.png) left repeat-y,
url(bg2.png) right repeat-y;
}

```

```
}
```

Exemplo completo no documento.
exemplo_multiplo_bg.html

Sobre posicoes de backgrounds

Exemplo de como sobrepor imagens,

```
body {
    text-align:center;
    min-height:600px;
    background:
    url(bg_01.png) left top no-repeat, /* 2 */
    url(bg_02.png) right bottom no-repeat, /* 1 */
    /*url(bg_01.png) right bottom no-repeat, */
    url(bg_00.png); /* 0 */
}
```

Exemplo completo no documento.
exemplo_multiplo_bg2.html

- Módulo Template Layout

A propriedade FLOAT, é uma peça muito importante na criação de layouts com CSS, porem não é a maneira correta de estruturação.

Template Layout, esse modulo criado pela w3c, proporciona uma nova forma de criar, organizar os elementos e as demais informações do layout.

O módulo Template Layout basicamente define slots de layout para que você encaixe e posicione seus elementos. O mapeamento dos slots é feito com duas propriedades que já conhecemos que este módulo adiciona mais alguns valores e funcionalidades, são as propriedades position e display.

A propriedade display irá definir como será o Grid. Quantos slots e etc.
A propriedade position irá posicionar seus elementos nestes slots.

Exemplo de estrutura HTML

```
<div class="geral">
    <nav class="menu">&diams;</nav>
    <aside class="lateral">&diams;</aside>
    <aside class="informacoes">&diams;</aside>
    <article>&diams;</article>
    <footer>&diams;</footer>
</div>
```

Exemplo de estrutura CSS

```
<style>
```

HTML5

```
.geral {
  display: "a          a          a"
          "b          c          d" /100px
          "e          e          e"
          100px      600px      100px;
}
nav.menu { position:a; }
aside.lateral {position:b; }
aside.informacoes {position:d;}
article {position:c; }
footer {position:e;}

</style>
```

Com a propriedade *display* define-se o GRID.

Então, o layout conta com três colunas.

a coluna A B E, terá 100px de largura

a coluna A C E terá 600px de largura

a coluna A D E terá 100px de largura.

Na coluna horizontal B C D, define-se a altura da mesma. No exemplo esta como 100px.

Para melhor visualização defina um background para cada elemento.

```
nav.menu {position:a; background:#000;}
aside.lateral {position:b; background:#999;}
aside.informacoes {position:d; background:#666;}
article {position:c; background:#f0f0f0;}
footer {position:e; background:#333;}
```

Um fator importante é que não importa o posicionamento dos elementos na pagina, tudo dependerá conforme o desenvolvedor posiciona-los.

Pseudo-elemento ::slot()

Utiliza-se o slot() para formatar um elemento específico, assim não atribui o estilo ao elemento e sim ao slot. Se necessário posicionar aquele elemento em outro lugar, ficará mais fácil.

Exemplo de utilização.

```
<style>
.geral {
  display: "a          a          a"
          "b          c          d" /100px
          "e          e          e"
          100px      600px      100px;
}

nav.menu {position:a; background:#000;}
aside.lateral {position:b; background:#999;}
aside.informacoes {position:d; background:#666;}
article {position:c; background:#f0f0f0;}
footer {position:e; background:#333;}

.geral::slot(b) { background: #F90 }

</style>
```

OBS.: Esta especificação é um estudo da W3C, os browsers ainda não a suportam. Um desenvolvedor criou um Javascript que interpreta o CSS e imprime os resultados.

Use com cautela e sempre teste nos principais navegadores.

Exemplo completo no documento. Combinado com:

- @font-face
- transition
- column
- gradient

exemplo_modulo_template.html

- Cores

A mais conhecida maneira de trabalhar com cores sólidas em desenvolvimento web é com o formato hexadecimal, porém não é o único formato existente. Alguns desenvolvedores preferem utilizar o formato RGB. o formato é simples, são três conjuntos de número que iniciam-se em 0 até 255 ou utiliza-se porcentagem 0% a 100%.

Sintaxe de aplicação na CSS:

COR VERMELHA

```
.classe {  
    rgb(255,0,0)  
}
```

```
.classe {  
    rgb(100%,0%,0%)  
}
```

Agora o CSS3 apresenta o RGBA, que no caso continua a mesma sintaxe, porém com o canal ALPHA, propriedade na qual cuida da opacidade.

```
.classe {  
    rgba(255,0,0, 0.5)  
}
```

```
.classe {  
    rgba(100%,0%,0%,0.5)  
}
```

Exemplo completo no documento.

exemplo_cores.html

Outra novidade interessante se tratando de produtividade é a *currentColor*.

Essa propriedade faz um aproveitamento da última cor utilizada dentro do seletor.

Exemplo de utilização:

```
.classe {  
    color:rgba(255,255,0, 0.5);  
    border:1px solid currentColor;  
}
```

Obs.: currentColor funciona em qualquer propriedade que utilize cor e esteja dentro do mesmo seletor.

- Paged media

Essa função serve de auxílio ao desenvolvedor que terá no em seus trabalhos, usuários com a necessidade de imprimir o conteúdo. Até o momento não tínhamos uma maneira correta de formatar uma página para impressão, suprimindo essa necessidade contamos com @page.

Com o CSS3, construiremos o modelo de página.

Page Model (modelo de página)

O page box tem algumas áreas simples de se entender que facilitará a explicação. Veja abaixo uma imagem e uma explicação de suas respectivas áreas:

Page box

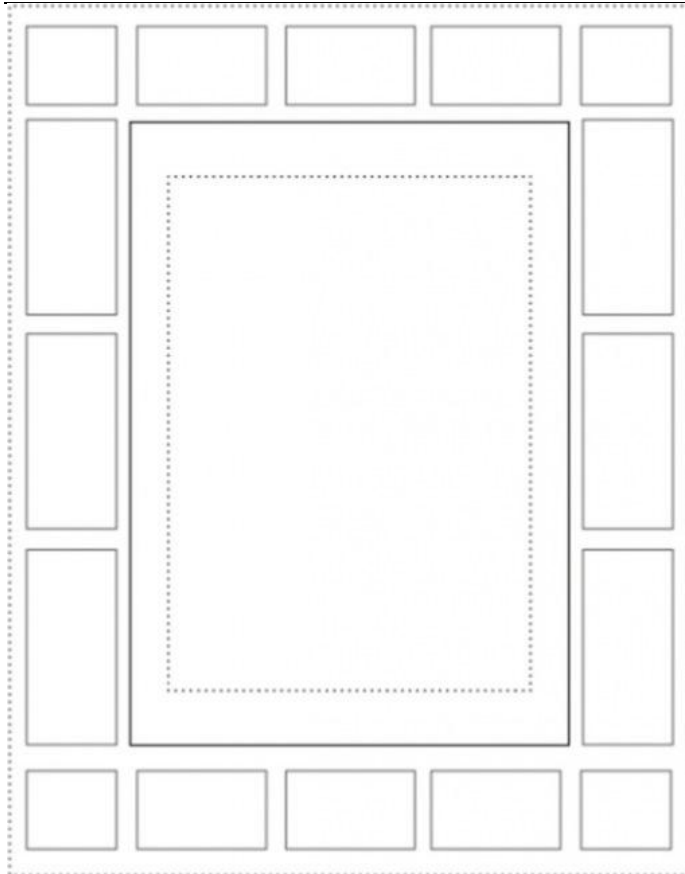
O page box é onde tudo acontece. Tenha em mente que o page box é o viewport das mídias impressas. É lá que conterá as Áreas de margem, padding, border e onde o texto será consumido. A largura e altura do page box é determinada pela propriedade size. Em um caso simples, o page box tem a largura e a altura de uma folha. Entretanto em casos complexos onde page box difere das folhas de papel em valores e orientações já que você pode personalizar de acordo com sua necessidade.

Page area

A page area é a área de conteúdo (content area) do page box.

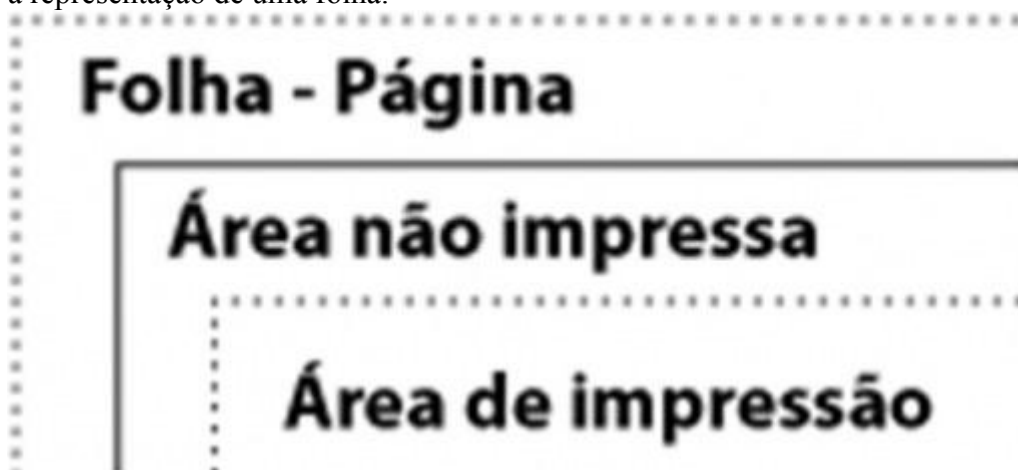
Margin box

Margin boxes contém boxes para header e footer. São conjunto de 16 boxes onde você pode inserir conteúdo útil como número da página, título do livro, etc, etc, etc. Essas áreas ficam fora do Page area. Cada um tem suas margins, paddings e bordas individuais. Veja o diagrama abaixo para visualizar melhor.



Page sheet

A folha, a página, a superfície que será impresso o conteúdo. A ilustração abaixo mostra a representação de uma folha.



Área não impressa

A área de não impressão é a área onde o dispositivo de impressão não é capaz de imprimir. Esta área depende do dispositivo que você está utilizando. O page box fica dentro da área de impressão.

Área de impressão

A área impressa é onde o dispositivo de impressão é capaz de imprimir. A área de impressão é o tamanho da page sheet menos a área de não impressão. Como a área de não impressão, a área útil de impressão depende muito do dispositivo. O dispositivo pode ajustar o conteúdo para que seja impresso sem problemas nessa área. Cada dispositivo tem seu meio de ajuste.

Propriedade size

A propriedade size especifica o tamanho e a orientação da área do de conteúdo, o page box. O tamanho do page box pode ser definida com valores absolutos (px) ou relativos (%).

Você pode usar Três valores para definir a largura e a orientação do page box:

auto

O page box irá ter o tamanho e orientação do page sheet escolhido pelo usuário.

landscape (modo paisagem)

O page box neste caso tem o mesmo tamanho da página, mas o Lado maior é o horizontal.

Portrait (modo retrato)

O page box neste caso tem o mesmo tamanho da página, mas o lado maior é o vertical.

Veja um exemplo abaixo:

```
@page {  
  size: auto;  
  margin: 10%;  
}
```

Como nessa caso a margem é variável, ela está sendo relativa às dimensões da página. Logo se a página uma A4, com as dimensões: 210mm x 297mm, as margens serão 21mm e 29.7mm.

Outro exemplo:

```
@page {  
  size: 210mm 297mm; /* definem o page-sheet para um tamanho de A4*/  
}
```

Page-size

O page-size pode ser especificado utilizando um dos media names abaixo. Isso é o equivalente a utilizar os valores escritos diretamente na propriedade size. Contudo é muito melhor utilizar o nome de um formato de formato de papel.

Formato Descrição

A5 A página deve ser definida para o tamanho ISO A5: 148mm x 210mm.

HTML5

A4 A página deve ser definida para o tamanho ISO A4: 210 mm x 297 mm.

A3 A página deve ser definida para o tamanho ISO A3: 297mm x 420mm.

B5 A página deve ser definida para o tamanho ISO B3 media: 176mm x 250mm.

B4 A página deve ser definida para o tamanho ISO B4: 250mm x 353mm.

letter A página deve ser definida para o tamanho papel carta: 8.5 pol x 11 pol

A4 é um dos formatos mais utilizados para impressão de documentos.

Exemplo:

```
@page {  
  size: A4 landscape;  
}
```

Exemplo completo no documento.

exemplo_@page.html

- @font-face

O design do website está pronto, imagens, cores e fontes. Descobre-se que a fonte utilizada no layout não é uma fonte padrão do sistema e provavelmente o usuário não a terá instalada. Fique tranquilo, o CSS3 tem a solução

A propriedade @font-face possibilita utilizar fonts externas em websites.

```
@font-face {  
  font-family: 'LeagueGothic';  
  src: url(http://site.com/fonts/LeagueGothic.otf);  
}
```

```
@font-face {  
  font-family: 'Orienta';  
  src: url(Orienta.woff);  
}
```

```
header {  
  font-family: 'Orienta';  
}
```

Referencias

<http://www.w3.org>

<http://pt.wikipedia.org>



GINEAD



Semeando Conhecimento