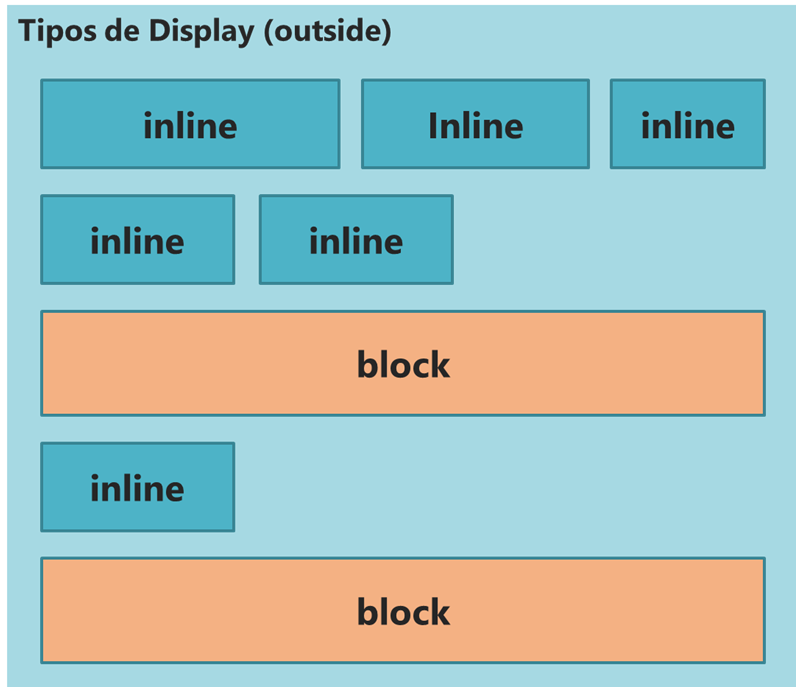
**PROPRIEDADE DISPLAY**

A maneira como um elemento e seus filhos serão dispostos em uma página pelo Navegador, em termos de layout, é definida pela propriedade display. A propriedade display pode receber diversos valores, sendo que alguns podem se referir a maneira como o elemento em questão será organizado em relação aos elementos irmãos (outside), e outros valores podem se referir a como os elementos filhos vão ser dispostos dentro de um elemento pai (inside). São várias opções e vamos passar com calma por todas elas.

Inicialmente, por padrão, os elementos HTML são divididos em duas categorias: inline ou block. Os elementos inline (Ex: <a>, <span>, <img>, <button>, <input>, entre outros) são dispostos na mesma linha, um em frente ao outro, enquanto houver espaço na tela. Os elementos block (Ex: <div>, <h1> .. <h6>, <p>, <form>, <canvas>, <table>, entre outros) ocupam toda a próxima linha. Um exemplo da diferença entre estas categorias é mostrada na imagem a seguir.



Para especificar o comportamento de um elemento do lado de fora (outside) ou alterar o comportamento padrão, a propriedade display pode receber os valores **inline** ou **block**. Um caso típico de alteração do comportamento padrão de um elemento é quando queremos centralizar uma imagem que é um elemento inline por padrão. Para fazer isso, basta alterar o comportamento da imagem para block e ajustar as margens laterais de forma automática, como na seguinte regra:

**img { display: block; margin: 0 auto; }**

Se a propriedade display receber o valor **none**, o elemento é retirado da visualização pelo Navegador, tornando-se invisível pelo usuário.

Para especificar o comportamento de elementos filhos de um elemento (inside), podemos atribuir à propriedade display os valores: **table**, **flex** ou **grid**. Nesse caso, o elemento do lado de fora terá automaticamente o comportamento de um elemento **block**, ocupando toda a linha.

Para elementos com a propriedade display com valor **table**, os elementos filhos devem ajustar a propriedade display de forma a trabalhar com o elemento pai. Nesse caso, a propriedade display dos filhos pode receber os valores: **table-row, table-cell, table-column, table-caption, table-row-group, table-header-group ou table-footer-group**. Neste caso, o elemento pai servirá como um agrupador dos elementos filhos que vão se comportar como elementos de uma tabela HTML.

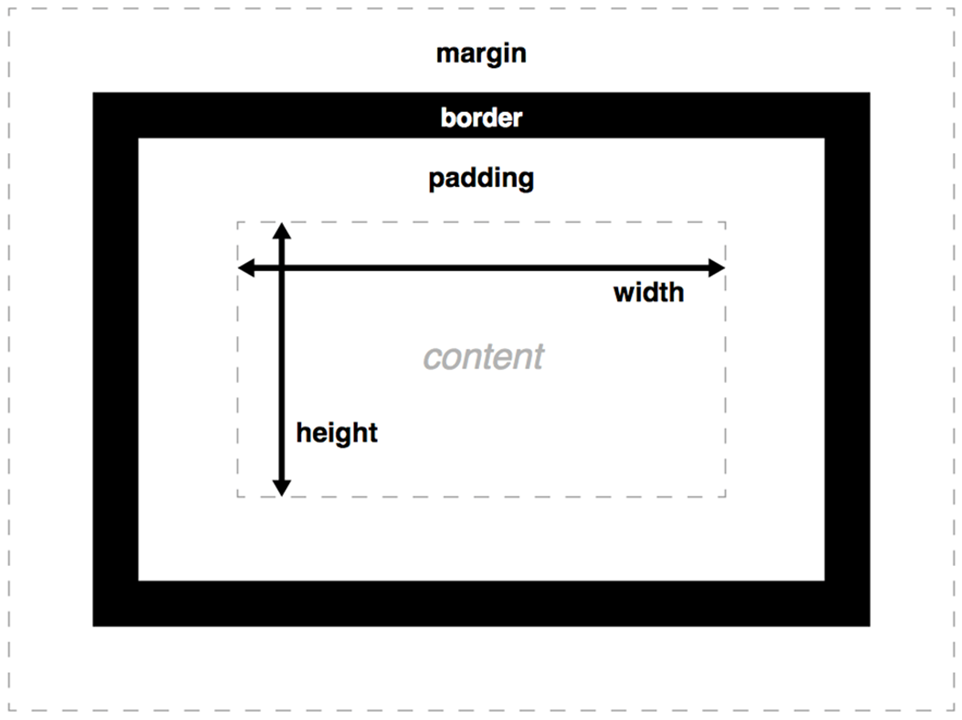
Para elementos com a propriedade display com valor **flex**, os elementos filhos podem ser controlados para tornar o layout fluído em função da largura da janela do Navegador.

Para elementos com a propriedade display com valor **grid**, é possível definir regiões em que os elementos filhos serão dispostos proporcionando um controle muito interessante ao desenvolvedor.

As abordagens de display **flex** e **grid** são muito utilizadas em layouts de sites responsivos e serão tratados em uma sessão de estudo específica para isso. Algumas referências de tutoriais para estas duas abordagens foram colocadas ao final dessa sessão.

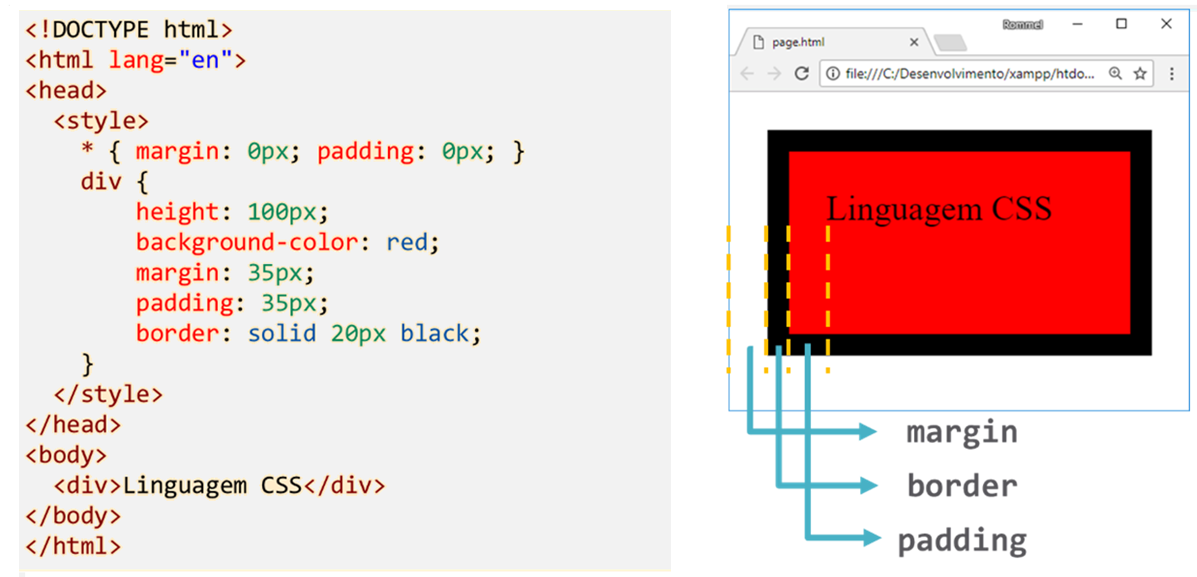
**BOX MODEL**

Diversas propriedades fazem parte do modelo denominado Box Model que define como os limites de um elemento vão funcionar. Nesse modelo estão propriedades associadas à margem, à borda, ao espaçamento interno (padding) do elemento, bem como a sua largura e altura, além de definir o local aplicável para o fundo do elemento (background). A imagem a seguir mostra como cada uma destas propriedades:

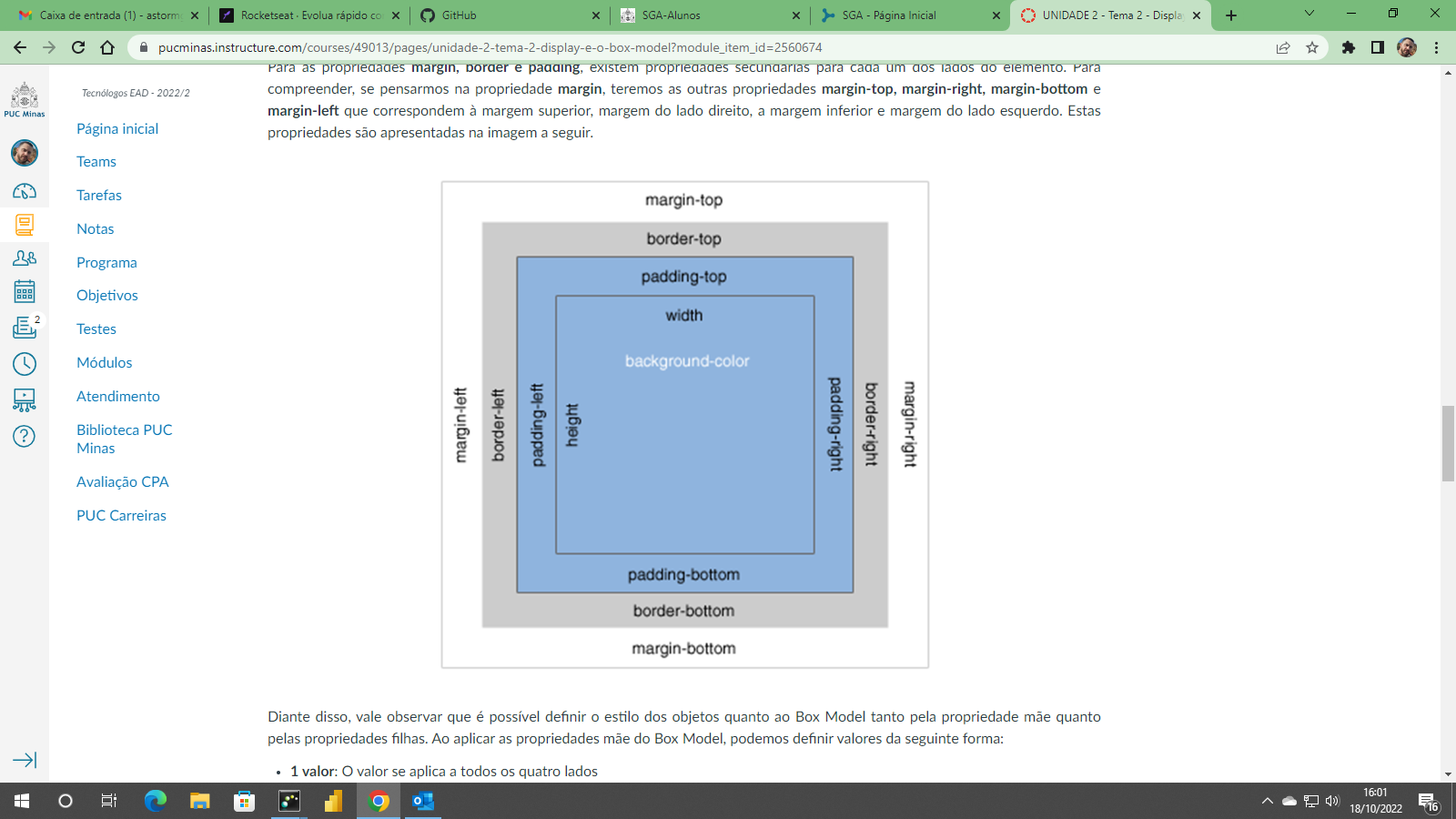


Fonte: [https://developer.mozilla.org/Links para um site externo.](https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/The_box_model)

A imagem a seguir apresenta um exemplo de aplicação das propriedades do Box Model e o resultado mostrado no Navegador.



Para as propriedades **margin, border e padding**, existem propriedades secundárias para cada um dos lados do elemento. Para compreender, se pensarmos na propriedade **margin**, teremos as outras propriedades **margin-top, margin-right, margin-bottom** e **margin-left** que correspondem à margem superior, margem do lado direito, a margem inferior e margem do lado esquerdo. Estas propriedades são apresentadas na imagem a seguir.



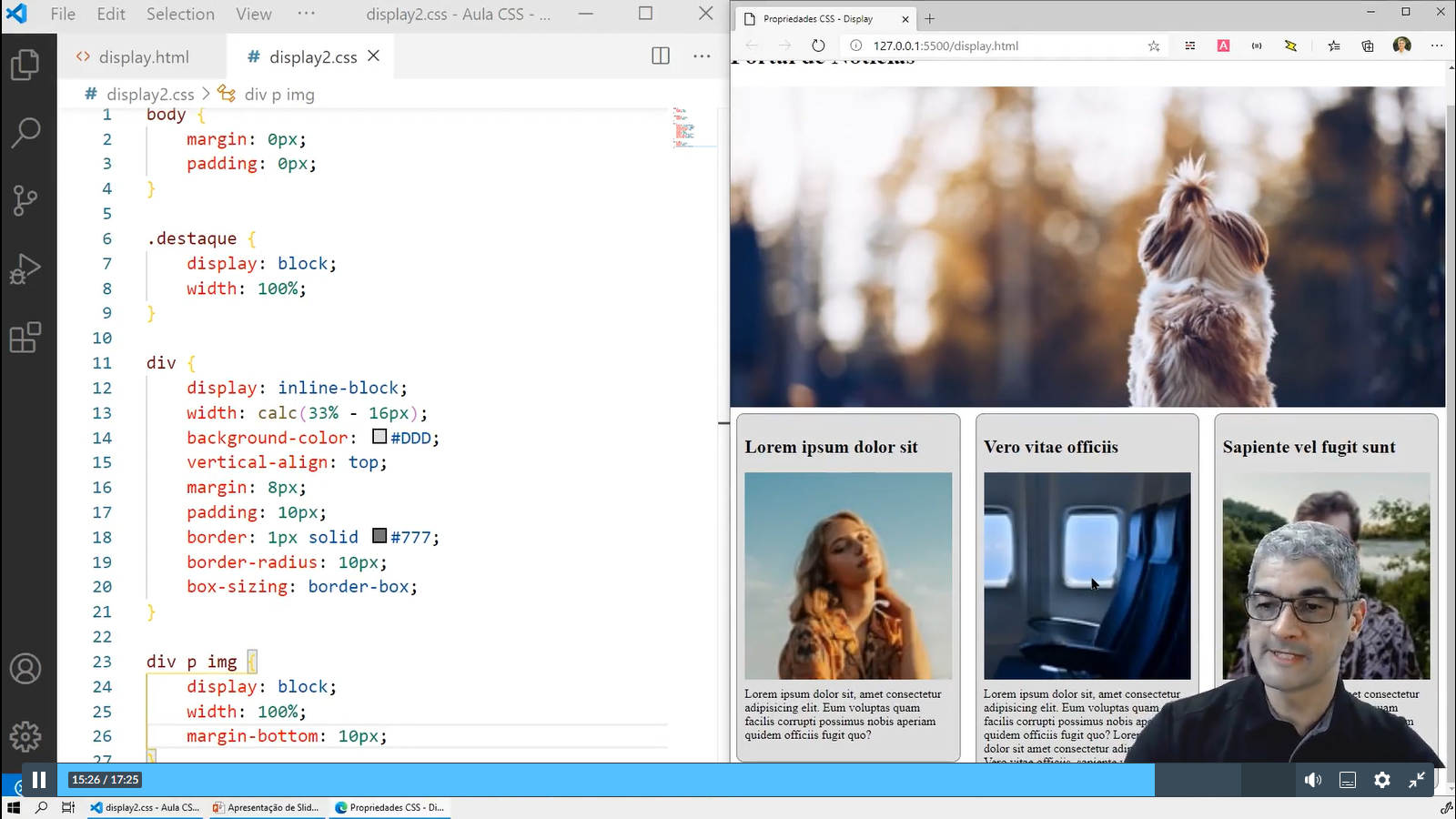
Diante disso, vale observar que é possível definir o estilo dos objetos quanto ao Box Model tanto pela propriedade mãe quanto pelas propriedades filhas. Ao aplicar as propriedades mãe do Box Model, podemos definir valores da seguinte forma:

* **1 valor**: O valor se aplica a todos os quatro lados  
  Ex: margin: 10px;
* **2 valores:** o primeiro valor se aplica à parte superior e inferior; e o segundo valor se aplica aos lados direito e esquerdo  
  Ex: margin: 5px   20px;
* **3 valores:**o primeiro valor se aplica à parte superior, o segundo valor se aplica ao lado direito e esquerdo; e o terceiro valor se aplica ao lado inferior  
  Ex: margin: 5px  20px  10px;
* **4 valores:** cada valor se aplica a um dos lados na seguinte ordem: lado superior, lado direito, lado inferior e lado esquerdo (ordem do horário)  
  Ex: margin: 5px 20px 10px 20px;

**FUNDO DE ELEMENTOS (BACKGROUND)**

A todo elemento de uma página Web podemos aplicar definições de fundo por meio do conjunto de propriedades background, conforme relação seguinte:

* **background-color:**cor de fundo do elemento  
  Ex: Background-color: yellow;
* **background-image:** Imagem ou gradiente de fundo do elemento.  
  Ex: background-image: url(http://imagens.com/fundo.jpg);  
  Ex: background-image: linear-gradient (red, blue);
* **background-repeat:** indica como repetir a imagem no fundo caso essa seja menor que o elemento. Opções: no-repeat | repeat | repeat-x | repeat-y | space | round.  
  Ex: background-repeat: no-repeat
* **background-position:** posição inicial do fundo em relação ao elemento HTML. Opções: top | left | center | right | bottom.  
  Ex: background-position: top



**PROPRIEDADES BÁSICAS DE TEXTO**

Na Linguagem CSS existem várias propriedades voltadas para a formatação de trechos de texto nas páginas *Web*. A tabela a seguir traz as principais propriedades envolvidas nesse propósito.

|  |  |  |
| --- | --- | --- |
| **Propriedade** | **Descrição** | **Valores** |
| font-family | Tipo de fonte empregada no texto | arial, helvatica, sans-serif |
| font-size | Especifica o tamanho da letra | xx-small | x-small | small | medium | large | x-large | xx-large | smaller | larger | 10px | 80% |
| font-style | Estilo da fonte | normal | italic | oblique |
| font-weigth | Indica a largura dos caracteres | normal | bold | bolder | lighter | 100 | 200 |
| font-variant | Indica se os caracteres minúsculos devem ser mostrados como maiúsculas | normal | small-caps |
| line-height | Altura da linha | normal | 1.6 | 80% | 200% |
| color | altera a cor do texto | [cor] |
| text-align | Define o alinhamento do texto | left | right | center | justify |
| text-shadow | Define se o texto terá sombra | none | x y z [cor]  x - distância horizontal  y – distância vertical z – difusão da sombra |
| text-decoration-line | Define o tipo de decoração do texto | none | underline | overline | line-through |
| text-decoration-style | Define o estilo da linha da decoração do texto | solid | wavy | dashed | dotted | double |
| text-decoration-color | Define a cor da linha de decoração do texto | [cor] |
| letter-spacing | Espaçamento entre letras em um bloco de texto | normal  | 2px | 0.1em |
| word-spacing | Espaçamento entre palavras em um bloco de texto | normal  | 2px | 0.1em |
| text-transform | Define se o texto deve vir em maiúscula, minúscula ou capitalizado | none | capitalize | uppercase | lowercase |

**FONTES DE TEXTO NA *WEB***

Por padrão, o CSS oferece as seguintes palavras-chave para identificar o tipo de fonte ou de letras para textos nos *sites* **Web:  serif, sans-serif, monospace, cursive e fantasy**.

Os diversos navegadores nos mais diversos dispositivos possuem um conjunto restrito de fontes suportadas e que podem lhe garantir uma exibição padronizada para todos os usuários que acessam o seu *site*. Ao definir a fonte de texto empregada nos *sites* que desenvolvemos é preciso estar atento a isso.

Para garantir a compatibilidade do seu *site* com as ofertas restritas de um navegador mais antigo, é possível definir uma lista de fontes possíveis e próximas. Se o navegador não possuir a primeira, ele tenta a segunda e daí em diante. Veja um exemplo de código que ilustra essa possibilidade:

As chamadas Web Safe Fonts, que são oferecidas pela maioria dos navegadores, podem variar um pouco, mas geralmente incluem as seguintes:

* **Sem serifas**
  + Arial
  + o Helvetica
  + o Trebuchet MS
  + o Verdana
* **Com serifas**
  + Garamond
  + Georgia
  + Times New Roman
* **Monoespaçada**
  + Courier
  + Courier New

A alternativa para utilizar uma fonte diferente das Web Safe Fonts é aplicar um recurso do CSS para importar as chamadas Web Fonts. Elas são disponibilizadas em alguns repositórios na Internet e podem ser referenciados no seu código. Você ainda pode disponibilizar a fonte diretamente a partir do seu *site*, garantindo independência dos repositórios como o Google Fonts, DaFont e Adobe.