

# Desenvolvimento de Sistemas WEB

## 02 - CSS

Arthur Porto - IFNMG Campus Salinas

*arthur.porto@ifnmg.edu.br*  
*arthurporto.com.br*

# Sumário I

- 1 Introdução
  - Regras do CSS
- 2 Introduzindo o CSS no código
  - Interno
  - Externo
- 3 Seletores
  - Seletor de Tipo
  - Seletor de ID
  - Seletor de Classe
  - Seletor de Atributo
  - Seletor universal
  - Grupo de elementos
  - Seletor descendente
  - Seletor filho
  - Regras do CSS

# Sumário II

## 4 Cores

## 5 Fonte

- Propriedades de fonte
- Propriedades

## 6 Pseudo-Classes

## 7 Atividade

## 8 Blocos

- Dimensões
- Bordas - border
- Preenchimento - padding
- Margem - margin
- Centralizando
- Sombreamento - CSS3
- Borda arredondada - CSS3

# Sumário III

- Pesquisa

9 Atividade

10 Lista

11 Tabelas

12 Pesquisa

13 Atividade

14 Layout

15 Posição dos elementos

- Sobreposição

- Flutuação

16 Tamanho das telas

- Layout fixo X Layout fluido

17 Layout HTML5

18 Atividade

# Sumário IV

19 Pesquisa

20 Referências

# Introdução

- O que é o CSS?
  - São Folhas de Estilo em Cascata (*Cascading Style Sheets*)
  - Descreve como os elementos HTML devem ser exibidos na tela
  - Pode controlar o *layout* e *design* de várias páginas de uma só vez
  - Já pensou se tivéssemos que usar o atributo *style* em todos os elementos?

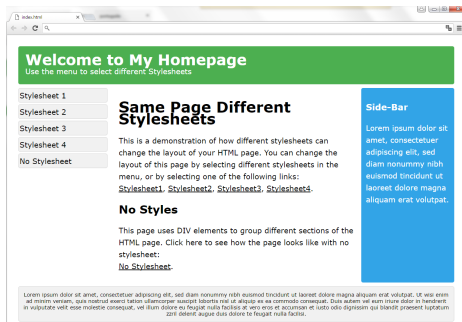


Figura 1: w3schools.com

## The Cottage Garden

The *cottage garden* is a distinct style of garden that uses an informal design, dense plantings, and a mixture of ornamental and edible plants.

The Cottage Garden originated in *England* and its history can be traced back for centuries, although they were re-invented in 1870's England, when stylized versions were formed as a reaction to the more structured and rigorously maintained *English estate gardens*.

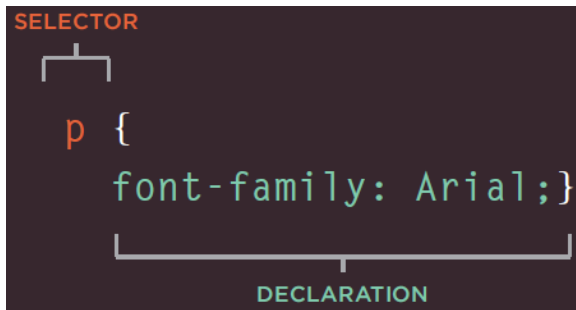
The earliest cottage gardens were more practical than their modern descendants, with an emphasis on vegetables and herbs, along with some fruit trees.

- O CSS age como se existisse linhas invisíveis ao redor de cada elemento formando uma “caixa”, aplicando o *estilo* aquele espaço

# Introdução

## Regras do CSS

- O CSS associa as regras de estilo aos elementos HTML [1]
- Usando sempre de **seletores** e **declarações**



- O **seletor** indica o elemento
- A **declaração** indica o estilo que esse elemento receberá



# Introdução

## Regras do CSS

As declarações ainda se dividem em **propriedades** e **valores**

```
h1, h2, h3 {  
    font-family: Arial;  
    color: yellow;}  
└────────┘ └────────┘  
PROPERTY  VALUE
```

- A **propriedade** indica os aspectos do elemento a serem mudados
- O **valor** é o elemento que ajusta a propriedade

# Introduzindo o CSS no código Interno

- A forma de colocar o código interno é dentro do `< head >` utilizando a tag `< style >`.
  - Especificando o atributo **type** que identificará o tipo de código.

```
1 <!doctype html>
2 <html lang="pt-br">
3 <head>
4   <meta charset="utf-8">
5   <title> CSS </title>
6   <style type="text/css">
7     h1{
8       color: red;}
9     h2{
10      color: blue;}
11   </style>
12 </head>
13 <body>
14   <h1> Vermelho </h1>
15   <h2> Azul </h2>
16 </body>
17 </html>
```



# Introduzindo o CSS no código Externo

- A forma de colocar o código externo é dentro do `< head >` utilizando a tag `< link >`.
  - Que fará uma ligação entre a página HTML e o arquivo de código CSS.
  - Com os atributos: rel (tipo de relação), type (tipo do arquivo) e href (caminho)

```
1 <!doctype html>
2 <html lang="pt-br">
3 <head>
4   <meta charset="utf-8">
5   <title> CSS </title>
6   <link rel="stylesheet" type="text/css" href="css/externo.css">
7 </head>
8 <body>
9   <h1> Vermelho </h1>
10  <h2> Azul </h2>
11 </body>
12 </html>
```

```
1 h1{
2   color: red;}
3 h2{
4   color: blue;}
```

# Introduzindo o CSS no código Externo

- Vantagens
  - Evita a repetição de código
  - Deixa o HTML separado e de mais fácil compreensão
  - Utilização em múltiplas páginas

- É o modo de acessar os elementos HTML.
- Como selecionar os elementos HTML para aplicar as regras CSS?
- É através dos seletores que os elementos HTML serão achados para se aplicar as regras de estilo.

# Seletores

## Seletor de Tipo

- Seleciona o tipo do elemento (TAG) HTML para receber o estilo.

### Seletor de tipo

```
1 | p {  
2 |   color: red;  
3 | }
```

# Seletores

## Seletor de ID

- Seleciona apenas o elemento com um determinado identificador.
- O elemento que tem o atributo **id** com o valor especificado.

### Seletor de id

```
1 | #meuId {  
2 |     font-style: italic;  
3 | }
```

- O **id** é identificado pelo caractere # (jogo da velha)

# Seletores

## Seletor de Classe

- Seleciona apenas o elemento com um determinado identificador.
- O elemento que tem o atributo **id** com o valor especificado.

### Seletor de classe

```
1 | .minhaClasse {  
2 |     margin-left: 10px;  
3 | }
```

- As **classes** são identificadas pelo • (ponto)



- Seleciona apenas os elementos que tenham um **atributo** com **valor** específico.

### Seletor de atributo

```
1 | [type="submit"] {  
2 |     width: 100px;  
3 | }
```

# Seletores

## Seletor universal

- Qualquer elemento receberá essas características.

### Seletor universal

```
1 | * {  
2 |   border: 1px solid green;  
3 | }
```

- Identificado com o \* (asterisco)

# Seletores

## Grupo de elementos

- Basta separar os seletores por vírgulas.

### Seletor de grupo

```
1 | p, #idname, .classname {  
2 |     border-bottom: 1px dotted orange;  
3 | }
```

# Seletores

## Seletor descendente

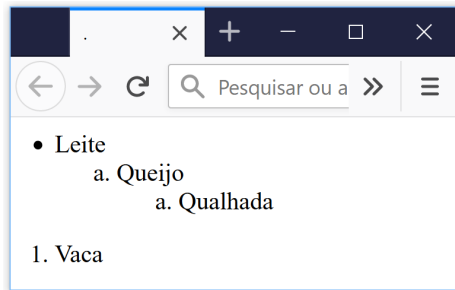
- Seleciona os elementos que são descendentes de outro.

```
1 | ul ol { list-style-type: lower-alpha; }
```

- Somente os `<ol>` que estão dentro de um `<ul>`

### Seletor descendente

```
1 <ul>
2   <li> Leite </li>
3   <ol>
4     <li> Queijo </li>
5     <ol>
6       <li> Qualhada </li>
7     </ol>
8   </ol>
9 </ul>
10 <ol>
11   <li> Vaca </li>
12 </ol>
```



# Seletores

## Seletor filho

- Seleciona apenas os elementos que são, diretamente, filhos de outro.

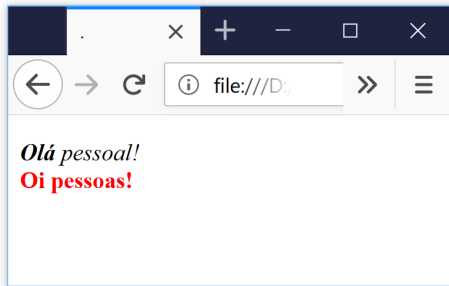
### Seletor de filho

```
1 | p > b { color:red; }
```

- Apenas os `<b>` que são filhos de `<p>` que receberão a cor vermelha.

### Seletor de filho

```
1 <p>  
2   <i> <b> Olá </b> pessoal! </i> <br />  
3   <b> Oi pessoas! </b>  
4 </p>
```

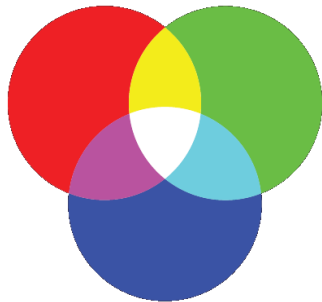


# Seletores

## Regras do CSS

- Em caso de redundância de um seletor, a última regra é a que predomina
- A regra com maior especificação é predominante
  - Um seletor de tipo predomina sobre o seletor universal
- Pode-se acrescentar a expressão “*!important*” depois de uma propriedade para fazer com que essa seja considerada mais importante.
- Algumas propriedades são herdadas para os elementos filhos, outras não. Usando a expressão “*inherit*” você força a herança

- Definição das cores
  - No padrão *RGB* - *color: rgb(100,100,90);*
  - Códigos em **hexadecimal** - *color: #ff5500;*
  - Nome da cor (147 predefinições) - *color: red;*
- Cor de primeiro plano
  - Propriedade: ***color***
- Cor de fundo
  - Propriedade: ***background-color***
- Opacidade (CSS3)
  - Propriedade: ***opacity***
  - Varia de 0.0 a 1.0



# Fonte

## Propriedades de fonte

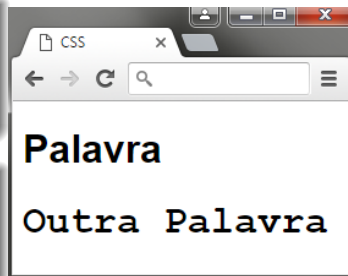
### • *font-family*

#### HTML

```
1 <body>
2   <h1> Palavra </h1>
3   <h1 id="b"> Outra Palavra </h1>
4 </body>
```

#### CSS

```
1 h1{
2   font-family: Arial, Verdana, sans-serif;
3 }
4 #b{
5   font-family: "Courier New", Courier, monospace;
6 }
```





# Fonte

## Propriedades de fonte

### • *font-size*

#### HTML

```
1 <body>
2   <h1> Fazenda </h1>
3   <h2> Sejam Bem-vindos! </h2>
4   <p> Essa é a fazenda... </p>
5 </body>
```

#### CSS

```
1 h1 { font-size: 30px;}
2 h2 { font-size: 75%;}
3 p { font-size: 0.9em;}
```



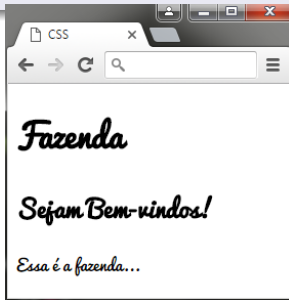
# Fonte

## Propriedades de fonte

### • **@font-face**

```
1 @font-face {  
2   font-family: 'Pacifico';  
3   src: url('fonts/Pacifico.ttf') format('truetype');  
4 }  
5 body{  
6   font-family: Pacifico, Arial, Georgia, serif;  
7 }
```

- [Google Fonts](#)
- [Font Squirrel](#)
- [Font Library](#)



BROWSER	FORMAT			
	eot	woff	ttf / otf	svg
Chrome (all)				●
Chrome 6+		●	●	●
Firefox 3.5			●	
Firefox 3.6+		●	●	
IE 5 - 8	●			
IE 9+	●	●	●	
Opera 10+			●	●
Safari 3.1+			●	●
iOS <4.2				●
iOS 4.2+			●	●

# Fonte

## Propriedades

- Negrito - **font-weight**
  - Valores: *bold*, *normal*
- Itálico - **font-style**
  - Valores: *normal*, *italic*, *oblique*
- Caixa da fonte - **text-transform**
  - Valores: *uppercase*, *lowercase*, *capitalize*
- **text-decoration**
  - Valores: *none*, *underline*, *overline*, *line-through*, *blink*
- Altura das linhas - **line-height**
  - Valores ex.: *15px*, *75%*, *0.8em*
- Espaço entre letras - **letter-spacing**
  - Valores ex.: *15px*, *0.8em*
- Espaço entre palavras - **word-spacing**
  - Valores ex.: *15px*, *0.8em*

# Fonte

## Propriedades

- Alinhamento horizontal - ***text-align***
  - Valores: *left, right, center, justify*
- Alinhamento vertical - ***vertical-align***
  - Valores: *baseline, sub, super, top, text-top, middle, bottom, text-bottom*
  - Obs.: Funciona com ***img*** ou dentro de células de tabelas. Ou utilizar a propriedade ***padding*** com valores iguais no eixo vertical
- Sombras (CSS3) - ***text-shadow***
  - Posição da sombra
  - Cor da sombra

## CSS

```
1 h1{
2   text-shadow: 1px 1px 3px #777;
3 }
```

- Especifica um estado especial do elemento a ser selecionado
- Links ainda não visitados - *:link*
- Links já visitados - *:visited*
- Mouse sobre - *:hover*
- **Pseudo-Classes**

## CSS

```
1 a:link{
2     color: red;
3     font-weight: bold;
4     text-decoration: none;
5     background-color: #777;
6 }
7 a:visited{
8     color: green;
9     font-weight: normal;
10    text-decoration: underline;
11    background-color: #ace;
12 }
13 a:hover{
14     background-color: blue;
15     color: #fff;
16 }
```

# Atividade 05

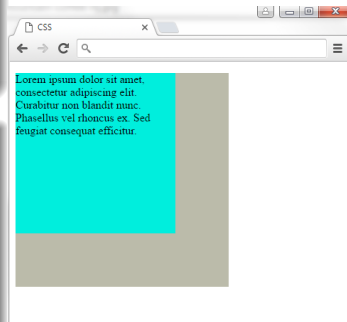
# Blocos

## Dimensões

- **width** (largura), **height** (altura)
  - Pode se usar píxeis (px) ou % como unidades de medida

```
1 <div>
2   <p>
3     Lorem ipsum dolor sit amet, consectetur adipiscing
4     elit. Curabitur non blandit nunc. Phasellus vel
5   </p>
6 </div>
```

```
1 div {
2   height: 300px;
3   width: 300px;
4   background-color: #bba;
5 }
6 p {
7   height: 75%;
8   width: 75%;
9   background-color: #0ed;
10 }
```



### Outras propriedades

- Largura máxima - ***max-width***
- Largura mínima - ***min-width***
- Altura máxima - ***max-height***
- Altura mínima - ***min-height***
  - Ex.: max-width - Quando a "tela" é menor do que o max-width definido a largura se ajusta a tela. Porém, quando o espaço é maior a largura vai até o máximo estabelecido.





- ***Border*** - A borda é o que separa as extremidades de um elemento à outro.
- ***Margin*** - A margem fica por fora da borda entre os elementos.
- ***Padding*** - Preenchimento é o espaço entre a borda e o conteúdo dentro do elemento.

## *COM* margin e padding

### **Moog**

Moog synthesisers were created by Dr. Robert Moog under the company name Moog Music. Popular models include the Moog Modular, Minimoog, Micromoog, Moog Rogue, and Moog Source.

### **ARP**

ARP Instruments Inc. was set up by Alan Peralman, and was the main competitor for Moog during the 1970's. Popular models include the Arp 2600 and the ARP Odyssey.

### **Sequential Circuits**

Sequential Circuits Inc was founded by Dave Smith, and the company was pivotal in the creation of MIDI. Famous models include the Prophet 5, Prophet 600, and Pro-One.

## *SEM* margin e padding

### **Moog**

Moog synthesisers were created by Dr. Robert Moog under the company name Moog Music. Popular models include the Moog Modular, Minimoog, Micromoog, Moog Rogue, and Moog Source.

### **ARP**

ARP Instruments Inc. was set up by Alan Peralman, and was the main competitor for Moog during the 1970's. Popular models include the Arp 2600 and the ARP Odyssey.

### **Sequential Circuits**

Sequential Circuits Inc was founded by Dave Smith, and the company was pivotal in the creation of MIDI. Famous models include the Prophet 5, Prophet 600, and Pro-One.

- Largura da borda - ***border-width***

- Valores: *px* ou *thin*, *medium*, *thick*
- Outras propriedades
  - *border-top-width*
  - *border-right-width*
  - *border-bottom-width*
  - *border-left-width*

- Estilo da borda - ***border-style***

- Valores: *solid*, *dotted*, *dashed*, *double*, *groove*, *ridge*, *inset*, *outset*, *hidden*/*none*
- Outras propriedades
  - *border-top-style*
  - *border-right-style*
  - *border-bottom-style*
  - *border-left-style*

- Cor - ***border-color***

- Outras propriedades
  - *border-top-color*
  - *border-right-color*
  - *border-bottom-color*
  - *border-left-color*

- Atributo ***border***

- É possível definir as características de *Largura*, *Estilo* e *Cor* em um só comando
  - Ex.: ***border: 3px dotted #0088dd;***

Here is a simple chord sequence  
played on a Hammond organ  
through a Leslie speaker.

# Blocos

## Preenchimento - padding

```
1 <p>
2 Analog synths produce
3 a wave sound, whereas the...
4 </p>
5 <p class="p2">
6 Analog synths produce a wave
7 sound, whereas the ...
8 </p>
```

- Atributo - **padding**

Ex.: **padding: 5px;** (Todos os lados)

Ex.: **padding: 5px 10px;**(vert. hori.)

Ex.: **padding: 5px 3px 7px 5px;**

- Outras propriedades

- padding-top
- padding-right
- padding-bottom
- padding-left

```
1 p {
2   width: 275px;
3   border: 2px solid #0088dd;
4 }
5 p.p2 {
6   padding: 10px;
7 }
```

Analog synths produce a wave sound,  
whereas the sounds stored on a digital  
synth have been sampled and then  
turned into numbers.

Analog synths produce a wave sound,  
whereas the sounds stored on a digital  
synth have been sampled and then  
turned into numbers.

# Blocos

## Margem - margin

```
1 <p>
2   Analog synths produce a wave sound, whereas
3 </p>
4 <p class="p2">
5   Analog synths produce a wave sound, whereas
6 </p>
```

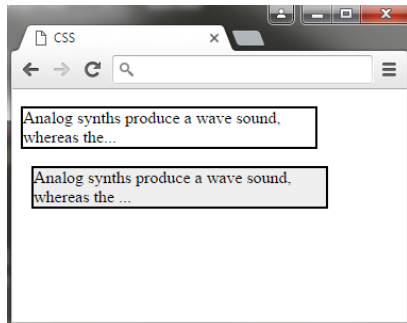
```
1 p {
2   width: 275px;
3   border: 2px solid #000;
4 }
5 p.p2 {
6   margin: 10px;
7   background-color: #eee;
8 }
```

- Atributo - **margin**

Ex.: **margin: 5px;**

Ex.: **margin: 5px 10px;**

Ex.: **margin: 5px 10px 5px 10px;**



- Outras propriedades

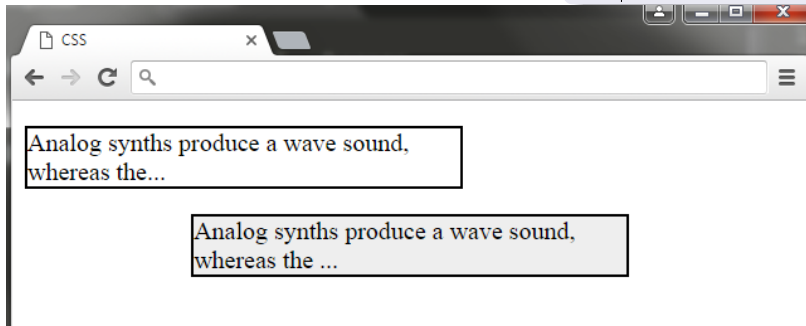
- margin-top
- margin-right
- margin-bottom
- margin-left

# Blocos

## Centralizando

```
1 <p>
2   Analog synths produce a wave sound, whereas
3 </p>
4 <p class="p2">
5   Analog synths produce a wave sound, whereas
6 </p>
```

```
1 p {
2   width: 275px;
3   border: 2px solid #000;
4 }
5 p.p2 {
6   margin: 0px auto;
7   background-color: #eee;
8 }
```



- Atributo - ***box-shadow***

### CSS

```
1 | p {  
2 |   -moz-box-shadow: 5px 5px 5px #777;  
3 |   -webkit-box-shadow: 5px 5px 5px  
   |     #777;  
4 |   box-shadow: 5px 5px 5px #777;  
5 | }
```



- Atributo - ***border-radius***

### CSS

```
1 p {  
2   border-radius: 10px;  
3   -moz-border-radius: 10px;  
4   -webkit-border-radius: 10px;  
5 }
```

Pet Sounds featured a number of unconventional instruments such as bicycle bells, buzzing organs, harpsichords, flutes, Electro-Theremin, dog whistles, trains, Hawaiian-sounding string instruments, Coca-Cola cans and barking dogs.



- Transbordar - ***overflow***
  - Valores ex.: *hidden, scroll*
- Visibilidade - ***visibility***
  - Valores ex.: *hidden, visible*
- Comportamento de visualização - ***display***
  - Valores ex.: *inline, block, none ...*

# Atividade 06

## ***list-style-type***

- Tipo do marcador da lista
- Valores
  - Lista não-ordenada - *none, disc, circle, square*
  - Lista ordenada - *decimal, lower-alpha, upper-alpha, lower-*

## ***list-style-position***

- Posição do marcador na lista
- Valores - *outside, inside*

## ***list-style***

- Tipo e posição

## CSS

```
1 ol {  
2     list-style-type: lower-roman;  
3     list-style-position: outside;  
4 }  
5 ul {  
6     list-style: inside circle;  
7 }
```

## ***empty-cells***

- Mostra células vazias ou não
- Valores - *show, hide*

## ***border-spacing***

- Espaço entre as bordas da tabela
- Valores - *px* (mesma ideia do *margin*)

## ***border-collapse***

- Sem espaço entre as bordas
- Valores - *collapse*

## CSS

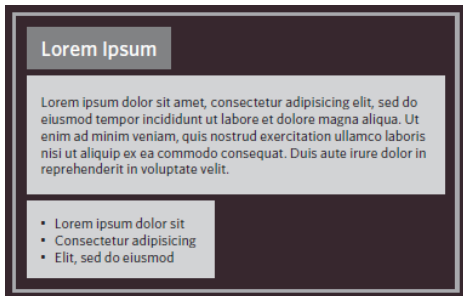
```
1 | table.um{  
2 |     empty-cells: hide;  
3 |     border-spacing: 5px 15px;  
4 | }  
5 | table.dois{  
6 |     empty-cells: hide;  
7 |     border-collapse: collapse;  
8 | }
```

- Marcadores com imagens - ***list-style-image***

# Atividade 07

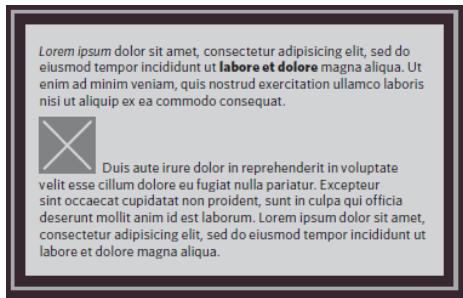
## Elementos em Bloco

- Elementos que quebram linha
- Exemplos: `< h1 >`, `< p >`, `< ul >`,  
`< li >`



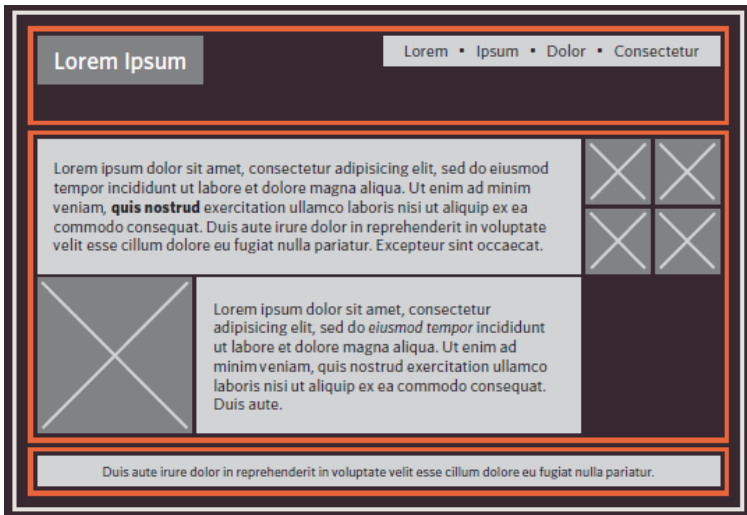
## Elementos em linha

- Elementos que **não** quebram linha
- Exemplos: `< img >`, `< b >`, `< i >`



# Layout

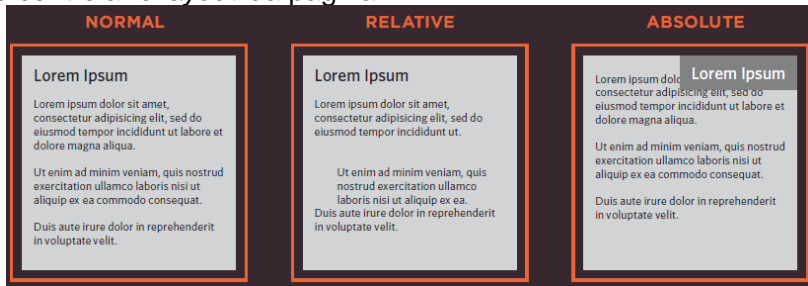
- Um elemento de bloco pode **conter** diversos outros elementos





# Posição dos elementos

- Propriedade: ***position***
  - Permite controlar o *layout* da página



- Valores
  - ***static*** - É o padrão. Cada bloco aparece em uma nova linha, sem interferência de posições
  - ***relative*** - O bloco se move de acordo com a posição estabelecida
  - ***absolute*** - O bloco se move a partir da posição do bloco que o contém

# Posição dos elementos

- Propriedade: ***position***
  - Valores
    - ***fixed*** - O bloco se move a partir da posição do *Browser*



## Movimentação dos elementos

- Propriedades: ***top, right, bottom, left***
  - Valores: ***px, em, %***

# Posição dos elementos

## Sobreposição

- Propriedade: ***z-index***
  - Valores: ***números inteiros***
  - Permite variação de sobreposição dos elementos, definindo qual conteúdo fica por cima ou por baixo.

# Posição dos elementos

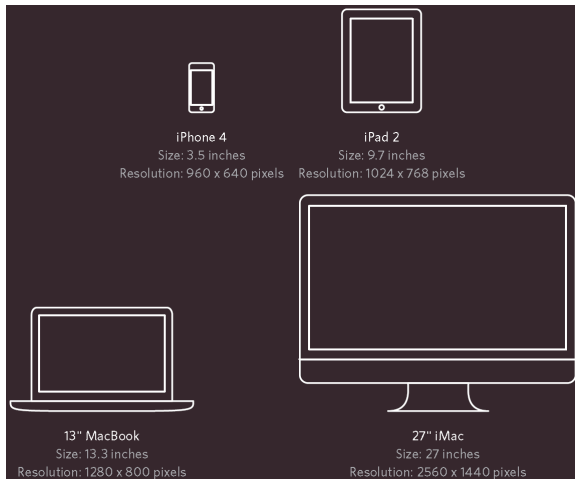
## Flutuação

- Propriedade: ***float***
  - Valores: ***left, right***
  - Permite que os elementos saiam do fluxo normal se posicionado à esquerda ou direita do elemento que o contém.

### Cancelar a flutuação

- Propriedade: ***clear***
  - Valores: ***left, right, both***

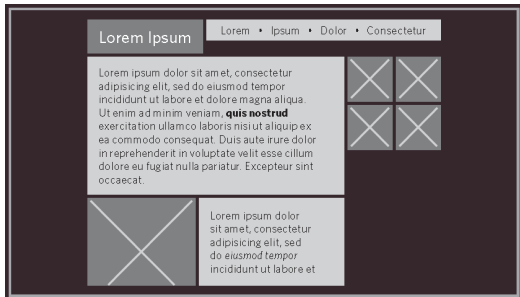
# Tamanho das telas



# Tamanho das telas

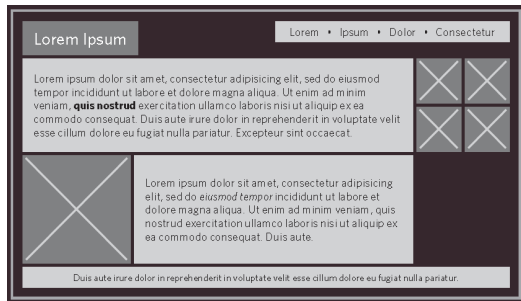
## Layout fixo X Layout fluido

### Fixo (PX)



- Vantagens
  - Controle de *Design*
- Desvantagens
  - Interação do usuário (Diferentes Telas)

### Fluido (%)



- Vantagens
  - Interação do usuário (Diferentes Telas)
- Desvantagens
  - Controle de *Design*

# Layout HTML5

`< header > e < footer >`

- Usados para cabeçalhos e rodapés

`< nav >`

- Menu principal do site

`< article >`

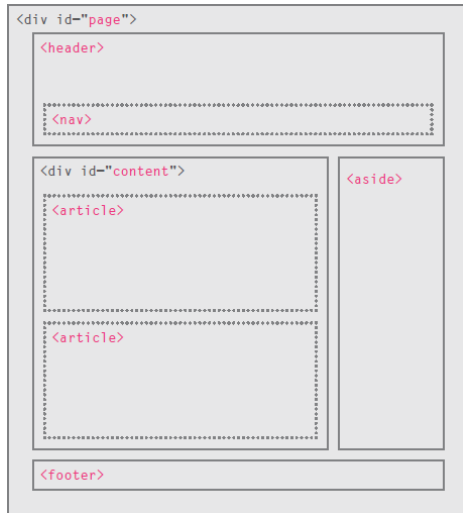
- Artigos contidos dentro de uma sessão

`< aside >`

- Conteúdo relacionado a página (posicionado à direita ou esquerda)

`< session >`

- A sessão agrupa elementos relacionados



# Atividade 08



## Flexbox

- ***display:flex;***
- Links
  - [W3Schools](#)
  - [ORIGAMID](#)

## Sass

- Links
  - [W3Schools](#)
  - [ORIGAMID](#)

## CSS Frameworks

- Bootstrap
- Foundation
- Skeleton

## Ferramentas

- [Verificar se site é responsivo](#)



J. Duckett.

*HTML & CSS: projete e construa websites.*

ALTA BOOKS, 2016.