

Programação Web

Cascading Style Sheets

Ivo Calado

Instituto Federal de Educação, Ciência e Tecnologia de Alagoas

22 de Fevereiro de 2016

Roteiro

- 1 Introdução
- 2 Sintaxe CSS
- 3 Tipos de CSS
- 4 Propriedades CSS
- 5 Casos de uso

Qual o problema com o HTML?

Suponha que desejamos criar uma aplicação Web, quais seriam os passos?

- Definimos o conteúdo que vamos querer exibir

Qual o problema com o HTML?

Suponha que desejamos criar uma aplicação Web, quais seriam os passos?

- Definimos o conteúdo que vamos querer exibir
- Definimos a estruturação do conteúdo na tela

Qual o problema com o HTML?

Suponha que desejamos criar uma aplicação Web, quais seriam os passos?

- Definimos o conteúdo que vamos querer exibir
- Definimos a estruturação do conteúdo na tela
- Fazemos uso de tags para definirmos a hierarquia do conteúdo (tags `<h>`, por exemplo)

Qual o problema com o HTML?

Suponha que desejamos criar uma aplicação Web, quais seriam os passos?

- Definimos o conteúdo que vamos querer exibir
- Definimos a estruturação do conteúdo na tela
- Fazemos uso de tags para definirmos a hierarquia do conteúdo (tags <h>, por exemplo)
- Definimos a formatação a ser adotada (cores, tamanhos de fonte, espaçamentos etc.)

Qual o problema com o HTML?

Suponha que desejamos criar uma aplicação Web, quais seriam os passos?

- Definimos o conteúdo que vamos querer exibir
- Definimos a estruturação do conteúdo na tela
- Fazemos uso de tags para definirmos a hierarquia do conteúdo (tags <h>, por exemplo)
- Definimos a formatação a ser adotada (cores, tamanhos de fonte, espaçamentos etc.)

Qual o problema com a abordagem anterior?

Qual o problema com o HTML?

Suponha que desejamos criar uma aplicação Web, quais seriam os passos?

- Definimos o conteúdo que vamos querer exibir
- Definimos a estruturação do conteúdo na tela
- Fazemos uso de tags para definirmos a hierarquia do conteúdo (tags <h>, por exemplo)
- Definimos a formatação a ser adotada (cores, tamanhos de fonte, espaçamentos etc.)

Qual o problema com a abordagem anterior?

Estamos acomodando o conteúdo a ser exibido de sua formatação!

Qual o problema com o HTML?

Suponha que desejamos criar uma aplicação Web, quais seriam os passos?

- Definimos o conteúdo que vamos querer exibir
- Definimos a estruturação do conteúdo na tela
- Fazemos uso de tags para definirmos a hierarquia do conteúdo (tags <h>, por exemplo)
- Definimos a formatação a ser adotada (cores, tamanhos de fonte, espaçamentos etc.)

Qual o problema com a abordagem anterior?

Estamos acomodando o conteúdo a ser exibido de sua formatação!
... considere um site que todos os anos altera o seu layout!



Qual o problema com o HTML?

CSS: a solução!

- Separa a marcação do HTML da apresentação do site
- Agora a responsabilidade do HTML é apenas estruturar o conteúdo enquanto que o visual é de responsabilidade do CSS
- HTML marca e estrutura textos, cabeçalhos, parágrafos, links, botões, formulários, imagens e demais elementos da página
- CSS define cores, posicionamento na tela, estilos de linhas, bordas e tudo o mais relacionado à apresentação

O que é?

- Acrônimo de **C**ascading **S**tyle **S**heets
- Define como os elementos HTML serão exibidos
- Foi adicionado ao HTML 4.0
- Podem ser definidos tanto dentro do arquivo HTML quanto em um arquivo separado (*.css)

O que é?

- Acrônimo de **C**ascading **S**tyle **S**heets
- Define como os elementos HTML serão exibidos
- Foi adicionado ao HTML 4.0
- Podem ser definidos tanto dentro do arquivo HTML quanto em um arquivo separado (*.css)

Quando utilizar dentro do próprio HTML e quando deve estar em um arquivo separado?

O que é?

- Acrônimo de **C**ascading **S**tyle **S**heets
- Define como os elementos HTML serão exibidos
- Foi adicionado ao HTML 4.0
- Podem ser definidos tanto dentro do arquivo HTML quanto em um arquivo separado (*.css)

Quando utilizar dentro do próprio HTML e quando deve estar em um arquivo separado?

Se a formatação representar poucos elementos, pode ser utilizado dentro do próprio arquivo. Porém, se a formatação do CSS abranger todo o conteúdo da página e/ou a aplicação Web for composta de diversas páginas, deve ser utilizado em um arquivo separado de modo a possibilitar reutilização!

A regra CSS e sua sintaxe

- Uma regra CSS é uma declaração que segue uma sintaxe própria e que define como será aplicado estilo a um ou mais elementos HTML
- Um conjunto de regras CSS formam uma Folha de Estilos.

Uma regra CSS, na sua forma mais elementar, compõe-se de três partes: um seletor, uma propriedade e um valor e tem a seguinte sintaxe

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

Definindo os elementos

- **Seletores:** é o elemento HTML para qual a regra será válida. Pode ser uma tag, uma classe, ou por uma ID etc. (por exemplo: <p>, <h1>, <form>, .minhaclasse, etc...)
- **Propriedade:** é o atributo do elemento HTML ao qual será aplicada a regra (por exemplo: font, color, background, etc...)
- **Valor:** é a característica específica a ser assumida pela propriedade



Definindo os elementos

- **Seletores:** é o elemento HTML para qual a regra será válida. Pode ser uma tag, uma classe, ou por uma ID etc. (por exemplo: <p>, <h1>, <form>, .minhaclasse, etc...)
- **Propriedade:** é o atributo do elemento HTML ao qual será aplicada a regra (por exemplo: font, color, background, etc...)
- **Valor:** é a característica específica a ser assumida pela propriedade

A atribuição dos valores às propriedades é feito através do operador ':' e não do operador '='!

Definindo os elementos

- **Seletores:** é o elemento HTML para qual a regra será válida. Pode ser uma tag, uma classe, ou por uma ID etc. (por exemplo: <p>, <h1>, <form>, .minhaclasse, etc...)
- **Propriedade:** é o atributo do elemento HTML ao qual será aplicada a regra (por exemplo: font, color, background, etc...)
- **Valor:** é a característica específica a ser assumida pela propriedade

A atribuição dos valores às propriedades é feito através do operador ':' e não do operador '='!

O operador ';' é opcional se houver apenas uma propriedade ou no caso da última!

Exemplos

Listing 1: Elementos 'p' e 'body' customizados

```
p {
font-size: 12px;
}

body {
color: #000000;
background: #FFFFFF;
font-weight: bold;
}
```

Listing 2: Alteração da fonte para o conteúdo do elemento 'h3'

```
h3 {
font-family: "Comic Sans MS"
}
```



Comentários

É possível introduzir comentários ao longo das regras CSS. Para tal, colócase entre os elementos `/* */`

```
seletor
{
    propriedade: valor; /*O que estiver aqui não será
                        considerado!*/
}
```

O que é e como fazer?

- Muitas vezes precisamos definir a mesma regra para vários seletores
- Deste modo, é necessário agrupar todos os elementos em uma única regra

Listing 3: Exemplo de agrupamento de seletores

```
h1, h2, h3, h4, h5, h6 {  
  color: #00FF00;  
}
```



O seletor **class**

- Até agora vimos a aplicação de estilos às *tags*
- Mas quais são as limitações disso?

O seletor **class**

- Até agora vimos a aplicação de estilos às *tags*
- Mas quais são as limitações disso?
- Não é possível definir diferentes estilos ao mesmo *tag* ao mesmo tempo!
- Nestes casos fazem-se uso dos seletores **class** e **id**

O seletor *class*

O seletor **class** permite que você defina seus próprios identificadores que são utilizados nas regras de estilo.

```
elemento.minhaclasse {
    propriedade: valor;
}
```

```
p.cor-um {
    color:#000;
}
```

```
p.cor-dois {
    color:#0ff;
}
```



Usando o seletor *class*

```
<p class = "cor-um"> este parágrafo será na cor preta.</p>
```

```
<p class = "cor-dois">este parágrafo será na cor azul.</p>
```

Extendendo o seletor *class* para várias *tags*

Ao criar uma classe você talvez queira que ela seja aplicável a qualquer elemento HTML. Neste caso basta que se omita o nome do elemento antes da classe.

```
.cor-tres {
    color: #00f;
}
```

Usando o seletor id

- O seletor ID difere do seletor classe, por ser ÚNICO
- Um seletor ID de determinado nome só pode ser aplicado a UM e somente UM elemento HTML dentro do documento

```
#meuID {
    propriedade: valor;
}
```

```
p#cor {
    color: #00f;
}
```



Usando o seletor id

`<p id = "cor"> este parágrafo será na cor preta.</p>`

Quando utilizar o seletor **id** e quando utilizar **class**

Quando utilizar o seletor **id** e quando utilizar **class**

O seletor **id** deve ser usado quando apenas 1 elemento deverá ser alterado. O elemento **class** será utilizado quando muitos elementos serão customizados

Inserção de folhas de estilo

Existem 3 formas de inserção de código CSS

- Definição em arquivo externo
- Definição no elemento **header**
- Definição inline

Folhas de estilo externos

- Definido em um arquivo com extensão .css
- Após isso, ele é referenciado de duas formas

```
<head>
```

```
...
```

```
  <link rel="stylesheet" type="text/css" href="
    estilo.css">
```

```
...
```

```
</head>
```

```
<head>
```

```
...
```

```
<style type="text/css">
```

```
  @import url("estilo.css");
```

```
</style>
```

```
...
```

Folha de estilo definido no header

- As folhas de estilos são definidos dentro da table **style**

```
<head> <style type="text/css">
<!--
body {
    background: #000;
    url(imagens/minhaimagem.gif);
}
h3 {color: #f00;}
p {
    margin-left: 15px;
    padding: 1.5em;
}
-->
</style>
</head>
```


Folha de estilo inline

- O estilo é definido dentro do parâmetro **style**

Folha de estilo inline

- O estilo é definido dentro do parâmetro **style**

```
<p style="color:#000; margin: 5px;">
```

Aqui um parágrafo de cor preta e com 5px nas 4 margens.

```
</p>
```



Conflito entre estilos

O que acontece quando dois estilos são aplicados ao mesmo elemento em escopos diferentes?

Conflito entre estilos

O que acontece quando dois estilos são aplicados ao mesmo elemento em escopos diferentes?

Há uma situação de conflito.

Conflito entre estilos

O que acontece quando dois estilos são aplicados ao mesmo elemento em escopos diferentes?

Há uma situação de conflito.

Neste caso é utilizado o efeito Cascata (*Cascade*)

Então o que acontece?

Conflito entre estilos

O que acontece quando dois estilos são aplicados ao mesmo elemento em escopos diferentes?

Há uma situação de conflito.

Neste caso é utilizado o efeito Cascata (*Cascade*)

Então o que acontece? É utilizado o valor mais interno!



Customizando a fonte

As propriedades para as fontes, definem as características (os valores na regra CSS) das letras que constituem os textos dentro dos elementos HTML

- **color:** cor da fonte
- **font-family:** tipo da fonte
- **font-size:** tamanho da fonte
- **font-style:** estilo da fonte
- **font-variant:** fontes maiúsculas de menor altura
- **font-weight:** nível de negrito

Color

Existem três formas de representar cores:

Color

Existem três formas de representar cores:

- código hexadecimal: #FFFFFF
- código rgb: rgb(255,235,0)
- nome da cor: red, blue, green...etc

Ver exemplo

Font-family

Define o nome das fontes a serem utilizadas

```
<style type="text/css">
<!--
h2 {font-family: arial, helvetica, serif;}
p {font-family: sans-serif;}
-->
</style>
```

Font-size

Especifica o tamanho da fonte:

- xx-small
- x-small
- small
- medium
- ...
- length: uma medida reconhecida pelas CSS (px, pt, em, cm, ...)
- %

Ver exemplo

Font-style

Define estilos de fontes como itálico, oblicuo

```
<style type="text/css">
<!--
h1 {font-style: italic;}
h2 {font-style: normal;}
p {font-style: oblique;}
-->
</style>
```

font-variant

Define variações para fontes. Existem duas variantes principais:

- **normal:** fonte normal
- **small-caps:** transforma em maiúsculas de menor altura

```
<style type="text/css">
<!--
h3 {font-variant: normal;}
p{font-variant: small-caps;}
-->
p</style>
```

font-weight

Define a intensidade da cor (nível de negrito)

- normal
- bold
- bolder
- lighter
- 100
- 200
- ...

Ver exemplo

Formatação de texto

Além das fontes, é possível realizar diversas customizações no texto propriamente dito

- **letter-spacing:** espaçamento entre letras
- **word-spacing:** espaçamento entre palavras
- **text-align:** alinhamento
- **text-decoration:** decoração de texto (underline, overline, blink etc)
- **text-indent:** indentação do texto
- **text-transform:** transformação de texto (maiúsculo para minúsculo e vice-versa)
- **white-space:** especifica como os espaços em branco serão tratados

letter-spacing

- **normal:** é o espaçamento default
- **length:** uma medida reconhecida pelas CSS (px, pt, cm, ...) São válidos valores negativos

```
<style type="text/css">
<!--
h2 { letter-spacing: 1.2cm; }
p { letter-spacing: 0.4cm; }
-->
</style>
```



word-spacing

Segue a mesma estrutura de espaçamento de palavras

```
<style type="text/css">
<!--
h2 {word-spacing: 1.8em;}
p {word-spacing: 80px;}
-->
```

text-align

Define o estilo de alinhamento

- left
- right
- center
- justify

Ver demais exemplos na página...

text-indent

Especifica o recuo de um texto em termo de graduação

```
<style type="text/css">
<!--
h3 {text-indent: 80px;}
p {text-indent: 3em;}
-->
</style>
```

Margens

A propriedade para margens, define um valor para espessura das margens dos elementos HTML

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

```
<style type="text/css">
<!--
p {margin-top: 2cm;}
-->
</style>
```



Bordas

- As propriedades para as bordas, definem as características (os valores na regra CSS) das quatro bordas de um elemento HTML
- Existem em torno de 20 diferentes propriedades que podem ser customizadas!!
- Todavia, todas as propriedades estão relacionadas à largura, estilo e cor das bordas

```
<style type="text/css">
<!--
p {
    border-width: thin;
    border-top-style: hidden;
    border-right-color: rgb(255,235,0);
}
```

Padding

Uma outra propriedade que pode ser configurada é a distância entre a borda e o elemento HTML

São definidas as propriedades: **padding-top**, **padding-right**, **padding-bottom**, **padding-left**

```
<style type="text/css">
<!--
p {
    padding-top: 2cm;
}
-->
</style>
```



Outras propriedades CSS

Além das propriedades já apresentadas, diversas outras opções de configuração são possíveis, como:

- Background
- Listas
- Entrelinhas
- etc

Visibilidade

- É possível tornar um elemento “invisível” a partir do CSS
- Existem duas opções com resultados levementes diferentes

```
h1.hidden { visibility: hidden; }
h1.hidden { display: none; }
```



Posicionamento

- Css possibilita customizar a posição do elemento pretendido
- É possível posicionar um elemento a esquerda, direita, cima, embaixo e inclusive com sobreposição entre elementos
- O posicionamento de elementos podem ser fixos ou relativo

Elementos Fixos

- Elementos não se movem com a barra de rolagem

```
p.pos_fixed  
{  
  position:fixed;  
  top:30px;  
  right:5px;  
}
```

Posicionamento relativo

- Aqui o posicionamento é relativo ao que deveria ser sua posição “original”

```
h2.pos_left
{
  position: relative;
  left: -20px;
}
h2.pos_right
{
  position: relative;
  left: 20px;
}
```



Posicionamento absoluto

- Com posicionamento absoluto é possível definir onde cada elemento vai se encontrar exatamente
- Neste tipo de posição pode haver sobreposição de elementos

```
h2
{
  position: absolute;
  left: 100px;
  top: 150px;
}
```



Sobreposição de elementos

- Conforme apresentado com o posicionamento absoluto, é possível ocorrer sobreposição de elementos
- Para tal, pode-se definir a ordem de qual elemento será exibido através do parâmetro **z-index**

```
img
{
  position: absolute;
  left: 0px;
  top: 0px;
  z-index: -1;
}
```



Pseudoclasses

Em CSS pseudoclasses são usada para adicionar efeitos especiais a alguns seletores

Sintaxe:

```
selector:pseudo-class {property:value;}
selector.class:pseudo-class {property:value;}

```

Diversos tipos de pseudoclasses podem ser utilizadas

- **Anchor** (âncoras)
- **first-child**
- Outros: focus, first-line, before, after etc.

Ancoras

Links podem ser exibidos de diferentes formas em navegadores que suportam CSS

```
a:link {color:#FF0000;} /* unvisited link */
a:visited {color:#00FF00;} /* visited link */
a:hover {color:#FF00FF;} /* mouse over link */
a:active {color:#0000FF;} /* selected link */
```

Notas

- **a:hover** DEVE ser apresentado após **a:link** e **a:visited** na definição de CSS para que haja efeito
- **a:active** DEVE ser apresentado após **a:hover** na definição CSS para que haja efeito

A pseudoclasse :first-child

Objetivo: customizar o primeiro elemento dentro de um outro elemento

Listing 4: Encontra o primeiro elemento p

```
<style type="text/css">
p: first-child
{
color: blue;
}
</style>
```



Barra de Navegação Horizontal

- A criação de barras de navegação é importante em qualquer Website
- Com CSS é possível transformar menus HTML em barras de navegação estilizadas

Barra de Navegação Horizontal

- A criação de barras de navegação é importante em qualquer Website
- Com CSS é possível transformar menus HTML em barras de navegação estilizadas

Barras de navegação == Lista de links!!

Uma barra de navegação é basicamente uma lista de links

Passo 1: criação da lista de links

```
<ul>
<li><a href="home.html">Home</a></li>
<li><a href="noticias.html">News</a></li>
<li><a href="contatos.html">Contact</a></li>
<li><a href="sobre.html">About</a></li>
</ul>
```



Passo 2: remover marcadores da lista e espaços

```
ul
{
list-style-type:none; /*Remove os marcadores*/
margin:0; /*Remove espaço entre a margem*/
padding:0; /*Remove espaço interno da célula*/
}
li
{
float:left;
}
```



Passou 3: definido comportamento dos links

```
a: link , a: visited
{
    display: block;
    width: 120px;
    font-weight: bold;
    color: #FFFFFF;
    background-color: #98bf21;
    text-align: center;
    padding: 4px;
    text-decoration: none;
    text-transform: uppercase;
}
a: hover , a: active
{
    background-color: #7A991A;
}
```

Barra de navegação vertical

Para transformar nosso exemplo em uma barra vertical temos apenas de retirar a flutuação dos elementos

```
li  
{  
  float : left ;  
}
```

Dúvidas?

Exercício I

- Com base no HTML apresentado aqui
<http://paste.pocoo.org/raw/382563/>, crie uma página como esta: <http://tinyurl.com/5t8fg4d>
- Altere o código apresentado em
<http://paste.pocoo.org/raw/382566/> retirando todo o conteúdo de formatação e movendo-o para um conteúdo CSS
- Altere o html (http://maujor.com/temas/html-padroao_zengarden-maujor.zip) adicionando as devidas formatações CSS. Usem inserção de imagens, posicionamento, transparência, alteração de fontes etc. Sejam criativos! 