

CSS - Introdução

- Folhas de estilos em cascata — Cascading Style Sheets (CSS) — é uma ferramenta fantástica para construção do layout dos seus websites.
- Um editor de texto simples é o ideal para o aprendizado de HTML e de CSS porque eles não afetam e nem modificam o código que você digita.

O que fazer com CSS?

- É uma linguagem para estilos que define o layout de documentos HTML.
- CSS controla fontes, cores, margens, linhas, alturas, larguras, imagens de fundo, posicionamentos e muito mais.
- HTML pode ser (in)devidamente usado para definir o layout de websites. Contudo CSS proporciona mais opções e é mais preciso e sofisticado. CSS é suportado por todos os navegadores atuais.

Diferença entre CSS e HTML

- HTML é usado para estruturar conteúdos.
- CSS é usado para formatar conteúdos estruturados.

Os benefícios concretos do uso de CSS incluem:

- controle do layout de vários documentos a partir de uma simples folha de estilos;
- maior precisão no controle do layout;
- aplicação de diferentes layouts para servir diferentes mídias (tela, impressora, etc.);
- emprego de variadas, sofisticadas e avançadas técnicas de desenvolvimento.

Como funciona CSS?

- Muitas propriedades usadas em Cascading Style Sheets (CSS) são semelhantes àsquelas do HTML.

Sintaxe básica das CSS

seletor { **propriedade**: **valor** }

↑
Em qual tag(s)
HTML será
aplicada a
propriedade
(p. ex.: "body")

↑
A propriedade pode
ser por exemplo: cor
do fundo
("background-color")

↖ O valor da
propriedade cor do
fundo, pode ser por
exemplo: vermelha
("#FF0000")

Como funciona CSS?

- Um exemplo de folha de estilos com apenas uma regra.

H1 {font-size: 48pt}

- Nesta regra, H1 é o **seletor** e {font-size: 48pt} é o bloco da **declaração**, que estabelece um tamanho de fonte (propriedade font-size) para todos os objetos marcados com <H1>.
- Se usa dois-pontos (:) e não igual (=) para aplicar um valor a uma propriedade. Pode haver mais de uma declaração de estilo para um seletor.
- Isto pode ser feito em outro bloco. Cada linha acrescenta ou sobrepõe declarações feitas em linhas anteriores.

Como funciona CSS?

- No exemplo a seguir, o texto marcado com <H1> será azul e terá tamanho de 18pt porque a regra H1 { font-size: 18pt } ocorreu depois da regra H1 { font-size: 24pt }

H1 { font-size: 24pt }

H1 { color: blue }

H1 { font-size: 18pt }

Múltiplas declarações e seletores

- Várias declarações de estilo podem ser aplicadas de uma vez a um seletor.
- As declarações precisam ser separadas por ponto-e-vírgula (;) :

H1 {font-size: 18pt; color: blue; font-family: Caslon, serif }

BODY { background : navy; color : white }

- Os espaços em branco são ignorados pelo browser na hora de interpretar uma folha de estilos e podem ser usados para melhorar a sua legibilidade.

Múltiplas declarações e seletores

- As duas linhas do exemplo anterior ficariam mais legíveis da forma:

```
BODY {background : navy;  
      color : white }
```

```
H1 {color: blue;  
    font-size: 18pt;  
    font-family: Caslon, serif }
```


Múltiplas declarações e seletores

- Uma declaração só termina com uma fecha-chaves (}) ou com um ponto-e-vírgula (;).
- Dependendo da propriedade, esta pode ter vários valores separados por vírgulas ou valores compostos com as palavras separadas por espaços:

P { font: 12pt "Times New Roman" bold }

H2 { font-family: Arial, Helvetica, Sans-serif }

As aspas devem ser usadas quando uma palavra que tem espaços precisar ser usada.

Múltiplas declarações e seletores

- Assim como um seletor pode ter várias propriedades definidas para ele, um mesmo con-junto de propriedades pode ser aplicada a um grupo de seletores, separando-os com vírgulas:

```
H1, H2, H3 { color: blue;  
             font-size: 18pt;  
             font-family: Arial, Helvetica, Sans-serif }
```

Múltiplas declarações e seletores

- As declarações de estilo podem ser aplicadas em quase qualquer descritor HTML.
- Na prática, muitos browsers ainda têm problemas de compatibilidade, e não implementam a especificação.
- Ao utilizar folhas de estilo, deve-se respeitar os elementos HTML que possuem descritores finais freqüentemente ignorados, como `</P>`, ``, etc.
- A falta do `</P>` pode causar o "vazamento" das declarações de estilo para fora do parágrafo em alguns browsers.