Programação Web

Professor: Euclides Paim

euclides.paim@ifc.edu.br

Fundamentos de CSS

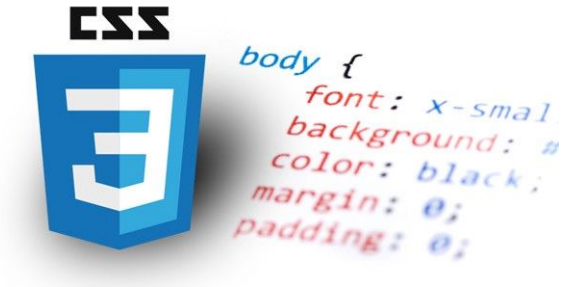
Professor: Euclides Paim

euclides.paim@ifc.edu.br

Programação Web

Sumário

- **Fundamentos de CSS**
  - Introdução;
  - Sintaxe CSS (revisão) ;
  - *Box model*;
  - Propriedade *Display*
  - *CSS Backgrounds*;
  - *CSS Borders;*
  - *CSS Margins;*
  - *CSS Padding;*
- **Resumo**



Web Design

Web Design

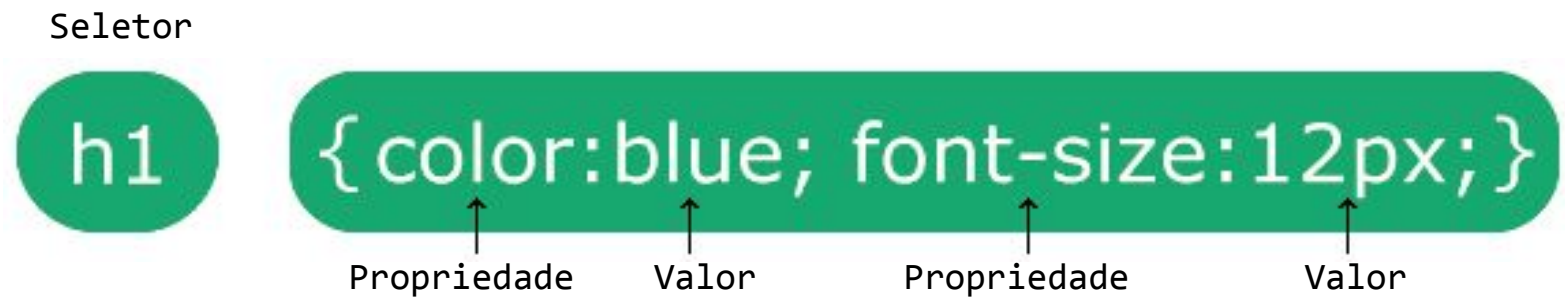
Introdução

- **O que estudamos até agora?**
  - HTML
    - Editores;
    - Elementos;
    - Atributos;
    - Imagens, tabelas, listas;
    - *Links;*
    - Classes;
    - Ids;
    - *Iframes;*
    - Elementos semânticos;
    - Formulários, elementos de formulários, tipos de *input.*
  - CSS
    - Sintaxe CSS;
    - Seletores;
    - Formas de inserir CSS, ordem de cascateamento.

Web Design

Sintaxe CSS

- **Sintaxe CSS relembrando:**
  - A sintaxe do CSS consiste em uma coleção de seletores associados a instruções.
  - O **seletor** aponta para o **elemento** HTML que desejamos estilizar.
  - O bloco de declaração contém uma ou mais **declarações** separadas por ponto e vírgula.
  - Cada declaração inclui um **nome de propriedade** CSS e um **valor**, separados por dois pontos.
  - Várias declarações CSS são separadas por ponto-e-vírgula e cercados por chaves.



Web Design

Box Model

- **Box Model**

No CSS, todos os elementos são representados como caixas retangulares. O Box Model define como o espaço de um elemento é calculado.

Componentes do Box Model:

1. **Content** – O conteúdo real do elemento (texto, imagem, etc.).
2. **Padding** – Espaço interno entre o conteúdo e a borda.
3. **Border** – A borda ao redor do elemento.
4. **Margin** – Espaço externo ao redor da borda, separando o elemento dos outros.

Web Design

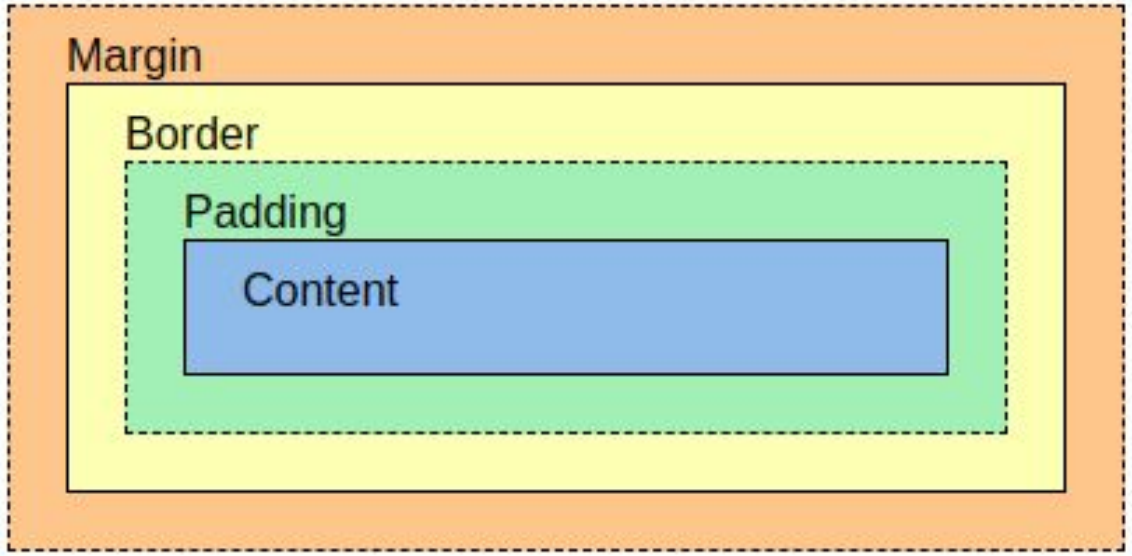
Box Model

- **CSS Box Model...**
  - Em CSS, o termo *"box model"* é usado para falar sobre *design* e *layout*.
  - O modelo de caixa CSS é essencialmente uma caixa que envolve todos os elementos HTML.
  - Consiste em:
    - **margens;**
    - **bordas;**
    - **preenchimento;**
    - **conteúdo real.**

Nota: Outros comportamentos do *box model* podem ser definidos usando a propriedade CSS ***box-sizing***. Esta propriedade define como a largura e a altura de um elemento serão calculadas.

ox Model...

- A imagem a seguir ilustra o ***box model***:



- Explicação das diferentes partes:
  - *Content:* O **conteúdo** da caixa, onde o texto e as imagens aparecem.
  - *Padding:* Reserva uma área ao redor do conteúdo com **preenchimento** transparente.
  - *Border:* Uma **borda** que circunda o preenchimento e o conteúdo.
  - *Margin:* Reserva uma área fora da borda. A **margem** é transparente.

Web Design

Box Model

- **CSS Box Model…**
  - **Fórmula do tamanho total da caixa:**

Largura Total = largura do conteúdo + (2 * padding) + (2 * border) + (2 * margin) .

Dica:

Para facilitar o cálculo, use box-sizing: border-box;, que faz com que width e height incluam padding e border, evitando a soma manual.

Web Design

Box Model

- **CSS Box Model...**
  - O modelo de caixa nos permite adicionar uma borda ao redor dos elementos e definir o espaço entre os elementos.
  - Exemplo:

```
div {
  width: 300px;
  border: 15px solid green;
  padding: 50px;
  margin: 20px;
}
```

Web Design

Box Model

- ***Width* e *Height* de um Elemento**
  - Para definir a **largura** e a **altura** de um elemento corretamente em todos os navegadores, precisamos saber como funciona o modelo de caixa.

Importante: ao definir as propriedades de largura e altura de um elemento com CSS, estamos apenas definindo a largura e a altura da **área de conteúdo**.

  - Para calcular o tamanho total de um elemento, **devemos** adicionar **preenchimento, bordas e margens**.

Web Design

Box Model

h e *Height* de um Elemento...

- Aqui está o cálculo:
  - 320px (largura)
  - + 20px (preenchimento esquerdo + direito)
  - + 10px (borda esquerda + direita)
  - + 0px (margem esquerda + direita)
  - = 350px

- A **largura** total de um elemento deve ser calculada assim:

Largura total do elemento = largura preenchimento esquerdo + preenchimento direito + borda esquerda + borda direita + margem esquerda + margem direita

- A **altura** total de um elemento deve ser calculada assim:

Altura total do elemento = altura preenchimento superior + preenchimento inferior + borda superior + borda inferior + margem superior + margem inferior

Calcule a largura total:

A figura acima tem 350px de largura. A largura total deste elemento também é 350px.

Web Design

Propriedade Display

Tipos de Display no CSS

O display define como um elemento será renderizado na página. Os principais valores são:

1. block

- Ocupa toda a largura disponível e sempre começa em uma nova linha.
- Exemplos: , <p>, <h1>, <section>.
- Exemplo CSS:

```css
p {
  display: block;
  width: 50%;
  margin: auto;
}
```

Web Design

Propriedade Display

Tipos de Display no CSS:

2. inline

- Ocupa apenas o espaço necessário e não permite definir width e height.
- Exemplos: , <a>, .
- Exemplo CSS:

```css
span {
  display: inline;
  color: blue;
}
```

Web Design

Propriedade Display

Tipos de Display no CSS:

3. inline-block

- Comporta-se como inline, mas permite definir width e height.
- Exemplo CSS:

```css
button {
  display: inline-block;
  width: 100px;
  height: 50px;
  background: lightblue;
}
```

Web Design

Propriedade Display

Tipos de Display no CSS:

4. flex (Flexbox)

- Organiza os elementos dentro de um contêiner de forma flexível.
- Usado para criar layouts mais dinâmicos.
- Exemplo CSS:

```css
.container {
  display: flex;
  justify-content: space-between;
}
```

Tipos de Display no CSS:

5. grid (CSS Grid)

- Divide o layout em uma grade de linhas e colunas.
- Ideal para layouts mais estruturados.
- Exemplo CSS:

```css
.container {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
}
```

parência.

- A propriedade `opacity` especifica a opacidade / transparência de um elemento. Pode assumir um valor de 0.0 a 1.0. Quanto menor o valor, mais transparente:

```
div {
  background-color: green;
  opacity: 0.3;
}
```

Nota: Ao usar a propriedade opacity para adicionar transparência ao plano de fundo de um elemento, todos os seus elementos filho herdam a mesma transparência. Isso pode tornar o texto dentro de um elemento difícil de ler.

- Se **não** quisermos aplicar opacidade aos elementos filhos, como no exemplo acima, devemos usar valores de cor **RGBA**. O exemplo a seguir define a opacidade da cor de fundo e não do texto:

```
div {
  background: rgba(0, 128, 0, 0.3) /* Green background with 30% opacity */
}
```

Web Design

CSS Backgrounds

- ***CSS background-image***
  - A propriedade `background-image` especifica uma imagem a ser usada como plano de fundo de um elemento. Por padrão, a imagem é repetida para cobrir todo o elemento.

```
body {
  background-image: url("bkg.png");
}
```

  - **Nota**: Ao usar uma imagem de plano de fundo, use uma imagem que não perturbe o texto.
  - A imagem de fundo também pode ser definida para elementos específicos, como o elemento <p>:

```
p {
  background-image: url("paper.gif");
}
```

Web Design

CSS Backgrounds

- ***CSS background-repeat***
  - Por padrão a propriedade `background-repeat` repete uma imagem horizontal e verticalmente. Algumas imagens devem ser repetidas apenas horizontal ou verticalmente, ou ficarão estranhas.
  - Para repetir uma imagem horizontalmente, usamos:
    - `background-repeat: repeat-x;`
  - Para repetir uma imagem verticalmente, usamos:
    - `background-repeat: repeat-y;`
  - Mostrar a imagem de fundo apenas uma vez também é especificado pela propriedade background-repeat:
    - `background-repeat: no-repeat;`

- ***CSS background-attachment***
  - A propriedade `background-attachment` especifica se a imagem de fundo deve rolar ou ficar fixa (não rolar com o resto da página):
  - Exemplo onde a imagem de fundo fica fixa:

```
body {
  background-image: url("img_tree.png");
  background-repeat: no-repeat;
  background-position: right top;
  background-attachment: fixed;
}
```

Web Design

CSS Backgrounds

- ***CSS background-position***
  - A propriedade `background-position` é usado para especificar a posição da imagem de fundo.
  - Exemplo onde a imagem de plano de fundo é posicionada no canto superior direito:

```
body {
  background-image: url("img_tree.png");
  background-repeat: no-repeat;
  background-position: right top;
}
```

- ***CSS background* (forma abreviada)**
  - Para encurtar o código, também é possível especificar todas as propriedades do fundo em uma única propriedade. Isso é chamado de propriedade abreviada (*shorthand property*).
  - Ao invés de escrever:

```
body {
  background-color: #ffffff;
  background-image: url("img_tree.png");
  background-repeat: no-repeat;
  background-position: right top;
}
```

  - Podemos usar a propriedade abreviada `background`:

```
body {
  background: #ffffff url("img_tree.png") no-repeat right top;
}
```

Web Design

CSS Borders

- ***CSS borders***
  - As propriedades CSS ***border*** permitem que você especifique o estilo, a largura e a cor da borda de um elemento.
  - A propriedade `border-style` especifica o tipo de borda a ser exibida:
    - *dotted* - Define uma borda pontilhada;
    - *dashed* - Define uma borda tracejada;
    - *solid* - Define uma borda sólida.
    - *double* - Define uma borda dupla;
    - *groove* - Define uma borda com ranhuras 3D. O efeito depende do valor da cor da borda;
    - *ridge* - Define uma borda com estrias 3D. O efeito depende do valor da cor da borda;
    - *inset* - Define uma borda inserida 3D. O efeito depende do valor da cor da borda;
    - *outset* - Define uma borda de início 3D. O efeito depende do valor da cor da borda;
    - *none* - Define sem borda;
    - *hidden* -Define uma borda oculta.

Nota: **Nenhuma** das OUTRAS propriedades CSS border (sobre as quais vamos aprender nessa aula) terá QUALQUER efeito, a menos que a propriedade *border-style* seja definida!

Web Design

CSS Borders

- ***CSS border-width***
  - A propriedade `border-width` especifica a largura das quatro bordas. A largura pode ser definida como um tamanho específico (em, *px*, *pt*, *cm*, *em*, etc) ou usando um dos três valores predefinidos: *thin, medium,* ou *thick.*
  - A propriedade `border-width` pode ter de **um** a **quatro** valores (para a borda superior, borda direita, borda inferior e borda esquerda):

```
p.three {
  border-style: solid;
  border-width: 25px 10px 4px 35px; /* top, right, bottom e left */
}
```

Web Design

CSS Borders

- ***CSS border-color***
  - A propriedade `border-color` é usada para definir a cor das quatro bordas:

```
p.one {
  border-style: solid;
  border-color: red;
}
```

  - A propriedade `border-color` pode ter de um a quatro valores (para a borda superior, borda direita, borda inferior e borda esquerda).

```
p.one {
  border-style: solid;
  border-color: red green blue yellow; /* top, right, bottom and left */
}
```

Nota: Se a cor da borda não for definida, ela herda a cor do elemento.

Web Design

CSS Borders

- ***CSS border*** (Lados Individuais)
  - A partir dos exemplos nas páginas anteriores, você viu que é possível especificar uma borda diferente para cada lado.

```
p {
  border-top-style: dotted;
  border-right-style: solid;
  border-bottom-style: dotted;
  border-left-style: solid;
}
```

  - O exemplo acima aplica o mesmo efeito que a declaração a seguir:

```
p {
  border-style: dotted solid;
}
```

Web Design

CSS Borders

- ***CSS border*** (Lados Individuais)**...**
  - Então, é assim que funciona: Se a propriedade border-style tiver **quatro** valores:
    - **`border-style: dotted solid double dashed;`**
      - borda superior é: pontilhada;
      - borda direita é: sólida;
      - borda inferior é: dupla;
      - borda esquerda é: tracejada.
  - Se a propriedade border-style tiver **três** valores:
    - **`border-style: dotted solid double;`**
      - borda **superior** é: pontilhada;
      - borda **direita** e **esquerda** são: sólidas;
      - borda **inferior** é: dupla.

Web Design

CSS Borders

- ***CSS border*** (Lados Individuais)**...**
  - Se a propriedade border-style tiver **dois** valores:
    - **`border-style: dotted solid;`**
      - borda **superior** e **inferior** são: pontilhadas;
      - borda **direita** e **esquerda** são: sólidas.
  - Se a propriedade border-style tiver **um** valores:
    - **`border-style: dotted;`**
      - **todas** as bordas são: pontilhadas.

Web Design

CSS Borders

- ***CSS border*** (forma abreviada)
  - Para encurtar o código, também é possível especificar todas as propriedades individuais da borda em uma propriedade.
  - A propriedade `border` é uma propriedade abreviada para as seguintes propriedades de borda individuais:
    - *border-width*
    - *border-style* (obrigatório)
    - *border-color*.
  - Exemplo:

```
p {
  border: 5px solid red;
}
```

Web Design

CSS Borders

- ***CSS border-radius***
  - A propriedade `border-radius` é usada para adicionar bordas arredondadas a um elemento:
  - Exemplo:

```
p {
  border: 2px solid red;
  border-radius: 5px;
}
```

Web Design

CSS Margin

- ***CSS margin***
  - As margens são usadas para criar espaço **ao redor** dos elementos, **fora** de quaisquer bordas definidas
  - Com CSS, temos controle total sobre as margens. Existem propriedades para definir a margem para cada lado de um elemento (superior, direita, inferior e esquerda):
    - *margin-top*
    - *margin-right*
    - *margin-bottom*
    - *margin-left*

  - Todas as propriedades de margem podem ter os seguintes valores:
    - ***auto*:** o navegador calcula a margem
    - ***length:*** especifica uma margem em px, pt, cm, etc.
    - **%:** especifica uma margem em% da largura do elemento que o contém
    - ***Inherit:*** especifica que a margem deve ser herdada do elemento pai

- ***CSS margin* (forma abreviada)**
  - Para encurtar o código, é possível especificar todas as propriedades de margem em uma propriedade. A propriedade `margin` é uma propriedade abreviada para as seguintes propriedades de margem individuais:
    - *margin-top*
    - *margin-right*
    - *margin-bottom*
    - *margin-left*
  - Se a propriedade `margin` tiver quatro valores:
    - **`margin: 25px 50px 75px 100px;`**
      - Margem **superior** é: 25px.
      - Margem **direita** é: 50px.
      - Margem **inferior** é: 75px.
      - Margem **esquerda** é: 100px.

Web Design

CSS Margin

- ***CSS margin* (forma abreviada)**
  - Se a propriedade `margin` tiver **três** valores:
    - **`margin: 25px 50px 75px;`**
      - Margem **superior** é: 25px.
      - Margem **direita** e **esquerda** são: 50px.
      - Margem **inferior** é: 75px.
  - Se a propriedade `margin` tiver **dois** valores:
    - **`margin: 25px 50px;`**
      - Margem **superior** e **inferior** são: 25px.
      - Margem **direita** e **esquerda** são: 50px.
  - Se a propriedade `margin` tiver **um** valor:
    - **`margin: 25px;`**
      - Todas as margens tem: 25px.

Web Design

CSS Margin

- **O valor *auto***
  - Podemos definir a propriedade `margin` como ***auto*** para centralizar horizontalmente o elemento em seu contêiner. O elemento então ocupará a largura especificada e o espaço restante será dividido igualmente entre as margens esquerda e direita.
  - Exemplo:

```
div {
  width: 300px;
  margin: auto;
  border: 1px solid red;
}
```

Web Design

CSS Margin

- **O valor *inherit***
  - Este exemplo permite que a margem esquerda do elemento `<p class = "ex1">` seja **herdada** do elemento pai (`<div>`).
  - Exemplo:

```
div {
  border: 1px solid red;
  margin-left: 100px;
}

p .ex1 {
  margin-left: inherit;
}
```

- **Redução da margem (*collapse*)**
  - As margens superior e inferior dos elementos às vezes são reduzidas em uma única margem que é igual à maior das duas margens. Isso não acontece nas margens esquerda e direita! Apenas as margens superior e inferior! Exemplo:

```
h1 {
  margin: 0 0 50px 0;
}

h2 {
  margin: 20px 0 0 0;
}
```

  - No exemplo acima, o elemento <h1> tem uma margem inferior de 50px e o elemento <h2> tem uma margem superior definida para 20px. O bom senso pode sugerir que a margem vertical entre <h1> e <h2> seria um total de 70px (50px + 20px). Mas, devido ao colapso (***collapse***) da margem, a margem real acaba sendo 50px.

Web Design

CSS Padding

- **CSS *Padding***
  - O preenchimento ***(padding)*** é usado para criar espaço ao redor do conteúdo de um elemento, dentro de quaisquer bordas definidas.
  - Existem propriedades para definir o preenchimento para cada lado de um elemento (superior, direito, inferior e esquerdo):
    - *padding-top*
    - *padding-right*
    - *padding-bottom*
    - *padding-left*
  - Todas as propriedades `padding` podem ter os seguintes valores:
    - ***length:*** especifica uma margem em px, pt, cm, etc.
    - **%:** especifica uma margem em% da largura do elemento que o contém
    - ***Inherit:*** especifica que a margem deve ser herdada do elemento pai.

Nota: Valores negativos não são permitidos.

Web Design

CSS Padding

- **CSS *Padding* (forma abreviada)**
  - A propriedade padding é uma propriedade abreviada para as seguintes propriedades individuais de preenchimento:
    - *padding-top*
    - *padding-right*
    - *padding-bottom*
    - *padding-left*

  - Se a propriedade `padding` tiver **quatro** valores:
    - **`padding: 25px 50px 75px 100px;`**
      - Preenchimento **superior** é: 25px.
      - Preenchimento **direita** é: 50px.
      - Preenchimento **inferior** é: 75px.
      - Preenchimento **esquerda** é: 100px.

Web Design

CSS Padding

- **CSS *Padding* (forma abreviada)**
  - Se a propriedade `padding` tiver **três** valores:
    - **`padding: 25px 50px 75px;`**
      - Preenchimento **superior** é: 25px.
      - Preenchimento **direita** e **esquerda** são: 50px.
      - Preenchimento **inferior** é: 75px.
  - Se a propriedade `padding` tiver **dois** valores:
    - **`padding: 25px 50px;`**
      - Preenchimento **superior** e **inferior** são: 25px.
      - Preenchimento **direita** e **esquerda** são: 50px.
  - Se a propriedade `padding` tiver **um** valor:
    - **`padding: 25px;`**
      - **Todos** os preenchimentos serão: 25px.

Web Design

Resumo

- **Fundamentos de CSS**
  - Introdução;
  - Sintaxe CSS (revisão) ;
  - *Box model*;
  - Propriedade *Display*
  - *CSS Backgrounds*;
  - *CSS Borders;*
  - *CSS Margins;*
  - *CSS Padding;*
- **Resumo**

Referências

Referências Básicas

SILVA, Maurício Samy. CSS3: desenvolva aplicações web profissionais com uso dos poderosos recursos de estilização das CSS3. São Paulo: Novatec, 2012.

SILVA, Maurício Samy. HTML 5: a linguagem de marcação que revolucionou a web. São Paulo: Novatec, 2011.

NIEDERAUER, Juliano. Desenvolvendo websites com PHP: aprenda a criar websites dinâmicos e interativos com PHP e bancos de dados. 2. ed. rev. e atual. São Paulo: Novatec, 2011.

Referências Complementares

FLANAGAN, David. **o guia definitivo**. . O Really. 2012

SILVA, Maurício Samy. **Criando sites com HTML: sites de alta qualidade com HTML e CSS**. . Novatec. 2010

SOARES, Walace. **PHP 5: conceitos, programação e integração com banco de dados**. . Érica. 2010

DALL'OGLIO, Pablo. **PHP: programando com orientação a objetos**. . Novatec. 2009

DEITEL, Paul J. Ajax,. **Rich Internet applications e desenvolvimento Web para programadores**. . Pearson Prentice Hall. 2009

IEPSEN, Edécio Fernandes. **Lógica de Programação e Algoritmos com JavaScript.** Novatec. 2018.

Referências na Internet

https://www.w3schools.com

https://developer.mozilla.org/pt-BR/docs/Web

https://illustrated.dev/advancedjs