

CSS CLASSES E IDS

CLASSES E IDS

Para possibilitar uma organização melhor da página e da seleção de elementos por CSS, podemos utilizar **classes** e/ou **ids** para identificar Tags;

Uma id, como o nome diz, é uma identificação única: só pode ser utilizada uma vez no documento inteiro;

Normalmente é utilizada para identificar elementos estruturais da página;

Uma classe é reutilizável: pode se repetir na página e também combinar-se com outras (podemos pôr mais de uma classe em um elemento).

CLASSES E IDS

É perfeitamente possível fazer um site apenas com ids, apenas com classes, com uma combinação das duas, ou sem nenhuma das duas;

Tudo irá depender do seu estilo de fazer o código das páginas. O importante é definir um padrão e segui-lo;

Por exemplo, vamos imaginar um site com três blocos de conteúdo que aparecem em todas as páginas do site:

CLASSES E IDS

```
<div>
  Aqui vai o cabeçalho da Página
</div>
<div>
  Aqui vai o conteúdo da Página
</div>
<div>
  Aqui vai o rodapé da Página
</div>
```

LME - CSS

Como estes elementos sempre aparecem nas páginas e aparecem apenas uma vez em cada página, podemos utilizar uma id para cada um, através do atributo id, da seguinte forma:

```
<div id="cabecalho">
  Aqui vai o cabeçalho da Página
</div>
<div id="conteudo">
  Aqui vai o conteúdo da Página
</div>
<div id="rodape">
  Aqui vai o rodapé da Página
</div>
```

LME - CSS

Quanto a classes, podemos imaginar um site onde temos vários parágrafos de texto:

```
<p>
```

```
....
```

```
</p>
```

```
<p>
```

```
....
```

```
</p>
```

```
<p>
```

```
....
```

```
</p>
```

LME - CSS

Se utilizarmos o seletor `p` para aplicar estilos, estes serão aplicados a todos os parágrafos. Se quisermos, por exemplo, destacar um ou mais parágrafos, podemos criar uma classe com o nome "destaque" e colocá-la apenas nos parágrafos que desejamos, através do atributo `class` como no código abaixo:

```
<p class="destaque">
    ...
</p>
<p>
    ...
</p>
<p class="destaque">
    ...
</p>
```

CLASSES E IDS

Desta forma, quando selecionarmos a classe "destaque" no CSS, estaremos aplicando os estilos apenas nos parágrafos onde a classe está aplicada;

Outra vantagem é que podemos aplicar a classe "destaque" para outros tipos de elementos também, como links, entre outros;

Classes são mais versáteis que ids e por isso muitos desenvolvedores optam pelo uso de classes nos seus sites;

Veja o exemplo:

LME - CSS

```
<!DOCTYPE HTML>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <title>Classes e ID's</title>
  <link href="css/estilos.css" rel="stylesheet">
</head>
  <body>
    <div id="cabecalho">
      Aqui vai o cabeçalho da Página
    </div>
    <div id="conteudo">
      Aqui vai o conteúdo da Página
    </div>
    <div id="rodape">
      Aqui vai o rodapé da Página
    </div>

    <p class="destaque">
      Paragrafo 1
    </p>
    <p>
      Paragrafo 2
    </p>
    <p class="destaque">
      Paragrafo 3
    </p>
```

SELETORES

- Principais tipos de seletores:

Seletor de tipo: este é o tipo de seletor que utilizamos nos exemplos até agora. Com este seletor, selecionamos todas as Tags de um mesmo tipo. Por exemplo, se digitamos **a** estaremos selecionando todas as Tags **a (links)** da página e poderemos aplicar estilos a elas. Útil para estilos gerais, mas para maior especificidade utilizamos outros seletores.

SELETORES

- Principais tipos de seletores:

Seletor descendente: com este seletor, podemos escolher um ou mais elementos que estão dentro de outro, ou seja, que são descendentes do elemento principal. Exemplo: **p strong**. Com isso, selecionamos apenas **tags strong** que estão dentro de parágrafos.

SELETORES

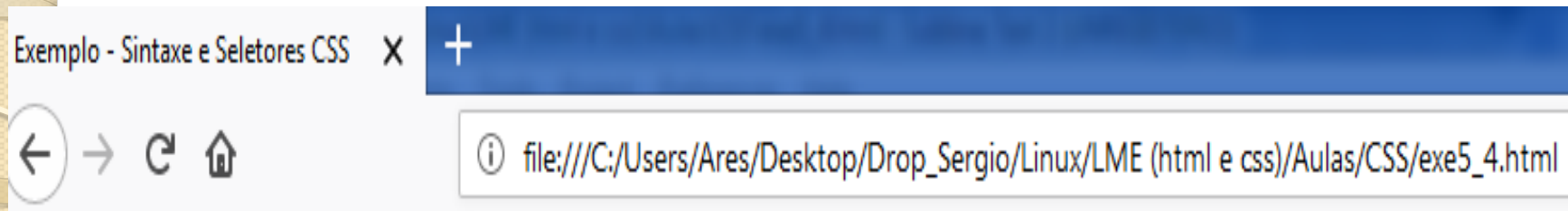
Principais tipos de seletores:

Seletor de classe: seleciona elementos com uma classe específica aplicada. Exemplo: **.destaque** seleciona todos os elementos com a classe "destaque".

Seletor de id: seleciona o elemento com a id especificada. Exemplo: **#cabecalho** irá selecionar o elemento com a id "cabecalho". Cada id é única e não pode ser repetida no mesmo documento.

LME - CSS

Exemplo de Seletores



Este é um link e ele está sendo selecionado com o seletor a.

Esta tag strong está dentro de um elemento p e está sendo selecionada com o seletor `p strong`.

Este é um parágrafo com a classe "destaque" e ele está sendo selecionado com o seletor `.destaque`.

Este é um parágrafo com a id "unico" e ele está sendo selecionado com o seletor `#unico`.

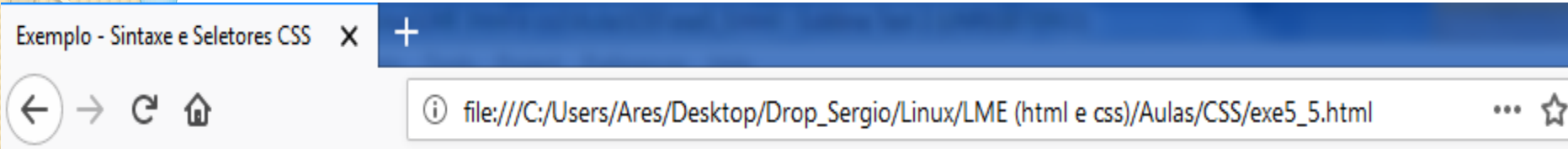
COMBINANDO SELETORES

Podemos também combinar os seletores que aprendemos acima, para conseguir diferentes elementos e partes mais específicas de nossos sites. Alguns exemplos:

- `p.destaque` - seleciona apenas os parágrafos que possuem a classe "destaque".
- `div#cabecalho h1` - seleciona tags `h1` que estejam dentro da div com a id "cabecalho".
- `#conteudo ul li a` - seleciona links (tag `a`) dentro de itens de lista dentro de tags `ul` que estejam dentro de um elemento com a id "conteudo".

LME - CSS

Exemplo de Combinando Seletores



Este é um parágrafo com a classe "destaque". Duas regras estão selecionando este elemento: `p.destaque` e `p.destaque, h1, a.saiba-mais`.

Esta é uma tag `h1` que está dentro de uma `div` com a `id` "cabecalho". Ela
seletor `div#cabecalho h1`.

- Este é um link com a classe "saiba-mais", que está dentro de uma tag `li`, que está dentro de uma `ul`, que está dentro de uma `div` com a `id` "conteudo".
`#conteudo ul li a p.destaque, h1, a.saiba-mais`.

ATIVIDADE

Pesquise outros seletores usados em CSS, pelo menos 5(cinco), mostre um exemplo de uso de cada e realize teste.

CSS

PSEUDO-CLASSES

• PSEUDO-CLASSES

Pseudo-classes são tipos de classes especiais que não são definidas pelo programador Web (já são pré-definidas).

A sua utilização é com links (tags a) para definir estilos diversos para quando o link é novo, visitado, não visitado...

O seletor de pseudo-classe é escrito com o *nome do elemento + dois pontos + nome da pseudo-classe*.

PSEUDO-CLASSES

Formas de pseudo-classes de links:

- **a:link** - estilizamos apenas os links não-visitados, ou seja, links no seu estado normal;
- **a:visited** - estilizamos apenas links visitados, ou seja, que já foram clicados;
- **a:hover** - estilizamos links quando o mouse está em cima do mesmo;
- **a:focus** - estilizamos links quando os selecionamos com o teclado, através da tecla Tab;
- **a:active** - estilizamos um link quando o mouse está sendo clicado ou pressionamos Enter, ativando o link.

PSEUDO-CLASSES

Formas de pseudo-classes de links:

Link normal (:link)

Link visitado (:visited)

Link com o cursor em cima (:hover)

Link focado com o teclado (:focus)

Link ativado (com o clique ou tecla enter) (:active)

CSS

CASCATA (E HERANÇA)

CASCATA (E HERANÇA)

A cascata é outra das características importantes das CSS. Tão importante, que está até no nome: **Cascading** Style Sheets. O que "Cascading" quer dizer é que a maioria das propriedades que aplicamos a um elemento, irá passar aos seus "herdeiros".

CASCATA (E HERANÇA)

```
<!-- HTML -->
<div class="texto">
  <p>
    Texto exemplo <strong>com um destaque</strong>.
  </p>
</div>
```

```
/* CSS */
.texto {
  color: #F00;
}
```

Note que, apesar da propriedade color ter sido aplicada à classe "texto", esta propriedade passou para o parágrafo e para o também.

CASCATA (E HERANÇA)

A utilização mais comum da cascata é definir uma fonte para a página inteira.

```
body {  
    font-family: Arial, sans-serif;  
}
```

Nesta regra, todos os elementos que estão dentro de body (ou seja, todo o conteúdo do site) irá usar a fonte Arial ou, se não estiver disponível, a fonte sans-serif padrão do sistema operacional.

CSS

ESPECIFICIDADE E PRECEDÊNCIA

ESPECIFICIDADE E PRECEDÊNCIA

O CSS obedece algumas regras de **precedência**. Quando várias regras se aplicam a um mesmo elemento, o browser precisa decidir qual delas tem precedência.

Veja o exemplo a seguir:

ESPECIFICIDADE E PRECEDÊNCIA

```
<div id="sidebar">
  <p class="descricao">
    Qual a cor do Texto?
  </p>
</div>
```

Vamos supor que temos duas regras aplicadas à div.

```
#sidebar {
  color: red;
}

div#sidebar p {
  color: blue;
}
```

Qual delas será aplicada?

ESPECIFICIDADE E PRECEDÊNCIA

Que pensou na "mais específica", acertou!

O seletor **div.sidebar p** tem precedência. Por que ele é mais específica?

Para compreender melhor sobre especificidade, há algumas regras gerais:

ESPECIFICIDADE E PRECEDÊNCIA

• Regras de Precedência:

- Regras em tags style tem precedência sobre regras em arquivos externos CSS;
- Regras em linha (atributo style) tem precedência sobre regras em tags style ou arquivos externos CSS;
- Seletores de elemento (por exemplo, p) tem a menor precedência (podemos dizer que, numa escala, este tipo de seletor vale “1”);
- Seletores de classe (por exemplo, .destaque) tem mais precedência (vale "10");
- Seletores de id (por exemplo, #conteudo) tem maior precedência (vale “100”).

ATIVIDADE

Pesquise sobre cálculo de especificidade em CSS?

CSS

Margens, Bordas e Box Model

MARGENS, BORDAS E BOX MODEL

Uma página é feita de blocos. Estes blocos são "empilhados" de cima para baixo, de acordo com a ordem do código: o que está no início do código fica em cima e os elementos subsequentes vão se acomodando em baixo uns dos outros.

Através de CSS podemos mudar esta ordem (Aula de Layout).

Inicialmente iremos estudar as principais propriedades que podemos aplicar a estes "blocos" que compõem a página.

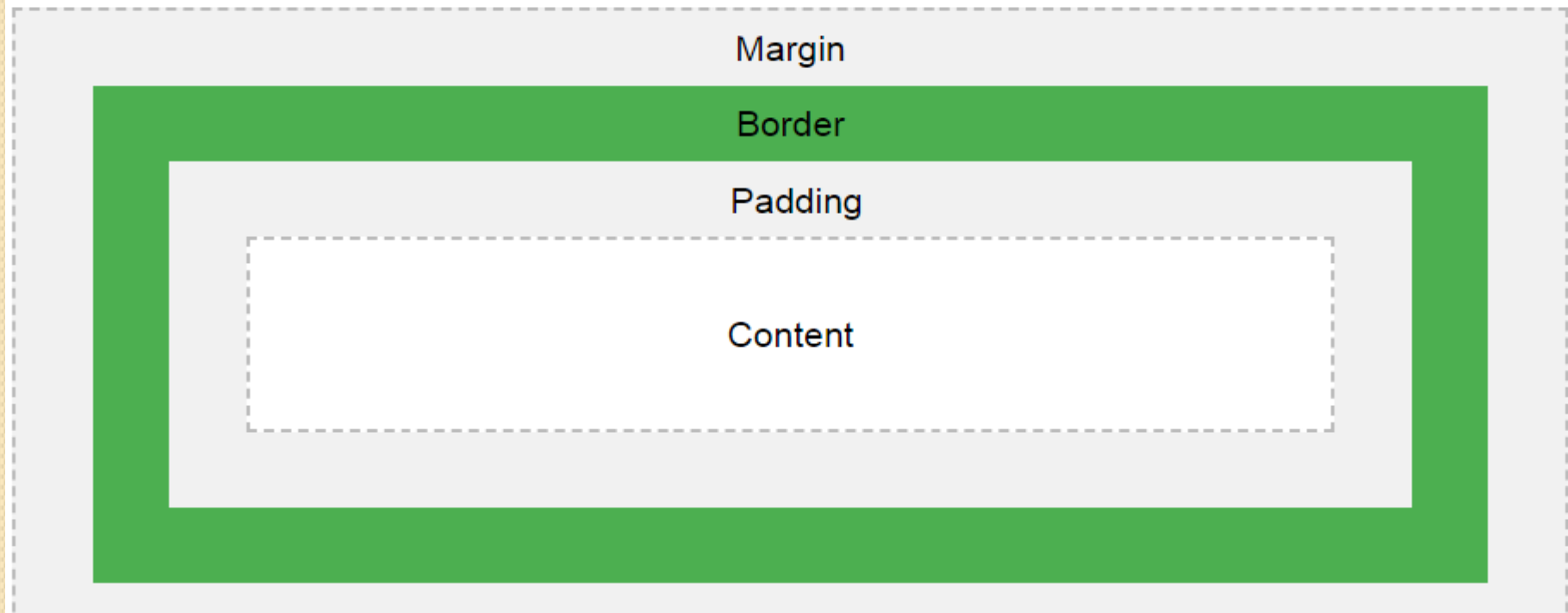
MARGENS, BORDAS E BOX MODEL

Cada elemento pode ter uma margem interna (*padding*), margem externa (*margin*) e uma borda (*border*). A forma como as margens e bordas se comportam junto com os elementos constitui o que é chamado de **box model**.

Box Model

Todos os elementos HTML podem ser considerados como caixas. Em CSS, o termo “Box Model” é usado quando se fala sobre design e layout. O Box Model CSS é essencialmente uma caixa que envolve todos os elementos HTML. Consiste em: margens, bordas, preenchimento e o conteúdo real.

Box Model



Margem

Vamos trabalhar com uma div com fundo colorido para exemplificar:

```
div.margem {  
    background: #900;  
    color: #FFF;  
}
```

Margem

Vamos aplicar uma margem interna:

```
div.margem2 {  
    background: #900;  
    color: #FFF;  
    padding: 10px;  
}
```

Margem

No exemplo anterior as divs estão “juntas” uma na outra. Vamos aplicar `margin: 10px;`

```
div.margem3 {  
    background: #900;  
    color: #FFF;  
    padding: 10px;  
    margin: 10px;  
}
```

Margem

Podemos utilizar uma forma abreviada, veja o exemplo a baixo:

```
margin: 5px 10px 20px 30px
```

(ordem cima/direita/baixo/esquerda)

```
margin: auto
```

(margem automática para todos os lados)

Bordas

A propriedade `border`, podes adicionar bordas ao redor de qualquer elemento.

Veja os componentes desta propriedade.
No exemplo a seguir, aplicamos uma borda no topo de um parágrafo:

Bordas

```
p.borda1 {  
    border-top-width: 3px;  
    border-top-style: solid;  
    border-top-color: #000;  
}
```

Podemos resumir as três declarações acima em apenas uma (*shorthand*), como o exemplo abaixo:

```
p.borda1 {  
    border-top: 3px solid #000;  
}
```

Bordas

Para controlar as bordas em um elemento inteiro, podemos utilizar as propriedades `border-width`, `border-style` e `border-color`:

```
p.borda2 {  
    /* a ordem dos lados é a mesma que com margens */  
    border-width: 2px 5px 3px 6px; /* aqui, cada lado possui uma borda de largura diferente */  
    border-style: dashed dotted; /* estilos de borda diferentes para cima-baixo e esquerda-direita */  
    border-color: #000 #090 #900 #009; /* uma cor de borda para cada lado */  
    padding: 20px;  
}
```

Bordas

Por fim podes utilizar a propriedade border para definir as bordas uniformemente no elemento:

```
p.borda3 {  
    border: 1px solid #000;  
}
```

Neste caso, foi utilizado pixels, mas podes utilizar qualquer outra unidade ou palavras-chave como thin (fina), medium (média) e thick (grossa). Depois, escolher um tipo de borda. Os principais tipos são:

solid, dashed e dotted....

Exercícios

