



Programação para Internet

Módulo 2 – Parte 2

Introdução à Linguagem CSS

Prof. Dr. Daniel A. Furtado - FACOM/UFU

Conteúdo protegido por direito autoral, nos termos da Lei nº 9 610/98

A cópia, reprodução ou apropriação deste material, total ou parcialmente, é proibida pelo autor

Conteúdo da Aula

1. Propriedades CSS
2. Centralização de Conteúdo
3. Pseudo-Classes
4. Pseudo-Elementos
5. Cascade, Especificidade e Herança
6. Dicas e Validação

Algumas Propriedades de Ajuste de Fundo

Propriedade	Descrição	Exemplos de Valores
background-color	Define a cor de fundo	<i>lightgray</i>
background-image	Define uma imagem de fundo	<i>url("images/bgImage.png")</i>
background-repeat	Modo de repetição da imagem de fundo	<i>no-repeat, repeat-x, repeat-y, repeat</i>
background-size	Tamanho da imagem de fundo	<i>100% 100%, cover</i>
background-position	Posição da imagem de fundo	<i>right top</i>

Há também a propriedade abreviada **background** para definir tudo de uma vez

Propriedades de Ajuste de Fundo - Exemplo

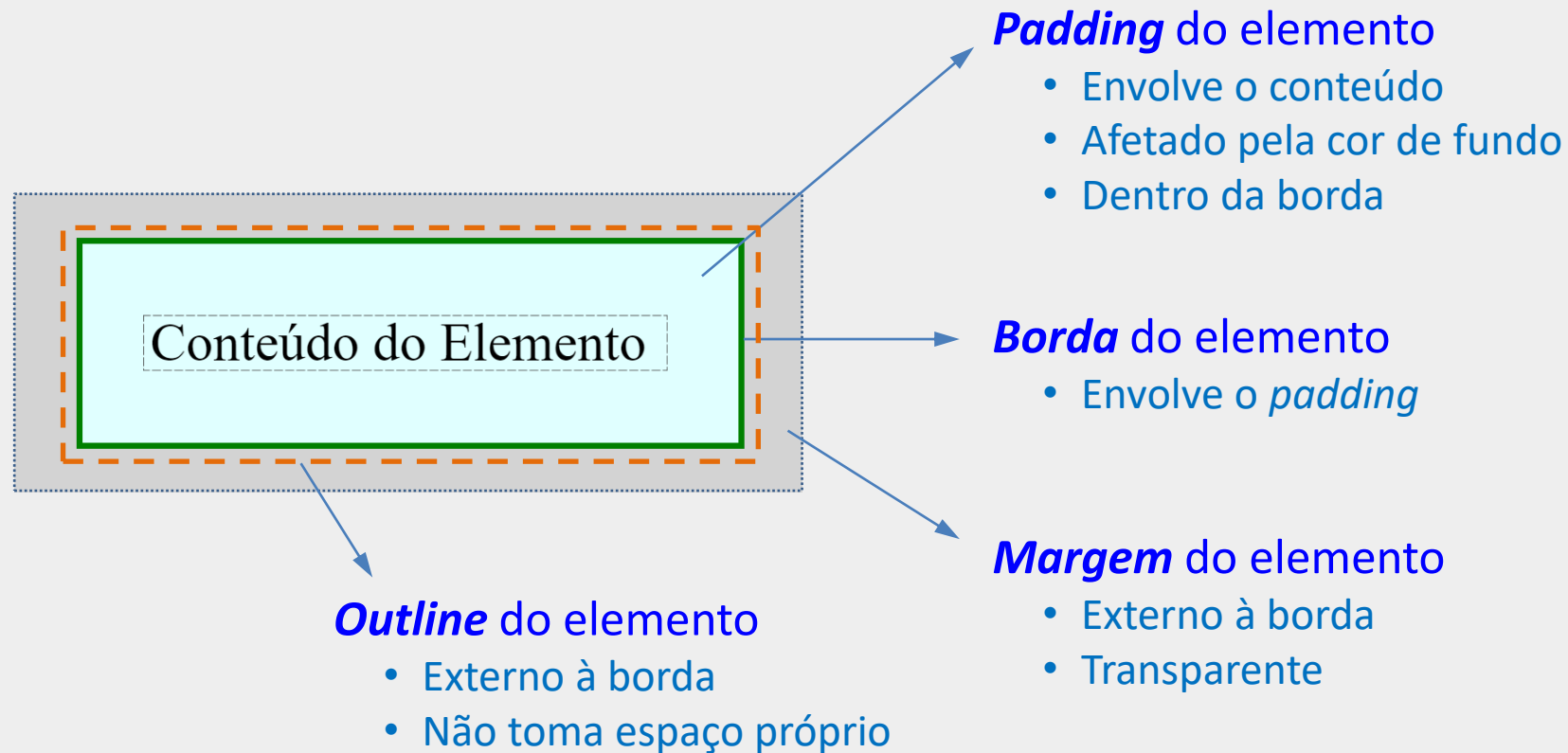
```
<style>
  body {
    background-image: url("images/bg2.jpg");
    background-repeat: no-repeat;
    background-size: cover;
  }
</style>
```

Testando imagem de fundo

Lorem ipsum dolor sit amet consectetur adipisicing elit. Libero numquam at atque ipsam voluptas magni in nihil laborum est, magnam quia repellat quisquam recusandae dolor nisi possimus quas? Exercitationem, assumenda!



Conteúdo, Padding, Borda e Margem



Propriedade *padding*

- Propriedade Abreviada

`padding: 20px 50px 80px 100px;` (superior, direita, inferior e esquerda)

`padding: 20px 40px 20px;` (superior, laterais, inferior)

`padding: 20px 40px;` (superior/inferior, laterais)

`padding: 20px;` (todos os *padding*s)

- Propriedades Individuais

`padding-top`

`padding-bottom`

`padding-left`

`padding-right`

Propriedade *margin*

- Propriedade Abreviada

`margin: 20px 50px 80px 100px;` (superior, direita, inferior e esquerda)

`margin: 20px 40px 20px;` (superior, laterais, inferior)

`margin: 20px 40px;` (superior/inferior, laterais)

`margin: 20px;` (todas as margens)

- Propriedades Individuais

`margin-top`

`margin-bottom`

`margin-left`

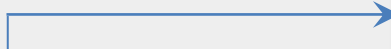
`margin-right`

Propriedade *margin*

- Margens entre dois elementos de bloco vizinhos não se somam
- Prevalecerá a maior delas

Propriedade ***border***

- Propriedade Abreviada
- Permite definir a borda por completo (4 lados)
- Sintaxe mais comum:
`border: espessura estilo cor;`
Ex.: `border: 1px solid blue;`
- A ordem dos valores não importa



solid
dotted
dashed
double
none

Definindo as Bordas Individualmente

- **border-color** define a cor de cada borda individualmente
- **border-style** define o estilo de cada borda individualmente
- **border-width** define a espessura de cada borda individualmente

Definindo as Bordas Individualmente

- Outras propriedades abreviadas (semelhantes a *border*)
 - `border-top`
 - `border-bottom`
 - `border-left`
 - `border-right`
- Propriedades constituintes
 - `border-top-color`
 - `border-top-style`
 - `etc.`

Ajustes de Borda - Exemplo

```
#p1 {  
  border: 1px solid blue;  
}  
  
#p2 {  
  border-color: red green blue black;  
  border-style: dashed dotted double solid;  
  border-width: 4px 6px 8px 10px;  
}  
  
#p3 {  
  border-color: gray;  
  border-style: solid none;  
  border-width: 2px;  
}  
  
#p4 {  
  border-top: 1px solid blue;  
}
```

Programação para Internet

Programação para Internet

Programação para Internet

Programação para Internet

Propriedade *border-radius*

- Define a **borda** e o *background* com cantos arredondados
- Algumas formas de uso:

`border-radius: 3px;`

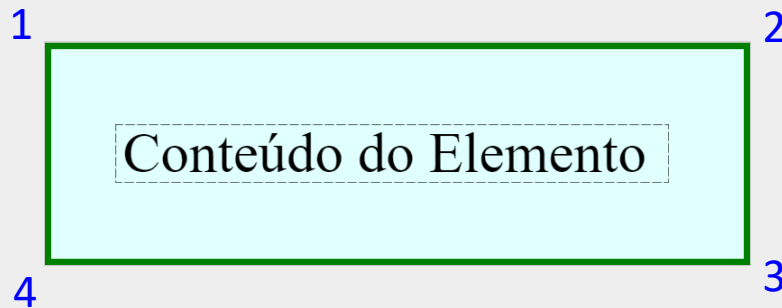
4 cantos com mesma curvatura, em pixels

`border-radius: 10%;`

4 cantos com mesma curvatura, em porc.

`border-radius: 10px 20px 30px 40px;`

cada canto com curvatura própria,
na ordem dos números a seguir:



border-radius - Exemplo

```
#p1 {  
  border-radius: 10px;  
}  
  
#p2 {  
  border-radius: 10px 20px 40px 60px  
}  
  
#p3 {  
  border-radius: 10px 40px;  
  border: none;  
}  
  
#p4 {  
  border-radius: 20% 50% 50% 20%;  
}
```

Programação para Internet (P1)

Programação para Internet (P2)

Programação para Internet (P3)

Programação para Internet (P4)

Propriedade *box-shadow*

`box-shadow: 20px 10px;`

Programação para Internet

`box-shadow: 5px 5px gray;`

Programação para Internet

`box-shadow: 5px 5px 5px gray;`

Programação para Internet

`box-shadow: 0 0 10px cyan;`

Programação para Internet

Há também outras formas de uso, com outros parâmetros

Propriedades para Ajuste de Largura e Altura

`width`
`height`

Definem a **largura** e a **altura** de um elemento (conteúdo)

`min-width`
`min-height`

Definem a **largura mínima** e a **altura mínima** do elemento

`max-width`
`max-height`

Definem a **largura máxima** e a **altura máxima** do elemento

- Não contabilizam margens, bordas e *padding*s
- Valores em pixels, porcentagem, *ems*, etc.
- Outros valores: *auto*, *max-content*, *min-content*, *fit-content*

Propriedade *display*

- Altera o modo de apresentação do elemento. Alguns valores:

inline

- exibição em nível de linha (como o)
- *width* e *height* não terão efeito em alguns elementos
- margens e *padding*s **sup.** e **inf.** de alguns elem. **não são** respeitados

block

- exibição em nível de bloco (como o <div>)
- começará em nova linha e ocupará, por padrão, toda a largura

inline-block

- exibição em nível de linha
- possibilidade de usar *width* e *height*
- margens e *padding*s **superiores** e **inferiores** **respeitados**

none

- oculta completamente o elemento, removendo do layout
- libera o espaço para outros

Propriedade *display*

- Outros valores comuns:

flex elemento tratado como container de bloco tipo flex

grid elemento tratado como container de bloco tipo grid

inline-flex elemento tratado como container de linha tipo flex

inline-grid elemento tratado como container de linha tipo grid

Há ainda vários outros possíveis valores

Propriedade *visibility*

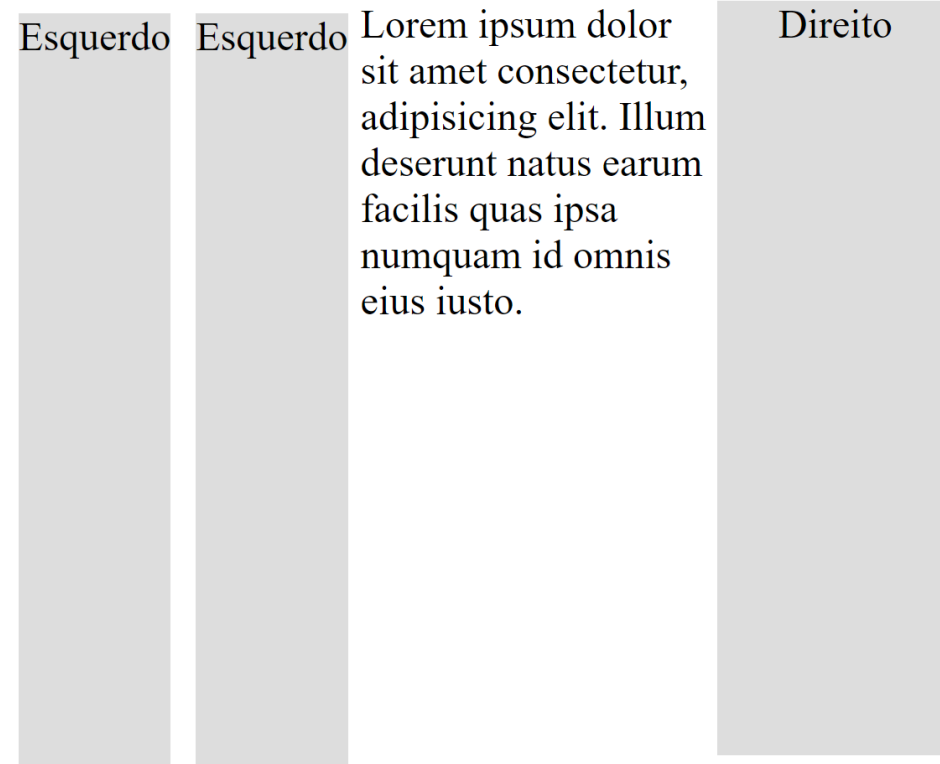
- Mostra ou oculta um elemento **sem alterar o layout**
- Valores possíveis
 - **visible** elemento visível
 - **hidden** elemento oculto, mas espaço ocupado permanece
 - **collapse** comumente usado para ocultar linhas e/ou colunas de tabelas. O espaço é removido sem alterar os tamanhos das linhas e colunas.

Propriedade *float*

- Posiciona o elemento no lado esquerdo ou direito de seu container
- Permite que texto e elementos *inline* se posicionem à sua volta
- Alguns valores
 - **left** elemento "flutua" no **lado esquerdo** do *container*
 - **right** elemento "flutua" no **lado direito** do *container*
 - **none** elemento não "flutua"

Propriedade *float* - Exemplo

```
<style>
  .esquerdo {
    background-color: #ddd;
    width: 15%;
    height: 300px;
    margin: 5px;
    float: left;
  }
  .direito {
    background-color: #ddd;
    width: 25%;
    height: 300px;
    float: right;
  }
</style>
</head>
<body>
  <aside class="esquerdo">Esquerdo</aside>
  <aside class="esquerdo">Esquerdo</aside>
  <aside class="direito">Direito</aside>
  <main><p>Lorem ipsum dolor...</p></main>
</body>
```



Propriedade *overflow*

- Trata o conteúdo do elemento quando ele extrapola a borda
- Propriedade abreviada de `overflow-x` e `overflow-y`
- Alguns valores
 - `visible` conteúdo sempre visível, ainda que fora dos limites (*default*)
 - `hidden` conteúdo cortado, se necessário, para caber no espaço
 - `scroll` barras de rolagens são sempre apresentadas
 - `auto` barras de rolagens apenas se necessário

Propriedade *overflow* - Exemplo

Sistemas de Informação

Curso de graduação oferecido pela Faculdade de Computação da Universidade Federal de Uberlândia. Possui duração de 4 anos.

Curso de graduação oferecido pela Faculdade de Computação da Universidade Federal de Uberlândia. Possui duração de 4 anos.

`overflow: visible;`

Sistemas de Informação

Curso de graduação oferecido pela Faculdade de Computação da Universidade Federal de Uberlândia. Possui duração de 4 anos.

Curso de graduação oferecido pela Faculdade de Computação

`overflow: hidden;`

Sistemas de Informação

Curso de graduação oferecido pela Faculdade de Computação da Universidade Federal de Uberlândia. Possui duração de 4 anos.

Curso de graduação oferecido

`overflow: auto;`

Propriedade *position*

- Define como o elemento é posicionado na página
- Normalmente utilizada em conjunto com *top*, *left*, *right* e *bottom*
- Valores possíveis
 - *static*
 - *relative*
 - *absolute*
 - *fixed*
 - *sticky*
- O elemento é dito **posicionado** quando *position* tem valor diferente de *static*

Propriedade *position*

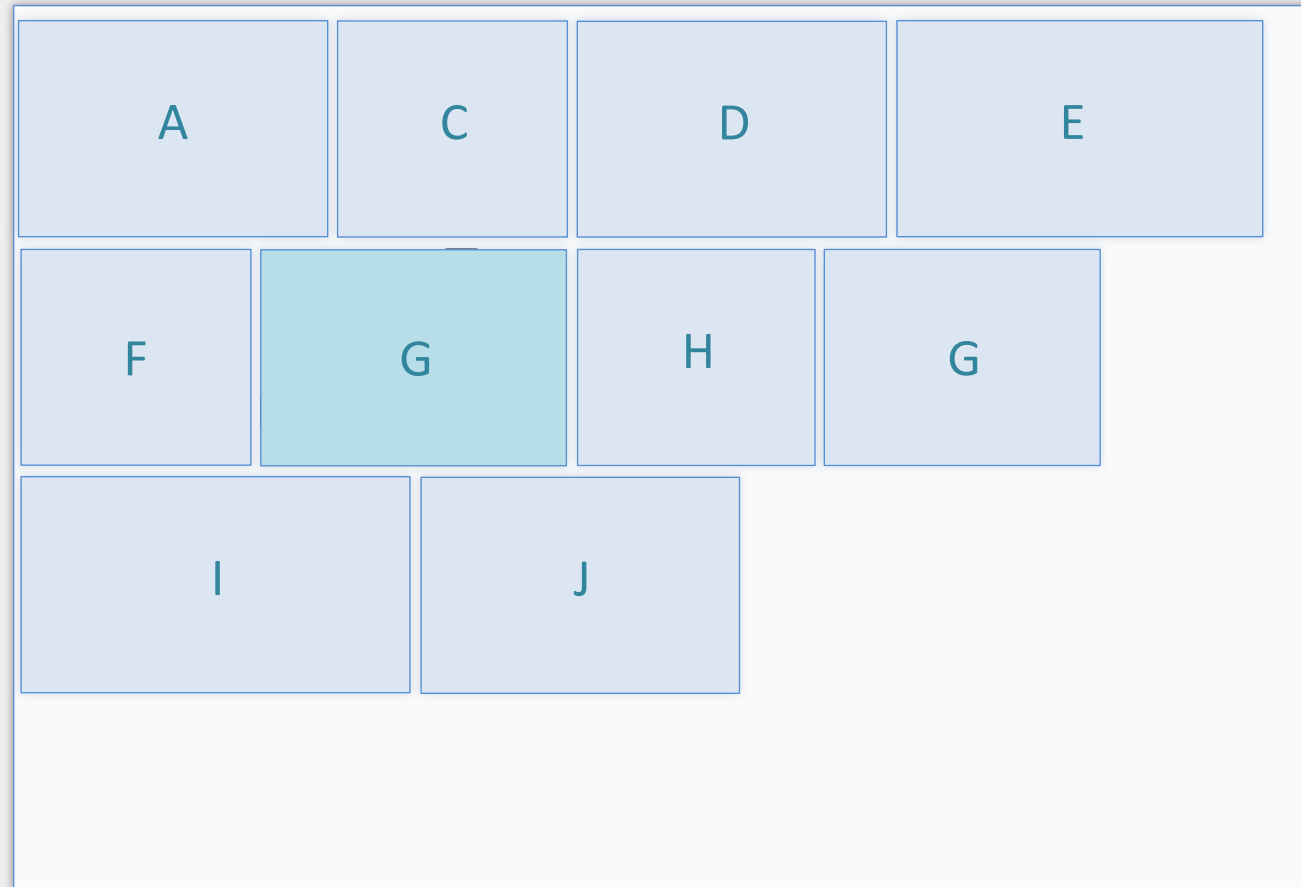
position: *static*;

- Valor padrão
- Elemento posicionado de acordo com fluxo normal do documento

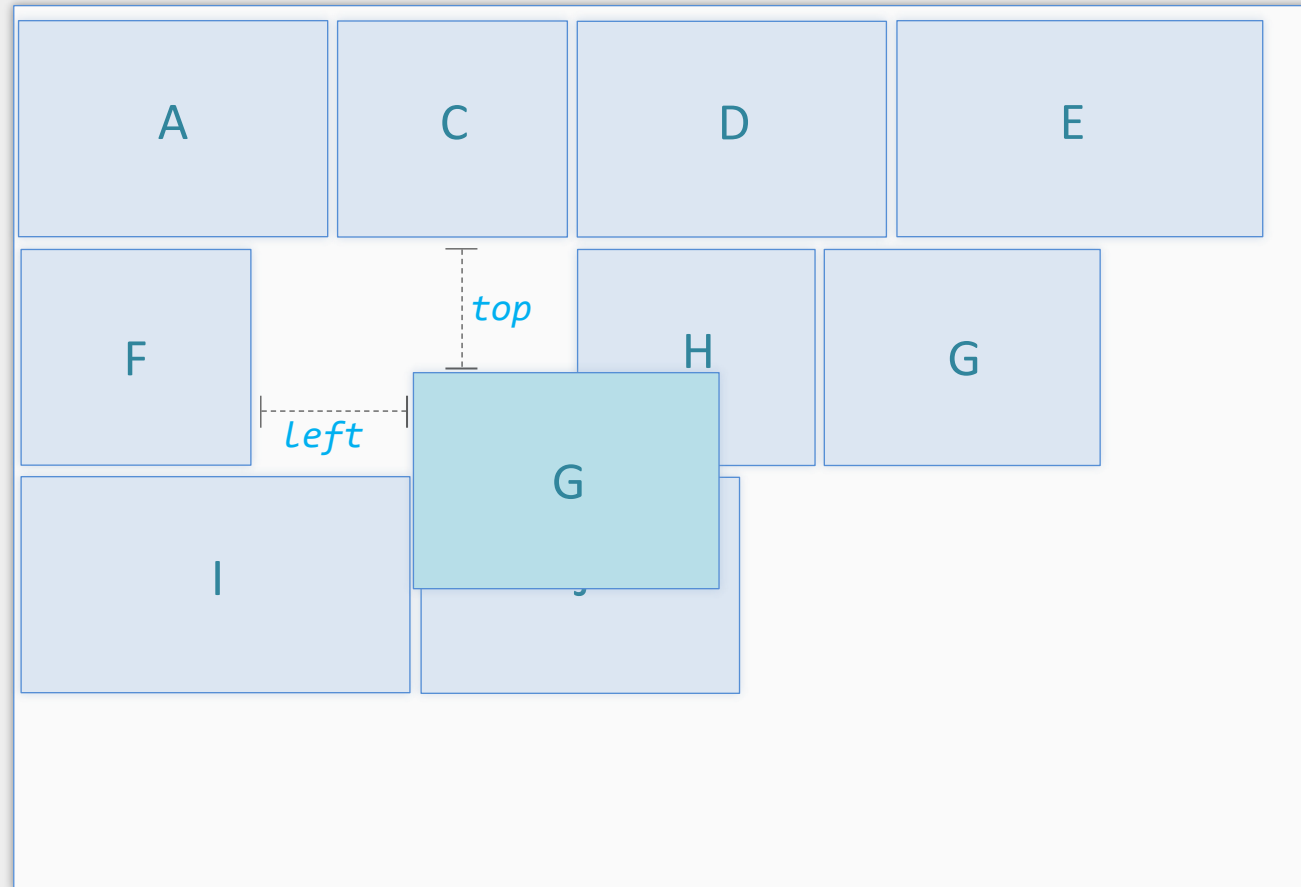
position: *relative*;

- Primeiramente posicionado de acordo com o fluxo normal
- Em seguida, é deslocado da sua posição com *top*, *left*, *right* e *bottom*
- O deslocamento não afeta a posição dos demais elementos
 - Layout geral é o mesmo do posicionamento *static*

Propriedade *position*: *relative*



Propriedade *position*: *relative*



Propriedade ***position*** - Valores

position: **absolute**;

- Elemento removido do fluxo normal
- Nenhum espaço no layout é criado para o elemento
- Posicionamento com *top*, *left*, *right* e *bottom* relativo ao:
 - Ancestral mais próximo **posicionado**, se houver
 - Caso contrário, com relação ao container inicial (elemento <html>)

Propriedade ***position*** - Valores

position: **fixed**;

- Elemento posicionado com relação à *viewport* (janela do navegador)
- Posição **não se altera com a rolagem** da página
- Posicionamento com as propriedades *top*, *left*, *right* e *bottom*
- Elemento **removido do fluxo normal** - nenhum espaço é criado no layout
- Quando impresso, aparecerá na mesma posição em todas as páginas

position: **sticky**;

- Elemento posicionado de acordo com fluxo normal do documento
- **Ocupa espaço no layout**
- Elemento "gruda" no primeiro ancestral com mecanismo de rolagem

Propriedade *transform*

- Permite mover, rotacionar, torcer e dimensionar um elemento
- Exemplos:

<code>transform: translateX(50px);</code>	move o elemento 50 pixels em x (hor.)
<code>transform: translateY(50px);</code>	move o elemento 50 pixels em y (vert.)
<code>transform: translate(30px,10px);</code>	move o elemento 30 pixels em x e 10 em y
<code>transform: rotate(45deg);</code>	rotaciona o elemento em 45 graus
<code>transform: scale(2);</code>	dobra o tamanho nas duas direções

Há muitas outras formas de uso

Centralizando um Elemento de Bloco na **Horizontal**

- Define-se uma largura com *width*
- Coloca-se as margens laterais no automático (*auto*)

```
body {  
    width: 60%;  
    margin: 0 auto;  
}
```

Para centralizar o texto dentro de um elemento, utilize:
`text-align: center;`

Centralizando um Elemento de Bloco na **Vertical**

```
.blocoCentralizadoVert {  
  position: absolute;  
  top: 50%;  
  transform: translateY(-50%);  
}  
  
.elementoPai {  
  height: xxxx;  
  position: relative;  
}
```


Centralizando um Elemento de Bloco na **Vertical**

top=50%

transform: translateY(-50%);

Centralizando um Elemento de Bloco na **Vertical**

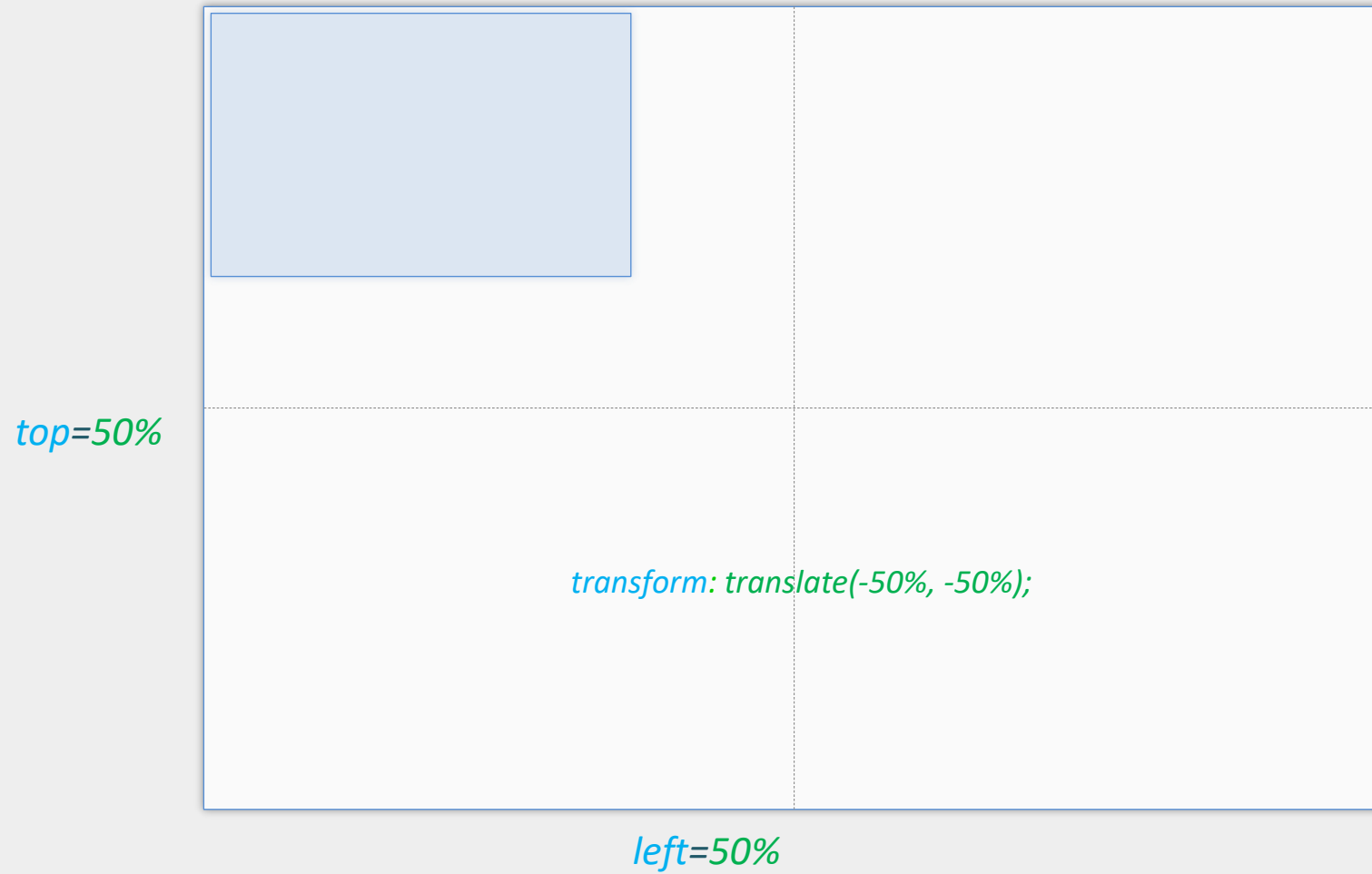
top=50%

transform: translateY(-50%);

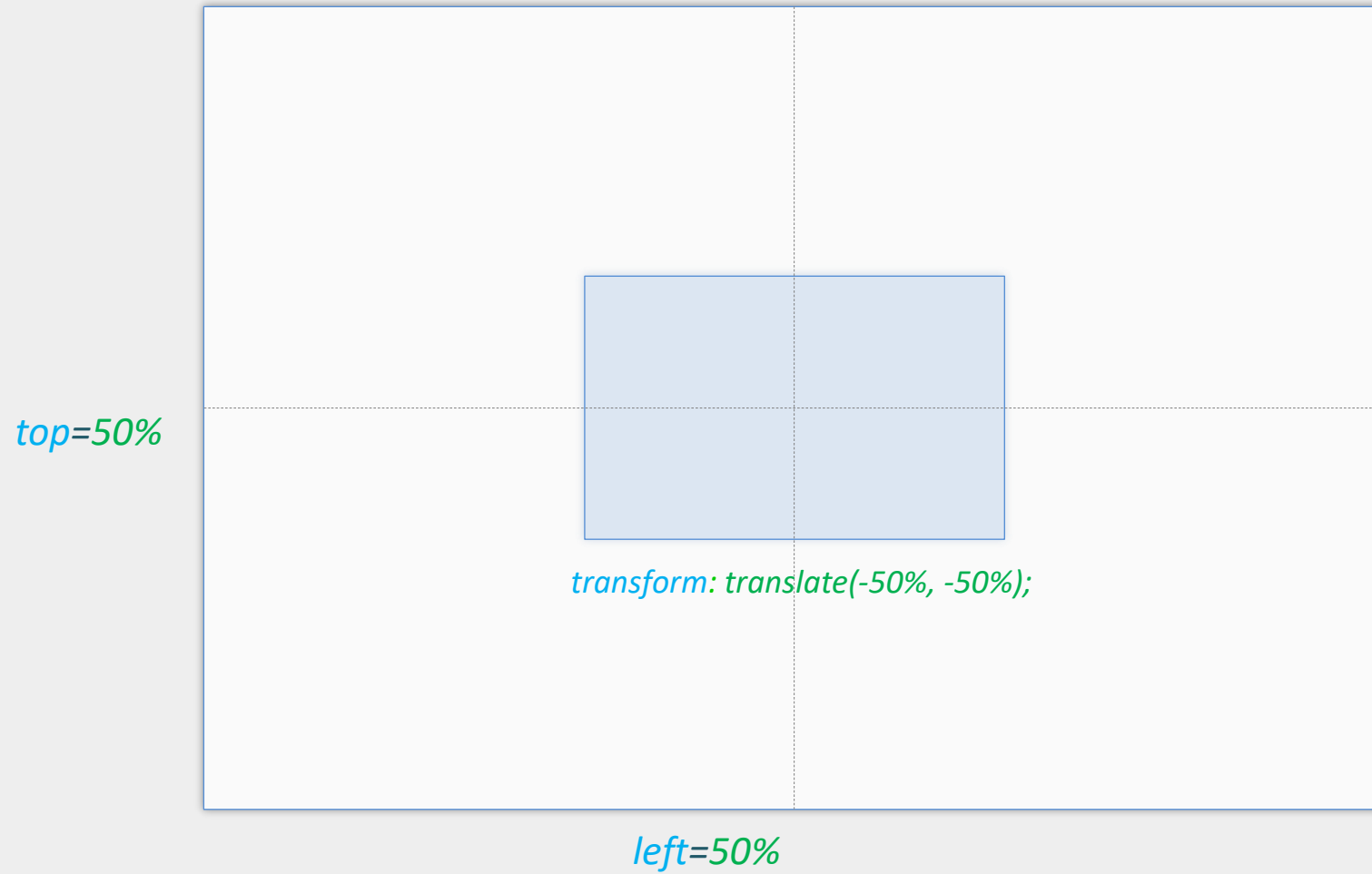
Centralizando na **Horizontal** e na **Vertical**

```
.blocoCentralizado {  
  position: absolute;  
  top: 50%;  
  left: 50%;  
  transform: translate(-50%, -50%);  
}  
  
.elementoPai {  
  height: xxxx;  
  position: relative;  
}
```

Centralizando na **Horizontal** e na **Vertical**



Centralizando na **Horizontal** e na **Vertical**



Centralizando na Horizontal e na Vertical - Exemplo

```
main {  
  width: 50%;  
  background-color: #eee;  
  border: 0.5px solid gray;  
  margin: 0;  
  padding: 2% 4%;  
  
  position: absolute;  
  top: 50%;  
  left: 50%;  
  transform: translate(-50%, -50%);  
}
```



Pseudo-Classes

- Indica um estado em particular do elemento selecionado
- Palavra-chave adicionada a um seletor
- Precedida pelo caracter dois-pontos (:)

```
a:visited {  
    color: gray;  
}
```

*Os links **que já foram visitados** aparecerão na cor cinza*

```
input:invalid {  
    border-color: red;  
}
```

*Os campos do tipo input aparecerão com borda vermelha **quando o conteúdo é inválido** (por ex., um campo de e-mail faltando o @)*

Exemplos de Pseudo-Classes

Pseudo-Classe	Descrição	Exemplos
:link	Estilo inicial do link	<i>a:Link</i> {color: blue;}
:hover	Usuário interage com elem. sem ativar Ex.: ponteiro do mouse sobre o elemento	<i>a:hover, button:hover</i> {...}
:active	Estado ativo do elemento Ex.: botão do mouse pres. sobre o elemento	<i>a:active, p:active</i> {...}

Exemplos de Pseudo-Classes

Pseudo-Classe	Descrição	Exemplos
:valid	Campo de form. com conteúdo válido	<i>input:valid</i> {...}
:invalid	Campo com conteúdo inválido	<i>input:invalid</i> {...}
:checked	<i>checkbox</i> , <i>radio</i> ou <i>option</i> selecionado	<i>radio:checked</i> {...}
:focus	Elemento recebendo foco	<i>input:focus</i> {...}
:first-child	Primeiro filho do elemento pai	<i>li:first-child</i> {color: blue;}
:last-child	Último filho do elemento pai	<i>li:last-child</i> {color: blue;}

Pseudo-Elementos

- Permite selecionar **uma parte específica** de um elemento
- Palavra-chave adicionada a um seletor
- Precedida por **::**

```
p::first-line {  
  text-transform: uppercase;  
}
```

A primeira linha de cada parágrafo será apresentada com letras maiúsculas

```
p::selected {  
  color: green;  
  background-color: black;  
}
```

O texto que o usuário selecionar nos parágrafos aparecerá na cor verde com fundo preto

OBS: Para cada pseudo-elemento há uma lista restrita de propriedades que podem ser utilizadas.

Outros Pseudo-Elementos

```
p::after {  
  content: 'exemplo after';  
}
```

Insere o texto 'exemplo after' como um pseudo-elemento **depois do conteúdo** do parágrafo

```
p::before {  
  content: 'exemplo before';  
}
```

Insere o texto 'exemplo before' como um pseudo-elemento **antes do conteúdo** do parágrafo

```
input::placeholder {  
  color: red;  
}
```

Altera a cor dos textos de **placeholder** dos campos *input*

OBS: Os pseudo-elementos `::after` e `::before` **não podem** ser utilizados em elementos sem conteúdo como `` ou `
`

Observação sobre Pseudo-Elementos

- A notação com 4 pontos (::) foi introduzida na CSS3
- :: é a forma atualmente recomendada para *pseudo-elementos*
- Diferencia **pseudo-elementos** de **pseudo-classes**
- Entretanto, na CSS2 utilizava-se apenas dois pontos (:)
 - Navegadores ainda suportam essa forma

Cascade, Especificidade e Herança

- Em caso de regras conflitantes, qual regra será aplicada?
- Problemas comuns:
 - Regras CSS sem efeito
 - Resultado muito diferente do esperado
- Nessas situações é fundamental conhecer como o navegador decide pelas regras
 - Ordem no código
 - Especificidade
 - Herança

Ordem das Regras

- Caso haja declarações CSS repetidas
- Caso sejam do mesmo tipo de seletor
- Prevalecerá a declaração definida por último

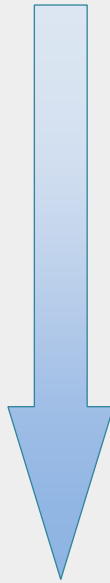
Especificidade

Seletor de Elemento

Seletor de Classe

Seletor de ID

Código Inline



Maior especificidade

Maior peso da declaração CSS

Mais provável que sobrescreva as anteriores

Herança

- Algumas propriedades herdam os valores do elemento pai
 - Ex.: `font-family`, `color` e `text-align`
 - Ex.: um `<p>` dentro de um `<div>` herda a fonte definida para o `<div>`
- Outras propriedades não herdam
 - Ex.: `width`, `margin` e `padding`
 - Ex.: um `<p>` dentro de um `<div>` não herda as margens definidas para o `<div>`
- Definir uma propriedade para o valor `inherit` ativa a herança

Valores Globais para Propriedades

inherit

- O valor da propriedade é herdado do elemento pai

initial

- Retorna a propriedade ao valor inicial

unset

- Retorna a propriedade ao valor *inherit*, caso ela herde naturalmente do pai
- Ou ao valor *initial*, caso contrário

Podem ser utilizados em todas as propriedades

Validação do Código CSS

- Exibição adequada no navegador não é garantia de código correto
 - O navegador pode ocultar erros e inconsistências
- Código fora da especificação
 - Apresentação inconsistente e imprevisível nos navegadores
- Serviço gratuito oferecido pelo W3C para validação de código CSS
 - jigsaw.w3.org/css-validator/

CSS e Cache do Navegador

- Eventualmente o navegador pode armazenar estilos CSS em memória
- Neste caso, mudanças nas regras CSS podem não ter efeito imediato
- Se preciso, tecle **Ctrl-F5** para forçar o navegador a recarregar os estilos

Referências

- www.w3.org/Style/CSS/
- developer.mozilla.org/en-US/docs/Web/CSS
- www.w3.org/Style/CSS/learning
- **HTML and CSS: Design and Build Websites**, Jon Duckett.