

<head>

</head>

<body>

Tópicos Especiais em Desenvolvimento de Software I

Aula 3

HTML5 e CSS3

</body>

<footer>

Instituto de Computação UFMT 2025

TEDS I - 2025/1 - Prof. Jivago
Medeiros Ribeiro

1 /

>

</footer>

<head>

HTML5 e CSS3

</head>

<body>

- Por que geralmente vemos as nomenclaturas HTML5 + CSS3?
 - Na primeira metade da primeira década do século XXI (por volta de 2004) começou a existir um grande descontentamento das empresas que desenvolviam os navegadores e dos *devs* com as especificações HTML e CSS:
 - Não havia especificações (HTML e CSS) voltada para o desenvolvimento de “interfaces ricas”;
 - Tecnologias como o Adobe (inicialmente Macromedia) Flash estavam em ascensão;
 - Navegadores tomavam iniciativas próprias, propondo novas propriedades CSS por exemplo;
 - O que causava uma enorme incompatibilidade entre os navegadores e dificultava o trabalho dos *devs*.
 - Formação de um grupo de trabalho “independente” ao W3C pela Mozilla, Apple e Opera, o WHATWG (Web Hypertext Application Technology Working Group);
 - W3C não seria capaz de aos anseios dos desenvolvedores?

</body>

<footer>

>

</footer>

<head>

Resposta do W3C

</head>

<body>

- O W3C se juntou ao WHATWG para definirem uma nova especificação do HTML (que seria o HTML5);
- Primeiros rascunhos lançados em 2008, versão oficial lançada em 2014.
- Mudanças do HTML5:
 - Descontinuou “tags de estilo”: **, *<big>*, *<center>* etc.;
 - Introduziu “tags semânticas”: *section*, *header*, *footer*, *article*;
 - Introduziu importantes “APIs Nativas” aos navegadores: APIs de Storage (*localStorage*, *sessionStorage*, *Indexed DB* etc), API de Geolocalização, *Canvas*, *WebSocket*, entre outras.

</body>

<footer>

>

</footer>

<head>

Atualmente...

</head>

<body>

- W3C “devolveu” a especificação do HTML para o WHATWG;
- Ou seja, o WHATWG é responsável (desde 2019) pela especificação do HTML: <https://html.spec.whatwg.org/>
- “Não existe mais” HTML5, ou HTML6, o WHATWG mantém o que é chamado de “*Living Standard*”. Uma “especificação viva e continuamente em evolução”.

</body>

<footer>

>

</footer>

<head>

Aonde foi parar o HTML5?

</head>

<body>

- A especificação mantida pelo W3C que estava na versão 5, mais especificamente na versão 5.3 foi retirada pela W3C para que a única especificação seja a *Living Standard* do WHATWG:

HTML 5.3 publication history

Date	Status
28 January 2021	Retired
18 October 2018	Working Draft
9 August 2018	Working Draft
3 July 2018	Working Draft
26 April 2018	Working Draft
15 March 2018	Working Draft
6 February 2018	Working Draft
14 December 2017	First Public Working Draft

<https://www.w3.org/standards/history/html53/>

</body>

<footer>

>

</footer>

<head>

Resumo

</head>

<body>

- “Não existe mais” o HTML5 da W3C;
- A especificação é mantida pelo WHATWG;
- A especificação atual não possui uma versão específica, é uma “living standard”;
- O termo HTML5 comumente é utilizado para se referir a “tecnologias frontend modernas”, apesar de que hoje nem é mais necessária essa distinção.

</body>

<footer>

>

</footer>

<head>

CSS3

</head>

<body>

- A especificação do CSS continua nas mãos do W3C;
- e...
- **Relembrando:**
 - O CSS3 dividiu a especificação em módulos que podem andar de forma independente.
 - Esses módulos, basicamente agrupam seletores e propriedades, ex: Selectors, Color, Background, Box, Media Queries, Namespaces, etc.
 - Ou seja, não é necessária uma “nova versão do CSS” para que a especificação dos módulos seja atualizada.

</body>

<footer>

>

</footer>

<head>

CSS3

</head>

<body>

- O objetivo da especificação do CSS3 foi aumentar a padronização entre os navegadores e atender a demandas dos desenvolvedores web, como por exemplo:
 - Bordas arredondadas, estilos relacionados ao background dos elementos, sombras (shadow), fonts, animações, entre outras.
 - Muitas dessas estilizações, anteriormente ao CSS3 era feito via **hacks** (ou gambiarras mesmo o.O) sendo que quase sempre era necessário um **hack** pra cada browser.

</body>

<footer>

>

</footer>

<head>

Voltando ao HTML(5?)

</head>

<body>

- Desde a O HTML5 foi introduzido um **maior número** de “*tags* semânticas” à linguagem HTML;
- Mas o que seriam “*tags* semânticas” ?
- É aquela *tag* (ou elemento) que possui um “significado” e consequentemente uma utilização explícita que deve ser obedecida, como por exemplo: ***section, header, article.***

</body>

<footer>

>

</footer>

<head>

Tags semânticas

</head>

<body>

- Mas por que “um maior número” ?
 - Porque já haviam *tags* semânticas no HTML, por exemplo: *form*, *img*, *table*, entre outros.
- Então quais *tags* não são semânticas?
 - Elementos como *div* e *span* não possuem nenhuma semântica enquanto elementos como *p*, *b*, *i*, *u* possuem pouca semântica.

</body>

<footer>

>

</footer>

<head>

Tags semânticas

</head>

<body>

- Por que deve-se dar preferência a utilização de *tags* semânticas?
 - Auxiliam na organização e entendimento do documento
 - Facilitam o trabalho de diferentes dispositivos, agentes autônomos e mesmo navegadores no acesso, renderizem e manipulem dos conteúdos e elementos de um documento HTML de acordo com as diferentes necessidades.

</body>

<footer>

>

</footer>

<head>

Tags semânticas

</head>

<body>

- **<article>** : Define um conteúdo que geralmente contém textos e imagens, como por exemplo, uma postagem, uma notícia, um comentário, etc.
<https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/article>
- **<aside>** : Define um conteúdo adjacente / ao lado de outro conteúdo.
<https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/aside>
- **<details> / <summary>** : Esse conjunto descreve um conteúdo com comportamento ocultar / mostrar. O conteúdo da tag summary será o conteúdo sempre visível. Importante: observar a compatibilidade dos navegadores.
<https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/details>
- **<figure> / <figcaption>** : Encapsula imagem e adiciona uma legenda com figcaption
<https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/figure>

</body>

<footer>

>

</footer>

<head>

Tags semânticas

</head>

<body>

- **<footer>** : Conteúdo referente ao rodapé de um documento ou seção.
<https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/footer>
- **<header>** : Conteúdo referente ao cabeçalho de um documento ou seção.
<https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/header>
- **<main>** : Tag que “encapsula” o conteúdo principal de um documento HTML. Importante: observar a compatibilidade dos navegadores.
<https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/main>
- **<mark>** : Tag para destacar (highlight) um trecho de texto.
<https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/mark>
- **<nav>** : Define um conjunto de links de navegação (ou menus).
<https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/nav>

</body>

<footer>

>

</footer>

<head>

Tags semânticas

</head>

<body>

- **<section>** : Define uma seção do documento. Geralmente uma seção encapsula cabeçalhos, menus, artigos, etc.
<https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/section>
- **<time>** : Informações referentes a data e hora, como por exemplo, a data de criação / publicação de um artigo.
<https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/time>
- **<address>** : Utilizado para fornecer informações de contato. Muitas vezes adicionada a um *article* e/ou no *body*
<https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/address>

</body>

<footer>

>

</footer>

<head>

Tags Semânticas

</head>

<body>

- A utilização das *tags* semânticas introduzidas no HTML5 deve ser capaz de diminuir a “***div-dependência***” que “contaminou” muitos desenvolvedores.
- Mas isso não significa que seja “proibido” utilizar *tags* não semânticas como por exemplo *div* e *span*.
 - Essas *tags* muitas vezes são necessárias para criarmos containers ou ainda conseguirmos um posicionamento específico em nosso documento HTML.

</body>

<footer>

>

</footer>

<head>

</head>

<body>

Voltando ao CSS3...

</body>

<footer>

Instituto de Computação UFMT 2025

TEDS I - 2025/1 - Prof. Jivago
Medeiros Ribeiro

16 /

>

</footer>

<head>

Bordas Arredondas

</head>

<body>

- Utilizamos a propriedades border-radius:
- Quanto maior o valor, mais arredondada será, exemplo:

```
border-radius: 4px;  
border-radius: 10px;  
border-radius: 30px;
```

Também é possível utilizar imagens como bordas:
border-image

<https://developer.mozilla.org/en-US/docs/Web/CSS/border-image>

</body>

<footer>

>

</footer>

<head>

Border-image

</head>

<body>

- Sintaxe básica:

border-image: [imagem] [recorte] [tamanho] [outset] [repeat]

- Importante:

- **border-image** não é uma propriedade trivial, dessa forma é recomendado definir de forma separada cada propriedade:

- border-image-source
- border-image-slice
- border-image-repeat
- border-image-width
- border-image-outset

- **border-image-source:** fonte da imagem que será utilizada como borda, pode ser um arquivo físico externo, pode ser base64 ou mesmo gradientes.

</body>

<footer>

>

</footer>

<head>

border-image-slice

</head>

<body>

- Um importante passo ao utilizar imagens como bordar é definir qual(is) parte(s) do "recorte da imagem" será utilizada na criação da borda. Essa definição é feita com a propriedade border-image-slice seguindo esta lógica:
 - O navegador reparte a imagem em 9 partes (conforme imagem a seguir);
 - Dizemos qual ou quais dessas nove partes irá compor a borda;
 - Podemos definir as partes para todas as bordas ou especificamente para o topo, o lado direito, a parte de baixo e o lado esquerdo;
 - Por padrão o meio da imagem (número 9) é retirado, mas se quisermos que ele seja utilizado como fundo usamos a palavra-chave **fill**

Ex: border-image-slice: 12;

Utiliza o "pedaço 1" e "pedaço 2" como borda de todo o elemento.

</body>

<footer>

>

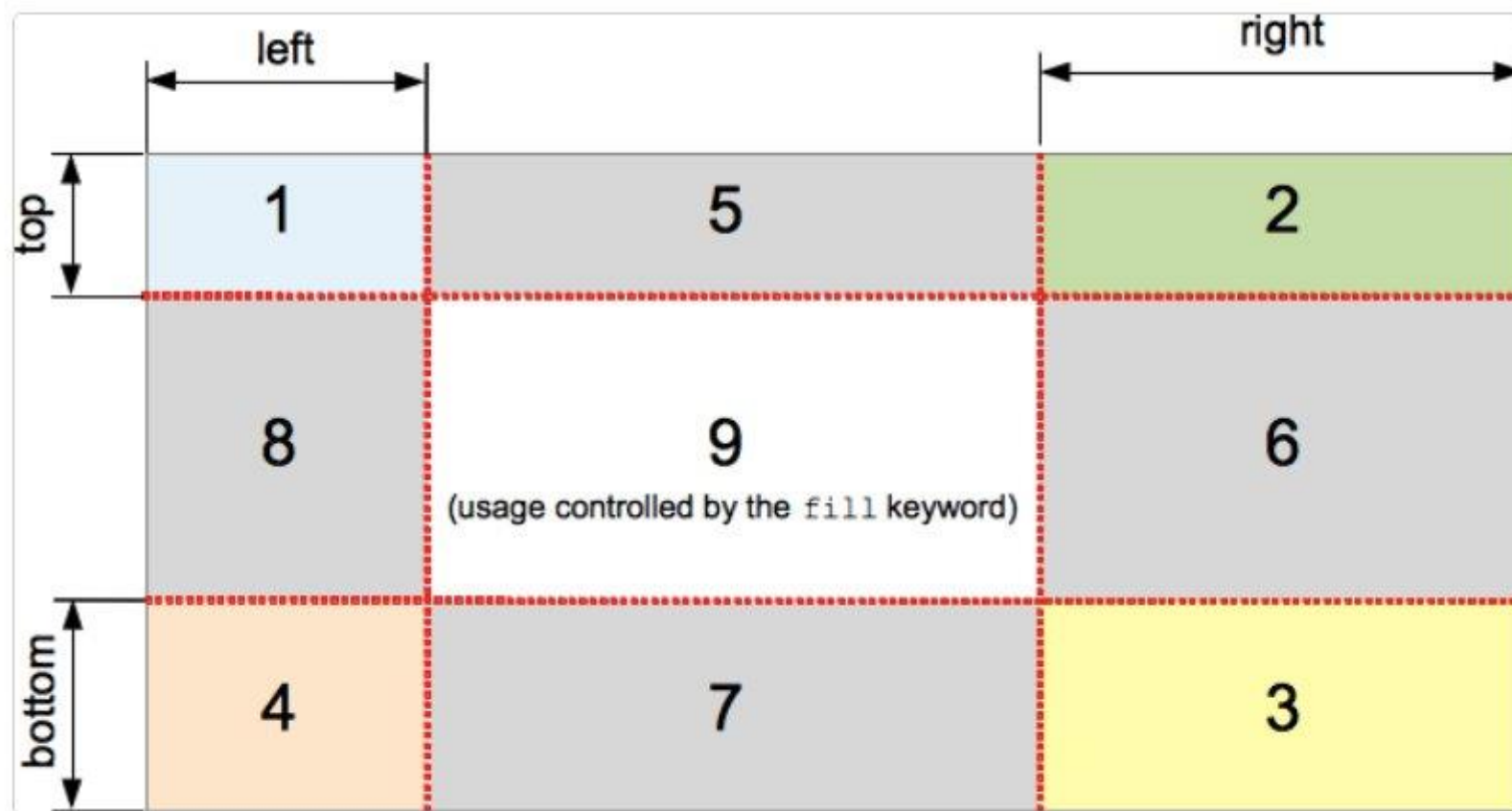
</footer>

<head>

“slice” da imagem no border-image

</head>

<body>



Fonte: <https://developer.mozilla.org/en-US/docs/Web/CSS/border-image-slice>

</body>

<footer>

>

</footer>

<head>

Gradientes

</head>

<body>

- O CSS3 adiciona a possibilidade de **gradientes** lineares e radiais:
- Gradiente Linear:

Pode-se utilizar **palavras chaves** como *to left*, *to right*, *to bottom*, *to up* ou ainda o **ângulo** da direção em graus, ex: **45deg**
Padrão: *to right*



background: *linear-gradient(direção, cor1, cor2, ...);*



Também é possível dizer o ponto de parada da cor, exemplo:
blue 10%

</body>

<footer>

>

</footer>

<head>

Gradientes

</head>

<body>

- O CSS3 adiciona a possibilidade de gradientes lineares e radiais:
- Gradiente Radial:

background: *radial-gradient(tipo,
corinicial, .., corfinal);*

A principal diferença em relação ao gradiente linear é que define-se o tipo (*circle* | *ellipse*) e pode-se definir a posição utilizando *at*

</body>

<footer>

>

</footer>

<head>

Sombras

</head>

- O CSS3 descreve sombras para texto e blocos

.Sintaxe:

text-shadow: *posh posv blur cor;*

box-shadow: *posh posv blur size cor;*

Para sombras internas, utilizamos a palavra-chave ***inset***:

box-shadow: *inset posh posv blur size cor;*

</body>

<footer>

>

</footer>

<head>

Fundo (background)

</head>

- Em relação a *background* (ou fundo dos elementos) as principais adições do CSS3 são:
- Múltiplos *backgrounds*
- E a propriedade ***background-size***

</body>

<footer>

>

</footer>

<head>

Backgrounds Múltiplos

</head>

- Backgrounds múltiplos era um anseio antigo dos desenvolvedores web, que passou a constar na especificação do CSS3

Utilização:

`background-image: url(imagem-1.png), url(imagem-2.jpg);`

`background-repeat: repeat, no-repeat;`

`background-position: left top, bottom right;`

Background 1

,

Background 2

Importante: os fundos definidos primeiro (mais a esquerda) ficam sobre os definidos depois (mais a direita).

</body>

<footer>

>

</footer>

<head>

CSS3

</head>

• Outras propriedades interessantes:

.font-face: permite adicionar fonts de arquivos externos

.transitions: Personaliza/especifica como ocorrerão as alterações em propriedades CSS

.transformation: Transformações 2D e 3D

.animation: permite criar desde animações simples até complexas

</body>

<footer>

>

</footer>

<head>

font-face

</head>

<body>

```
@font-face {  
  font-family: minhaFonte;  
  src: url('fonts/YoungRanger.ttf');  
}
```



Primeiro definimos o “nome” e o “local da fonte. Também é possível definir características como font-weight, font-style, entre outras.

```
p {  
  font-family: minhaFonte;  
}
```



Na sequência basta utilizarmos a fonte que acabamos de definir.

</body>

<footer>

>

</footer>

<head>

transitions

</head>

- A propriedade `transition` nos permite definir características relacionadas as transições em propriedades CSS, exemplo utilizando a pseudo-classe `:hover`:

```
div:hover { padding: 10px; opacity: 0.2; }
```

```
div {  
  transition: opacity 1.5s, padding 0.25s;  
}
```

Definimos como será a transição para a propriedade `opacity` (terá a duração de 1.5 segundos)

Para a propriedade `padding` definimos duração de 0.25 segundos

<footer>

>

</footer>

<head>

transitions

</head>

<body>

- Forma composta de definição da propriedade **transition**:

transition: <property> <duration> <timing-function> <delay>;

timing-function: “Padrão” que será utilizada para definir a transição, exemplo, uma função linear (padrão), onde a transição acontece de forma linearmente. Pode ser definida utilizando palavras chaves (*easing functions*) ou funções como *steps()*, *cubic-bezier()*, *frames()*.

<https://developer.mozilla.org/pt-BR/docs/Web/CSS/transition-timing-function>

<http://easings.net/pt>

delay: Atraso entre a começar entre a interação que dispara a transição e o início da transição em si

Lista de propriedades CSS que podem receber **transition**

https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_animated_properties

</body>

<footer>

>

</footer>

<head>

transformação

</head>

<body>

- A propriedade *transform* nos permite definir transformações (2D e 3D) como rotação, translação, escala, entre outros, exemplo:

```
div:hover { transform: rotate(30deg); }
```

<https://developer.mozilla.org/pt-BR/docs/Web/CSS/transform>

</body>

<footer>

>

</footer>

<head>

animação

</head>

<body>

- O CSS3 introduziu a propriedade ***animation***, utilizada para criar animações em diversas outras propriedades CSS; mesma lista da propriedade ***transition***. As animações em CSS são feitas por meio de quadros (frames) que são definidos dentro da regra *@keyframes*

</body>

<footer>

>

</footer>

<head>

Exemplo

</head>

<body>

```
div#um {  
  border: 1px solid #000;  
  background-color: #060;  
  height: 100px;  
  width: 100px;  
  animation: gira 4s infinite;  
}
```

```
@keyframes gira {  
  0% {  
    transform: rotate(0deg);  
  }  
  100% {  
    transform: rotate(360deg);  
  }  
}
```

Nesse exemplo, o **div#um** irá girar de 0° a 360° em 4s infinitamente.

Podemos definir o comportamento em muitos outros *frames* (10%, 20%, 30%, ..., 90%, 95%, etc). No caso de apenas dois *frames*, podemos utilizar as palavras **from** (0%) e **to** (100%).

</body>

<footer>

>

</footer>

<head>

Explicando

</head>

<body>

Duração da animação. Pode ser em milissegundos (ms)

animation: *gira 4s infinite*;



nome da animação
(utilizado na regra
@keyframe)

Número de iterações: pode
ser um inteiro ou a palavra-
chave infinite.

<https://developer.mozilla.org/pt-BR/docs/Web/CSS/animation>

</body>

<footer>

>

</footer>

<head>

Exemplo

</head>

- No exemplo de animation que fizemos, podemos colocar para "no meio do caminho" ele mudar de cor:

Dica: é possível pausar ou continuar uma animação em CSS utilizando JavaScript e alterando a propriedade `animation-play-state` (`obj.style.animationPlayState`) setando-a como *running* ou *paused*.

```
@keyframes gira {  
  0% {  
    transform: rotate(0deg);  
  }  
  50% {  
    background-color: #999900;  
  }  
  100% {  
    transform: rotate(360deg);  
  }  
}
```

</body>

<footer>

>

</footer>

<head>

CSS3 Seletores

</head>

<body>

- A especificação do CSS3 também trouxe alguns novos seletores, como por exemplo os seletores de elementos irmãos / subsequentes:
 - `div ~ p`: seleciona todos os elementos `p` precedidos (imediatamente ou não) por um elemento `div`
 - `div + p`: seleciona todos os elementos `p` que estão imediatamente após elementos `div`
- E também algumas novas pseudo-classes, a citar: `:checked`, `:disabled`, `:enabled`, `:empty`

https://developer.mozilla.org/en-US/docs/Web/CSS/General_sibling_selectors

https://developer.mozilla.org/en-US/docs/Web/CSS/Adjacent_sibling_selectors

</body>

<footer>

>

</footer>