

I've seen the
FUTURE
It's in my
BROWSER

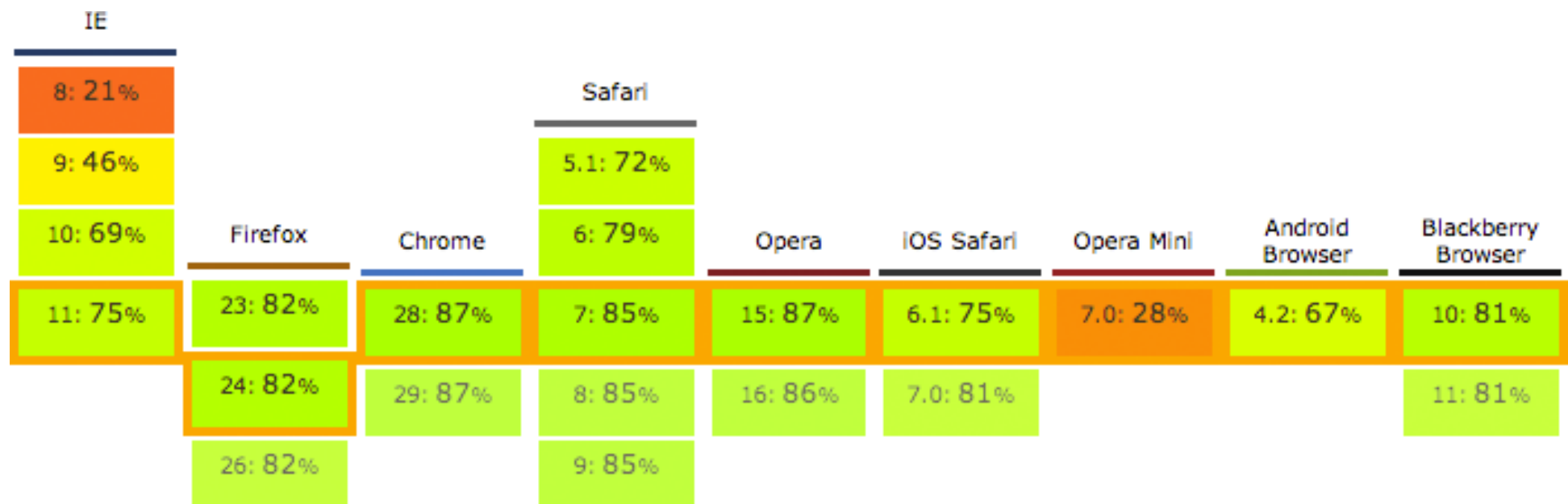


Programa Aula 2

- Visão geral de CSS3;
- Novos selectores;
- Box model;
- Transições;
- Animações;
- Borders;
- Backgrounds;
- Media Queries;

Visão geral de CSS3

Visão geral de CSS3



Visão geral de CSS3

- Terceira versão de Cascading Style Sheets (linguagem que permite descrever a formatação de um documento escrito em markup);
- É retro-compatível;
- Ainda não está finalizada;
- Já pode ser usada hoje!

Revisão rápida selectores

- element selector;
- .class selector;
- element.class selector;

Demo: http://bit.ly/VC_HTML5_Demo06

Selectores de atributos

- {element}[attribute^="value"]
 - Elemento cujo valor do atributo *attribute* começa por *value*;
- {element}[attribute\$="value"]
 - Elemento cujo valor do atributo *attribute* termina em *value*;
- {element}[attribute*="value"]
 - Elemento cujo valor do atributo *attribute* contem *value*;

Demo: http://bit.ly/VC_HTML5_Demo07

Pseudo classe *nth*-*

- {element}:nth-child(n)
 - Elemento *element* filho numero *n* de um *parent*;
- {element}:nth-last-child(n)
 - Elemento *element* filho numero *n* de um *parent*, a contar do último;
- {element}:nth-of-type(n)
 - Elemento *element*, número *n* dos seus irmãos;
- {element}:nth-last-of-type(n)
 - Elemento *element*, número *n* dos seus irmãos, contar do último;

Demo 8

Pseudo-classes *-child e empty

- {element}:last-child
 - Elemento *element* que seja o último filho do seu *parent*;
- {element}:only-child
 - Elemento *element* que não tenha irmãos;
- {element}:empty
 - Elemento *element* vazio

Demo 9

Pseudo-classes *-of-type

- {element}:first-of-type
 - Elemento *element* primeiro do seu tipo;
- {element}:last-of-type
 - Elemento *element* último do seu tipo;
- {element}:only-of-type
 - Elemento *element* único desse tipo dentro do seu *parent*;

Demo 10

Pseudo-classes

target, enabled, disabled e checked

- {element}:target
 - Elemento *element* que é o target de um fragmento (#)
- {element}:enabled
 - Elemento *element* que está *enabled*;
- {element}:disabled
 - Elemento *element* que está disabled;
- {element}:checked
 - Elemento *element* que está checked (radio + checkboxes);

Demo: http://bit.ly/VC_HTML5_Demo11

Pseudo-classe not e combinator ~

- {element}:not(.negative)
 - Elemento *element* que não tem a classe *negative*
- {element1} ~ {element2}
 - Qualquer *element2* precedente de um irmão *element1*;

Demo: http://bit.ly/VC_HTML5_Demo12

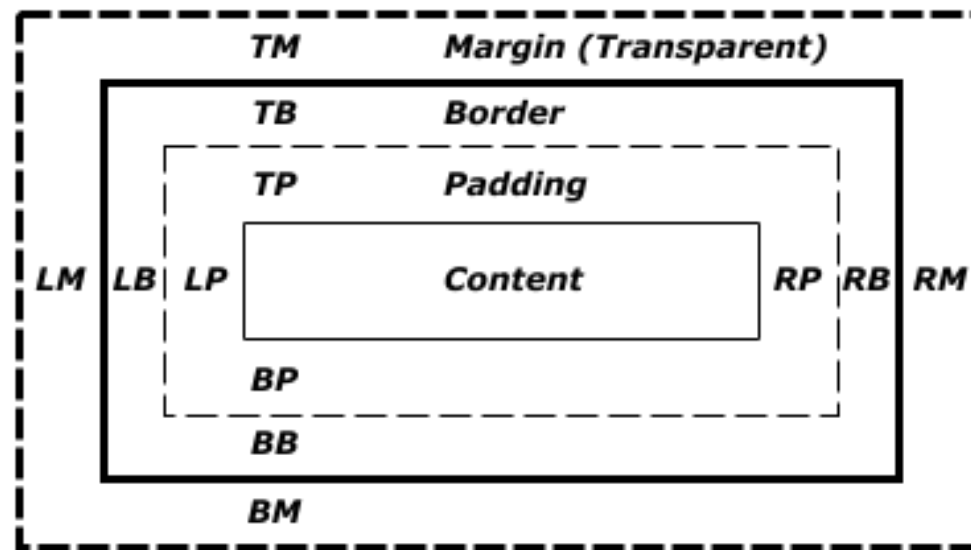
Mais info em: http://bit.ly/VC_HTML5_Selectors

Box model

Box-model - Contexto

- Qualquer elemento em web design é rectangular;
- A largura de um elemento é a soma da largura do seu conteúdo, com os *padding*s laterais e as *borders* laterais;
- Problema: definir de forma inteligente a largura de um elemento sem fazer demasiados cálculos.
- Solução: **box-model**!

Box-model - O que é



- Margin edge
- Border edge
- - - Padding edge
- Content edge

Box-model - Valores possíveis

- content-box
 - Valor por defeito. Ao definir a *width* de um elemento, definimos a largura da zona *content*;
- border-box
 - Ao definir a *width* de um elemento, definimos a largura máxima que a *content*, *padding* e *border* podem acumular;
- inherit
 - Herda o valor definido no *parent*.

Box-model - Suporte

- Todos os browsers modernos
- Internet Explorer 8
 - Sem suporte para max-width, min-width, max-height, min-height.
- Como utilizar

```
selector {  
    -moz-box-sizing: border-box;  
    -webkit-box-sizing: border-box;  
    box-sizing: border-box;  
}
```

Box-model - Utilização ideal

```
*, *:before, *:after {  
    -moz-box-sizing: border-box;  
    -webkit-box-sizing: border-box;  
    box-sizing: border-box;  
}
```

Demo: http://bit.ly/VC_HTML5_Demo13

Transições

Transições - Contexto

- Permitem animar propriedades de CSS, em vez de estarem num determinado estado e mudarem imediatamente de valor;
- Permitem definir:
 - As propriedades a animar - *transition-property*;
 - Quanto tempo devem demorar - *transition-duration*;
 - Tipo de aceleração da transição - *transition-timing-function*;
 - Quando devem animar - *transition-delay*;

Transições - Exemplo

transition: propriedade duração tipo-de-animação atraso

- transition: background-color 2s linear 0s;
- transition: color 3s ease 0.5s;
- transition: width 0.5s ease 2s;

Lista de propriedades animáveis:

- http://bit.ly/VC_HTML5_Transitions

Transições - Exercício

Criar quatro caixas no ecrã com 150x150px, com fundo vermelho e texto branco;

- Caixa 1: animar *background-color* em 2 segundos, com atraso de 0.5s;
- Caixa 2: animar *color* em 3 segundos com atraso de 0.5s;
- Caixa 3: animar *width* em 0.5segundos com atraso de 3s;
- Caixa 4:
 - *width* em 0.5s;
 - *height* em 5s;

Base: http://bit.ly/VC_HTML5_Base14

[Exercício 14](#)

Transformações

Transformações - Contexto

- Permitem:
 - mudar de posição - *translate*;
 - rodar - *rotate*;
 - escalar - *scale*;
 - enviesar - *skew*;

Transformações - Exemplo

transform: tipo-de-transformação(parâmetros)

- `translate(10px, 50px)`
 - 10px para a direita, 50px para baixo
- `rotateX(50deg)`
 - rotação de 50 graus no eixo do X
- `scale(0.5, 2)`
 - escala de 50% no eixo do X e 200% no eixo dos Y
- `skew(30deg):`
 - enviesado 30 graus no eixo do X

Transformações - Exercício

Criar quatro caixas no ecrã com 150x150px, com fundo vermelho e texto branco;

- Caixa 1: Mover 30px para a direita e 40px para cima;
- Caixa 2 : Rodar 180 graus no eixo dos Y;
- Caixa 3 : Aplicar escala de 200% no eixo dos X e 50% no eixo dos Y;
- Caixa 4 : Enviesar 15 graus no eixo dos Y

Base: http://bit.ly/VC_HTML5_Base15 [Exercício 15](#)

Animações

Animações - Contexto

- Semelhante as transições, mas permite definir estados intermédios para o elemento.
- São compostas por keyframes, e tem a seguinte sintaxe

```
@keyframes nomedaanimacao {  
    selector-key-frame { /* propriedades css */ }  
}
```

Animações - keyframes

- Os keyframes podem ser definidos por from & to (definindo o estado inicial e final), ou com percentagem, da seguinte forma:

```
@keyframes nomedaanimacao {  
  0% { color: black; }  
  25% { color: red; }  
  75% { color: blue; }  
  100% { color: white; }  
}
```

A percentagem é em função da duração definida.

Animações - Configuração

- *animation-delay* - quando deve começar;
- *animation-direction* - se deve animar no sentido contrário quando chega ao fim, ou se deve saltar logo para o início;
- *animation-duration* - duração da animação;
- *animation-iteration-count* - quantas vezes deve correr;
- *animation-name* - nome da animação;

Animações - Configuração

- *animation-play-state* - permite pausar a animação;
- *animation-timing-function* - tipo de aceleração da animação;
- *animation-fill-mode* - permite definir propriedades aplicados antes e depois da animação

Animações - Demonstração

- Criar animação com quatro keyframes:
- 25% - alterar a cor de fundo;
- 50% - alterar a cor do texto;
- 75% - alterar a largura da caixa;
- 100% - aplicar rotação;

Base: http://bit.ly/VC_HTML5_Base16

[Exercício 16](#)

Borders

Borders - Contexto

- border-image
 - Permite usar uma imagem como border
- border-radius
 - Permite colocar os cantos arredondados
- Box-shadow
 - Permite criar sombras nos elementos

Borders - border-image

border-image: url largura altura tipo

border-image: url(path.png) 30 40 repeat

- repeat
 - a imagem simplesmente é cortada e repetida a partir do centro
- round
 - a imagem é cortada, e ajusta para ficar sempre bem no início e no fim
- stretch
 - a imagem é simplesmente esticada

Borders - border-radius

border-radius: 50px

- border-top-right-radius
 - radius do canto superior direito
- border-bottom-right-radius
 - radius do canto inferior direito
- border-bottom-left-radius
 - radius do canto inferior esquerdo
- border-top-left-radius
 - radius do canto superior esquerdo

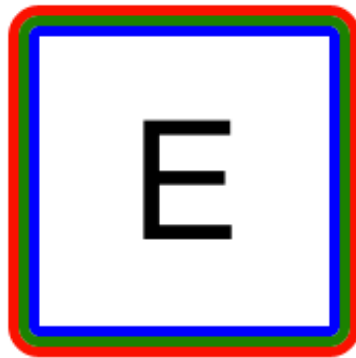
Borders - box-shadow

box-shadow: offset-x offset-y blur-radius spread-radius color type

- Pode-se definir o desvio em xx e yy do centro da sombra;
- blur-radius
 - raio do blur, quanto maior o valor, maior e mais clara fica a sombra. 0 é parecido com border
- spread-radius
 - raio de expansão da sombra
- type
 - outset/inset - sombra para dentro ou para fora

Borders - Exercício

- Definir uma regra CSS que faça uma caixa semelhante à seguinte:



- Dica: um elemento pode ser múltiplas sombras

Base: http://bit.ly/VC_HTML5_Base17

[Exercício 17](#)

Backgrounds

Backgrounds - Revisão

- background-image:
 - caminho para a imagem
- background-position:
 - posição da imagem (ex: top left)
- background-repeat:
 - definição das repetições (ex: repeat-x, repeat-y, no-repeat)
- background-size:
 - tamanho do background (*widthpx heightpx*)

Backgrounds - múltiplos backgrounds

- As propriedades são separadas por vírgulas
- A ordem é importante, porque definem a posição dos backgrounds em relação ao utilizador (profundidade)
 - background-image: url(mais-proximo), (médio), (mais-distante)

Backgrounds - exemplo

- background-image
 - url(path1), url(path2);
- background-position
 - 0 0, bottom right;
- background-repeat
 - repeat-x, repeat-y;
- background-size
 - 50px 50px, 300px 10px;

Backgrounds - Exercício

- Criar um elemento que tem duas imagens de fundo (uma no topo direito, e outra no topo esquerdo).

Base: http://bit.ly/VC_HTML5_Base18 [Exercício 18](#)

Media queries

Media queries - Contexto

- Permitem aplicar regras CSS diferentes mediante diferentes contextos:
 - largura / altura do navegador
 - largura / altura do dispositivo
 - densidade do ecrã
 - orientação do ecrã

Media queries - sintaxe

Num ficheiro CSS incluído de forma normal:

```
@media (validação) {  
    /* CSS a aplicar */  
}
```

Validando a inclusão do ficheiro:

```
<link href="path.css" media="(validação)" />
```

Media queries - validações comuns

- max-width : largura máxima do browser
- min-width : largura mínima do browser
- max-device-width : largura máxima do dispositivo
- min-device-width : largura mínima do dispositivo
- orientation : orientação do dispositivo
- device-pixel-ratio : rácio de resolução

Media queries - Exemplo

- `@media all and (max-width: 960px) { /* ... */ }`
- `@media all and (orientation: landscape) { /* ... */ }`
- `@media all and (device-pixel-ratio: 2) { /* ... */ }`
- `@media all and (max-width: 320px) and (orientation: portrait) { /* ... */ }`

Media queries - Exercício

- Criar quatro caixas no ecrã com 150 por 150 píxeis
- Largura default (superior a 960) o fundo das caixas é cinzento
- Quando a largura é inferior a 960, o fundo passa par azul
- Quando a largura é inferior a 650, o fundo passa para vermelho, e as caixas ficam com 100x100

Base: http://bit.ly/VC_HTML5_Base19 [Exercício 19](#)