



CSS 3

—

Prof. Victor Farias

V 1.3

Referências

w3schools.com



Introdução

CSS 3

- CSS 3 é a especificação mais recente do CSS
- Ele foi dividido em módulos que incorporam a especificação antiga do CSS:
 - Selectors
 - Box Model
 - Backgrounds and Borders
 - Image Values and Replaced Content
 - Text Effects
 - 2D/3D Transformations
 - Animations
 - Multiple Column Layout
 - User Interface

CSS 3



Bordas Arredondadas

Bordas Arredondadas

- Propriedade **border-radius**

- É possível colocar de 1 a 4 valores:

- Um valor: todos os cantos são arredondados igualmente
 - Dois valores: primeiro valor é aplicado no top-left e no bottom-right. O segundo valor é aplicado ao top-right e ao bottom-left
 - Três valores: o primeiro valor é aplicado ao top-left, o segundo é aplicado ao top-right e o bottom-left e o terceiro é aplicado ao bottom-right
 - Quatro valores: o primeiro é aplicado ao top-left, o segundo é aplicado ao top-right, o terceiro é aplicado ao bottom-right e o quarto é aplicado ao bottom-left

```
border-radius: 25px;
```

- **border-radius** é atalho para **border-top-left-radius**, **border-top-right-radius**, **border-bottom-right-radius** e **border-bottom-left-radius**

Sombras

Sombra

- Podemos criar sombras em texto e em elementos bloco
 - **text-shadow**
 - **box-shadow**

Sombra

- Propriedade **text-shadow**
 - Quatros valores:
 - tamanho horizontal
 - tamanho vertical
 - tamanho do blur (opcional)
 - cor (opcional)

```
h1 {  
  
    color: white;  
  
    text-shadow: 2px 2px 4px #000000;  
  
}
```

Ou múltiplos shadows

```
h1 {  
  
    text-shadow: 0 0 3px #FF0000, 0 0 5px #0000FF;  
  
}
```

Sombra

- Propriedade **box-shadow**
 - Adiciona sombra em elementos
 - Mesmos quatro valores de **text-shadow**

```
div {  
  
  color: white;  
  
  box-shadow: 1px 2px 4px rgba(0, 0, 0, .5);  
  
}
```

Transformações



Transformações 2D

- Transformações CSS permitem transladar, rotacionar, dimensionar e inclinar elementos
 - `translate()`
 - `rotate()`
 - `scale()`
 - `skewX()`
 - `skewY()`
 - `matrix()`

Transformações 2D

- Método **translate()** move elemento em relação à sua posição original

```
div {  
  
    -ms-transform: translate(50px, 100px); /* IE 9 */  
  
    -webkit-transform: translate(50px, 100px); /* Safari */  
  
    transform: translate(50px, 100px);  
  
}
```

Transformações 2D

- Método **rotate()** rotaciona elemento segundo um dado angulo

```
div {  
  
    -ms-transform: rotate(20deg); /* IE 9 */  
  
    -webkit-transform: rotate(20deg); /* Safari */  
  
    transform: rotate(20deg);  
  
}
```

Transformações 2D

- Método **scale()** aumenta ou diminui o elemento em relação ao seu tamanho original

```
div {  
  
    -ms-transform: scale(2, 3); /* IE 9 */  
  
    -webkit-transform: scale(2, 3); /* Safari */  
  
    transform: scale(2, 3);  
  
}
```


Transformações 2D

- Métodos **skewX()** e **skewY()** para inclinar elemento no eixo X ou Y

```
div {  
  
  -ms-transform: skewY(20deg); /* IE 9 */  
  
  -webkit-transform: skewY(20deg); /* Safari */  
  
  transform: skewY(20deg);  
  
}
```

Ou os dois juntos (eixo X e Y)

```
div {  
  
  -ms-transform: skew(20deg, 10deg); /* IE 9 */  
  
  -webkit-transform: skew(20deg, 10deg); /* Safari */  
  
  transform: skew(20deg, 10deg);  
  
}
```

Transformações 2D

- Método **matrix()** agrupa todos os métodos
 - `matrix(scaleX(),skewY(),skewX(),scaleY(),translateX(),translateY())`:

```
div {
```

```
-ms-transform: matrix(1, -0.3, 0, 1, 0, 0); /* IE 9 */
```

```
-webkit-transform: matrix(1, -0.3, 0, 1, 0, 0); /* Safari */
```

```
transform: matrix(1, -0.3, 0, 1, 0, 0);
```

```
}
```

Transformações 3D

- Métodos **rotateX()**, **rotateY()** e **rotateZ()** para rotacionar elementos nos eixos X, Y e Z

```
div {  
  
  -webkit-transform: rotateX(150deg); /* Safari */  
  
  transform: rotateX(150deg);  
  
}
```

Transições



Transições

- Transições CSS permitem mudar valores de propriedade suavemente em um tempo determinado
- Deve ser especificado, pelo menos, duas coisas:
 - A propriedade CSS na qual o efeito será adicionado
 - A duração do efeito

```
div {  
  width: 100px;  
  height: 100px;  
  background: red;  
  transition: width 2s;  
}  
div:hover {  
  width: 300px;  
}
```

Ou, para várias propriedades:

```
div {  
  -webkit-transition: width 2s, height 4s; /* Safari */  
  transition: width 2s, height 4s;  
}
```

Transições

- Propriedade **transition-timing-function**
 - Define curva de velocidade do efeito de transição
 - Valores:
 - ease - começo lento, meio rápido e final lento
 - linear - mesma velocidade do começo ao fim
 - ease-in - começo lento
 - ease-out - fim lento
 - ease-in-out - começo e fim lento
 - cubic-bezier(n,n,n,n) - você pode definir os valores para a função cúbica de bezier
- Propriedade **transition-delay**
 - Define tempo para começar o efeito de transição

Transições

- Propriedade **transition**
 - Propriedade atalho
 - 4 valores:
 - Propriedade
 - Tempo de duração
 - Curva da velocidade do efeito
 - Tempo de delay

```
div {  
  
  transition: width 2s linear 1s;  
  
}
```

Obs: se não definir tempo de duração, o padrão é 0s

Animacão



Animação

- Animações CCS permitem trocar um estilo para outro de forma gradual
- Deve-se especificar quais são os keyframes da animação
- Regra **@keyframes** para definir estilos dos keyframes

/* The element to apply the animation to */

```
div {  
  width: 100px;  
  height: 100px;  
  background-color: red;  
  animation-name: example;  
  animation-duration: 4s;  
}
```

/* The animation code */

```
@keyframes example {  
  from {background-color: red;}  
  to {background-color: yellow;}  
}
```

Obs: se animation-duration não for especificado, o valor padrão 0s será usado

Animação

- Também é possível usar a porcentagem de conclusão para definir os keyframes

```
/* The element to apply the animation to */
```

```
div {  
    width: 100px;  
    height: 100px;  
    position: relative;  
    background-color: red;  
    animation-name: example;  
    animation-duration: 4s;  
}
```

```
/* The animation code */
```

```
@keyframes example {  
    0% {background-color: red; left:0px; top:0px;}  
    25% {background-color: yellow; left:200px; top:0px;}  
    50% {background-color: blue; left:200px; top:200px;}  
    75% {background-color: green; left:0px; top:200px;}  
    100% {background-color: red; left:0px; top:0px;}  
}
```

Animação

- Propriedade **animation-iteration-count**
 - Define quantas vezes a animação vai rodar
 - Pode ser infinito

```
div {  
  width: 100px;  
  height: 100px;  
  position: relative;  
  background-color: red;  
  animation-name: example;  
  animation-duration: 4s;  
  animation-iteration-count: 3;  
}  
ou  
div {  
  width: 100px;  
  height: 100px;  
  position: relative;  
  background-color: red;  
  animation-name: example;  
  animation-duration: 4s;  
  animation-iteration-count: infinite;  
}
```

Media Queries

Media Queries

- A regra **@media** permite mudar o estilo de acordo com as características do equipamento que está renderizando a página
- Media queries conseguem checar vários atributos como:
 - altura e largura do viewport
 - orientação do tablet/telefone (landscape ou portrait)
 - resolução

Sintaxe:

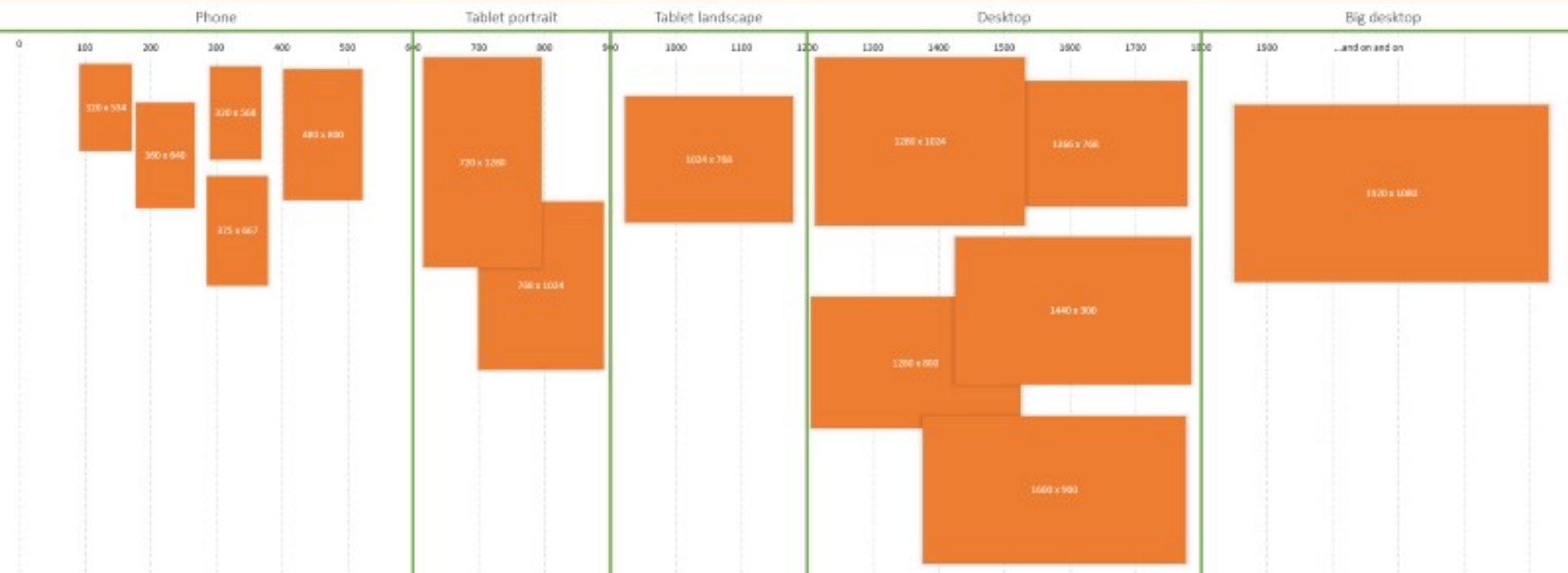
```
@media not|only mediatype and (expressions) {  
    CSS-Code;  
}
```

Exemplo:

```
@media screen and (min-width: 480px) {  
    body {  
        background-color: lightgreen;  
    }  
}
```

Media Queries

- Tipos de media:
 - all - todos
 - print - impressora
 - screen - telas de computador, tablets, celulares, etc
 - speech - screenreaders
- Características da media:
 - max-height
 - min-height
 - max-width
 - min-width
 - min-resolution
 - max-resolution



Um pixel não é um pixel

Box sizing



Box sizing

- Por padrão, a largura completa do elemento é dado por:
 - width + padding + border
- Ou seja, o elemento parece maior do que você definiu
- Com **box-sizing** é possível incluir border e padding no width e height

```
.div1 {  
  width: 300px;  
  height: 100px;  
  border: 1px solid blue;  
  box-sizing: border-box;  
}
```

```
.div2 {  
  width: 300px;  
  height: 100px;  
  padding: 50px;  
  border: 1px solid red;  
  box-sizing: border-box;  
}
```

calc

—

Box sizing

- Como fazer cálculo de medidas?
 - Função calc!

```
#div1 {  
    position: absolute;  
    left: 50px;  
    width: calc(100% - 100px);  
    border: 1px solid black;  
    background-color: yellow;  
    padding: 5px;  
    text-align: center;  
}
```

Exemplos

- Loading
- Botões 3d
- Accordeon
- Image overlay
- Layouts responsivos



Perguntas?

Prof. Victor Farias