

## Módulo V - Elementos de Página

# O professor

Adriano Rivolli

- Desenvolvedor web desde 2002
- Professor (UTFPR)
- Mestrado e doutorado (USP)

# Visão geral do módulo

- CSS
  - Layout
  - Formatação de textos, listas, tabelas
  - Gradientes e sombras
- HTML
  - Elementos de formulário
  - Multimídia
  - Iframes

## Módulo V - Elementos de Página

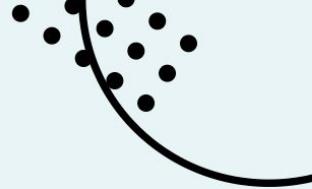
### Regras e Media Queries

# Regras

- Uma regra é uma declaração iniciado com o caractere @ e termina com um ;
- Regras:
  - @charset
  - @import
  - @media

# Charset

- Especifica a codificação de caracteres utilizada na folha de estilo
- Precisa ser declarada no início do arquivo
- Sintaxe
  - @charset "charset";
- Principais charset
  - UTF-8
  - iso-8859-15



# Import

- Utilizado para importar regras de outros arquivos CSS
- Precisa ser declarado no início do arquivo (depois de @charset)
- Sintaxe
  - `@import url("outro-arquivo.css");`

# Media Queries

- Utilizado para restringir o escopo das folhas de estilo
- Com isso regras podem ser aplicadas para diferentes dispositivos
  - Celular
  - Tablets
  - Monitor

# Exemplo

- Inclusão de arquivos

```
<link rel="stylesheet" media="(max-width: 800px)" href="teste.css">
```

- Dentro do HTML

```
<style>  
@media (max-width: 600px) {  
    ...  
}  
</style>
```

# Restrições

- altura e largura da janela
- altura e largura do dispositivo
- orientação (retrato e paisagem para dispositivos móveis)
- resolução

# Sintaxe

```
@media not|only mediatype and (mediafeature and|or|not mediafeature)  
{
```

Código CSS

```
}
```

- **Mediatype:** all | print | screen | speech
- **Media features:**
  - (max/min) height, width, resolution
  - orientation: landscape ou portrait
  - ...

[https://www.w3schools.com/cssref/css3\\_pr\\_mediaquery.asp](https://www.w3schools.com/cssref/css3_pr_mediaquery.asp)

# Exemplo

- <https://codepen.io/rivolli/pen/LYQOmBR>

## Módulo V - Elementos de Página

### Display

# Display

- Define o comportamento com que os elementos HTML serão apresentados
- Distingue os elementos de linha dos elementos de bloco
- É possível modificar o comportamento padrão dos elementos

# Tipos de display

- Nenhum
  - `p { display: none; }`
- Em linha
  - `p { display: inline; }`
- Em bloco
  - `p { display: block; }`
- Em bloco/linha
  - `p { display: inline-block; }`

# Display None

- Quando o valor da propriedade display é igual a none o elemento não ocupa nenhum espaço na interface
- No entanto, é possível apenas não exibir o elemento alterando sua visibilidade
  - `p { visibility: hidden; }`
- Outros valores da propriedade visibility são
  - `visible`
  - `collapse`

# Display Inline-block

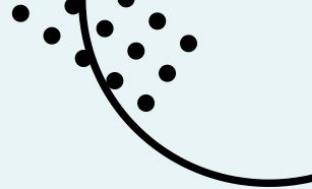
- Mostra um elemento como um elemento de linha
- Permite aplicar as propriedades dos elementos de bloco
  - altura e largura

# Exemplos

- <https://codepen.io/rivolli/pen/gOvoawY>
- <https://codepen.io/rivolli/pen/dydJYdL>

## Módulo V - Elementos de Página

### Layout



# Layouts

- Um site é composto de seções (partes)
  - Cabeçalho
  - Menu de navegação
    - Menu secundário
  - Extras
  - Conteúdo principal
  - Rodapé

# Layout principal



# Conteúdo

1 coluna:



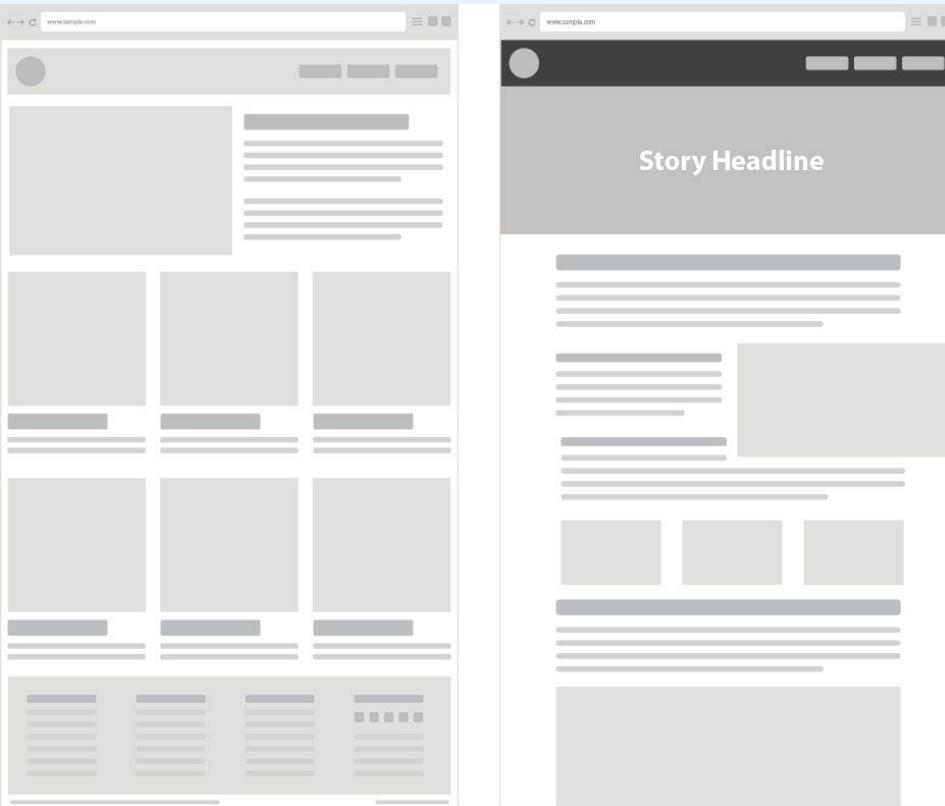
2 colunas:



3 colunas:

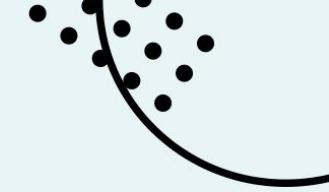


# Outros formatos



Fonte: <https://webflow.com/blog/examples-of-unique-website-layouts>

# Outros formatos



## The Dominion of the Air

The Story of Aerial Navigation

Excerpts from the book by J.M. Bacon

### Chapter listing

- [The Dawn of Aeronautics](#)
- [The Invention of the Balloon](#)
- [The First Balloon Ascent in England](#)
- [The Development of Balloon Philosophy](#)
- [Some Famous Early Voyagers](#)
- [Charles Green and the Nassau Balloon](#)
- [John Wise - The American Aeronaut](#)
- [The Balloon in the Service of Science](#)
- [Some Noteworthy Ascents](#)
- [The Highest Ascent on Record](#)

### The First Balloon Ascent in England

Following his own account, Lunardi's first act on finding himself fairly above the town was to fortify himself with some glasses of wine, and to devour the leg of a chicken. He describes the city as a vast beehive, St. Paul's and other churches standing out prominently; the streets shrunk to lines, and all humanity apparently transfixed and watching him. A little later he is equally struck with the view of the open country, and his ecstasy is pardonable in a novice. The verdant pastures eclipsed the visions of his own lands. The precision of boundaries impressed him with a sense of law and order, and of good administration in the country where he was a sojourner.

By this time he found his balloon, which had been only two-thirds full at starting, to be so distended that he was obliged to unlace the mouth to release the strain. He also found that the condensed moisture round the neck had frozen. These two statements point to his having reached a considerable altitude, which is intelligible enough. It is, however, difficult to believe his further assertion that by the use of his single oar he succeeded in working himself down to within a few hundred feet of the earth. The descent of the balloon must, in point of fact, have been due to a copious outrush of gas at his former altitude. Had his oar really been effective in working the balloon down it would not have needed the discharge of ballast presently spoken of to cause it to reascend. Anyhow, he found himself sufficiently near the earth to land a passenger who was anxious to get out. His car had not been comfortable in the cold upper regions, and now at its urgent appeal was deposited in a corn field, which was the point of first contact with the earth. It was carefully received by a country-woman, who promptly sold it to a gentleman on the other side of the hedge, who had been pursuing the balloon.

This book is available to read in full on the Project Gutenberg website.

Fonte: <https://gridbyexample.com/examples/page-layout/>

## Example layout

- [Link One](#)
- [Link Two](#)
- [Link Three](#)
- [Link Four](#)

This is an example layout using the Skeleton Framework but replacing the Skeleton Grid with Grid Layout, and also using Flexbox for elements.

### Panel 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam sed mauris a urna ultrices ultrices et nec leo. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Duis faucibus fringilla sapien luctus mattis. Ut ut dui in est lacinia mollis.

### Panel 2

Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Sed fermentum ligula neque, non suscipit ipsum elementum non.

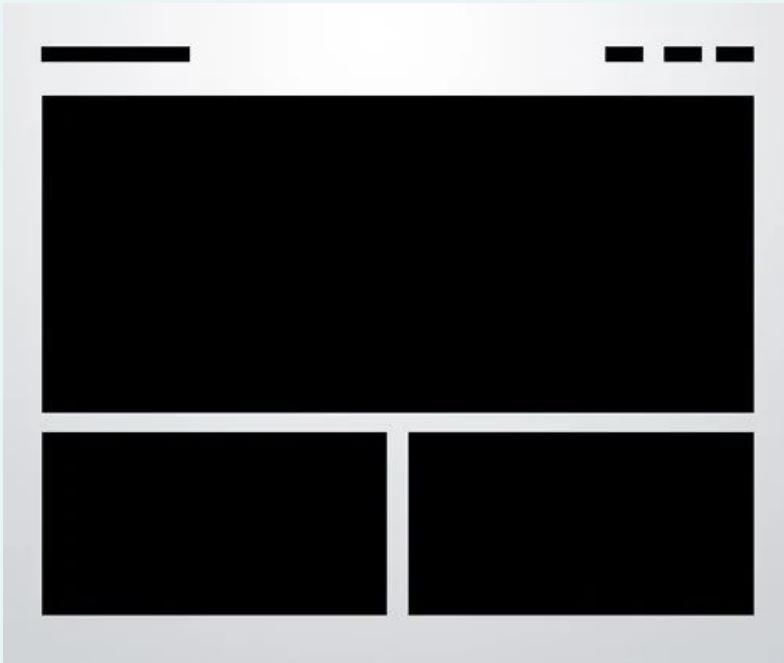
### Panel 3

Aliquam a convallis nulla. Nam mollis mauris urna, et dapibus lacus aliquam vel.

Find out about Skeleton [here](#).



# Outros formatos



Fonte: <https://designshack.net/articles/layouts/10-rock-solid-website-layout-examples/>

# Ferramentas

- Imagens
  - [https://www.4devs.com.br/gerador\\_de\\_imagem\\_lorem\\_pixel](https://www.4devs.com.br/gerador_de_imagem_lorem_pixel)
- Gerador de textos
  - <https://br.lipsum.com/feed/html>

# Como criar layouts?

- Definir as estruturas principais no papel
- Usar IDs para os elementos principais
- Usar bordas, margens, cor de fundo e tamanhos
- Praticar, praticar, praticar...

# Exemplos

- [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_website\\_layout\\_blog](https://www.w3schools.com/css/tryit.asp?filename=trycss_website_layout_blog)
- [https://www.w3schools.com/css/css\\_templates.asp](https://www.w3schools.com/css/css_templates.asp)

# Módulo V - Elementos de Página

## Posicionamento

# Posicionamento

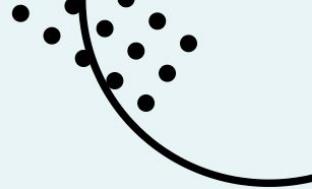
- Os elementos são posicionados por meio das propriedades **top, bottom, left, right**
- Estas propriedades irão funcionar de acordo com o posicionamento definido
- Há 5 tipos de posicionamento no HTML
  - Estático
  - Relativo
  - Fixo
  - Absoluto
  - Sticky

# Posicionamento estático

- Este é o comportamento padrão de posicionamento
- Ignora as propriedades **top**, **bottom**, **left**, **right**
- Elementos estáticos seguem o fluxo de elementos disponíveis na página
- Esta propriedade é utilizada para desfazer um posicionamento atribuído por outra regra a um elemento HTML
- Exemplo:
  - `p { position: static; }`

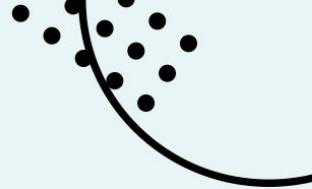
# Posicionamento relativo

- Um elemento com posicionamento relativo é posicionado em relação a sua posição normal
- As propriedades **top**, **bottom**, **left**, **right** não serão ignorados
- Os espaços deixados pelo posicionamento não irão ser preenchidos
- Exemplo:
  - `p { position: relative; }`



# Posicionamento fixo

- Um elemento com posicionamento fixo é fixado em alguma posição da tela
- Sempre irá ocupar o mesmo espaço independentemente do conteúdo
- É necessário definir pelo menos uma das propriedades **top**, **bottom**, **left**, **right**
- Este tipo de posicionamento não irá deixar espaços
- Exemplo:
  - `p { position: fixed; right: 0; bottom: 0; }`



# Posicionamento absoluto

- Um elemento com posicionamento absoluto irá ser posicionado nas coordenadas definidas a partir do seu ancestral
- É necessário definir pelo menos uma das propriedades **top**, **bottom**, **left**, **right**
- Este tipo de posicionamento não irá deixar espaços e pode sobrescrever elementos
- Exemplo:
  - `p { position: absolute; right: 0; bottom: 0; }`

# Posicionamento sticky

- Um elemento com posicionamento sticky é posicionado com base na posição de rolagem do usuário.
- Funciona como um elemento relativo e fixo de acordo com a barra de rolagem
- É necessário definir pelo menos uma das propriedades **top, bottom, left, right**
- Este tipo de posicionamento não irá deixar espaços e pode sobrepor elementos
- Exemplo:
  - `p { position: sticky; top: 0; }`

# Exemplos

- <https://codepen.io/rivolli/pen/RwQxrzL>

## Módulo V - Elementos de Página

### Tamanho da caixa

# Modelo de caixa



# Tamanho da caixa

- Por padrão o tamanho de um elemento é calculado por:
  - Largura = width + padding + border
  - Altura = height + padding + border
- Ao definir uma altura ou largura o elemento terá um valor maior do que o valor definido
- No entanto, existe uma propriedade chamada de **box-sizing** que pode corrigir este comportamento

# Box-sizing

- Os valores que a propriedade aceita são:
  - **content-box** - comportamento padrão
  - **border-box** - Inclui o espaçamento interno e a borda como parte do tamanho
- Exemplo:
  - `p { box-sizing: border-box; }`
  - `* { box-sizing: border-box; }`

# Exemplos

- <https://codepen.io/rivolli/pen/abqEZZY>

## Módulo V - Elementos de Página

### Float

# Flutuando elementos

- A propriedade **float** define como um elemento deve flutuar
- Este recurso é utilizado para posicionar o formatar um conteúdo
- As opções de valores para esta propriedade são:
  - **none** - O elemento não irá flutuar (valor padrão)
  - **left** - O elemento irá flutuar para a esquerda da caixa em que está inserido
  - **right** - O elemento irá flutuar para a direita da caixa em que está inserido

# Cancelando o efeito do float

- Para cancelar o efeito da propriedade float é necessário utilizar a propriedade clear
- Os valores que a propriedade aceita são:
  - **none** - nada acontecerá (valor padrão)
  - **left** - cancela os efeitos de um elemento que está flutuando a esquerda
  - **right** - cancela os efeitos de um elemento que está flutuando a direita
  - **both** - cancela os efeitos de um elemento que flutua

# Usando float para definir layouts

- Definir blocos divs com tamanhos (altura/largura)
- Fazer estes elementos flutuar
- Definir margem e espaçamento necessário

# Exemplos

- <https://codepen.io/rivolli/pen/vYdpGja>
- <https://codepen.io/rivolli/pen/vYdpKBv>

## Módulo V - Elementos de Página

### Flexbox

# Flexbox

- *Flexible box Layout Module*
- Torna mais simples a criação de layouts sem usar float e posicionamento
- Visão geral
  - Flex container
  - Flex Itens

# Flex container

- Para se criar um container utiliza-se a propriedade display
  - `.container { display: flex; }`
- As propriedades de um container são:
  - Direção
  - Quebra
  - Alinhamento

# Flex container (direção)

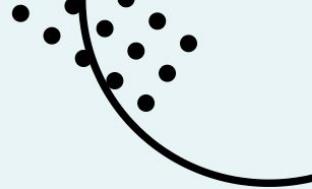
- Qual direção os elementos serão dispostos
- A propriedade utilizada é **flex-direction**:
  - `.container { display: flex; flex-direction: column; }`
- Os valores possíveis são:
  - column
  - column-reverse
  - row
  - row-reverse

# Flex container (quebra)

- Define se os elementos deverão quebrar de linha/coluna ou não
- A propriedade utilizada é **flex-wrap**:
  - `.container { display: flex; flex-wrap: wrap; }`
- Os valores possíveis são:
  - wrap
  - wrap-reverse
  - nowrap

# Flex container (alinhamento horizontal)

- Define como os elementos deverão ser alinhados horizontalmente
- A propriedade utilizada é **justify-content**:
  - `.container { display: flex; justify-content: center; }`
- Os valores possíveis são:
  - center
  - flex-start
  - flex-end
  - space-around
  - space-between



# Flex container (alinhamento vertical)

- Define como os elementos deverão ser alinhados verticalmente
- A propriedade utilizada é **align-items**:
  - `.container { display: flex; align-items: center; }`
- Os valores possíveis são:
  - center
  - flex-start
  - flex-end
  - stretch
  - baseline

# Exemplos

- <https://codepen.io/rivolli/pen/rNJpMOQ>

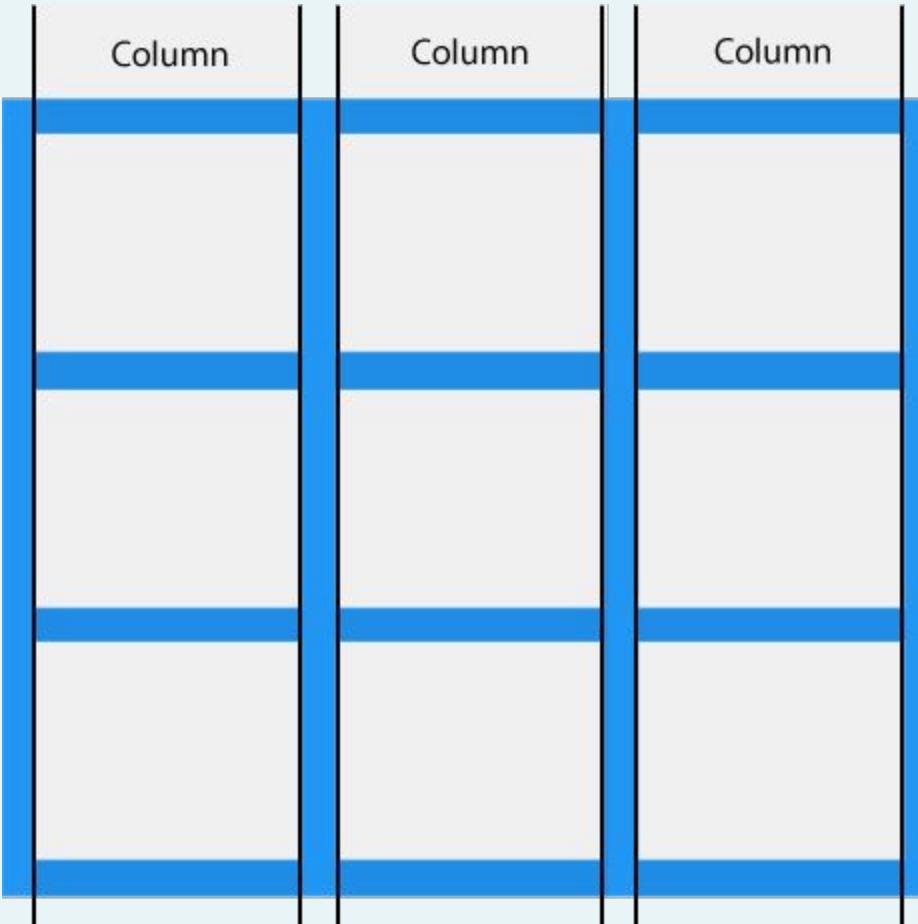
## Módulo V - Elementos de Página

### Grid

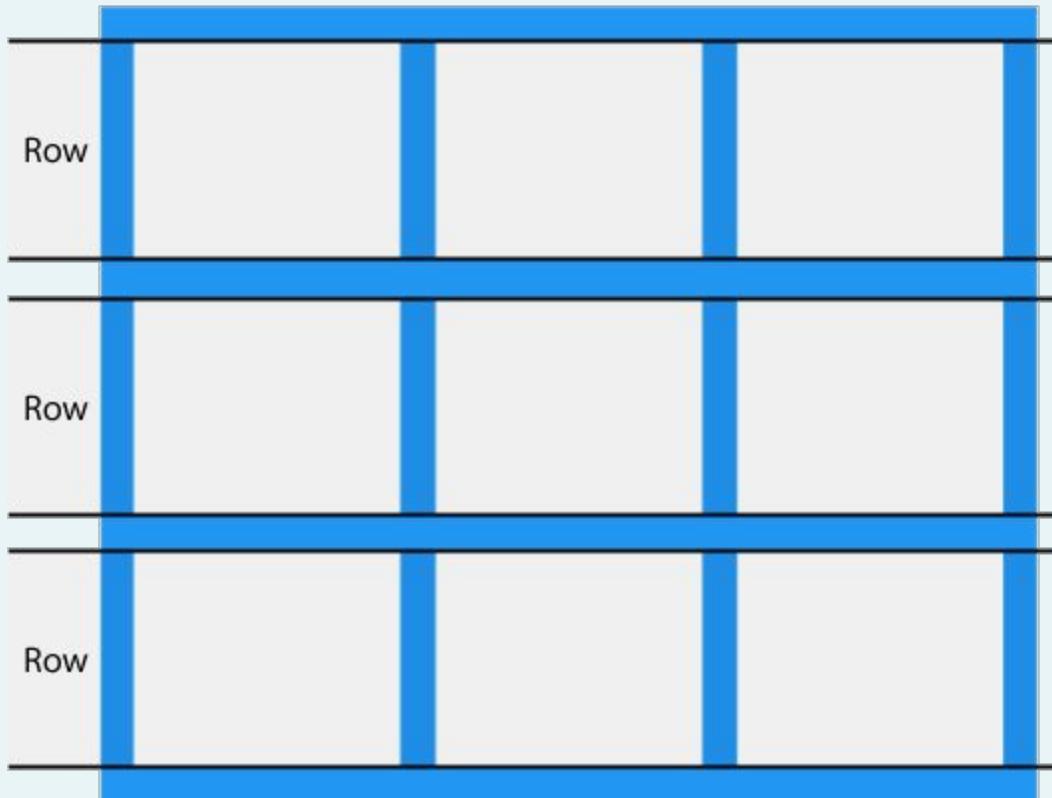
# Grid

- Diferente do Flexbox o sistema de Grid permite a definição de linhas/colunas
- Torna mais simples a criação de layouts sem usar float e posicionamento
- Visão geral
  - Grid container
  - Grid Itens

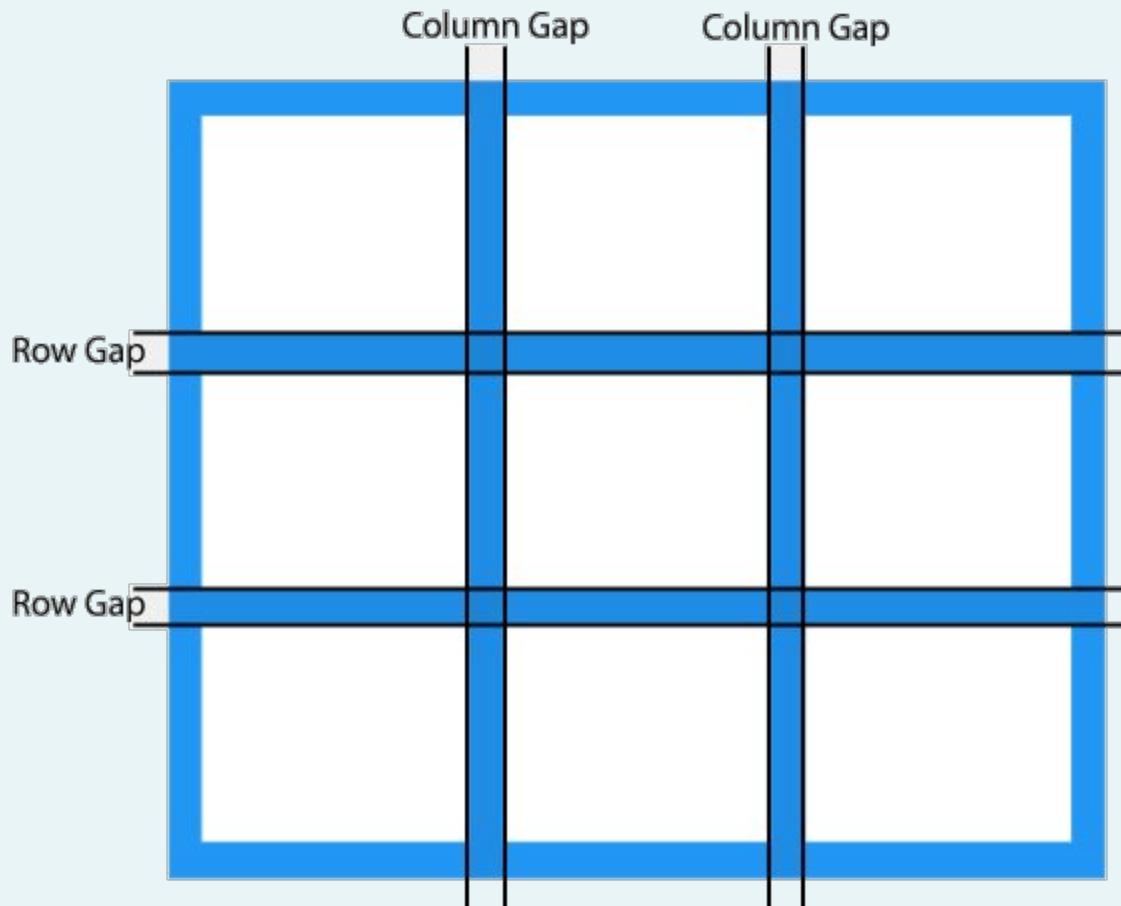
# Colunas



# Linhas



# Espaços



# Grid container

- Definir um Grid container:
  - `.container { display: grid; }`
- Para definir o tamanho da grid utilize:
  - `grid-template-columns: 80px 200px auto 40px;`
  - `grid-template-rows: 60% auto auto;`

# Grid container (alinhamento horizontal)

- Define como os elementos deverão ser alinhados na horizontal
- A propriedade utilizada é **justify-content**:
  - `.container { display: grid; justify-content: center; }`
- Os valores possíveis são:
  - space-evenly
  - space-around
  - space-between
  - center
  - start
  - end

# Grid container (alinhamento vertical)

- Define como os elementos deverão ser alinhados na vertical
- A propriedade utilizada é **align-items**:
  - `.container { display: grid; align-items: center; }`
- Os valores possíveis são:
  - center
  - space-evenly
  - space-around
  - space-between
  - start
  - end

# Expandindo colunas

- Uma célula pode ocupar o espaço de outras células em diferentes colunas
- A propriedade **grid-column** define onde o item vai começar e terminar
- Ocupar 2 colunas (começando na 1 e terminando antes da 3)
  - `.item1 { grid-column: 1 / 3; }`
- Ocupar 2 colunas (começando na 3)
  - `.item2 { grid-column: 3 / span 2; }`

# Expandindo linhas

- Uma célula pode ocupar o espaço de outras células em diferentes linhas
- A propriedade **grid-row** define onde o item vai começar e terminar
- Ocupar 2 linhas (começando na 1 e terminando antes da 3)
  - `.item1 { grid-row: 1 / 3; }`
- Ocupar 2 colunas (começando na 3)
  - `.item2 { grid-row: 3 / span 2; }`

# Exemplos

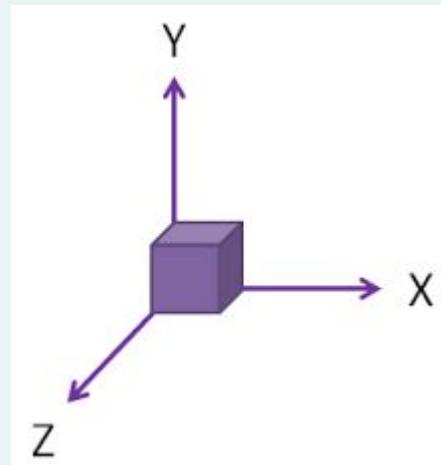
- <https://codepen.io/rivolli/pen/VwQyKKv>

## Módulo V - Elementos de Página

### Pilha dos elementos

# Pilha de elementos

- Diferentes elementos em CSS podem ocupar o mesmo espaço (x, y)
- Por padrão os elementos serão empilhados em um eixo z
- O último elemento declarado será o elemento visível



# Alterando a ordem dos elementos no eixo z

- A propriedade **z-index** permite definir qual a ordem de visualização dos elementos
- Funciona apenas para elementos posicionais (absoluto, relativo, fixo e sticky)
- Quanto maior o valor mais a frente ele estará
- Exemplo:
  - `p { z-index: -1; }`

# Overflow

- É possível controlar o comportamento dos elementos de blocos quando o conteúdo é maior do que os limites do elemento
- Isso é realizado a partir da propriedade **overflow**
- Os valores possíveis para esta propriedade são:
  - visible
  - hidden
  - scroll
  - auto

# Exemplos

- <https://codepen.io/rivolli/pen/eYVybpi>

## Módulo V - Elementos de Página

### Fontes

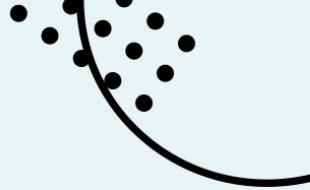
# Fontes

- Uma característica visual importante para um site
- As fontes podem criar uma identidade com a marca
- Podem tornar a leitura do texto uma tarefa mais fácil ou difícil

# Família de fontes

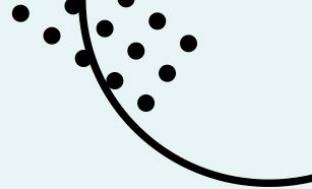
- Existem 5 famílias genéricas de fontes
  - Serif (impresão)
  - **Sans-serif (tela)**
  - Monospace
  - Cursive
  - Fantasy





# Fontes

Família da Fonte	Nome das fontes
Serif	Times New Roman Georgia Garamond
Sans-serif	Arial Verdana Helvetica
Monospace	Courier New Lucida Console Monaco
Cursive	<i>Brush Script MT</i> <i>Lucida Handwriting</i>
Fantasy	❖□□□*□□●♦▼* ☆*□□◆▲



# Fontes

- As fontes devem estar presentes no cliente
- A propriedade que define o nome da fonte é **font-family**
- Diferentes nomes de fontes devem ser especificados
- Exemplo:
  - `p { font-family: Arial, Helvetica, sans-serif; }`

# Fontes seguras

- Arial (sans-serif)
  - Verdana (sans-serif)
  - Helvetica (sans-serif)
  - Tahoma (sans-serif)
  - “Trebuchet MS” (sans-serif)
  - “Times New Roman” (serif)
  - Georgia (serif)
  - Garamond (serif)
  - Courier New (monospace)
- Brush Script MT (cursive)

Fonte: [https://www.w3schools.com/css/css\\_font\\_websafe.asp](https://www.w3schools.com/css/css_font_websafe.asp)

# Estilo itálico

- Para definir o estilo itálico da fonte, utilize a propriedade **font-style**
- Os valores permitidos são:
  - normal / italic / oblique
- Exemplo
  - `p { font-style: italic; }`

# Estilo negrito

- Para definir o estilo negrito da fonte, utilize a propriedade **font-weight**
- Os valores permitido para **font-weight** são:
  - normal
  - bold
  - bolder
  - lighter
  - Número entre 100 e 900
- Exemplo
  - `p { font-weight: bold; }`

100 – Thin

200 – Extra Light (Ultra Light)

300 – Light

400 – Normal

500 – Medium

600 – Semi Bold (Demi Bold)

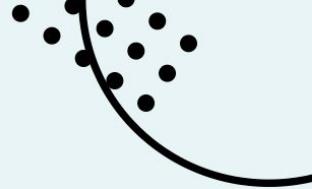
700 – Bold

800 – Extra Bold (Ultra Bold)

900 – Black (Heavy)

# Estilo caixa alta

- Para definir como a transformação de caixa alta utilize **font-variant**
- Os valores permitidos para esta propriedade são:
  - normal
  - small-caps
- Exemplo:
  - `p { font-variant: small-caps; }`



# Estilo condensado/expandido

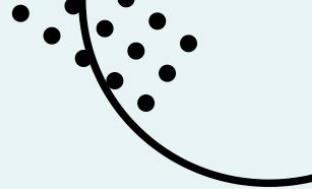
- Para definir o texto condensado ou expandido use **font-stretch**
- Os valores permitidos para esta propriedade são:
  - ultra-condensed | extra-condensed | condensed | semi-condensed | normal | semi-expanded | expanded | extra-expanded | ultra-expanded
- Exemplo:
  - `p { font-stretch: expanded; }`

# Tamanho

- Para definir o tamanho da fonte utilize **font-size**
- Os valores permitidos para esta propriedade são um valor numérico e uma unidade de medida
  - `p { font-size: 1.5em; }`

# Google Fonts

- <https://fonts.googleapis.com/>
- Para incluir uma fonte utilize
  - @import url(<https://fonts.googleapis.com/css?family=NomeDaFonte>);
- Utilize a fonte normalmente



# Pareamento de fontes

- Georgia e Verdana
- Helvetica e Garamond
- Google fonts:
  - Merriweather e “Open Sans”
  - Ubuntu e Lora
  - “Abril Fatface” e Poppins
  - Cinzel e “Fauna One”
  - “Fjalla One” e “Libre Baskerville”

[https://www.w3schools.com/css/css\\_font\\_pairings.asp](https://www.w3schools.com/css/css_font_pairings.asp)

# Web fontes

- Possível disponibilizar uma fonte própria
- O uso de web-fontes permite ao desenvolvedor “criar” sua própria fonte
- Isso ocorre pelo uso da regra **@font-face**
- Formatos de fonte
  - TrueType Fonts (TTF)
  - OpenType Fonts (OTF)
  - The Web Open Font Format (WOFF e WOFF 2.0)
  - SVG Fonts/Shapes
  - Embedded OpenType Fonts (EOT)

# @font-face

- Utilizado para definir o nome de uma fonte
- Sintaxe

```
@font-face {  
    font-family: NomeDaMinhaFonte;  
    src: url(caminho-da-fonte.woff);  
}
```

- Exemplo de uso

```
div { font-family: NomeDaMinhaFonte; }
```

# Exemplo

- <https://codepen.io/rivolli/pen/JjpMxpN>

# Módulo V - Elementos de Página

## Textos

# Textos

- Além da fonte, existem algumas propriedades para formação de textos
- São elas:
  - Cor
  - Alinhamento
  - Decoração
  - Transformação
  - Espaçamento

# Alinhamento horizontal

- O alinhamento horizontal é definido pela propriedade **text-align**
- Os possíveis valores para esta propriedade são:
  - center
  - left
  - right
  - justify
- A propriedade **text-align-last** é aplicado para a última linha de um bloco de texto

# Alinhamento vertical

- O alinhamento vertical é definido pela propriedade **vertical-align**
- Os possíveis valores para esta propriedade são:
  - baseline
  - text-top
  - text-bottom
  - sub
  - super

# Alinhamento vertical

- O alinhamento vertical é definido pela propriedade **vertical-align**
- Os possíveis valores para esta propriedade são:
  - baseline
  - text-top
  - text-bottom
  - sub
  - super

# Decoração

- A decoração trata de um traço que acompanha o texto
- As seguintes propriedades podem ser utilizadas
  - **text-decoration-line** - Onde a linha irá aparecer
  - **text-decoration-color** - Cor da linha
  - **text-decoration-style** - Estilo da linha
  - **text-decoration-thickness** - Espessura da linha
  - **text-decoration** - Todas as propriedades em uma única declaração

# Decoração da linha

- **text-decoration-line** - Onde a linha irá aparecer
  - overline | underline | line-through | none
- **text-decoration-style** - Estilo da linha
  - solid | double | dotted | dashed | wavy

# Transformação

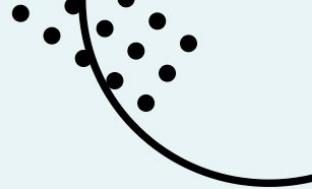
- Define como o texto é apresentado
- A propriedade utilizada para modificar o texto é **text-transform**
- Os seguintes valores podem ser utilizadas
  - **none** - A forma como o texto foi digitado
  - **uppercase** - Texto em maiúsculo
  - **lowercase** - Texto em minúsculo
  - **capitalize** - Primeira letra em maiúsculo

*Lembrete:*

*Diferente de **font-variant***

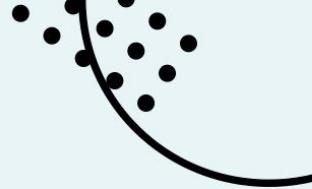
# Espaçamento

- Existem diferentes propriedades para alterar o espaçamento do texto
- As seguintes propriedades podem ser utilizadas
  - Parágrafo
  - Altura da linha
  - Espaçamento das letras
  - Espaçamento entre as palavras
  - Espaço em branco



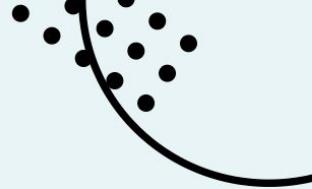
# Parágrafo

- A propriedade que define o tamanho do recuo do parágrafo é **text-indent**
- O valor deve ser uma tamanho (positiva ou negativa)
  - `p { text-indent: 3em; }`
  - `p { text-indent: -2em; }`



# Altura da linha

- A propriedade que define a altura da linha é **line-height**
- O valor deve ser um tamanho
  - `p { line-height: 2em; }`
  - `p { line-height: 150%; }`



# Espaçamento

- A propriedade que define o espaçamento entre as letras é **letter-spacing**
- A propriedade que define o espaçamento entre as palavras é **word-spacing**
- O valor destas propriedades é um tamanho
  - `p { letter-spacing: 1px; word-spacing: 1em; }`

# Espaço em branco

- Múltiplos espaços em branco são ignorados em HTML
- É possível modificar este comportamento por meio da propriedade **white-space**
- Os possíveis valores para esta propriedade são:
  - nowrap
  - pre
  - pre-line
  - normal

# Texto overflow

- Permite alterar a forma como o conteúdo de texto será tratado quando o mesmo ultrapassar os limites de tamanho
- A propriedade **text-overflow** define como o excesso de texto será tratado
- Os possíveis valores para esta propriedade são:
  - clip | ellipsis | texto
- Para seu funcionamento é necessário definir as seguintes propriedades
  - white-space: nowrap;
  - overflow: hidden;

# Texto overflow

- Permite alterar a forma como o conteúdo de texto será tratado quando o mesmo ultrapassar os limites de tamanho
- A propriedade **text-overflow** define como o excesso de texto será tratado
- Os possíveis valores para esta propriedade são:
  - clip | ellipsis | texto
- Para seu funcionamento é necessário definir as seguintes propriedades
  - white-space: nowrap;
  - overflow: hidden;

# Outras propriedades

- text-justify
- word-wrap
- writing-mode

# Exemplo

- <https://codepen.io/rivolli/pen/ExQoqyr>

## Módulo V - Elementos de Página

### Formatação de listas

# Listas

- Usando CSS é possível definir o estilo da lista e customizar a apresentação das mesmas
- Em HTML há 2 tipos principais de listas
  - Ordenadas <ol>
  - Não ordenadas <ul>
- A definição do estilo da lista é definido pela propriedade **list-style-type**

# Estilo da lista

## Não ordenada

- disc
- circle
- none
- square

## Ordenada

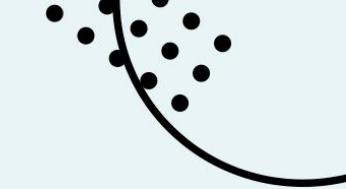
- decimal-leading-zero (01, 02, 03, etc.)
- lower-alpha (a, b, c, d, e, etc.)
- lower-latin (a, b, c, d, e, etc.)
- lower-roman (i, ii, iii, iv, v, etc.)
- upper-alpha (A, B, C, D, E, etc.)
- upper-latin (A, B, C, D, E, etc.)
- upper-roman (I, II, III, IV, V, etc.)

*Lista completa: [https://www.w3schools.com/cssref/pr\\_list-style-type.asp](https://www.w3schools.com/cssref/pr_list-style-type.asp)*

# Imagen como marcadores

- É possível definir uma imagem como marcador utilizando a propriedade **list-style-image**
- Exemplo
  - `ul { list-style-image: url('imagem.gif'); }`

# Posição da lista

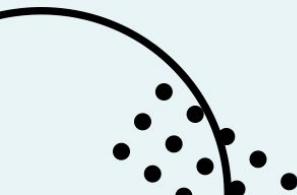


`ul { list-style-position: outside; }`

- Coffee - A brewed drink prepared from roasted coffee beans...
- Tea
- Coca-cola

`ul { list-style-position: inside; }`

- Coffee - A brewed drink prepared from roasted coffee beans...
- Tea
- Coca-cola



# Exemplo

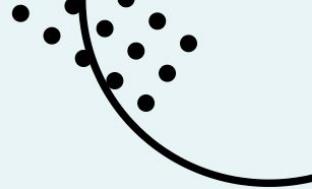
- <https://codepen.io/rivolli/pen/ExQoqmM>

## Módulo V - Elementos de Página

### Tabelas

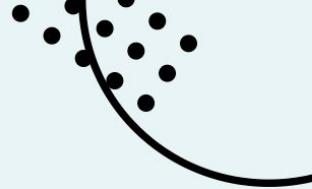
# Tabela

- As tabelas possuem diferentes elementos que podem ser formatados
  - table / tr / td / th
- As principais propriedades utilizadas são:
  - Borda
  - Cor de fundo
  - Tamanho
  - Alinhamento
  - Espaçamento



# Colapsando as bordas

- É possível unir as bordas entre as células de uma tabela
- A propriedade utilizada é **border-collapse** e o valor deve ser **collapse**
- Exemplo
  - `table { border-collapse: collapse; }`



# Linhas alteradas

- Para definir linhas alternadas é necessário utilizar o seletor
  - `tr:nth-child(even){background-color: #f2f2f2;}`
  - `tr:nth-child(odd){background-color: #e2f2f2;}`

# Exemplo

- [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_table\\_fancy](https://www.w3schools.com/css/tryit.asp?filename=trycss_table_fancy)
- <https://codepen.io/rivolli/pen/QWQaemm>

## Módulo V - Elementos de Página

### Bordas avançadas

# Bordas arredondadas

- A propriedade **border-radius** permite definir cantos arredondados para as bordas
- O valor que a propriedade recebe é um tamanho numérico
- É possível definir valores para os diferentes cantos
  - border-top-left-radius
  - border-top-right-radius
  - border-bottom-right-radius
  - border-bottom-left-radius

# Múltiplos valores

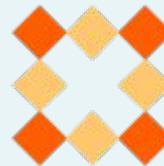
- 4 valores
  - topo-esquerda topo-direita base-direita base-esquerda
- 3 valores
  - topo-esquerda topo-direita/base-esquerda base-direita
- 2 valores
  - topo-esquerda/base-direita topo-direita/base-esquerda
- 1 valor
  - Para todos os lados

# Bordas elípticas

- É possível definir para cada lado de um canto
- Exemplo
  - border-radius: 15px / 50px;

# Borda com imagens

- A propriedade border-image permite definir uma imagem para a borda
- A imagem será segmentada em 9 partes
- Exemplo
  - [https://www.w3schools.com/css/css3\\_border\\_images.asp](https://www.w3schools.com/css/css3_border_images.asp)



# Exemplo

- <https://codepen.io/rivolli/pen/bGLaXmL>

## Módulo V - Elementos de Página

### Gradientes e sombras

# Gradientes

- São transições entre 2 ou mais cores
- Tipos:
  - Linear
  - Radial
  - Cônico

# Gradiente linear

- Sintaxe:
  - `background-image: linear-gradient(direção, cor1, cor2, ..., corN)`
- Direção
  - `to left / to right / to bottom / to top`
  - `to bottom right / to top left`
-

# Gradiente radial

- Sintaxe:
  - `background-image: radial-gradient(shape size, cor1, cor2, ..., corN)`
- Shape
  - circle / ellipse
- Size
  - closest-side / closest-corner / farthest-side / farthest-corner

# Gradiente cônico

- Sintaxe:
  - `background-image: conic-gradient(cor1, cor2, ..., corN)`

# Sombras

- Textos
  - text-shadow
- Caixas
  - box-shadow

# Sombras em textos

- Sintaxe
  - `text-shadow: horizontal vertical blur cor;`

- Exemplos
  - `text-shadow: 2px 2px red;`

**Text shadow effect!**

- `text-shadow: 2px 2px 5px red;`

**Text shadow effect!**

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

**Text shadow effect!**

# Sombras em caixas

- Sintaxe
  - `box-shadow: horizontal vertical blur tamanho cor;`
- Exemplos
  - `box-shadow: 10px 10px lightblue;`
  - `box-shadow: 10px 10px 5px lightblue;`
  - `box-shadow: 10px 10px 5px 12px lightblue;`

A `<div>` element with a lightblue box-shadow

A `<div>` element with a 5px blurred, lightblue box-shadow

A `<div>` element with a blurred, lightblue box-shadow, with a spread radius of 12px

# Exemplo

- <https://codepen.io/rivolli/pen/GRQQBeK>

## Módulo V - Elementos de Página

### Formulário em HTML

# Formulários

- São utilizados para receber dados do usuário
- Possui diversos elementos de entrada de dados
- Tag principal <form>
  - action = URL
  - method = (get ou post)
  - autocomplete = (on ou off)
- Exemplo

```
<form action="#" method="post">  
...  
</form>
```

# Entrada de texto

- Sintaxe:
  - <input type="text" name="nome" id="nome" placeholder="Nome completo">
- Principais atributos
  - **disabled** - Desabilitado
  - **maxlength** - Tamanho máximo
  - **readonly** - Somente leitura
  - **required** - Obrigatório
  - **size** - Tamanho
  - **value** - Valor

# Rótulos

- O elemento <label> é utilizado para definir um rótulo para uma entrada de texto
- Requer o atributo for cujo valor é o id de algum elemento
- Exemplo

```
<label for="nome">Nome</label>
```

```
<input type="text" id="nome" name="nome">
```

# Tipos de entrada

- Cor
- Data
- Email
- Número
- Senha
- Intervalo
- Busca
- Hora

URL

# Cor

- Utilizado para selecionar uma cor
- Sintaxe:
  - <input type="color" id="icor" name="cor" value="#e66465">

# Data

- Utilizado para receber uma data
- Sintaxe:

```
<input type="date" id="dtini" name="dtinicio" value="2018-07-22"  
min="2018-01-01" max="2018-12-31">
```

# Email

- Utilizado para receber um e-mail
- Sintaxe:

```
<input type="email" id="email" size="30" required>
```

# Número

- Utilizado para receber um valor numérico
- Sintaxe:

```
<input type="number" id="idade" name="idade"  
min="10" max="100" step="5">
```

# Senha

- Utilizado para receber uma senha
- Sintaxe:

```
<input type="password" id="pass" name="password"  
minlength="8" required>
```

# Intervalo

- Utilizado para selecionar um valor numérico em um intervalo
- Sintaxe:

```
<input type="range" id="volume" name="volume"  
min="0" max="11" value="7">
```

# Busca

- Utilizado para apresentar uma caixa de busca
- Sintaxe:

```
<input type="search" id="site-search" name="q">
```

# Hora

- Utilizado para selecionar uma hora (tempo)
- Sintaxe:

```
<input type="time" id="appt" name="appt"  
min="09:00" max="23:00" required>
```

# URL

- Utilizado para receber uma URL
- Sintaxe:

```
<input type="url" name="url" id="url"  
placeholder="http://www.site.com"  
size="30">
```

# Exemplo

- <https://codepen.io/rivolli/pen/bGLLxBq>

## Módulo V - Elementos de Página

### Caixa de texto e botões

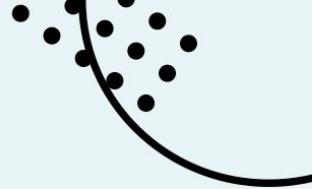
# Caixa de texto

- Utilizado para um texto longo
- Exemplo

```
<textarea id="texto" name="texto" rows="10" cols="60">
```

Era uma vez...

```
</textarea>
```



# Datalist

- Utilizado para permitir uma entrada de texto a partir de uma lista de valores
- Sintaxe:

```
<input type="text" list="sabores" id="isabores" name="sabores">  
  
<datalist id="sabores">  
  
    <option value="Chocolate">  
  
    <option value="Coco">  
  
    <option value="Menta">  
  
    <option value="Morango">  
  
    <option value="Baunilha">  
  
</datalist>
```

# Botões

- Utilizado para criar um botão
- Sintaxe:

```
<button type="button">Adicionar</button>
```

- Tipos:
  - submit - Envia os dados do formulário
  - reset - Volta os dados na configuração original
  - button - Botão para uso geral

# Botões com input

- Utilizado para criar um botão
- Sintaxe:

```
<input type="button" value="Adicionar">  
<input type="submit" value="Enviar">  
<input type="reset" value="Limpar formulário">
```

# Exemplo

- <https://codepen.io/rivolli/pen/RwQQYQr>

## Módulo V - Elementos de Página

### Caixas de seleção

# Grupo de elementos

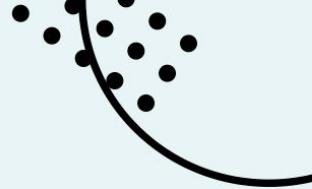
- Para reunir um grupo de elementos agrupados utilizamos o elemento **<fieldset>**
- O título do grupo é definido pelo elemento **<legend>**
- Exemplo

```
<fieldset>
```

```
    <legend>Grupo de elementos</legend>
```

```
    ...
```

```
    </fieldset>
```



# Caixa de seleção múltiplas

- Utilizado para permitir que o usuário selecione múltiplas opções
- Exemplo

```
<input type="checkbox" id="item1" name="item1" value="1">  
<label for="item1">Axé</label>
```

- O atributo **checked** seleciona uma caixa

# Caixa de seleção única

- Utilizado para permitir que o usuário selecione uma única opção
- Sintaxe:

```
<input type="radio" name="opcao" value="1" id="op1">
<label for ="op1">Opção 1</label>

<input type="radio" name="opcao" value="2" id="op2">
<label for ="op2">Opção 2</label>

<input type="radio" name="opcao" value="3" id="op3">
<label for ="op3">Opção 3</label>
```
- É necessário utilizar o mesmo nome para formar um grupo de opções
- O atributo **checked** seleciona uma caixa

# Exemplo

- <https://codepen.io/rivolli/pen/xxYYamv>

## Módulo V - Elementos de Página

### Combobox

# Caixa de seleção

- Utilizado para oferecer uma lista de escolhas para o usuário
- Normalmente utilizado há muitas opções para ser apresentado
- Sintaxe

```
<select name="opcao" id="opcoes">  
    <option value="1">Primeiro item</option>  
    <option value="2" selected>Segundo item</option>  
    <option value="3">Terceiro item</option>  
</select>
```

# Caixa de seleção com grupos

- Permite criar grupos de opções
- O elemento utilizado é o <optgroup>
- Sintaxe

```
<optgroup label="Grupo A">  
    <option value="1">Primeiro item</option>  
    <option value="2">Segundo item</option>  
    <option value="3">Terceiro item</option>  
    <option value="4" disabled>Quarto item</option>  
</optgroup>
```

# Opções

- O atributo **multiple** permite que múltiplas opções possam ser selecionadas
- O atributo **size** permite exibir um número maior de opções ao mesmo tempo

# Exemplo

- <https://codepen.io/rivolli/pen/wvyyYvy>