

HTML5, CSS3 E JAVASCRIPT

JEMERSON FERNANDO MAIA

**PÓS-GRADUAÇÃO EM DESENVOLVIMENTO DE
APLICAÇÕES WEB E MOBILE**

UNIPAR – UNIVERSIDADE PARANAENSE

EMENTA

HTML5 e CSS3; Fundamentos de HTML5 e CSS3; Responsividade com Media Queries; Seletores; SASS; UX; Javascript; ES6;

AGENDA

27/03: HTML5, Modernizr, HTML5 Boilerplate, Formulários e Validações, CSS3, Normalizer.css, Media Query, Framework Fron-End Bootstrap

10/04: SASS, JS, ES6, 2016, 2017, 2018, JQuery

24/04: API HTML5, UX, Admin Teampates, Acessibilidade, Testes de Responsividade, Performance, Admin Templates

Ferramentas

- Visual Studio Code (VSCode)
 - <https://code.visualstudio.com/>

Atividades e Avaliação

- GitHub
 - <https://github.com/>
- Dois repositório
 - Atividades
 - Projeto Avaliação

Material e Comunicação

- Whats
- Slack
 - <https://slack.com/intl/pt-br/>
 - https://join.slack.com/t/slack-pfn3604/shared_invite/zt-o40usk13-LDOfezki5D224BEeJn4R~Q
- Workspace: Pós Web & Dispositivos Móvis
- Canal: html5-css3-es6

Principal Referência

<https://www.w3schools.com/>

SOBRE MIM- FORMAÇÃO

Jemerson Fernando Maia

- Bacharel em Sistemas de Informação, 2004, UNIPAR – Universidade Paranaense, Toledo, PR
- Especialização em Tecnologias para Desenv. De Aplicações WEB, 2006, UNIPAR – Universidade Paranaense, Guáira, PR
- Especialização em Tecnologia Java, 2008, UNIPAN – União Pan-Americano de Ensino, Cascavel, PR
- Mestrando em Ciências da Computação, UNIOESTE – Universidade Estadual do Oeste do Paraná, Cascavel, PR

SOBRE MIM – EXPERIÊNCIA PROFISSIONAL

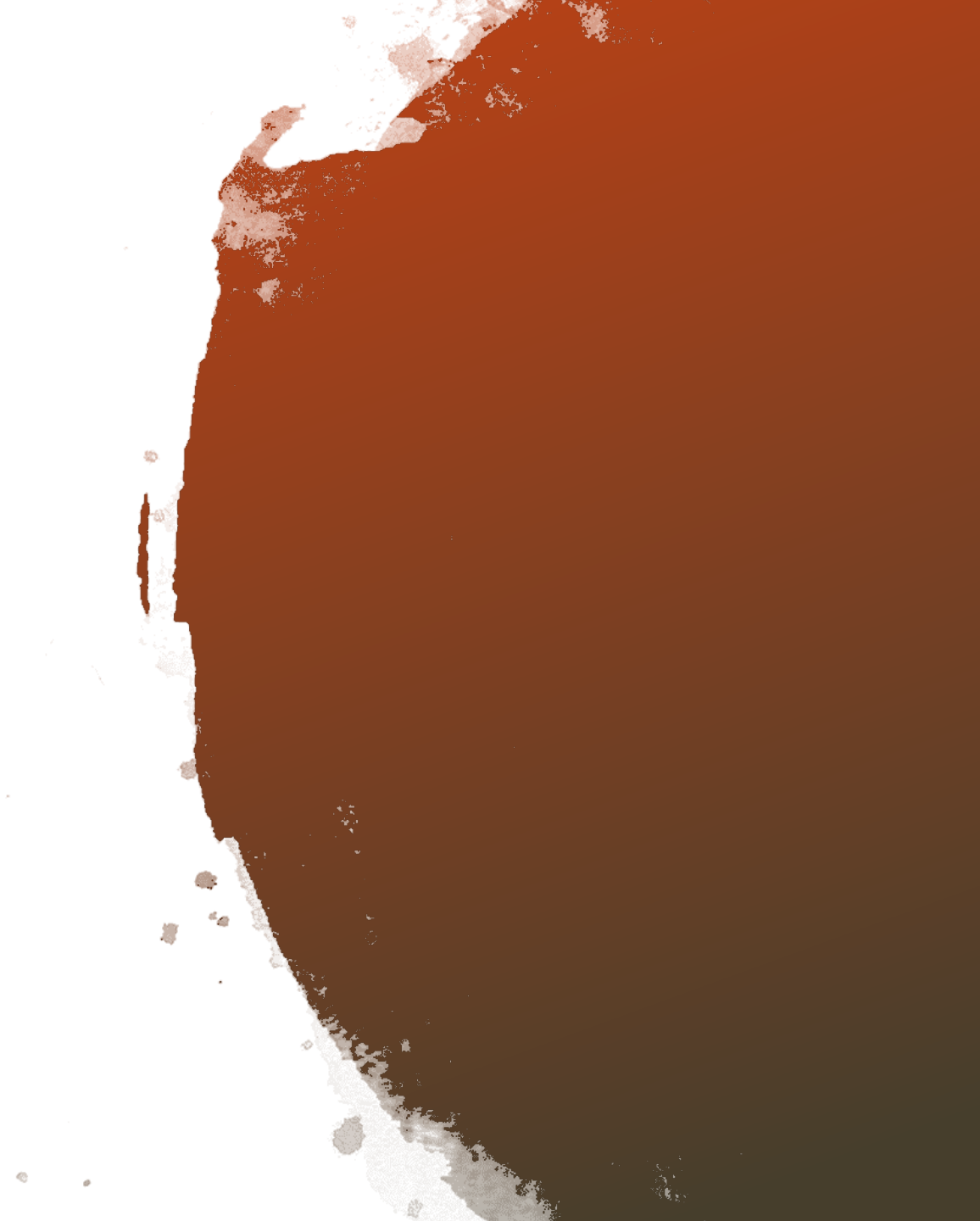
Jemerson Fernando Maia

- Desenvolvedor de Software, PRATI-DONADUZZI, Toledo, desde 2010
 - JEE, EJB, JPA, JSF, JSE, WildFly, Glassfish
 - Ruby, RubyOnRails, Apache, Passenger
 - HTML, CSS, JavaScript, CMS Joomla
 - MySQL, PostgreSQL, Oracle, SQLServer, Firebird
 - Integrações (Middleware e WS-REST, SOAP)
 - DEVOPS (Docker Swarm, CI/CD GitLab+Jenkins+SonarQube)
- Coordenador Curso de Graduação ADS – Faculdade Biopark, Toledo, 2019-2020
- Professor UNIPAR – Universidade Paranaense, 2008-2017
- Desenvolvedor, Datacoper Sistemas, Canscavel, 2009-2010
- Desenvolvedor, Maxicon Systems, Toledo, 2005-2009



Conhecer a turma

Por que
aprender HTML,
CSS e JS?



O que é um
desenvolvedor
front-end?

HTML + CSS + JavaScript

Informação

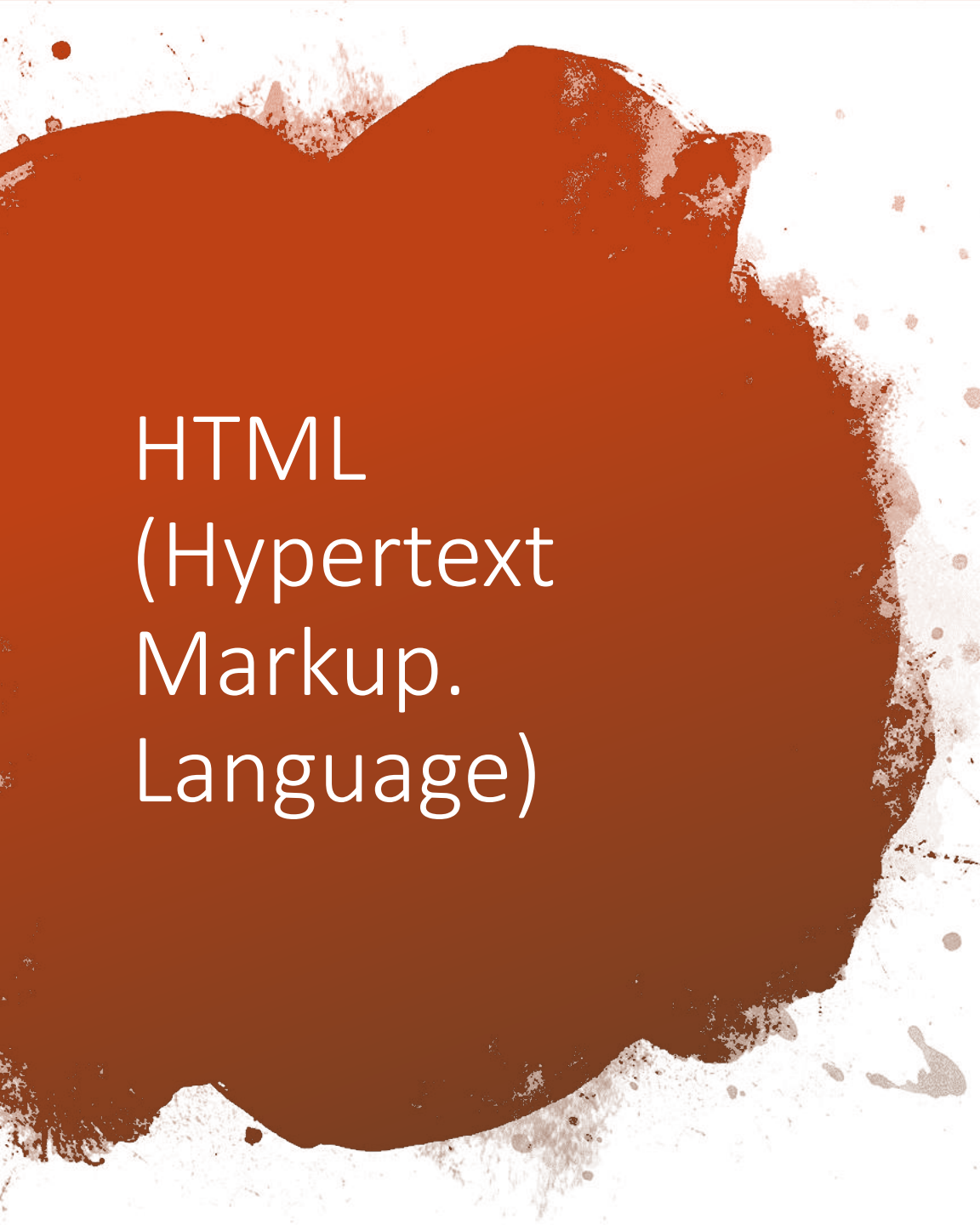
- marcação
HTML: ela define o que é cada informação

Formatação

- responsável por formatar a informação para que ela seja consumida em qualquer meio de acesso

Comportamento

- O JavaScript até hoje é o principal responsável



HTML (Hypertext Markup Language)

HTML é uma linguagem para publicação de conteúdo para Web.

Markup

Nós marcamos a informação, dando significado a estes objetos, tornando-os legíveis para os meios de acesso.

Hypertext

Hipertexto são conjuntos de elemento

Estes elementos conectados formam uma grande rede de informação

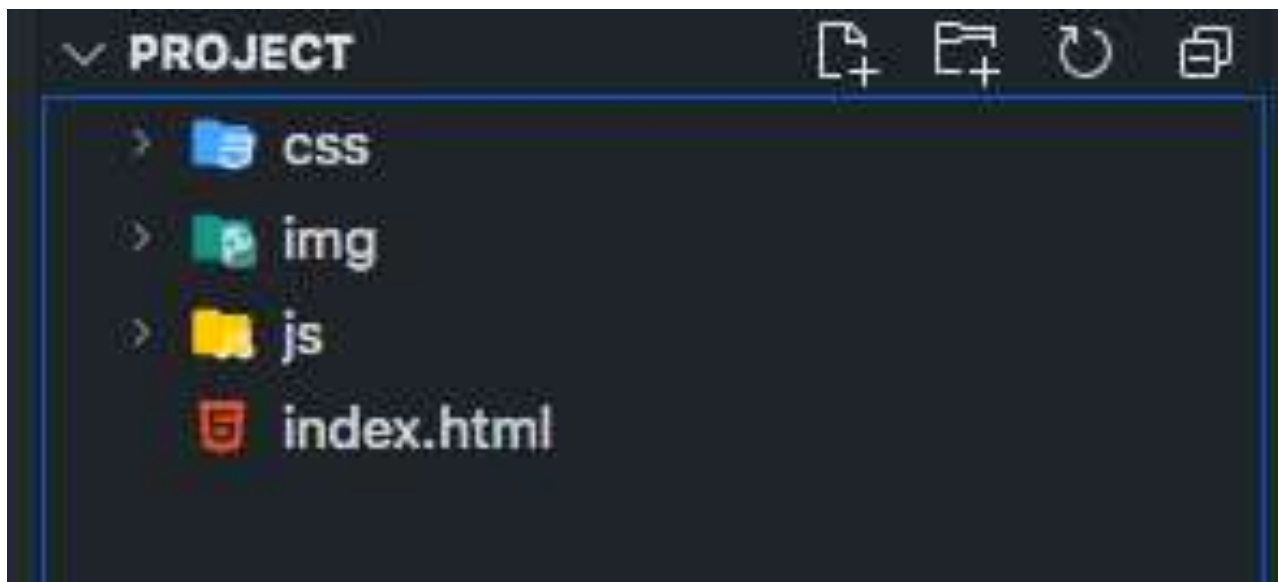
Hipermídia

HTML5

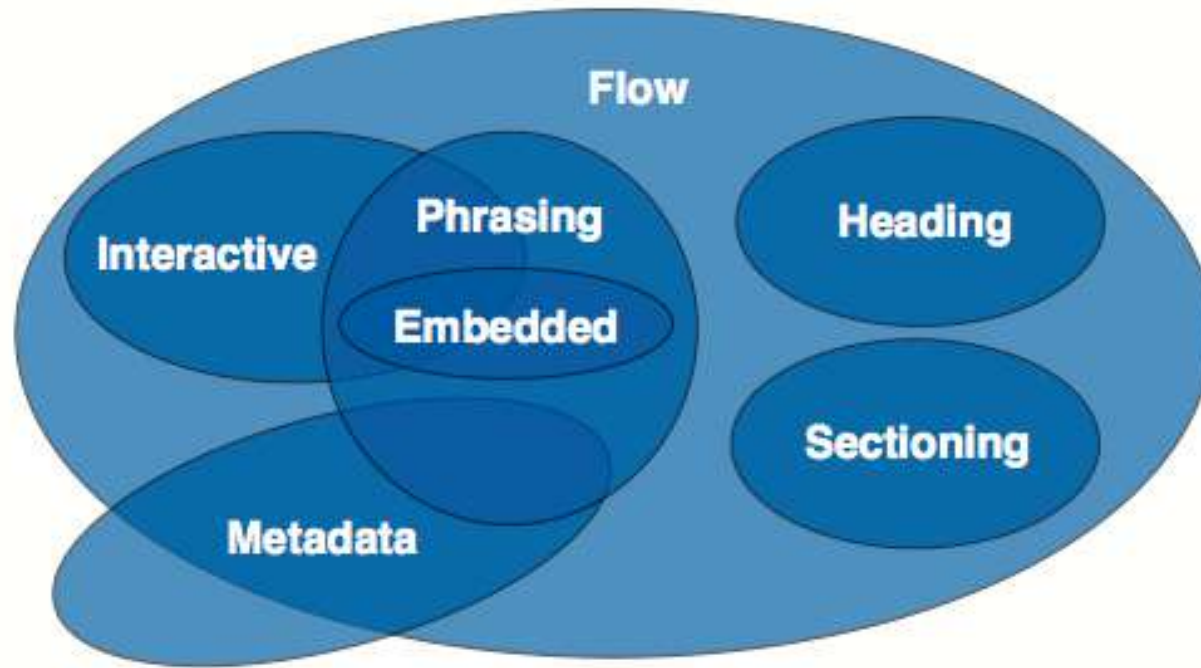
- Escrever HTML ficou mais simples.
- Novas tags.
- Novos atributos e atributos personalizados.
- Novas APIs

Estrutura Básica

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title></title>
    <meta charset="UTF-8">
    <link href="css/style.css" rel="stylesheet">
  </head>
  <body>
    Conteúdo aqui.
  </body>
</html>
```



Estrutura do Projeto



Categories Tags

Metadata content

base	command	link	meta	noscript
script	style	title		

Flow content

a	abbr	address	area	article
aside	audio	b	bdo	blockquote
br	button	canvas	cite	code
command	datalist	del	details	dfn
div	dl	em	embed	fieldset
figure	footer	form	h1 até h6	header
hgroup	hr	i	iframe	img
input	ins	kbd	keygen	label
link	map	mark	math	menu
meta	meter	nav	noscript	object
ol	output	p	pre	progress
q	ruby	samp	script	section
select	small	span	strong	style
sub	sup	svg	table	textarea
time	ul	var	video	wbr
text				

Sectioning content

header	footer	article	aside	nav	section
--------	--------	---------	-------	-----	---------

Heading content

h1 até h6	hgroup	header
-----------	--------	--------

Phrasing content

a	abbr	area	audio	b
bdo	br	button	canvas	vite
code	command	datalist	del	dfn
em	embed	i	iframe	img
input	ins	kbd	keygen	label
link	map	mark	math	meta
meter	noscript	object	output	progress
q	ruby	samp	script	sup
select	small	span	strong	sub
svn	textarea	time	var	video
textarea	time	var	video	wbr
Text				

Embedded content

audio	canvas	embed	iframe	img
math	objevt	svg	video	

Interactive content

a	audio	button	details	embed
iframe	img	input	keygen	label
menu	object	select	textarea	video

Block vs Inline

```
<body>
```

```
block
```

```
block
```

```
block
```

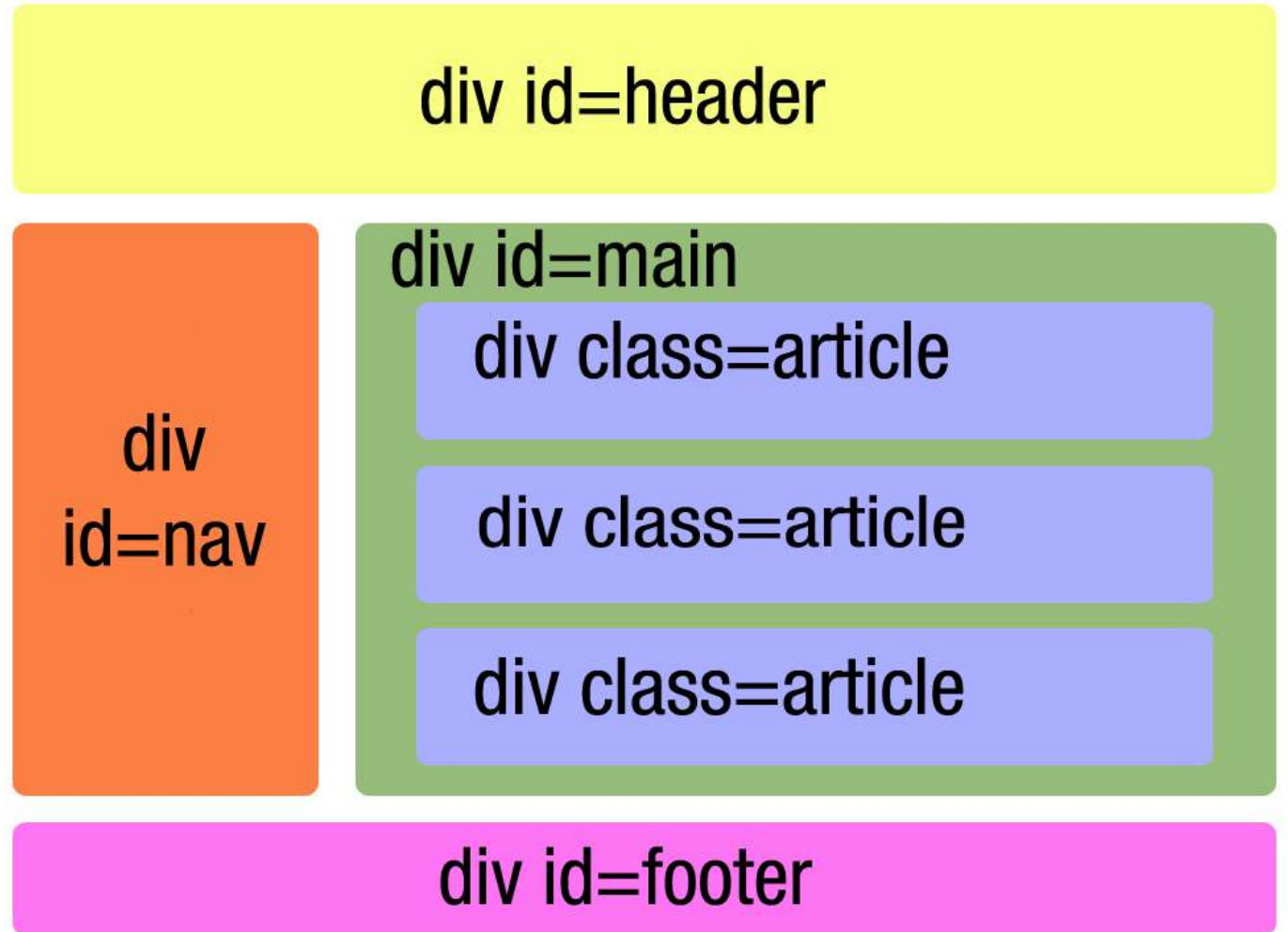
```
inline
```

```
inline
```

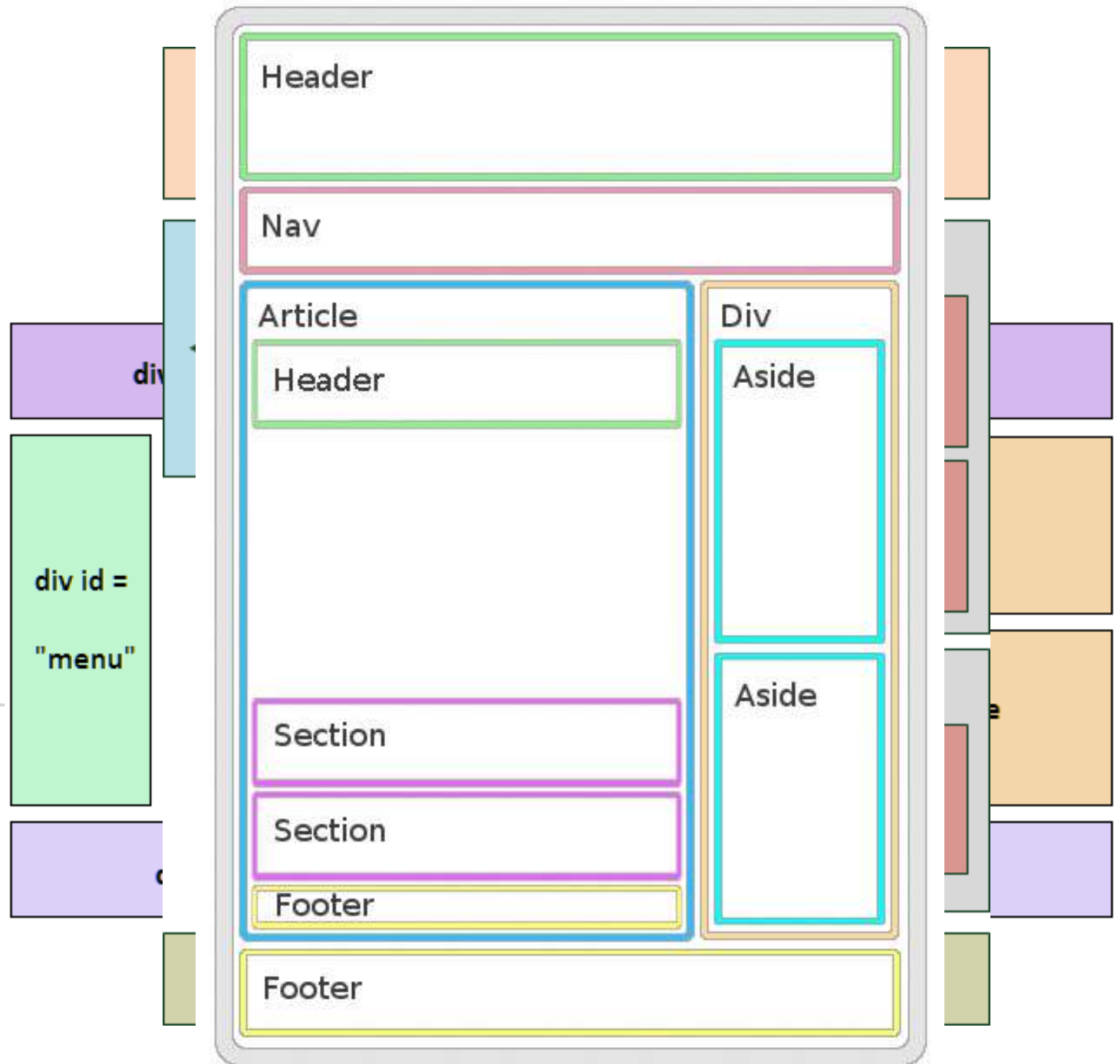
```
inline
```

```
</body>
```

Novos
elementos
semânticos



Novos elementos semânticos



Exemplo

Atividade

- Criar uma página contendo os seguintes elementos
 - Cabeçalho com título do website (HTML5, CSS3, JS)
 - Links de navegação para notícias e formulário
 - Área principal contendo 3 artigos
 - Título
 - Data e email(autor)
 - Imagem
 - Introdução
 - Link para notícia original
 - Rodapé
 - Nome – Nome da pós

Compatibilidade

- <https://caniuse.com/?search=html5>
- <https://html5test.com/results/desktop.html>
- <https://mobilehtml5.org/>
- <https://gs.statcounter.com/>

Modernizr e Html5 Boilerplate

- **Modernizr** é uma pequena biblioteca Javascript que detecta a disponibilidade das novas características do HTML5 e CSS3 nos browsers.- <https://modernizr.com/>
- **HTML5 Boilerplate** é um conjunto de arquivos de código aberto, disponível para download que fornecem uma base para criação de qualquer site - <https://html5boilerplate.com>

Fo

nts:

Name:

Email:

Please fill out this field.

Website:



input

- `<input type="button">`
- `<input type="checkbox">`
- `<input type="color">`
- `<input type="date">`
- `<input type="datetime-local">`
- `<input type="email">`
- `<input type="file">`
- `<input type="hidden">`
- `<input type="image">`
- `<input type="month">`
- `<input type="number">`
- `<input type="password">`
- `<input type="radio">`
- `<input type="range">`
- `<input type="reset">`
- `<input type="search">`
- `<input type="submit">`
- `<input type="tel">`
- `<input type="text">`
- `<input type="time">`
- `<input type="url">`
- `<input type="week">`

Com formatação e validação

Validadores

- autofocus
- placeholder
- required
- maxlength
- pattern - ex: `\d{5}-?\d{3}`

Exemplo

Atividade

- Criar um formulário para cadastro de notícias, contendo campos para:
 - Título
 - Cor do título
 - Data
 - E-mail
 - Imagem
 - Introdução
 - Endereço de origem (URL)
 - Botão enviar

Validadores Customizados

- setCustomValidity

```
<script>
  function vCPF(i){
    i.setCustomValidity(validaCPF(i.
value)?'':'CPF inválido!')
  }
</script>
```

```
<body>
  <form>
    <label>CPF: <input name="cpf"
oninput="vCPF(this)"></label>
    <input type="submit" value="Enviar">
  </form>
</body>
```


MATHML

- O HTML5 incorpora o padrão MathML. Trata-se de uma linguagem de marcação, baseada em XML, para representação de fórmulas matemáticas.

MATHML

$$x = \frac{\sqrt{-b \pm b^2 - 4ac}}{2a}$$

```
<body>
  <math xmlns="http://www.w3.org/1998/Math/MathML">
    <mrow>
      <mi>x</mi>
      <mo>=</mo>
      <mfrac>
```

```
>&minus;</mo>
/mo>
```

```
nes;</mo>
nes;</mo>
```

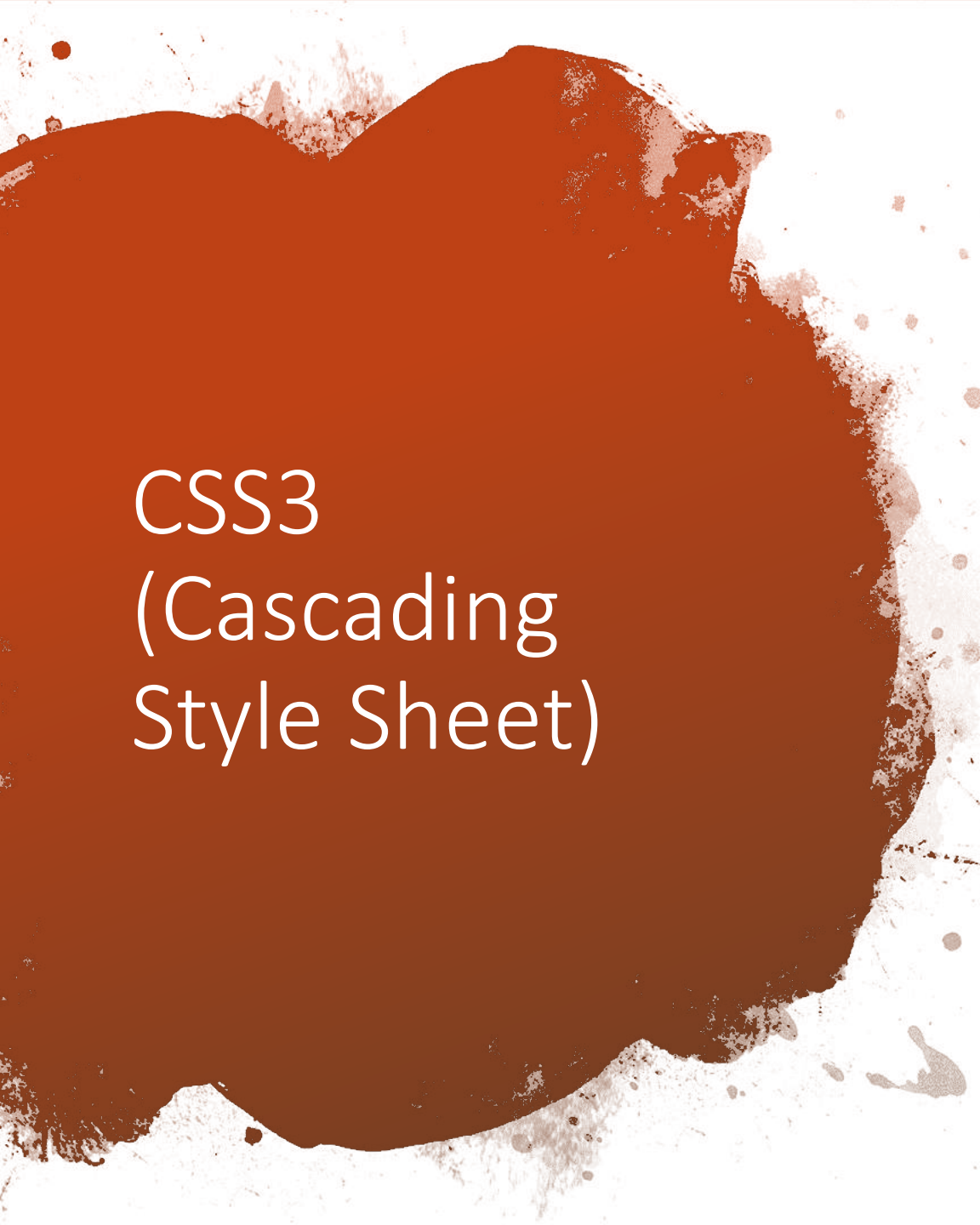
```
</mrow>
  <mrow>
    <mn>2</mn>
    <mo>&InvisibleTimes;</mo>
    <mi>a</mi>
  </mrow>
</mfrac>
</mrow>
</math>
</body>
```

SVG

- SVG é uma linguagem para marcação de gráficos vetoriais.

SVG

```
<body>
  <svg width="400" height="400">
    <!-- Um retângulo: -->
    <rect x="10" y="10" width="150" height="50"
          stroke="#000000" stroke-width="5" fill="#FF0000"></rect>
    <!-- Um círculo -->
    <circle cx="70" cy="240" r="60"
            stroke="#00FF00" stroke width="5" fill="#FFFFFF"></circle>
  </svg>
</body>
```



CSS3 (Cascading Style Sheet)

O CSS dá apresentação ao
conteúdo do website.

CSS3

- seleccionar primeiro e último elemento
- seleccionar elementos pares ou ímpares
- seleccionarmos elementos específicos de um determinado grupo de elementos
- gradiente em textos e elementos
- bordas arredondadas
- sombras em texto e elementos
- manipulação de opacidade
- controle de rotação
- controle de perspectiva
- animação
- controle básico de 3D

Seletores

- Um seletor representa uma estrutura. Essa estrutura é usada como uma condição para determinar quais elementos de um grupo de elementos serão formatados.



Vamos lembrar CSS

Seletor encadeado

```
div p strong a {  
  color: red;  
}
```

Seletor agrupado

```
strong, em, span {  
    color: red;  
}
```

Seletor complexos

```
h1 + p {  
    color:red;  
}
```

Seletor complexos

PADRÃO	SIGNIFICADO	CSS
elemento[atr]	Elemento com um atributo específico.	2
elemento[atr="x"]	Elemento que tenha um atributo com um valor específico igual a "x".	2
elemento[atr~="x"]	Elemento com um atributo cujo valor é uma lista separada por espaços, sendo que um dos valores é "x".	2
elemento[atr^="x"]	Elemento com um atributo cujo valor comece exatamente com string "x".	3
elemento[atr\$="x"]	Elemento com um atributo cujo valor termina exatamente com string "x".	3
elemento[atr*="x"]	Elemento com um atributo cujo valor contenha a string "x".	3
elemento[atr = "en"]	Um elemento que tem o atributo específico com o valor que é separado por hífen começando com EN (da esquerda para direita).	2
elemento:root	Elemento root do documento. Normalmente o HTML.	3
elemento:nth-child(n)	Selecione um objeto N de um determinado elemento.	3

Seletor complexos

elemento:nth-last-child(n)	Seleciona um objeto N começando pelo último objeto do elemento.	3
elemento:empty	Seleciona um elemento vazio, sem filhos. Incluindo elementos de texto.	3
elemento:enabled		
elemento:disabled	Seleciona um elemento de interface que esteja habilitado ou desabilitado, como selects, checkbox, radio button etc	3
elemento:checked	Seleciona elementos que estão checados, como radio buttons e checkboxes.	3
E > F	Seleciona os elementos E que são filhos diretos de F.	2
E + F	Seleciona um elemento F que precede imediatamente o elemento E.	2

<https://www.w3.org/TR/selectors-3/#selectors>

- As pseudo-classes manipulam um determinado estado do elemento.

https://www.w3schools.com/css/css_pseudo_classes.asp

Pseudo-classes

```
table tr:nth-child(odd) {background: gray;}
```

- Pseudo-elementos permitem que os autores selecionem informações inacessíveis e possibilitem os desenvolvedores um caminho para referenciar conteúdos que não existem.

https://www.w3schools.com/css/css_pseudo_elements.asp

Pseudo-elementos

```
p::first-letter {font-size:20px; font-weight: bold; }
```

Seletores por Classe e Identificador

.nome-classe

#id-elemento

CSS Avançado

- Gradiente
- Border Images
- Cantos arredondados
- Sombras
- 2D e 3D
- Transições
- Animações

<https://www.w3schools.com/css/default.asp>

@FONT-FACE

- A regra @font-face é utilizada para que você utilize fonts fora do padrão do sistema em seus sites.

```
@font-face {  
  font-family: helveticaneue;  
  src: url(helveticaNeueLTStd-U1tLt.otf);  
}
```

```
    p {font:36px helveticaneue, Arial, Tahoma,  
  Sans-serif;}
```

@FONT-FACE

String	Font Format	Common Extensions
truetype	TrueType	.ttf
opentype	OpenType	.ttf, .otf
truetype-aat	TrueType with Apple Advanced Typography extensions	.ttf
embedded-opentype	Embedded OpenType	.eot
woff	WOFF (Web Open Font Format)	.woff
svg	SVG Font	.svg, .svgz

Google Fonts

https://www.w3schools.com/css/css_font_google.asp

https://fonts.google.com/?preview.text_type=custom

Motor	Browser
Webkit	Safari, Google Chrome, Mobile Safari, Android Browser, alguns celulares da Nokia e da RIM.
Gecko	Firefox, Mozilla, Camino, Flock, Firefox para Android
Trident	Internet Explorer 4 ao 9
Presto	Opera 7 ao 10

Motores de renderização

Safari	-webkit-
Firefox	-moz-

```
div {  
  -webkit-border-radius: 10px;  
  -moz-border-radius: 10px;  
  border-radius: 10px;  
}
```

motores de renderização

Box model

- O **Box model** é como as propriedades de CSS compõem as dimensões, onde além do width e do height, as propriedades border e padding também influenciam no resultado final

Box mode

```
* {  
  box-sizing: border-box;  
}  
  
.box {  
  background-color: LimeGreen;  
  width: 250px;  
}  
  
.box-with-padding {  
  background-color: LightBlue;  
  padding: 0 25px;  
  width: 250px;  
}
```


Normalize.css

- **Normalize.css** estabiliza estilos para muitos elementos HTML, deixando-os iguais ou muito parecidos entre cada navegador - <https://necolas.github.io/normalize.css/>

Media Queries

- Podemos adicionar **design condicional** que só será aplicado em determinada situação
- Desing Responsivo

@media

CSS3 Media Types

Value	Description
all	Used for all media type devices
print	Used for printers
screen	Used for computer screens, tablets, smart-phones etc.
speech	Used for screenreaders that "reads" the page out loud

Media Types

```
@media screen and (min-width: 400px) {  
  
}
```

Media Queries- Sintaxe

```
@media only screen
and (min-device-width : 768px)
and (max-device-width : 1024px)
and (orientation : landscape) {
/*
  Além de verificar a largura do dispositivo,
  você pode conferir a orientação, utilizando
  "landscape" ou "portrait".
*/
}
```

Media Queries- Sintaxe

Media Queries

- Sintaxe

```
/* For desktop: */
.col-1 {width: 8.33%;}
.col-2 {width: 16.66%;}
.col-3 {width: 25%;}
.col-4 {width: 33.33%;}
.col-5 {width: 41.66%;}
.col-6 {width: 50%;}
.col-7 {width: 58.33%;}
.col-8 {width: 66.66%;}
.col-9 {width: 75%;}
.col-10 {width: 83.33%;}
.col-11 {width: 91.66%;}
.col-12 {width: 100%;}

@media only screen and (max-width: 768px) {
  /* For mobile phones: */
  [class*="col-"] {
    width: 100%;
  }
}
```

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

Media Queries- viewport

Layout Fluído

- Isso quer dizer não usar medidas fixas como pixels (ou pontos, centímetros, milímetros etc) pra programar o design
- As duas medidas flexíveis mais usadas são as porcentagens e o em

Layout Fluído

```
article {  
    /* o article ocupa 3/4 da página */  
    width: 75%;  
  
    /* e tem uma margem interna de 10% do tamanho do pai */  
    padding: 10%;  
}
```

Layout Fluído

```
html {  
    /* tem um font-size implícito que equivale  
       a 16px na maioria dos navegadores */  
}  
p {  
    /* parágrafos com tamanho base de 16px * 1.125 = 18px */  
    font-size: 1.125em;  
}  
h1 {  
    /* título principal com o dobro da fonte base */  
    font-size: 2em;  
}  
h2 {  
    /* título secundário é 50% maior que o valor base */  
    font-size: 150%; /* equivalente a 1.5em */  
}
```

Layout Fluído

O segredo de um layout com medidas flexíveis é pensar na proporção entre os elementos

Exemplo

Atividade

- Adicionar folha de estilo na página de notícias fazendo com que a página apresente a seguinte configuração:
 - Artigos fiquem fluídos ocupando:
 - 3 colunas
 - 1 coluna quando resolução máxima for 800px.

Frameworks Front-End

- ferramenta que auxilia o programador a trabalhar com determinada linguagem
- Fron-end auxiliam no desenvolvimento de GUI

Frameworks Front-End

- <https://purecss.io/>
- <https://materializecss.com/>
- <https://get.foundation/>

Bootstrap

<https://getbootstrap.com/>

- Um dos mais populares Framework Front-End

Referências

- <https://www.w3schools.com/html/default.asp>
- <https://www.w3schools.com/css/default.asp>
- <https://www.w3schools.com/bootstrap4/default.asp>
- <https://getbootstrap.com/>
- <https://devdocs.io/>