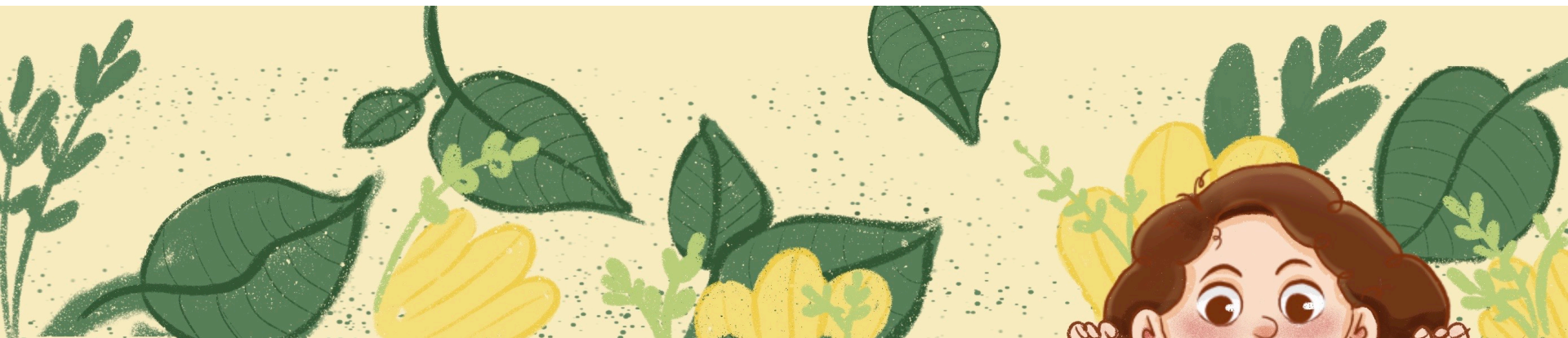


Desenvolvimento Web Básico





Gabi Grimm

gabrielle.grimm@up.edu.br

Doutoranda em Design - UFPR

Mestrado em Tecnologia – UTFPR

Bacharelado em Design -Programação visual – UP

Desenho Industrial – UTFPR

Áreas de interesse: ilustração, concept art, design digital, UX Design e tecnologias educacionais...

Ementa, objetivos e conteúdos

Conferir plano de aula



PLANO DE AULA

Componente curricular	DESENVOLVIMENTO WEB BÁSICO				
Professor(a)	GABRIELLE HARTMANN GRIMM				
CH (h-aula)	80	CH (h-relógio)	67	CH em extensão (h-aula)	-

EMENTA

Prática do desenvolvimento WEB básico. Ferramentas e linguagens de programação e marcação para criar páginas Web.

OBJETIVOS GLOBAIS DE APRENDIZAGEM PARA OS QUAIS A COMPONENTE CURRICULAR CONTRIBUI

O egresso será capaz de...

Aplicar conceitos, métodos, técnicas através de ferramentas para o desenvolvimento de sistemas web utilizando linguagens de programação para demonstração conceitual e aplicada aos contextos de mercado com a utilização de frameworks em nível básico e intermediário.
Projetar soluções usando tecnologia da informação nas organizações a fim de adequá-las às oportunidades de mudança.
Desenvolver projetos de software com a utilização de frameworks, garantindo a resolução algorítmica de problemas.

OBJETIVOS DE APRENDIZAGEM ESPECÍFICOS DA COMPONENTE CURRICULAR Ao concluir a componente curricular, o estudante será capaz de...	DIMENSÃO ¹	CONTEÚDOS
Criar páginas estáticas utilizando os principais recursos da linguagem de marcação HTML como marcadores, listas, imagens, tabelas e formulários.	3	Desenvolvimento de sites estáticos; Projeto (design) de interface com usuários; Programação HTML.

Objetivos de aprendizagem

- Criar páginas estáticas utilizando os principais recursos da linguagem de marcação HTML como marcadores, listas, imagens, tabelas e formulários.
- Utilizar a linguagem CSS e design responsivo para melhorar a organização, estruturação e a experiência do usuário em uma aplicação web.
- Utilizar HTML, CSS, Javascript e bibliotecas Javascript para criar páginas web dinâmicas e com validações.
- Compreender a estrutura de requisições de recursos na internet e identificar seus principais componentes: protocolos, servidores, linguagens, etc.
- Aplicar os diferentes tipos de métodos GET, POST, PUT e DELETE para realização de um cadastro básico.
- Integrar aplicações frontend e backend utilizando webservices e requisições síncronas e assíncronas.
- Publicar um site em um servidor web local ou remoto.
- Testar, debugar e versionar aplicações web.

Cronograma

- Plano de aula contém cronograma do semestre todo,
- Sujeito a alterações se necessário.

CRONOGRAMA DE ATIVIDADES

Data	Atividade prevista
02/03	Apresentação da disciplina e visão do desenvolvimento WEB sites estáticos
09/03	Introdução ao HTML5 (Tags, atributos e estrutura semântica de páginas)
16/03	Marcação de listas, tabelas, imagens e vídeos
23/03	Construção de formulários e introdução às requisições HTTP
30/03	Servidores WEB, estrutura de pastas e navegação entre páginas
06/04	Criação de layout de telas (UI) utilizando CSS
13/04	Folhas de estilos e propriedades avançadas do CSS
20/04	Responsividade e construção de layouts
27/04	Semana de Avaliação

Organização da disciplina e das aulas

- Aulas expositivas
- Exercícios práticos síncronos
- Atividades assíncronas
- Projetos em equipe

Aulas:

19h00 às 20h40: Conteúdo da aula;

20h40 às 20h50: Intervalo;

20h50 às 22h30: Exercícios e/ou atividades

Avaliações

1º bimestre (A1): 5,0 pontos

- Exercícios: 3,0 pontos
- Projeto em equipe: 2,0 pontos

2º bimestre (A2): 5,0 pontos

- Exercícios: 3,0 pontos
- Projeto em equipe: 2,0 pontos

A1 + A2: 10,0 pontos

Nota para aprovação= 6,0 pontos

Se não atingir a nota para aprovação, deve realizar a Avaliação Final **AF** (substitutiva).

Exercícios

- Em todas as aulas teremos exercícios como este de aquecimento de hoje!
- Salve todos eles, em pastas organizadas (nome do arquivo com a data da aula)
- Terão entregas no AVA para os exercícios

Desenvolvimento Web Básico

WEB

WEB?

A web consiste em um sistema **cliente-servidor**.

A comunicação entre os clientes e servidores se dá através do **protocolo HTTP**:

1. Usuário digita uma **URL** navegador (browser) ou clica em um link;
2. Navegador envia a requisição até o servidor de web;
3. Servidor de web envia o conteúdo requisitado para o navegador;
4. O Navegador apresenta a informação para o usuário.

PADRÕES

W3C

W3C (World Wide Web Consortium)

é um consórcio de empresas de tecnologia.

Padrões Web ou Web Standards é um conjunto de recomendações, produzidos pelo W3C e destinados a **orientar** fabricantes, **desenvolvedores e projetistas para o uso de práticas** que possibilitem a criação de uma Web **acessível a todos**, independentemente dos dispositivos usados ou de suas necessidades



Tim Berners-Lee
É o criador da World Wide Web dez/1990

Segundo a W3C a Web é baseada em três pilares

1 – URL

Um esquema de nomes para localização de fontes de informação na web, esse esquema chama-se URL

2 – HTTP

Um protocolo de acesso para acessar essas fontes, hoje o HTTP

3 – HTML

Uma linguagem de hipertexto para facilitar a navegação entre as fontes de informação: o HTML

HTML

HTML

abreviação para

HyperText Markup Language,

que pode ser traduzido como

Linguagem de Marcação para Hipertexto.

HTML

abreviação para

HyperText Markup Language,

que pode ser traduzido como

Linguagem de Marcação para Hipertexto.

HTML

abreviação para

HyperText Markup Language,

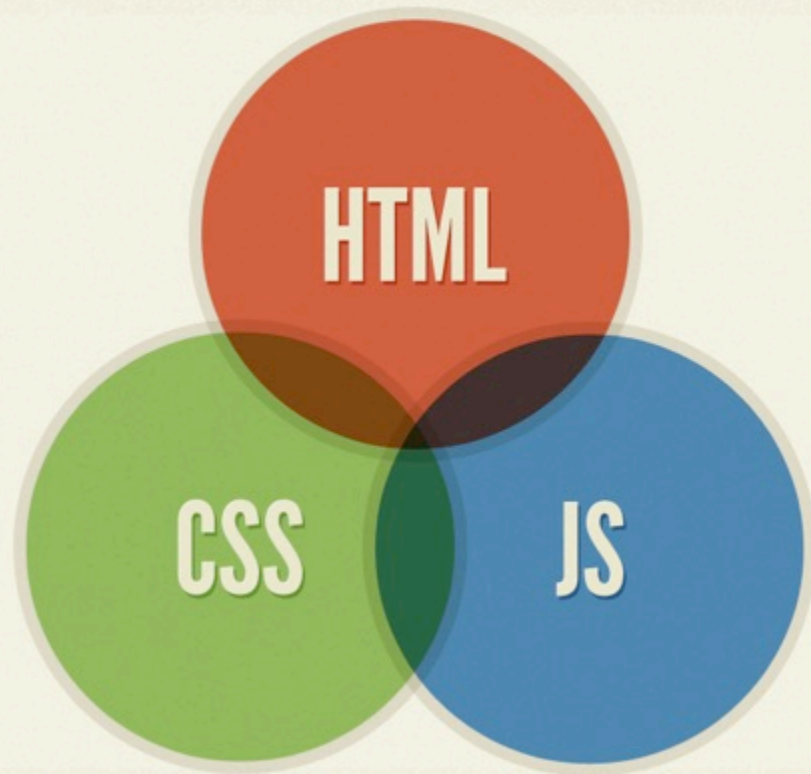
que pode ser traduzido como

Linguagem de Marcação para Hipertexto.

The background of the slide is a close-up, slightly blurred image of a computer screen displaying HTML code. The code is color-coded: opening and closing tags are in green, and attributes like 'class' and their values are in red. Visible code includes <body>, <div class="main">, <div cla..., <div..., <div..., , and <div... The left edge of the image shows a vertical sidebar with several red square icons, some containing white symbols like a minus sign. A semi-transparent dark grey rectangle is positioned on the right side of the image, containing the text for the HTML section.

HTML

Compõe a estrutura de uma página Web através de etiquetas (tags) e atributos.



**Design
responsivo**



Ferramentas



Visual Studio Code

Docs

Updates

Blog

API

Extensions

FAQ

Learn



Search Docs



Download

Welcome, Brackets users! [Learn about migrating to VS Code](#)



Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.



↓ **Windows**

Windows 7, 8, 10



↓ **.deb**

Debian, Ubuntu

↓ **.rpm**

Red Hat, Fedora, SUSE



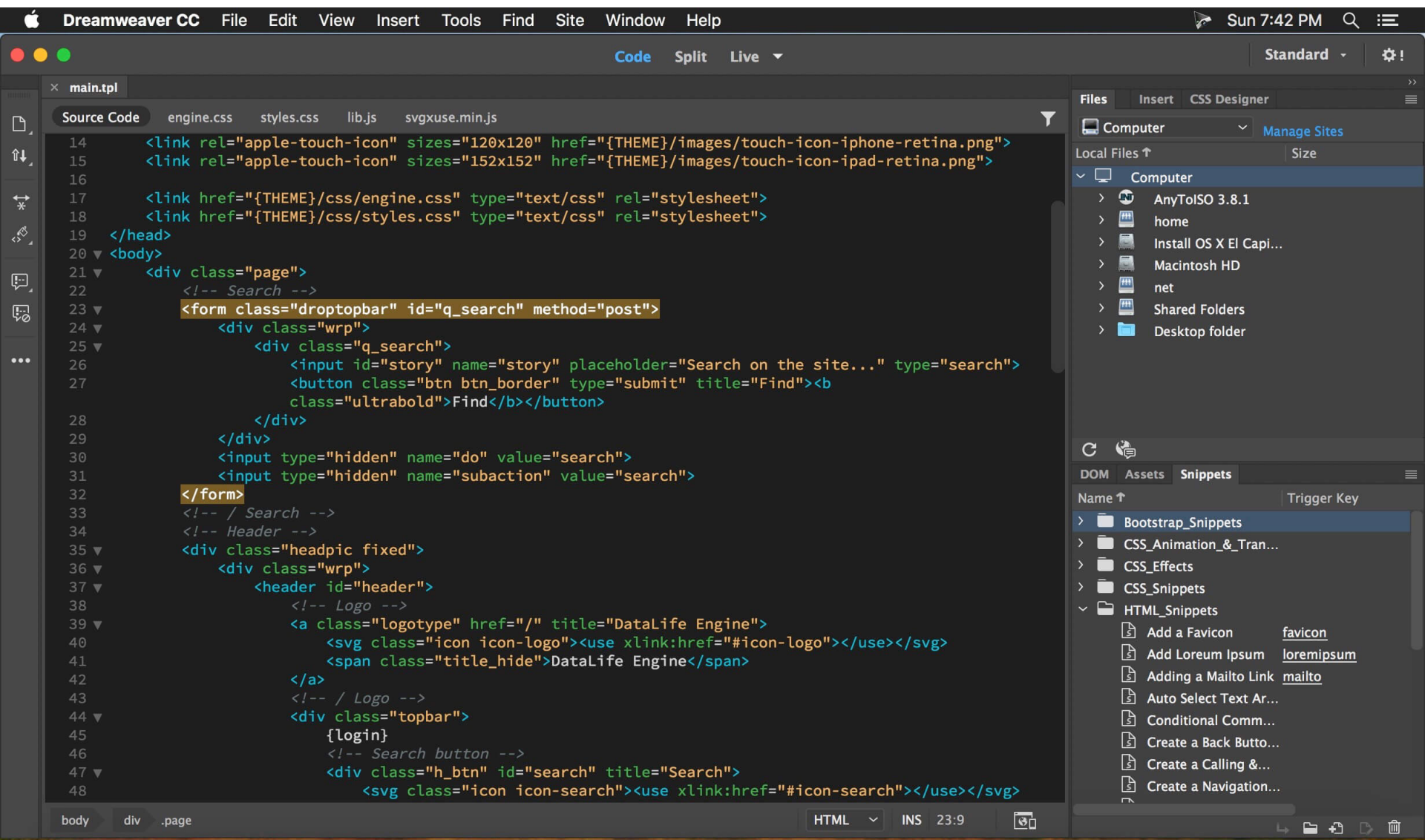
↓ **Mac**

macOS 10.10+

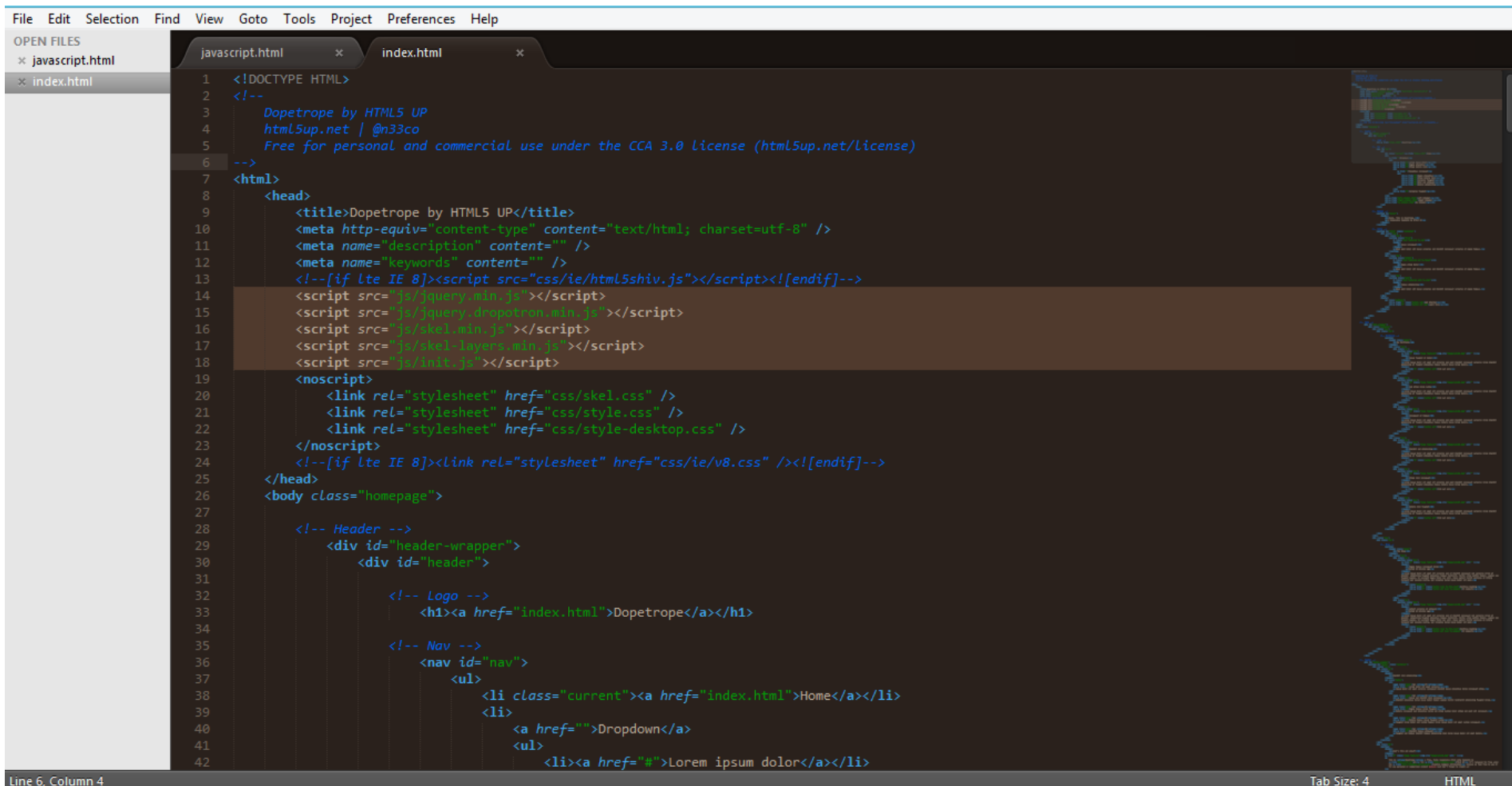
Visual Studio Code


```
1 extends layout
2
3 block content
4   header#site-header
5     .container-fluid
6       .row-fluid
7         #splash.span12.carousel.fade(data-interval='10000')
8         ol.carousel-indicators
9           li(data-target='#splash', data-slide-to='0').active
10          li(data-target='#splash', data-slide-to='1')
11          li(data-target='#splash', data-slide-to='2')
12          li(data-target='#splash', data-slide-to='3')
13          li(data-target='#splash', data-slide-to='4')
14          li(data-target='#splash', data-slide-to='5')
15          li(data-target='#splash', data-slide-to='6')
16        .carousel-inner
17          .item.active
18            img(src='images/header/new-system-screenshot.png')
19            .carousel-caption: p Following a tutorial on my new comp. Wheee!
20          .item
21            img(src='images/header/making-a-colorscheme.png')
22            .carousel-caption: p Designing a colorscheme with the help of colorschemedesigner.com.
23          .item
24            img(src='images/header/local-development-workflow.png')
25            .carousel-caption: p Building this portfolio, on localhost in a NodeJS server, using ExpressJS. Vim is awesome, if you have time to learn it.
26          .item
27            img(src='images/header/color-contrast-experiment.png')
28            .carousel-caption: p An experiment in color. Notice that a gradient between two colors doesn't produce the color between them around the color wheel; a gradient cuts a straight line across the wheel!
29          .item
30            img(src='images/header/ndss.png')
31            .carousel-caption: p A school project about my old high school. They never knew I made it, think I should try to sell it to them?
32          .item
33            img(src='images/header/picpost-gallery.png')
34            .carousel-caption: p Using <a href='http://isotope.metafizzy.co/'>Isotope</a> for an image gallery.
35          .item
36            img(src='images/header/picpost-upload.png')
37            .carousel-caption: p An upload page.
38        #middle
39          .container-fluid
40            .row-fluid
41              section(role='main')#main-content.span9
42                .row-fluid
43                  // posts, start an include when this gets variety
44                  for post in page.posts
45                    article(id = 'post-#{post.title.replace(/\s/g, '-').toLowerCase()}').span4
46                      .article-wrapper
47                        h3!= post.title
48                        div(role='content')!= post.content
49              section(role='complementary')#secondary-content.span3
50                // needed to keep top position consistent with section[role=main]
51                .row-fluid
52                  // messages
53                  for message in page.messages
54                    article(id = 'message-#{message.title.replace(/\s/g, '-').toLowerCase()}')
55                      .article-wrapper
56                        h3!= message.title
57                        div(role='content')!= message.content
58          footer
59            .container-fluid
60              .row-fluid
61                .span4
62                  .article-wrapper
63                    h2 find me on
64                    ul
65                      li <strong>github</strong>
66                        a(href='//github.com/d4goxn') d4goxn
67                      li <strong>stackexchange</strong>
68                        a(href='//stackexchange.com/users/178206/dan-ross?tab=accounts') Dan Ross
69                      li <strong>G+</strong>
70                        a(href='//plus.google.com/105196768292553852888') Dan Ross
71                      li <strong>twitter</strong>
72                        a(href='//twitter.com/ross9885') @ross9885.
73                        | I may actually use it, someday
74                      li <strong>facebook</strong>: Maybe if it paid me $10<sup>[4 to 6]</sup>. <a href='https://soundcloud.com/musicdotexe/dotexe-terminated'>Kill it with fire</a>!
75                .span4
76                  .article-wrapper
77                    h2 walk my webs
78                    ul
79                      li <a href='/library' class='inactive'><strong>reading</strong></a>
80                        | 1500 bookmarks and counting. Looking for a gift? Gimme a <a href='http://www.bookdepository.com/'>bookstore</a> gift certificate. Got too many books? Give 'em to <a href='http://www.literacynanaimo.org/support-literacy/'>Literacy Nanaïmo</a> to grow strong young grasshoppers and help eradicate ignorance
81                      li <a href='/politics' class='inactive'><strong>politics</strong></a>
82                        | practical, but basically green. Want my vote? Show me a plan with numbers that I can be proud of when I die
83                      li <a href='/giants' class='inactive'><strong>giants</strong></a>
84                        | whose shoulders I've hopped across
85                .span4
```

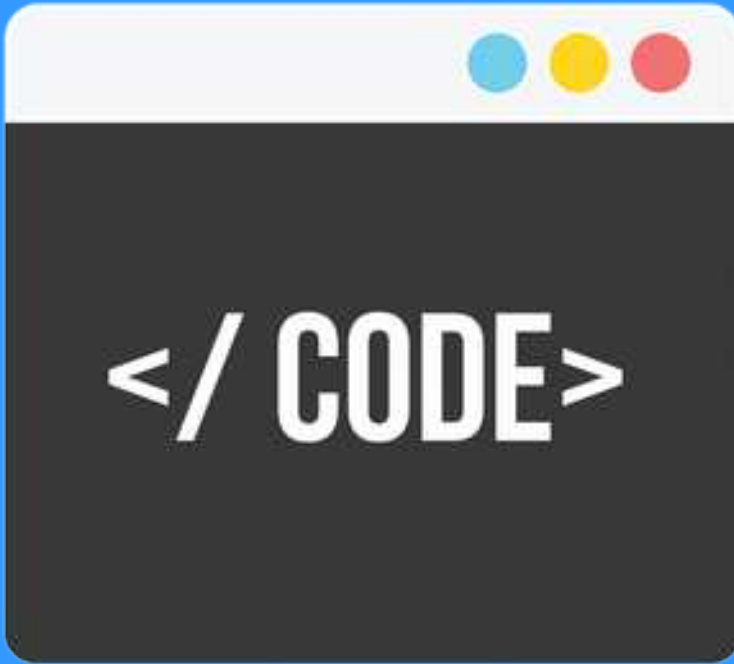
Vim



Dreamweaver

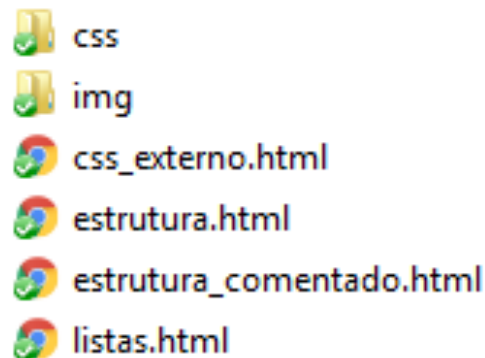


Sublime Text



HTML5

Lembre-se de manter os arquivos sempre organizados. Faça um diretório para todos os arquivos de um mesmo projeto, organize diretórios para imagens, fonts, e demais arquivos necessários.



Nomeações:

4 regras básicas:

- Prestar atenção na intercalação entre maiúsculas e minúsculas
- Sem caracteres especiais
- Sem acentuação ou pontuação
- Underlines e traços ao invés de espaço



HTML – Sintaxe básica

Rótulos usados para informar ao navegador como deve ser apresentado o website

<tag>Texto</tag>

Inicia o comando

Encerra o comando

Recebe o comando

HTML

Todo arquivo HTML apresenta marcações (**< e >**)

Toda tag tem um nome associado. Ex: **<body>** ou **<p>**

Tags podem ter conteúdo: Ex: um texto em **destaque**

Tags podem ser autocontidas. Ex: **
**

Tags podem conter atributos. Ex: ****

E esses atributos podem ser autocontidos como no exemplo acima

Estrutura

```
<html>
  <head>
    <title>Design de Interfaces</title>
  </head>
  <body>
    <p>Contruindo uma página HTML</p>
  </body>
</html>
```

HTML: Nó raiz

HEAD: Cabeçalho do documento – tudo o que é importante de informar ao navegador ou motores de busca mas não aparece no corpo do documento

TITLE: Título da página

BODY: Corpo do documento

HTML5

Doctypes

A sua função é "ligar" o modo de padrões web (standards mode) dos navegadores.

Não é uma tag e sim uma instrução

<!doctype html>



HTML Reference

[HTML by Alphabet](#)[HTML by Category](#)[HTML Attributes](#)[HTML Global Attributes](#)[HTML Events](#)[HTML Canvas](#)[HTML Audio/Video](#)[HTML Doctypes](#)[HTML URL Encode](#)[HTML Language Codes](#)[HTML Country Codes](#)[HTTP Messages](#)[HTTP Methods](#)[PX to EM Converter](#)[Keyboard Shortcuts](#)

HTML Tags

[<!-->](#)[<!DOCTYPE>](#)[<a>](#)[<abbr>](#)

HTML 5

```
<!DOCTYPE html>
```

HTML 4.01 Strict

This DTD contains all HTML elements and attributes, but does NOT INCLUDE presentational or deprecated elements (like font). Framesets are not allowed.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

HTML 4.01 Transitional

This DTD contains all HTML elements and attributes, INCLUDING presentational and deprecated elements (like font). Framesets are not allowed.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

HTML 4.01 Frameset

This DTD is equal to HTML 4.01 Transitional, but allows the use of frameset content.

HTML5

Codificação

Especificar a codificação dos caracteres

<meta charset="utf-8" />

A tag *meta* deve estar sempre dentro da tag *head* e representa vários tipos de metadados.

Metadados são informações que descrevem o conteúdo do seu arquivo.

Ou seja: **dados sobre seus dados**.

Exemplos de metadados são: palavras-chave, informações pro Google indexar o seu site, resumo do conteúdo, etc.

```
<!doctype html>
<html>
  <head>
    <title>Exercícios</title>
    <meta charset="utf-8" />
  </head>
  <body>
    <h1>Contruindo uma página HTML</h1>
  </body>
</html>
```

```
<meta charset="utf-8" />
```



Atributo de uma tag

Estrutura

<meta **charset**="utf-8" />

<tag **atributo**="valor" />

Ou se tiver algum conteúdo

<tag atributo="valor">Conteúdo da tag</tag>

Exercícios

Salvou esse exercício de hoje???

Não esqueça!

- Em todas as aulas teremos exercícios como este de aquecimento de hoje!
- Salve todos eles, em pastas organizadas (nome do arquivo com a data da aula)
- Terão entregas no AVA para os exercícios