


Sprint 1

1) HTML

$$H + m \mid 5$$

- +AGS

$\langle +ag \rangle$
↑
abre

contenido

$\langle 1+ag \rangle$
↑
FECHA

→ + algumas tags podem não precisar de "fechamento"

` new site `

↑ tag

↑ atributo href (referência)

↑ conteúdo

↑ fechada tag

<!DOCTYPE html> ← indica o tipo do documento

<html lang="pt-br"> ← começo / tipo da lingua

`<head>` → fica na tab do navegador 

met9 charset = "UTF-8" > tipo de char

`<+title> Nome </title>` → nome que ficava na tab

</head>

`<body>` → que aparece na página em si

$\langle n1 \rangle$ some $\langle /n1 \rangle$ titolo

`<h2> Nome </h2>` ← subtítulo

`<p> Nome </p>` → parágrafo

`</body>`

`</html>` → fim do documento

• pular linha → `
`

• comentário `<!-- Nome -->`

→ são links que apontam p/ outras páginas/docs

• Ancoras e links

↳ são links que marcam coisas próprias da página

` Título da Ancora `

tag de referência

↳ caminhos relativos precisam de # antes do nome

`<h1 id = Nome> título </h1>`

↳ tag que cria uma referência

` Nome `

↳ tag (target = "_blank")

cria uma nova tab p/ quando

o usuário clicar em um link

• Listas

• elemento
• elemento
• elemento

→ Não ordenadas

 título da lista

 elemento

 elemento



1. elemento
2. elemento
3. elemento

→ Ordenadas

 título da lista

 elemento

 elemento

• Imagem

<img src = " nome
doc .jpg "

ou { " nome
pasta / nome
doc .jpg " }
" caminho absoluto "

alt = " descrição da imagem " >

2) anotações de dúvidas : How to creat a Calculator... [Mediom] 🔍

• CSS → Margin → área transparente em volta de um elemento

→ define a posição de um elemento relativo a outros elementos

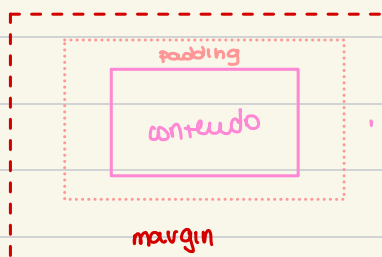
→ propriedades → valores

margin-top tamanhos (pixels, ems)

margin-bottom porcentagem (relativo a largura do pai)

margin-right auto

margin-left



• Margin x Padding

→ área transparente pl/rova da
margem do elemento

→ área dentro da margem
do elemento



• Sintaxe \rightarrow margin : _____ ;

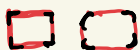
- 1) um valor \rightarrow aplica o **tamanho** p/ os quatro lados
- 2) dois valores \rightarrow aplica o 1º **tamanho** ao top + bottom e o 2º **tamanho** aplica ao right + left
- 3) tres valores \rightarrow aplica o 1º **tamanho** ao top, o 2º aos left + right e o 3º é para o bottom
- 4) quatro valores \rightarrow o **tamanho** cui de : top, right, bottom e left
 \hookrightarrow ex: margin: 15px 15px 15px 15px ;

• Unidades de tamanho

- 1) pixels (px)
- 2) porcentagem (%) \rightarrow Se baseia no tamanho do **pais** do elemento. Então, **margin: 10.1%** cria uma "margin" com 10.1% maior do que o elemento dentro dela.
- 3) (em) \rightarrow Se baseia no tamanho da fonte do elemento. 1em é igual ao tamanho atual da fonte. Bom para escalar o espaço do elemento com o texto.
- 4) (rem) \rightarrow Similar ao **em** mas se baseia na fonte do elemento em que está se modificando.
- 5) Viewport units (vh/vw) \rightarrow Se baseia no tamanho da janela do browser. 1vh = 1% da altura da janela de utilização ::

2.1) Box Sizing CSS - w3schools 🔍

- **box-sizing** é uma propriedade que deixa incluir **padding** e **border** no total da altura e largura de um elemento.



↳ `box-sizing: border-box;` → adiciona a **padding + border** como parte da altura e largura de um elemento

2.2) min-height CSS - w3schools 🔍

→ coloca a altura mínima de um elemento **<p>**

EX `.p1 { min-height: 200px; }` `.p2 { min-height: 100px; }`



→ se o conteúdo for menor que a altura mínima, a propriedade é ativada : → prevendo que o elemento fique menor que mínimo

2.3) VH em CSS - elementor blog 🔍

- viewport height (VH) → proporciona você deixar a altura de um elemento de acordo com a altura da janela do browser
↳ possibilitando o ajuste do html dependendo do dispositivo.

height =
EX 100 VH → significa que o elemento vai preencher qualquer tela (seja o dispositivo).

2.4) justify-content - CSS - w3schools 🔍

- justify-content → a propriedade alinha containers flexíveis quando o elemento não usa todo o espaço disponível na horizontal

na
* se quiser alinhar vertical usar → align-items

• valores da propriedade

- 1) flex-start → item é posicionado no começo do container
- 2) flex-end → item é posicionado no fim do container
- 3) center → item é posicionado no centro do container
- 4) space-between → os itens terão um espaço entre eles
- 5) space-around → os itens terão espaço em todos os 4 lados deles.

6) space-between → os itens terão uma quantidade igual de espaço entre eles.

2.5) A diferença de id e class no CSS

→ **id** é um identificador único para qualquer elemento

→ **class** é um identificador para qualquer elemento, que pode se repetir.

2.6) word-wrap - css - w3schools


↳ essa propriedade deixa palavras grandes sejam "quebradas" e colocadas na outra linha

word-wrap: break-word;

↳ deixa palavras "incabíveis" ir para a próxima linha

2.7) CURSOR - developer.mozilla.org 🔍

A propriedade **cursor** coloca o curso do mouse de algum modo quando o cursor esta sobre o elemento

EX `cursor: wait;` → 

2.8) :focus - css - developer.mozilla.org 🔍

:focus → a pseudo classe recebe um input e

elemento :focus {
..... }

faz algo quando tal input recebe "foco". Normalmente, o foco acontece quando um elemento é clicado.

2.9) guid CSS - w3schools 🔍