CSS e Responsividade

CSS (Layout

e

DEVinHouse

Parcerias para desenvolver a sua carreira

SENAI

<LAB365>

AGENDA

- Layout Float e Position
- Responsividade

CSS Layout

Além de estilos, CSS também posiciona os elementos na tela.
Existem várias formas de criar Layouts com CSS:

- Float
- Position
- Flexbox
- Grid

CSS Layout - Float

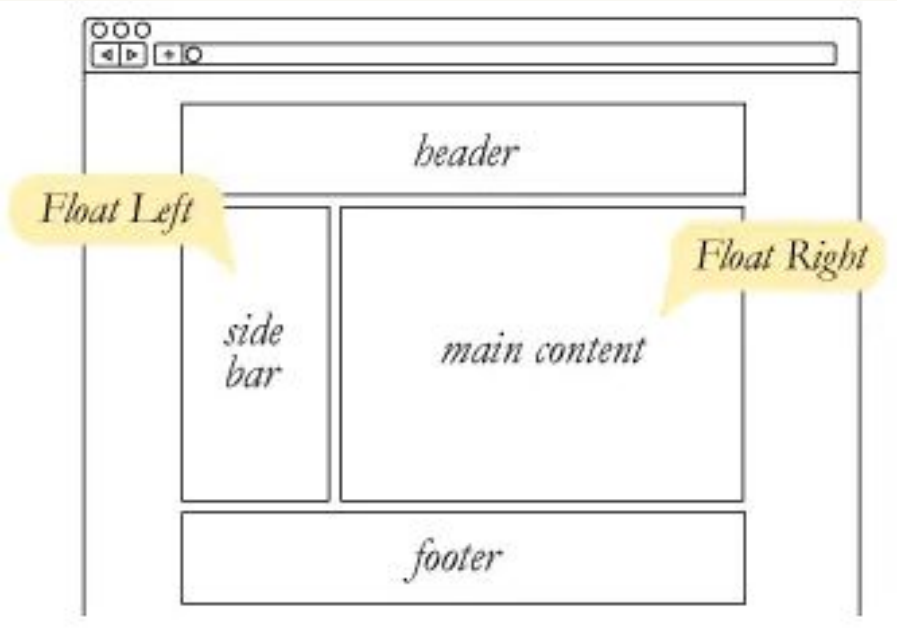
- Originalmente criado para simular layouts de jornais, onde textos "contornam" imagens.
- Retira o elemento de seu fluxo normal e coloca ao lado direito/esquerdo dentro do seu contêiner, ajustando textos e elementos inline ao seu redor

CSS Layout - Float

- Existem alguns valores para a propriedade float:

  left
  right
  none

- Também pode ser utilizado para criar layouts inteiros.



CSS Layout - Float

- https://www.w3schools.com/cssref/tryit.asp?filename=trycss_play_float

CSS Layout

- Ainda muito utilizado para Layouts, porém já existem ferramentas mais poderosas para criar Layouts CSS:

- Flexbox
- Grid

CSS Layout

- FLEXBOX - O layout flexível permite que os elementos responsivos dentro de um contêiner sejam organizados automaticamente, dependendo do tamanho da tela.

- GRID - O layout de grade CSS ou a grade CSS criam layouts de grade de design da Web complexos e responsivos com mais facilidade e consistência em todos os navegadores.

CSS Layout

- https://www.w3schools.com/css/css3_flexbox.asp
- https://www.w3schools.com/css/css_grid.asp

CSS Layout - Position

Remove elementos do fluxo normal do documento, posicionando de acordo com alguns valores:

- static: padrão, segue o fluxo normal.
- absolute: retira o elemento do fluxo normal e o posiciona em relação ao ancestral mais próximo. Define a distância para esse ancestral através dos atributos top, right, bottom e left.
- relative: mantém o elemento no fluxo normal, mas se utilizarmos os atributos top, right, bottom ou left, posiciona o elemento relativamente à sua posição original (que tem seu espaço preservado no documento).

CSS Layout - Position

- fixed: remove o elemento do fluxo normal e o posiciona relativamente ao contêiner inicial do documento (viewport), mantendo o elemento sempre visível, fixo nessa posição, mesmo com rolagem de tela.

- sticky: mantém o elemento no fluxo normal, preservando seu espaço no documento. Mas ao rolar a tela, o elemento "gruda" no viewport e se mantém sempre visível (respeitando as distâncias top/right/bottom/left).

CSS Layout - Position

- https://www.w3schools.com/css/tryit.asp?filename=trycss_position_static

Responsividade

- Termo cunhado por Ethan Marcotte em 2010
- Não é uma tecnologia separada
- Conjunto de práticas e técnicas que permite que páginas da Web alterem seu layout e aparência para se adequarem a diferentes tamanhos de dispositivos (smartphone, tablet, notebook)
- HTML e CSS são usados para, automaticamente: redimensionar, ocultar, reduzir, ampliar

Responsividade

- Fundamentado em três técnicas combinadas:
- grades fluidas, utilizando tamanhos relativos (%, em, vw, fr) e reposicionamento de elementos (float, flexbox, grid)
- imagens fluidas, utilizando max-width,
- <img srcset="" sizes=""> e <picture>
- media query, detectando tamanhos diferentes de tela e adaptando o CSS

Responsividade

- max-width: define uma largura máxima, mas não absoluta. Assim, a imagem será reduzida, se necessário, mas nunca excederá o tamanho máximo.
- Pode ser suficiente em alguns casos, mas não em todos.
- Conteúdo pode ficar pouco visível.
- Aparelhos menores carregando imagens muito grandes sem necessidade

Responsividade

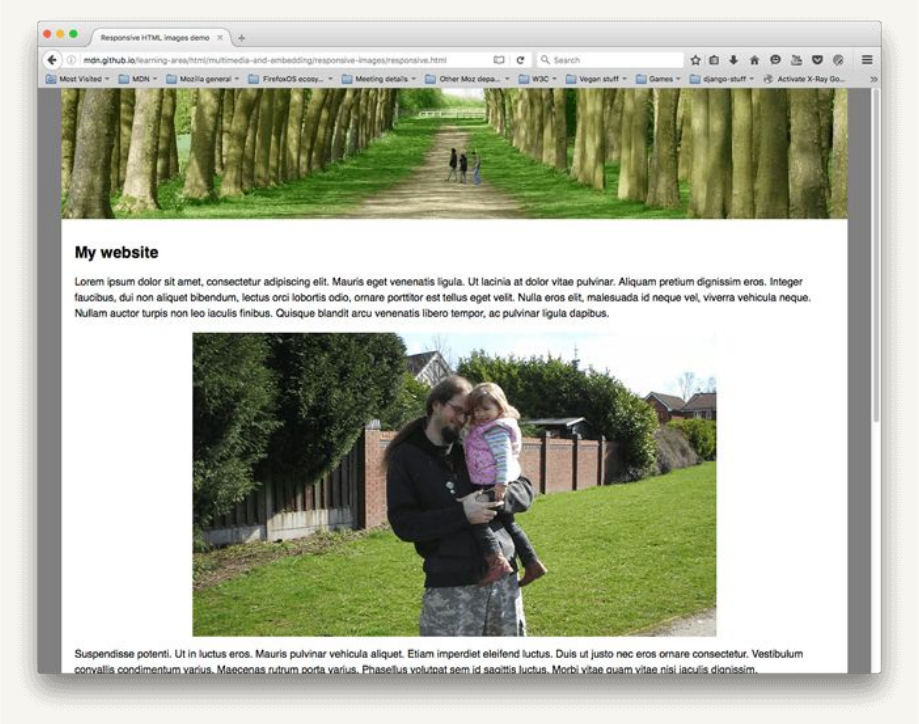
- srcset define o conjunto de imagens que nós iremos permitir ao navegador escolher, e qual tamanho cada imagem tem.
- A cada vírgula, escrevemos:
- nome do arquivo + espaço + largura da imagem em pixels.
- sizes define um conjunto de condições de mídia (ex.: largura da tela) e indica qual tamanho da imagem deveria ser a melhor escolha.
- A cada vírgula, escrevemos: condição de mídia + espaço + largura do espaço que a imagem vai preencher naquela condição.

Responsividade

```
<img src="elva-fairy-800w.jpg" alt="Elva dressed as a fairy">
```

```
<img
  srcset="elva-fairy-320w.jpg 320w,
          elva-fairy-480w.jpg 480w,
          elva-fairy-800w.jpg 800w"
  sizes="(max-width: 320px) 280px,
         (max-width: 480px) 440px,
         800px"
  src="elva-fairy-800w.jpg"
  alt="Elva dressed as a fairy"
>
```

Responsividade



Responsividade

- Relembrando: Defina sempre a tag meta sobre viewport!
- <meta name="viewport" content="width=device-width, initial-scale=1.0">
- Alguns navegadores mobile “mentem” sobre sua largura,
- e carregam as páginas com uma largura muito maior.
- A utilização dessa linha no HTML força esses navegadores
- a adotarem a sua largura real para carregarem as páginas.

Responsividade

- Relembrando: Defina sempre a tag meta sobre viewport!
- <meta name="viewport" content="width=device-width, initial-scale=1.0">
- Alguns navegadores mobile “mentem” sobre sua largura,
- e carregam as páginas com uma largura muito maior.
- A utilização dessa linha no HTML força esses navegadores
- a adotarem a sua largura real para carregarem as páginas.

Responsividade

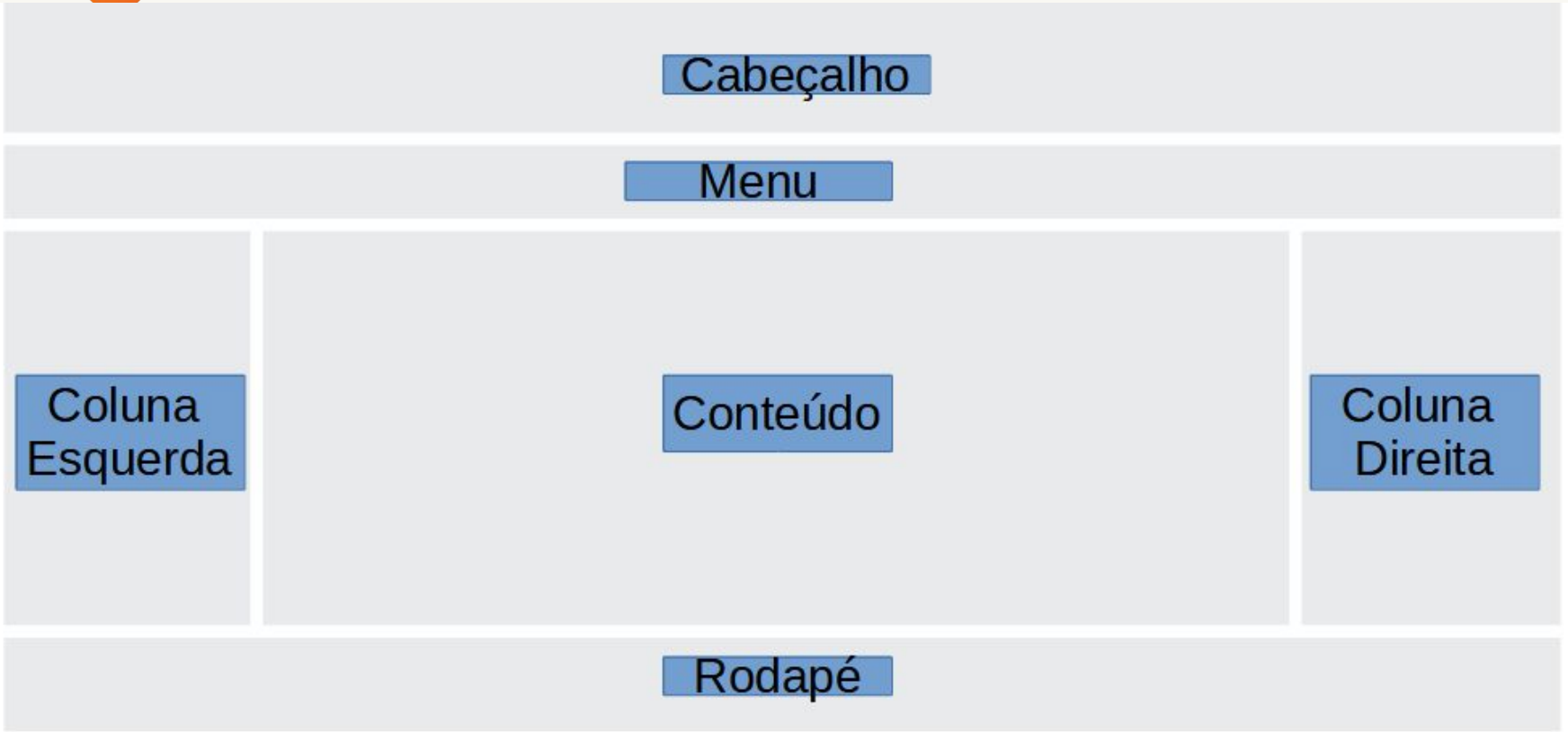
- Tipografia responsiva: assim como imagens, também podemos tratar responsivamente os tamanhos de fonte.
- A propriedade font-size de qualquer elemento de texto pode ser definida para diversos tamanhos de tela diferentes, utilizando media query.
- Também pode ser definida apenas uma vez utilizando,
- por exemplo, “vw”.

Responsividade

- Media Queries são úteis quando se deseja modificar seu site/app de acordo com o tipo (all, print, screen) ou alguma característica específica (por exemplo, largura da tela).

- Se os parâmetros definidos na query forem verdadeiros, o navegador aplica os estilos definidos.

Mão na massa!


Cabeçalho
Menu
Coluna Esquerda
Conteúdo
Coluna Direita
Rodapé

Responsividade

- How Floats and Clears work | https://youtu.be/LrdkRMZhgZg
- Entendendo sobre position no CSS | https://youtu.be/Y7NeqpwLM2g
- Learn CSS Position In 9 Minutes | https://youtu.be/jx5jmI0UlXU
- srcset and sizes attributes | https://youtu.be/2QYpkrX2N48
- Como utilizar Media Queries para sites Responsivos | https://youtu.be/AltqAPZzAqo
- Learn CSS Media Query In 7 Minutes | https://youtu.be/yU7jJ3NbPdA

DEVinHouse

Parcerias para desenvolver a sua carreira

OBRIGADO!

SENAI

<LAB365>