**HTML, CSS, JAVASCRIPT**

**<html> estrutura**

**<head> cabecalho onde sera linkado as coisa**

**<div class=””> cria uma campo div**

**<div id=””> cria um campo id**

**. -> representa div**

**# -> representa id**

**<img src=”image.png”> coloca imagem**

**CSS Responsivo**

**Img { max-width: 100% } //coloca largura total da tela.**

**@media screen and (min-width: 800px) { //se tonra responsivo conforme o tamanho atribuido.**

**.campoSelecionado {**

**Display: table; //coloca o display em forma de tabela.**

**Table-layout: fixed; //a largura das linhas serao divididas igualmente na tabela.**

**}**

**. campoSelecionado , .campoSelecionado {**

**Display: table-cell; //a ordem das colunas muda.**

**Vertical-align: middle //definimos a posicao que ficara na tabela.topo = top,meio = middle, rodape=bottom**

**Text-align: center; //centralizamos o texto no meio da tabela, left = esquerda, rigth = direita**

**}**

**.campoSelecionado {**

**Padding: 5rem; //5rem equivale a 50px, padding espacamento de dentro do componente.**

**Font-size: 2.5rem //rem equivale a 25px aumenta letra**

**}**

**}**

**CSS RESONSIVO**

**Body {**

**Background-color: black;**

**}**

**@media screen and (min-width: 320px) { // media responsive 320**

**Body {**

**Background-color: orange;**

**}**

**}**

**@media screen and (min-width: 550px) {// media responsive 550**

**Body {**

**Background-color: orange;**

**}**

**}**

**@media screen and (min-width: 768px) {// media responsive 768**

**Body {**

**Background-color: orange;**

**}**

**}**

**@media screen and (min-width: 968px) {// media responsive 968**

**Body {**

**Background-color: orange;**

**}**

**}**

**Margin: 0 auto //**margens automáticas na esquerda e na direita para centralizar o bloco horizontalmente na tela.

**Display: inline-block** // deixa alinhado da esquerda para a direita.

**Text-transform: uppercase;** //coloca os texto em letra maiscula.

**Display: flex** //deixa alinhado da esquerda para a direita

**Align-item: center** // alinha os componente no centro

**Justify-content: center** // centraliza no meio do container

**Justify-content: space-between** // iguala espaca conforme tamanho da tela no container

**Text-decoration: none** // quando for link tira a selecao de link

**Flex-direction: row-reverse** //reverte o alinhamento

**Flex-direction: column;** //deixa em forma de coluna

**Display: inline-flex;** //coloca em forma de linha flexivels

**Align-self: Flex-end** // coloca no final da altura do container

**Align-self: streach**// prenche o container automaticamente esticado

**Box-shadow:** **20px 20px 20px** white // voce passa tamanho e cor

**Border: 20px solid red** //na borda passamos tamanho estilo e cor

**Border: 20px dashed red** //na borda passamos tamanho estilo tracado e cor

Quando definimos uma pagina como display: flex podemos usar o

**Flex: 1 0 auto;** //redimensionao tamanho de cada contianer.

**Order: -1** //primeiro container a ser chamado 1 ultimo 0 meio

Quando denifimos ordem em todos os componentes podemos trocar.

Ex: order: 1 order: 2 order: 3 order: 4

Ex: order: 2 order: 3 order: 1 order: 4

**Flex-flow** : **row wrap** //coloca os coteudo em linha com 2 coluna

**Outline:** 3px dashed red //borda de contorno, pontilhada cor

**Text-ident: 1rem** //espacamento estilo paragrafo

**GRID**

**Display: grid; //torna o elemento um grid container**

**Display: grid; //torna um grid container com comportamento de linha**

**Grid-gap: 10px //espacamento da grid**

**Grid-template-rows: 200px 200px //define o tamanho baseado em linha**

**Grid-template-columns: 200px 200px //define o tamanho baseado em coluna**

**Grid-row: 1 //posicao que ficara o objeto confome a div**

**Grid-column: 1 //posicao que ficara o objeto confome a div**

**Ex:**

**Div 1**

**Grid-row: 1 //posicao que ficara o objeto confome a div**

**Grid-column: 1 //posicao que ficara o objeto confome a div**

**Div 2**

**Grid-row: 1 //posicao que ficara o objeto confome a div**

**Grid-column: 2 //posicao que ficara o objeto confome a div**

**Div 3**

**Grid-row: 2 //posicao que ficara o objeto confome a div**

**Grid-column: 1 //posicao que ficara o objeto confome a div**

**Div 4**

**Grid-row: 2 //posicao que ficara o objeto confome a div**

**Grid-column: 2 //posicao que ficara o objeto confome a div**

**Grid-auto-rows: 200px //para todas as div dentro desse container tera o tamanho predefinido**

**Grid-auto-columns: 200px //para todas as div dentro desse container tera o tamanho predefinido**

**Gap: 30px //deixao as grid juntas**

**Grid-template-columns: repeat(4, 1fr) //repetir a grid 4 vezes o tamanho da tela ocupando 4 colunas na tela**

**Grid-auto-flow: row //coloca no flow de linha primeiro as linha**

**Sem aplicar 1234**

**Ex: 12**

**34**

**Grid-auto-flow: column //coloca no flow de linha primeiro as linha**

**Sem aplicar 1234**

**Ex: 13**

**24**

**Grid-template-areas: ‘one two tree’ ‘for fice six’ ‘seven eight nine’ // define os padroes das areas**

**Ex: elemento1 elemento 2 elemento 3**

**Elemento4 elemento 5 elemento 6**

**Elemento7 elemento 8 elemento 9**

**elemento1 {**

**Grid-area: five // troca o valor elemento no lugar do five e five no um**

**}**

**Seletor CSS**

**Img[alt] {} //seleciona o alt da img**

**Img[alt=”roupas”] {}//seleciona alt=” roupas”**

**Ex:**

**<li data=”campo”>texto</li>**

**[data^=”campo”] { //stylo} //seletor so pega o primeiro componente**

**[data\*=”campo”] {//stylo} //pega todos os componentes**

**[data$=”campo”]{//stylo}//pega todos os componentes**

**.one+.item {} //soma os componente pega o que ta na 2 posicao**

**.item:nth-child(3){stylo} //pega somente o componente 3**

**.item:nth-child(3) ~ .iitem {//stylo} //pega os compoente do item 3 pra frente**

**.item:not(.not-item) {} //desceliona o stylo nele**

**.item:empty {display:none} //retira o que tinha no componente**

**Efeitos**

**Transition: box-shadow 1s //mostra a sombra apos um segundo**

**A:hover{ box-shadow: inset 0 -3px 0 #red} //chama o efeito em cor vermelha apos fica com o mouse em cima**

**Existe 4 parametros dentro de transition**

transition-property: all;

transition-duration: 1s;

transition-timing-function: ease;

transition-delay: 0s;

**porem podemos chamalo em uma so chamada**

**style{**

**transition: all 1s ease 0s;**

}

**Transform: translate(-50%, -50%) //tranforma o conteudo para a nova posicao.**

**Transform: translate(-50%, -50%) //tranforma o conteudo para a nova posicao.**

**Perspective: 400px; //muda a perspectiva do objeto 3d dando uma visaio mais longa ou proxima**

**Transition: transform 1s //transforma o objeto dentro de uma trnasicao de 1segundo**

**Transform-style: preserve-3d //transformando objeto em 3d**

**Compónente:hover: {**

**Box-shadow: 0px 0px 0px white;**

**}**

**Js**