Sites para escolher cores:

Coolors.co

Palleton

Adobe collors -> dá pra extrair cores de temas

Sugestões de cores para o front:

Interface gráfica do usuário

Descrição gerada automaticamente com confiança média  
Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Outras sugestões verdinho musgo: #373d20, #545A32, #717744

Para letras: 2a6c2e

Capturar cores de um site: adicionar extensão do google chrome -> Colorzilla

Configurações globais de CSS:

\* {

Heigth: 100%;

}

Gradiente:

Body {

Background-image: linear gradiente()

}

Sugestoes para fontes:   
titulo: serifada  
instruções: sem serifa, monespaçada talvez

* Placeholders: monoespaçada
* Verificar itálico na foto do celular
* Adicionar ícones nos botões
* Colocar o h2 numa div
* Escolher a fonte, formatar, alterar a cor, estilizar

SELETORES PERSONALIZADOS:

# = ID

. = class

: = pseudo classes

:: pseudo elementos (after, befor, first-letter, first-line…

> children

PSEUDO CLASSES

elemento:pseudoclasse {

}

Principais:

Hover, visited e active

Hover é quando passa o mouse em cima

TIPOS DE CAIXAS:

Box-level inline-level

Texto

Descrição gerada automaticamente com confiança média

Uma imagem contendo Interface gráfica do usuário

Descrição gerada automaticamente

LAYOUT DO DESAFIO 10:

[www.mockflow.com](http://www.mockflow.com)

Texto

Descrição gerada automaticamenteVARIÁVEIS EM CSS:

:root {

--cor02: #c5ebd6

--cor03

--cor04

}

Body {

Color-background: color: var(--cor01)

}

RESPONSIVIDADE PARA SITES:

O tamanho máximo de largura (width) é 1200, a partir disso, deve-se “travar” o conteúdo. Algo entre 1000 e 800 é o ideal.

O bang da responsavidade está em setar valores máximos e mínimos para largura, e deixar a margem ”auto”: (mas na real o media query é o ideal para responsividade).

*Max-width*

*Ex:max-width:300px;*

*(Tudo que estiver acima de 300px carregara a imagem grande,se diminuir ou chegar a 300 px, carrega a imagem pequena)*

*Min-width*

*Ex:min-width:300px;*

*(Tudo que estiver entre 1px e 299px carregara a imagem pequena,se ultrapassar os 300px carregara a imagem grande.)*

TRANSICAO INTERESSANTE PARA HOVER:

Para o hover depois ficar devagarinho

nav a {

    margin-left: 10px;

    margin-right: 15px;

    padding: 10px;

    border-radius: 5px;

    transition-duration: 0.6s;

    color: white;

    text-decoration: none;

    font-weight: bold;

}

nav > a:hover {

    background-color: #0bea6c;

    color: #0b4123;

}

ORGANIZAR LISTAS (UL/OL) EM COLUNAS:

aside ul {

   columns: 2;

}

PROPRIEDADES DE ESTILO PARA LISTAS:

Segue algumas Propriedades de Estilo extras apresentadas no vídeo para usarmos dentro de seletores de listas ( ex.: ul {...} ):

BOLINHAS DENTRO DA DIV

Quando criamos uma lista dentro de outro elemento, dentro de um aside por exemplo, geralmente as "bolinhas" da lista (marcadores sequenciais no caso de listas ordenadas) irão ficar para fora do elemento pai (aside)... Para resolver isso e manter a lista inteira dentro do elemento pai, devemos utilizar a Propriedade de Estilo " list-style-position " com o valor " inside " (por padrão marcadores de listas são posicionados com o valor " outside "). Essa propriedade define a posição dos marcadores da lista e dessa forma as "bolinhas" ficarão para dentro do elemento pai.

QUEBRAR A LISTA EM COLUNAS

Outra questão, às vezes temos uma lista com muitos itens, muito extensa... E queremos dividi-la em duas partes para manter uma formatação mais agradável... Para fazer isso, temos a Propriedade de Estilo " columns " com o valor " 2 ". Com isso iremos dividir a nossa lista em duas colunas (se quiser dividir em mais de duas colunas é só alterar o valor numérico da propriedade).

TROCAR O TIPO DE BOLINHAS

É possível trocar as "bolinhas" ou marcadores de listas por outro símbolo, um emoji por exemplo. Para fazer isso, devemos usar a Propriedade de Estilo " list-style-type "... Como valor dessa propriedade, devemos colocar entre aspas 'simples' o número do código do emoji que escolhermos (sem o "U+"), e após isso, para de fato carregar o emoji, adicionar uma contrabarra ( \ ) antes do código do emoji e ainda dentro das aspas 'simples'. Obs.: A troca de símbolos de lista não é 100% compatível com todos os navegadores.

ESPAÇO ENTRE BOLINHA E ITENS

Uma coisa que pode acontecer é o nosso símbolo ficar "grudado" no item da lista, para evitar isso é indicado acrescentar um "espaço" entre o emoji e o item de lista. Para fazer isso, ainda dentro das aspas 'simples' e DEPOIS do código do emoji, acrescente outra contrabarra ( \ ) e logo após isso o código " 00A0 " ou " 0020 " (comando de espaço ). Se você quiser, pode inserir mais de um espaço, é só repetir o processo. Obs.: Sempre que você quiser dar um "espaço" ou inserir emojis pelo CSS, para qualquer outro elemento e não somente listas, faça como foi orientado nesse texto.

Exemplo geral:

ul {

list-style-position: inside;

columns: 2;

list-style-type: '\1F5F8\00A0\00A0';

}

Nota: Ainda existem muitas outras Propriedades de Estilo para listas além dessas três apresentadas no vídeo e descritas nesse texto.

MARGIN NEGATIVA:

No nosso projeto, dentro da área " Saiba Mais " (aside), formatamos o nosso h3 de um jeito que parece que o subtítulo faz parte do topo da caixa. Para fazer isso, inicialmente atribuímos um destaque de cores ao subtítulo e adicionamos um pequeno padding de 10px no mesmo...

O resultado desejado seria que a caixa do h3 crescesse para "fora" e ocupasse a exata forma do topo da caixa do aside, resultando no efeito descrito.

Para isso, configuramos as margins laterais e superior do nosso h3 com um valor negativo de -10px, dessa forma a caixa do h3 irá crescer para "fora" e ocupará a forma do topo da caixa aside. A margin (espaçamento externo) foi configurada com um valor negativo, na prática essa margin está indo para dentro e se expandindo de forma inversa, abrangendo o topo do aside. Uma mesma configuração com um valor positivo no margin não daria o mesmo efeito.

Para fechar, posteriormente é só fazer o arredondamento da parte de cima da caixa do h3, para ficar condizente com a parte inferior da nossa caixa aside.

VIDEO RESPONSIVO:

aside.video {

    background-color: #1a5c38;

    margin: 0 -20px 30px -20px;

    padding: 20px;

    padding-bottom: 56.5%;

    position: relative;

}

aside.video > iframe {

    position: absolute;

    top: 5%;

    left: 5%;

    width: 90%;

    height: 90%;

}

PROPRIEDADE DE ESTILO - POSITION:

O posicionamento padrão de todo elemento no HTML é o "estático". Todo elemento estático é posicionado no canto superior esquerdo no corpo do documento ou do elemento onde ele está contido. Esta é a posição 0 de um elemento no corpo do documento.

A Propriedade de Estilo " position " é utilizada para determinar o posicionamento e hierarquia dos elementos na página do documento.

PROPRIEDADES AUXILIARES DO POSITION:

A Propriedade de Estilo "position" vem ACOMPANHADA e é usada em conjunto com quatro outras propriedades auxiliares (são propriedades e NÃO valores para position), que são:

top - (topo)

bottom - (baixo)

right - (direita)

left - (esquerda)

Essas quatro propriedades irão determinar a localização final do objeto, irão deslocar o elemento conforme for especificado.

Por exemplo, se usamos a propriedade "top" com o valor de 20px em um elemento, o mesmo vai se deslocar do topo num total de 20px para baixo.

VALORES DA PROPRIEDADE POSITION:

Para a propriedade position, é possível atribuir 5 valores, que são: static, relative, absolute, fixed e sticky.

Funções dos valores mais usados:

static - (estático):

O valor padrão da propriedade position é o static. Se não determinarmos nada, é esse valor que vai prevalecer por "debaixo dos panos". Ele não indica posicionamento algum, ou seja, é o valor para "nenhum posicionamento específico" do elemento. A função dele é manter a hierarquia de posicionamento conforme já estabelecido no documento HTML.

Isso significa que aquelas quatro outras propriedades já citadas (top, bottom, left e right) não funcionarão, não irão alterar a posição do elemento se usadas juntamente com esse valor... Visto que o position já está como "static" por padrão.

relative - (relativo):

O valor relative posiciona um elemento em relação à sua posição "normal"... Ou seja, as quatro propriedades auxiliares (top, bottom, left e right) vão fazer com que o elemento seja ajustado A PARTIR DA SUA POSIÇÃO ORIGINAL.

Após ter definido o tipo de position (relative nesse caso), a posição do elemento é definida através das quatro propriedades auxiliares do position: top, bottom, left e right. Que indicam a distância com relação ao topo, base, esquerda e direita, respectivamente, tomando como REFERÊNCIA DE POSICIONAMENTO (no caso do valor relative) a posição original do elemento. (se não configurarmos valores de posicionamento com as propriedades auxiliares, o elemento simplesmente fica no mesmo lugar)

absolute - (absoluto):

Um elemento com o valor absolute para position vai ser disposto, com as propriedades auxiliares, de forma relativa ao seu elemento pai... DESDE QUE ESSE ELEMENTO PAI TENHA UM POSITION COM UM VALOR DIFERENTE DE "STATIC" DEFINIDO!

Se o elemento pai não ter um position com um valor diferente de "static" definido, o valor absolute faz com que o elemento configurado " saia " da hierarquia estabelecida pelo HTML.

Um elemento com o position absolute é disposto em relação ao elemento de nível superior mais próximo dele que possui um position com um valor diferente de "static" definido... E como já foi dito, caso não exista nenhum elemento hierarquicamente de nível superior (pai ou "avô") que tenha um position sem ser o "static" definido, o position absolute será absoluto ao corpo da página (body).

Na prática, em elementos do tipo filho com o pai sem a propriedade position, o valor absolute "transforma" filhos de elementos em FILHOS DA PÁGINA (body). Assim, é possível posicionarmos um elemento filho (transformado) em qualquer lugar da página (fora do pai), independentemente do que temos lá. Isso pode ser útil em casos que precisamos posicionar um alerta em nossa página, por exemplo, e não queremos "quebrar" o layout. ---

Dessa forma, note que o position absolute possui dois comportamentos diferentes, que se manifestam dependendo da configuração do elemento pai ou elemento de nível superior.

Nota: Perceba que o Guanabara configurou o filho da div (iframe) com a declaração "position: absolute" e o pai (a div) com a declaração "position: relative"... Dessa forma o elemento filho (iframe) é posicionado, com as propriedades auxiliares, em relação ao seu pai (div).

COLOCANDO IMAGENS DE FUNDO

Observe a fonte utilizada URL

div#q3 {

            background-image: url(imagens/pattern001.png);

        }

Repetição de imagens de fundo:

Propriedade: background-repeat

Valores: repeat (que é o padrão), no repeat (aparece só uma imagem), repeat-x (repete em linha, repeat-y (repete em coluna.

body {

            background-image: url(https://www.acif.org.br/wp-content/uploads/2020/08/vector-abstract-beautiful-sunshine-gray-color-gradient-background.jpg);

            background-size: 100px 100px; /\* muda o tamanho da imagem de fundo \*/

            background-repeat: repeat-x

        }

O Ponto de partida é sempre o canto superior esquerdo (mas dá para mudar, é o que segue ).

Propriedade background-position:

Define o ponto de ancoragem (posição da imagem) dentro da div ou de qualquer outra caixa.

Valores:

Tabela

Descrição gerada automaticamente

div#q1 {

            background-position: left top;

        }

        div#q2 {

            background-position: left center;

        }

        div#q3 {

            background-position: right bottom;

        }

        div#q4 {

            background-position: center center;

        }

OBSERVAÇÃO:

Para fazer ocupar a caixa o tamanho inteiro da tela, ou seja, quando a caixa fica achatada, setar a propriedade height para o valor 100vh (view height):

height: 98vh;

Propriedade background-size:

Como o próprio nome diz: define o tamanho da imagem de fundo do site.

Setar a propriedade width em 100%, porem, isso pode causar algumas tretas tais como distorção da imagem

background-size: 100% 100%; -> preenche a tela, mas achata e distorce a imagem

countain; -> a imagem é 100% exibida, com laterais vazias as vezes

cover; -> cobre a tela toda mesmo que tenha que cortar a imagem