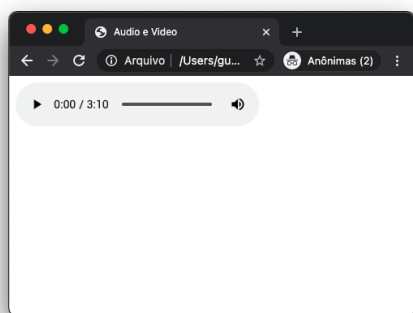


Atividade 1

Escreva um documento HTML que contenha um player que satisfaça as seguintes condições:

- a) O áudio deve estar em formato .mp3
- b) O player deve possuir controle e tocar a música em loop
- c) Deve mostrar uma mensagem ao usuário, caso o navegador não suporte áudio

Imagem de referência:



Link para *download* de exemplos de áudio: <https://www.learningcontainer.com/sample-audio-file/>

Atividade 2

Crie um documento HTML que contenha 2 vídeos, satisfazendo as seguintes condições:

- a) Um dos vídeos deve ser referência a um vídeo do Youtube
- b) Os vídeos devem estar lado a lado com margem de 20px entre eles
- c) Um dos vídeos deve estar em formato .mp4 (pode ser baixado no link disponibilizado a seguir).
- d) O vídeo deve possuir uma imagem quando não estiver sendo reproduzido

Link para *download* de exemplos de vídeo: <https://www.learningcontainer.com/mp4-sample-video-files-download/>

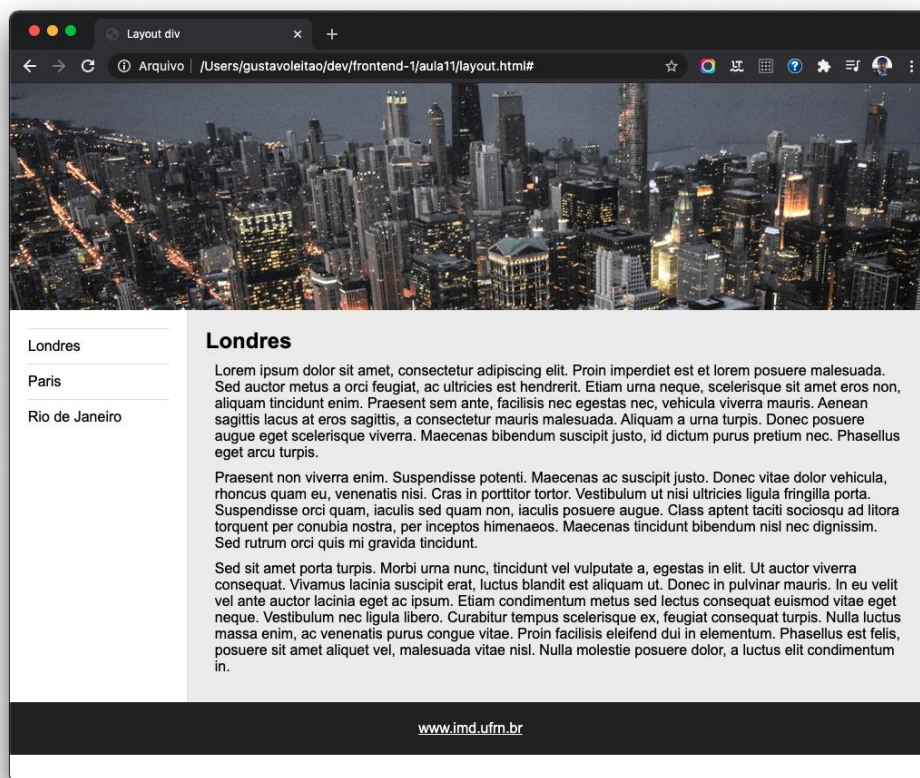


Atividade 1

Escreva um documento HTML e seu respectivo CSS, com a página inicial de um site que satisfaça as seguintes condições:

- d) Utilize a tag semântica <nav> para a navegação
- e) Utilize a tag semântica <footer> para rodapé
- f) Utilize a tag <article> para o conteúdo principal
- g) Utilize a tag <header> para o cabeçalho
- h) Utilize display flex para organizar o layout

Imagem de referência:



A imagem de referência é apenas um guia. Você pode criar de outra forma, desde que satisfaça os critérios postos. Use sua criatividade!

Questão 01

A utilização de Generics favorece a reutilização de componentes permitindo programas mais flexíveis. Cite 2 desvantagens de utilizar o tipo any em seu código.

Questão 02

Crie uma classe pessoa, com um atributo público nome que recebe um decorator validacao() de acordo com a figura abaixo:

```
1
2  class Pessoa{
3
4      @validacao()
5      public nome: string;
6  }
7
```

Crie o decorator validacao() que não permite nomes com mais de 10 caracteres, caso ocorra lance uma exceção.

Não esqueça de tratar a exceção com bloco try-catch.

Questão 01

A figura abaixo apresenta 4 funções que soma, multiplica, divide e subtrai dois valores passados como parâmetros.

```
1
2  function somar(num1:number, num2:number){
3      |   return num1 + num2;
4  }
5
6  function multiplicar(num1:number, num2:number){
7      |   return num1 * num2;
8  }
9
10 function dividir(num1:number, num2:number){
11     |   return num1 / num2;
12 }
13
14 function subtrair(num1:number, num2:number){
15     |   return num1 - num2;
16 }
17
18 let s = somar(6,3);
19 console.log(s)
20
21 s = multiplicar(6,3);
22 console.log(s)
23
24 s = dividir(6,3);
25 console.log(s)
26
27 s = subtrair(6,3);
28 console.log(s)
```

Organize este código utilizando um namespace chamado Calculadora, porém em outro arquivo. Altere o código onde consta as chamadas de funções de forma que funcione corretamente.