



inaC Merge pull request #2 from tryber/enhancement/add-resolution-resizin...



on 2 Mar

5

[View code](#)

README.md



Boas vindas ao repositório do projeto Carta Misteriosa!

Você já usa o GitHub diariamente para desenvolver os exercícios, certo? Agora, para desenvolver os projetos, você deverá seguir as instruções a seguir. Fique atento a cada passo e, se tiver qualquer dúvida, nos envie por *Slack*! #vqv 🚀

Aqui você vai encontrar os detalhes de como estruturar o desenvolvimento do seu projeto a partir deste repositório, utilizando uma branch específica e um *Pull Request* para colocar seus códigos.

O que deverá ser desenvolvido

Você deve desenvolver um site para que a pessoa que utilizá-lo, possa escrever uma frase e gerar uma carta que pareça ser retirada de várias fontes diferentes.

Abaixo você pode ver um exemplo de uma lista de tarefas. Utilize este exemplo como referência. Você poderá estilizar seu projeto da forma que desejar desde que todos os requisitos sejam cumpridos. Use sua imaginação!

Entregáveis

Para entregar o seu projeto você deverá criar um *Pull Request* para este repositório no **GitHub**.

Este *Pull Request* deverá conter, necessariamente, os arquivos `index.html`, `style.css` e `script.js`, que conterão seu código **HTML**, **CSS** e **JavaScript**, respectivamente. ⚠ É importante que seus arquivos tenham exatamente estes nomes! ⚠

Você pode adicionar outros arquivos se julgar necessário.

Requisitos do projeto

A seguir, estão listados todos os requisitos do projeto. Leia-os atentamente e siga à risca o que for pedido. Em particular, atente-se para os nomes de classes e ids que alguns elementos de seu projeto devem possuir. **Não troque ids por classes ou vice-versa**. O não cumprimento de um requisito, total ou parcialmente, impactará em sua avaliação.

Os requisitos do seu projeto são avaliados automaticamente, sendo utilizada a resolução `1366 x 768` (1366 pixels de largura por 768 pixels de altura). Logo, recomenda-se desenvolver seu projeto usando a mesma resolução, via instalação [deste plugin](#) do `Chrome` para facilitar a configuração dessa resolução.

Você tem liberdade para adicionar novos comportamentos ao seu projeto, seja na forma de aperfeiçoamentos em requisitos propostos ou novas funcionalidades, **desde que tais comportamentos adicionais não conflitem com os requisitos propostos**. Em outras palavras, você pode fazer mais do que foi pedido, mas nunca menos. Contudo, tenha em mente que **nada além do que for pedido nos requisitos será avaliado**. Esta é uma oportunidade de você exercitar sua criatividade e experimentar os conhecimentos adquiridos.

1 - Deve haver um `input` com o `id="carta-texto"` onde o usuário poderá digitar o conteúdo da carta.

2 - Deve haver um parágrafo com o `id="carta-gerada"` onde o usuário verá o resultado de sua carta misteriosa.

3 - Deve haver um botão com `id="criar-carta"` e ao clicar nesse botão, a carta misteriosa deve ser gerada.

Pontos importantes:

- * Cada palavra deve aparecer dentro de uma tag ``span``.
- * As tags ``span`` devem ser adicionadas como filha do parágrafo que possui o id ``carta-gerada``.

4 - Ao criar uma carta através do botão `id="criar-carta"`, o `input` com `id="carta-texto"` deve permanecer com o texto digitado.

5 - Crie a classe `newspaper` .

Pontos importantes:

- * Defina as propriedades:
 - ``background-color`` com o valor ``antiquewhite``
 - ``font-family`` com o valor ``Times New Roman``
 - ``font-weight`` com o valor ``bold``

6 - Crie a classe `magazine1` .

Pontos importantes:

- * Defina as propriedades:
 - ``background-color`` com o valor ``teal``
 - ``color`` com o valor ``white``
 - ``font-family`` com o valor ``Verdana``
 - ``font-weight`` com o valor ``900``
 - ``text-transform`` com o valor ``uppercase``

7 - Crie a classe **magazine2** .

Pontos importantes:

- * Defina as propriedades:
 - ``background-image`` com a imagem ``images/pink-pattern.png``
 - ``color`` com o valor ``fuchsia``
 - ``font-family`` com o valor ``Verdana``
 - ``font-weight`` com o valor ``900``

8 - Crie a classe **medium** .

Pontos importantes:

- * Defina as propriedades:
 - ``font-size`` com o valor ``20px``
 - ``padding`` com o valor ``8px``

9 - Crie a classe **big** .

Pontos importantes:

- * Defina as propriedades:
 - ``font-size`` com o valor ``30px``
 - ``padding`` com o valor ``10px``

10 - Crie a classe **reallybig** .

Pontos importantes:

- * Defina as propriedades:
 - ``font-size`` com o valor ``40px``
 - ``padding`` com o valor ``15px``

11 - Crie a classe `rotateleft` .

Pontos importantes:

- * Defina as propriedades:
 - ``transform`` com o valor ``rotate(-5deg)``

12 - Crie a classe `rotateright` .

Pontos importantes:

- * Defina as propriedades:
 - ``transform`` com o valor ``rotate(5deg)``

13 - Crie a classe `skewleft` .

Pontos importantes:

- * Defina as propriedades:
 - ``transform`` com o valor ``skewX(10deg)``;

14 - Crie a classe `skewright` .

Pontos importantes:

- * Defina as propriedades:
 - ``transform`` com o valor ``skewX(-10deg)``;

15 - Adicione as classes de forma aleatória afim de estilizar as palavras.

Pontos importantes:

- * As classes devem ser adicionadas às tags ``span`` de forma **`**aleatória**`**.
- * Sempre adicione mais de uma classe em uma palavra.
- * Utilize as classes:
 - ``newspaper``, ``magazine1``, ``magazine2`` (Grupo estilo)
 - ``medium``, ``big``, ``reallybig`` (Grupo tamanho)
 - ``rotateleft``, ``rotateright`` (Grupo rotação)
 - ``skewleft``, ``skewright`` (Grupo inclinação)

Não é necessário utilizar classes de todos os grupos em uma palavra, mas ****não**** utilize mais de uma classe do mesmo grupo.

Ou seja, se você utilizar as classes ``magazine1`` e ``skewright`` em uma palavra, as classes ``newspaper``, ``magazine2``, ``skeleft`` não devem ser usadas para essa mesma palavra.

BÔNUS

1 - Com uma carta misteriosa gerada, adicione a possibilidade de alterar o estilo de uma palavra específica ao clicar nela.

Pontos importantes:

- * Ao clicar em uma palavra, um novo estilo ****aleatório**** deve ser aplicado.
- * O número de mudanças deve ser ilimitado;

2 - Deve haver um parágrafo com o `id="carta-contador"` onde existirá um contador de palavras.

Pontos importantes:

- * Esse contador deve informar a quantidade de palavras presentes na carta misteriosa gerada.

Instruções para entregar seu projeto:

ANTES DE COMEÇAR A DESENVOLVER:

1. Clone o repositório



- `git clone https://github.com/tryber/sd-03-block5-project-mystery-letter.git`.

2. Crie uma branch a partir da branch `master`

- Verifique que você está na branch `master`
 - Exemplo: `git branch`
- Se não estiver, mude para a branch `master`

- Exemplo: `git checkout master`
- Agora, crie uma branch onde você vai guardar os `commits` do seu projeto
 - Você deve criar uma branch no seguinte formato: `nome-de-usuario-nome-do-projeto`
 - Exemplo: `git checkout -b exemplo-todo-list-project`
- 3. Adicione as mudanças necessárias para iniciar a implementação do seu projeto ao *stage* do Git e faça um `commit`
 - Verifique que as mudanças ainda não estão no *stage*
 - Exemplo: `git status` (devem aparecer listados os arquivos modificados em vermelho)
 - Adicione o novo arquivo ao *stage* do Git
 - Exemplo:
 - `git add .` (adicionando todas as mudanças - *que estavam em vermelho* - ao stage do Git)
 - `git status` (devem aparecer listados os arquivos modificados em verde)
 - Faça o `commit` inicial
 - Exemplo:
 - `git commit -m 'iniciando o projeto. VAMOS COM TUDO :rocket:'` (fazendo o primeiro commit)
 - `git status` (deve aparecer uma mensagem tipo *nothing to commit*)
- 4. Adicione a sua branch com o novo `commit` ao repositório remoto
 - Usando o exemplo anterior: `git push -u origin exemplo-todo-list-project`
- 5. Crie um novo `Pull Request` (*PR*)
 - Vá até a página de *Pull Requests* do [repositório no GitHub](#)
 - Clique no botão verde "*New pull request*"
 - Clique na caixa de seleção "*Compare*" e escolha a sua branch **com atenção**
 - Clique no botão verde "*Create pull request*"
 - Adicione uma descrição para o *Pull Request* e clique no botão verde "*Create pull request*"
 - **Não se preocupe em preencher mais nada por enquanto!**
 - Volte até a [página de Pull Requests do repositório](#) e confira que o seu *Pull Request* está criado

DURANTE O DESENVOLVIMENTO

-  **LEMBRE-SE DE CRIAR TODOS OS ARQUIVOS DENTRO DA PASTA COM O SEU NOME** 
- Faça `commits` das alterações que você fizer no código regularmente
- Lembre-se de sempre após um (ou alguns) `commits` atualizar o repositório remoto
- Os comandos que você utilizará com mais frequência são:
 - i. `git status` (para verificar o que está em vermelho - fora do stage - e o que está em verde - no stage)
 - ii. `git add` (para adicionar arquivos ao stage do Git)
 - iii. `git commit` (para criar um commit com os arquivos que estão no stage do Git)
 - iv. `git push -u nome-da-branch` (para enviar o commit para o repositório remoto na primeira vez que fizer o `push` de uma nova branch)
 - v. `git push` (para enviar o commit para o repositório remoto após o passo anterior)

DEPOIS DE TERMINAR O DESENVOLVIMENTO

Para "**entregar**" seu projeto, siga os passos a seguir:

- Vá até a página **DO SEU Pull Request**, adicione a label de "*code-review*" e marque seus colegas
 - No menu à direita, clique no *link* "**Labels**" e escolha a *label* **code-review**
 - No menu à direita, clique no *link* "**Assignees**" e escolha **o seu usuário**
 - No menu à direita, clique no *link* "**Reviewers**" e digite `students`, selecione o time `tryber/students-sd-03`

Se ainda houver alguma dúvida sobre como entregar seu projeto, [aqui tem um video explicativo](#).

REVISANDO UM PULL REQUEST



À medida que você e os outros estudantes forem entregando os projetos, vocês serão alertados **via Slack** para também fazer a revisão dos *Pull Requests* dos seus colegas. Fiquem atentos às mensagens do "*Pull Reminders*" no *Slack*!

Use o material que você já viu sobre [Code Review](#) para te ajudar a revisar os projetos que chegaram para você.

Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

Contributors 2



jeanpsv Jean Paulo Silva Vasconcelos



inaC