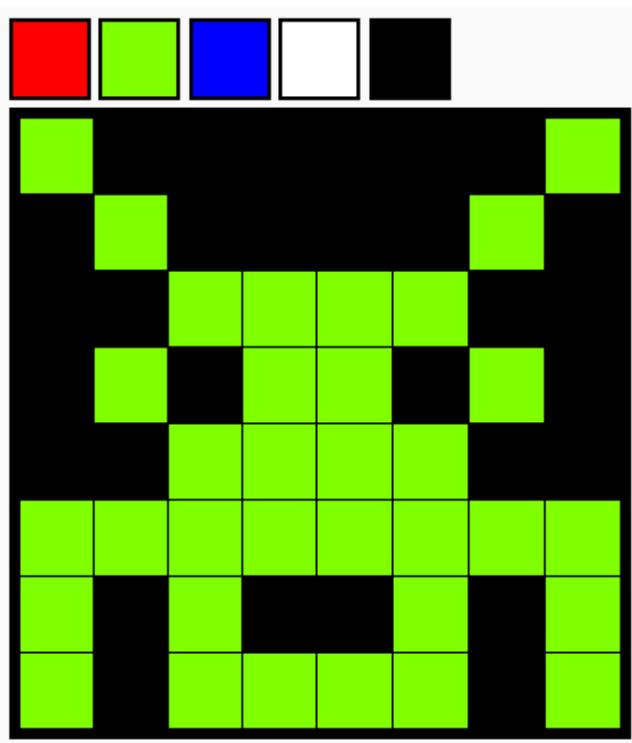


All Code Clubs must be registered. By registering your club we can measure our impact, and we can continue to provide free resources that help children learn to code. You can register your club at codeclubworld.org.

Introdução

Neste projeto, você vai criar um editor de arte em pixels. Além de usar conceitos de HTML e CSS, você vai aprender a usar JavaScript para adicionar interatividade ao seu projeto.



Como usar o editor: Clique em uma cor da paleta para escolher a cor de sua caneta e então clique nos pixels para alterar sua cor.



Lista de atividade

Siga estas **INSTRUÇÕES** uma a uma



Teste seu projeto



Salve seu projeto

Clique na bandeira verde para TESTAR

Certifique-se de SALVAR seu trabalho

Etapa 1: Criando uma malha quadriculada de pixels

Vamos criar uma malha quadriculada de pixels que pode ser usada para criar arte em pixels. O CSS fornece estilos de tabela para malha quadriculada e layouts de tabelas.

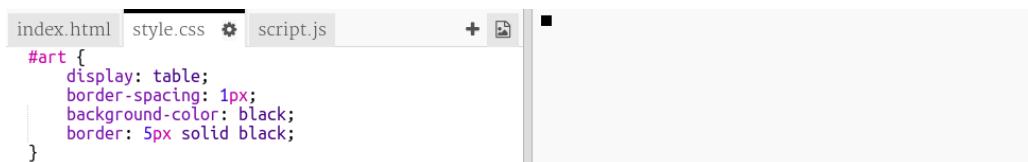
As tabelas contêm linhas que contêm células. Você vai criar uma tabela com fundo preto e então colocar pixels brancos nela.

Lista de atividades

- Abra este link de trinket: jumpto.cc/web-pixel. Se você está lendo isso on-line, você também pode usar o link incorporado abaixo.
- Adicione o seguinte código html ao corpo `<body>` do seu arquivo `index.html` para criar uma `<div>` que serve como um recipiente para sua arte em pixels e atribua a ela um id `art` para que você possa estilizá-la:

```
</head>
<body>
<div id="art"></div>
</body>
</html>
```

Agora, vá para o seu arquivo `style.css` e adicione o estilo de tabela para a `<div>` da arte. Observe que as três linhas de pixels são as mesmas. Digite a primeira e então copie e cole para criar as outras.



```
#art {
    display: table;
    border-spacing: 1px;
    background-color: black;
    border: 5px solid black;
}
```

Isso cria uma tabela com uma borda e define o espaçamento

3

dentro da malha quadriculada.

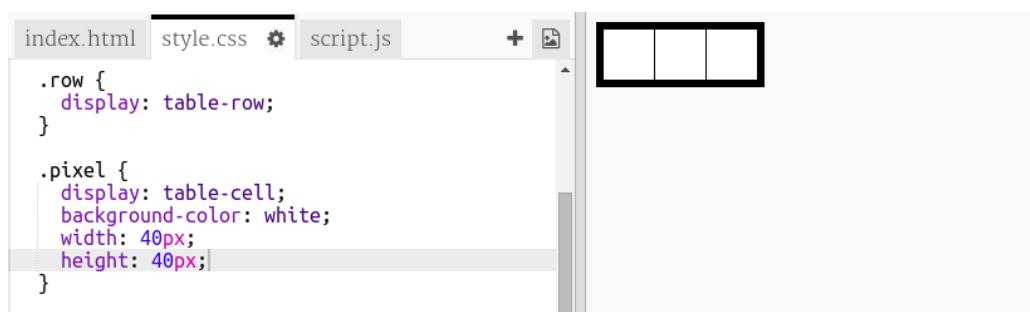
Ela ainda não parece muito interessante, você precisa colocar linhas de pixels dentro dela.

- Agora, retorne ao seu arquivo `index.html` e adicione uma linha de 3 pixels dentro da `<div>` da arte:

```
<div id="art">
  <div class="row">
    <div class="pixel"></div>
    <div class="pixel"></div>
    <div class="pixel"></div>
  </div>
</div>
```

Dessa vez, você vai usar classes para estilizar as divs, já que teremos várias delas.

Adicione o seguinte estilo para as linhas e células:



The screenshot shows a code editor with three tabs: index.html, style.css, and script.js. The style.css tab is active, displaying the following CSS code:

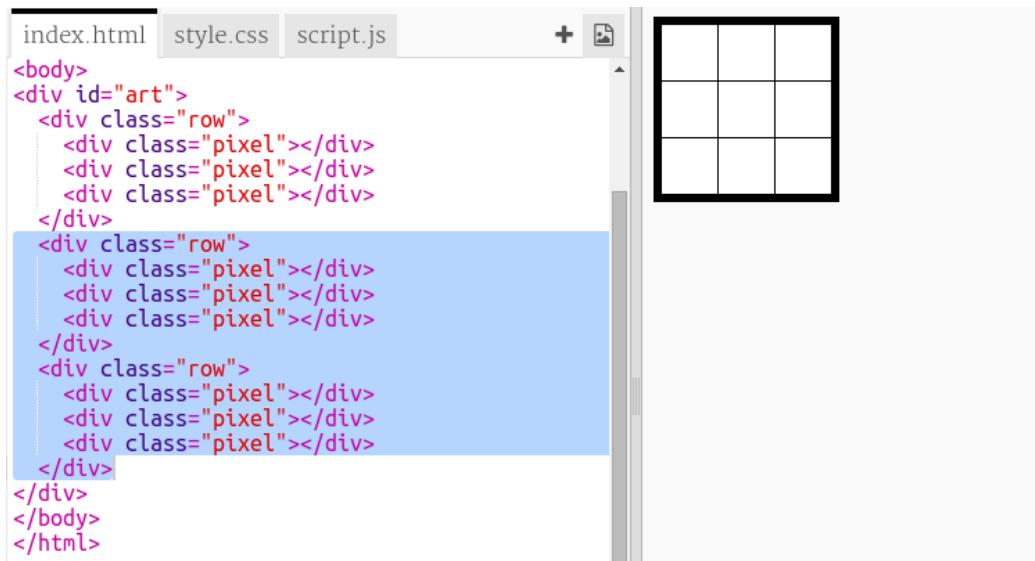
```
.row {
  display: table-row;
}

.pixel {
  display: table-cell;
  background-color: white;
  width: 40px;
  height: 40px;
```

To the right of the code editor, there is a preview window showing a 3x3 grid of white squares with black borders, representing the styled pixels.

Agora, seus pixels vão se alinhar em uma malha quadriculada com linhas pretas ao redor deles.

- Agora, adicione outras duas linhas de pixels para criar uma malha quadriculada com dimensões 3 x 3. Lembre-se de copiar e colar para ganhar tempo.



The screenshot shows a code editor with three tabs: index.html, style.css, and script.js. The index.html tab is active and contains the following HTML code:

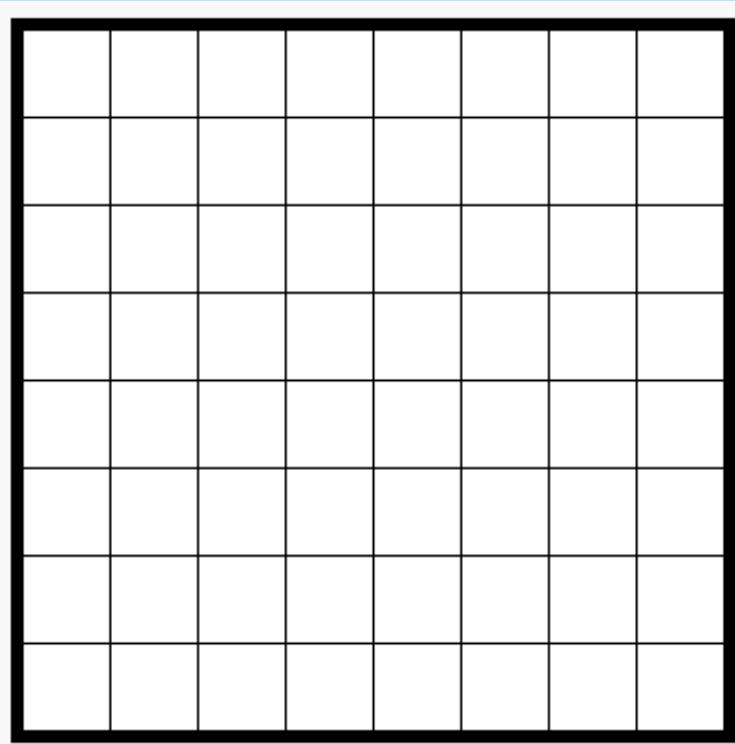
```
<body>
<div id="art">
  <div class="row">
    <div class="pixel"></div>
    <div class="pixel"></div>
    <div class="pixel"></div>
  </div>
  <div class="row">
    <div class="pixel"></div>
    <div class="pixel"></div>
    <div class="pixel"></div>
  </div>
  <div class="row">
    <div class="pixel"></div>
    <div class="pixel"></div>
    <div class="pixel"></div>
  </div>
</div>
</body>
</html>
```

To the right of the code editor is a preview window showing a 3x3 grid of white squares with black borders.

Desafio: Redimensione sua malha quadriculada

3x3 é uma malha quadriculada pequena para arte em pixels.
Você consegue fazer uma maior? 8x8 é um bom tamanho para arte em pixels.

Experimente copiar e colar ao invés de digitar tudo.





Etapa 2: Dando cor aos pixels

O HTML é usado para organizar seu conteúdo, enquanto o CSS serve para estilizá-lo. O JavaScript é uma linguagem de programação que pode ser usada para alterar uma página web conforme você interage com ela.

Você poderia usar HTML e CSS para definir a cor de fundo de pixels individuais, mas isso demoraria muito! Ao invés disso, você vai adicionar um código em JavaScript para colorir os pixels automaticamente quando clicar neles.

- No JavaScript, o código é colocado em uma função `function` que pode ser chamada quando queremos executar o código.

Você vai criar uma função chamada `setPixelColour`

A função `setPixelColour` precisa saber de qual pixel ela deve alterar a cor, isso é uma entrada `input`.

Adicione o seguinte código ao arquivo `script.js` para definir a cor de fundo de um pixel:

```
index.html style.css script.js
function setPixelColour(pixel)
{
    pixel.style.backgroundColor = 'black';
}
```

Observe que `backgroundColor` usa a pronúncia das cores em inglês americano.

- Agora, precisamos chamar a função quando um pixel for clicado.

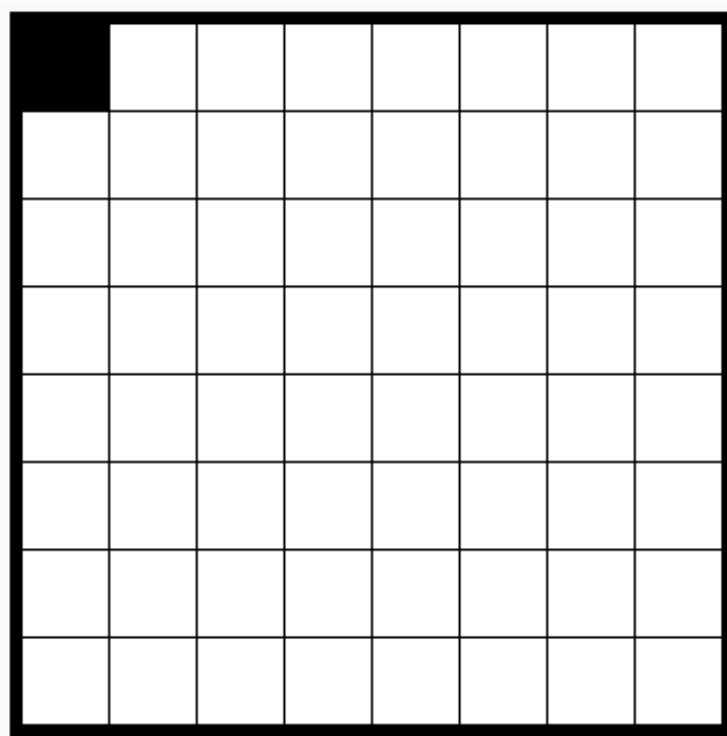
O HTML usa `onclick` para chamar uma função quando um elemento for

clicado. Você precisa passar ‘this’ como a entrada, assim sua função sabe de qual pixel ela deve alterar a cor.

Vá para `index.html` e adicione o seguinte código ao primeiro pixel:

```
<div id="art">
  <div class="row">
    <div class="pixel" onclick="setPixelColour(this)"></div>
    <div class="pixel"></div>
    <div class="pixel"></div>
```

Teste seu código clicando no primeiro pixel. Ele deve ficar preto:



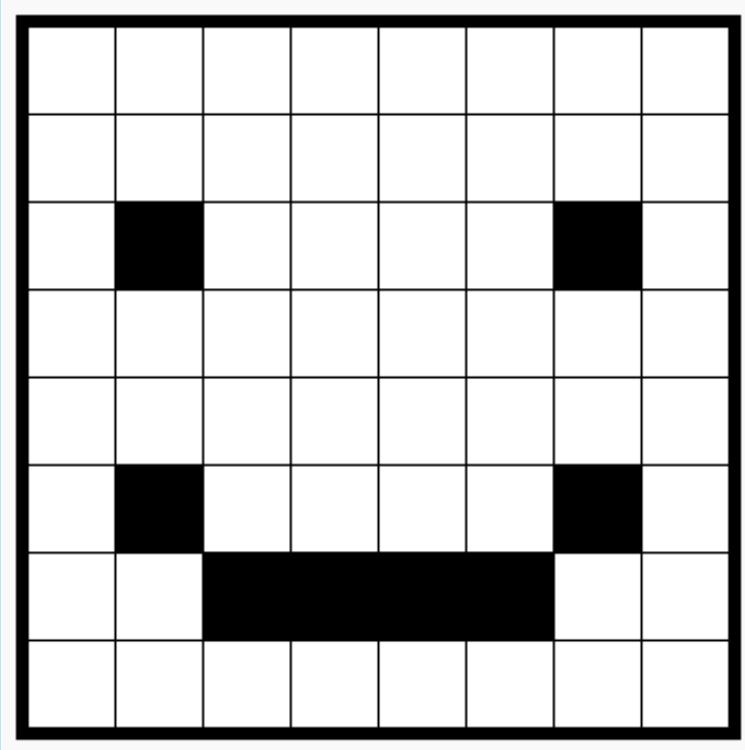
Você adicionou o código `onclick` apenas ao primeiro pixel, portanto ele ainda não funciona para os demais.

Desafio: Torne possível clicar em todos os pixels

Você consegue tornar possível clicar em todos os pixels?

Copie e cole para ganhar tempo.

Crie uma arte em pixels.



Dica: Você pode clicar em Autorun para limpar todos os pixels.

Etapa 3: Adicione uma paleta de cores

Você não achou chato não poder alterar novamente a cor de um pixel para branco caso tenha clicado errado? Vamos consertar isso criando uma paleta de cores, assim você pode clicar em uma cor para alterar a caneta.

- Primeiro, crie um estilo de caneta.

Adicione o código a seguir ao final do seu arquivo `style.css` :

```

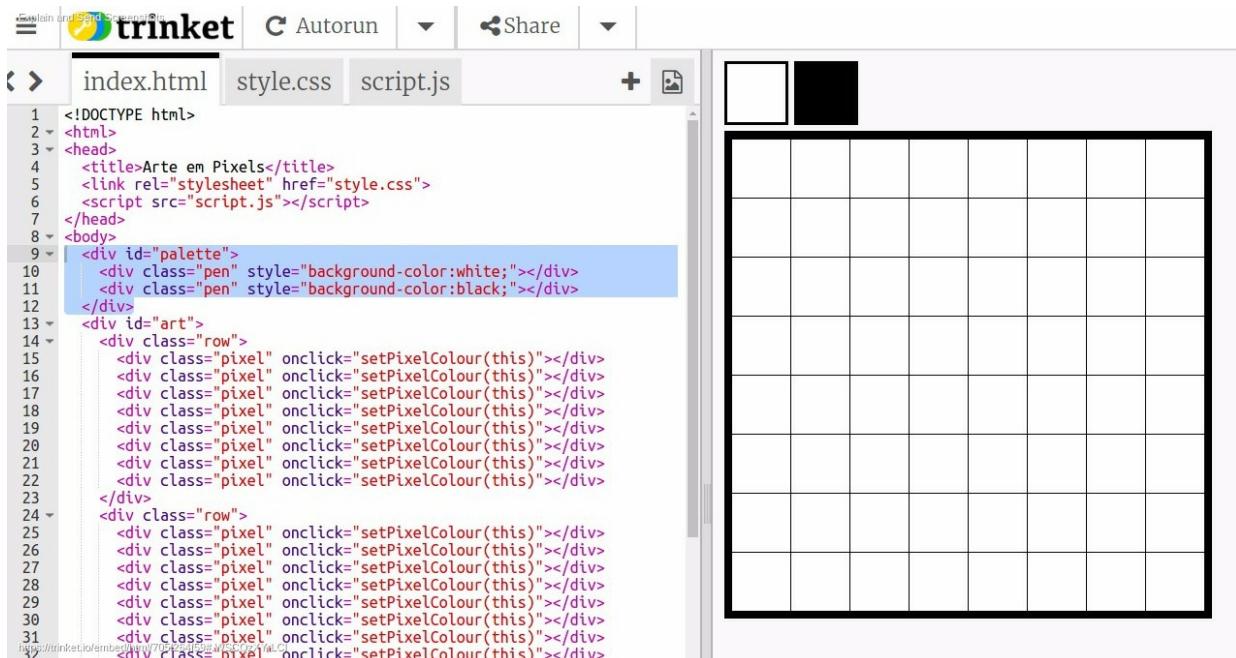
.pen {
    display: inline-block;
    width: 40px;
    height: 40px;
    border: 2px solid black;
}

```

- Agora, crie cores de caneta preta e branca que usam esse estilo.

Adicione o seguinte código ao seu arquivo `index.html` depois do corpo

`<body> :`



The screenshot shows the Trinket editor interface. On the left, the code editor displays the following HTML:

```

<!DOCTYPE html>
<html>
<head>
    <title>Arte em Pixels</title>
    <link rel="stylesheet" href="style.css">
    <script src="script.js"></script>
</head>
<body>
    <div id="palette">
        <div class="pen" style="background-color:white;"></div>
        <div class="pen" style="background-color:black;"></div>
    </div>
    <div id="art">
        <div class="row">
            <div class="pixel" onclick="setPixelColour(this)"></div>
            <div class="pixel" onclick="setPixelColour(this)"></div>
        </div>
        <div class="row">
            <div class="pixel" onclick="setPixelColour(this)"></div>
            <div class="pixel" onclick="setPixelColour(this)"></div>
        </div>
        <div class="row">
            <div class="pixel" onclick="setPixelColour(this)"></div>
            <div class="pixel" onclick="setPixelColour(this)"></div>
        </div>
        <div class="row">
            <div class="pixel" onclick="setPixelColour(this)"></div>
            <div class="pixel" onclick="setPixelColour(this)"></div>
        </div>
    </div>
</body>

```

The right side of the interface shows a 10x10 grid canvas where pixels can be set to white or black by clicking on them. Above the grid, there are two small preview boxes: one white and one black.

`style=` permite que você adicione CSS ao seu HTML, o que é conveniente aqui.

- Você quer poder alterar a cor da caneta quando uma cor da paleta for clicada.

As variáveis são usadas para armazenar informações. Vamos criar uma variável `penColour` em `script.js`.

Adicione o seguinte código no início do arquivo:

```
var penColour = 'black';
```

Então, adicione uma função para alterar a variável `penColour`:

```
var penColour = 'black';

function setPenColour(pen)
{
    penColour = pen;
}
```

- Você também vai precisar usar a cor da caneta quando alterar a cor de um pixel.

Altere a função `setPixelColour` para usar a variável `penColour` ao invés da cor preta `black`:

```
function setPixelColour(pixel)
{
    pixel.style.backgroundColor = penColour;
}
```

- Agora, você precisa chamar a função `setPenColour` que define a cor da caneta quando ela for clicada.

Adicione o código `onclick` em destaque às cores de sua caneta:

```
<body>
<div id="palette">
    <div class="pen" style="background-color:white;" onclick="setPenColour('white')"></div>
    <div class="pen" style="background-color:black;" onclick="setPenColour('black')"></div>
</div>
<div id="art">
```

- Agora, teste para ver se você pode alterar a cor da caneta entre preto e branco para preencher ou remover pixels.



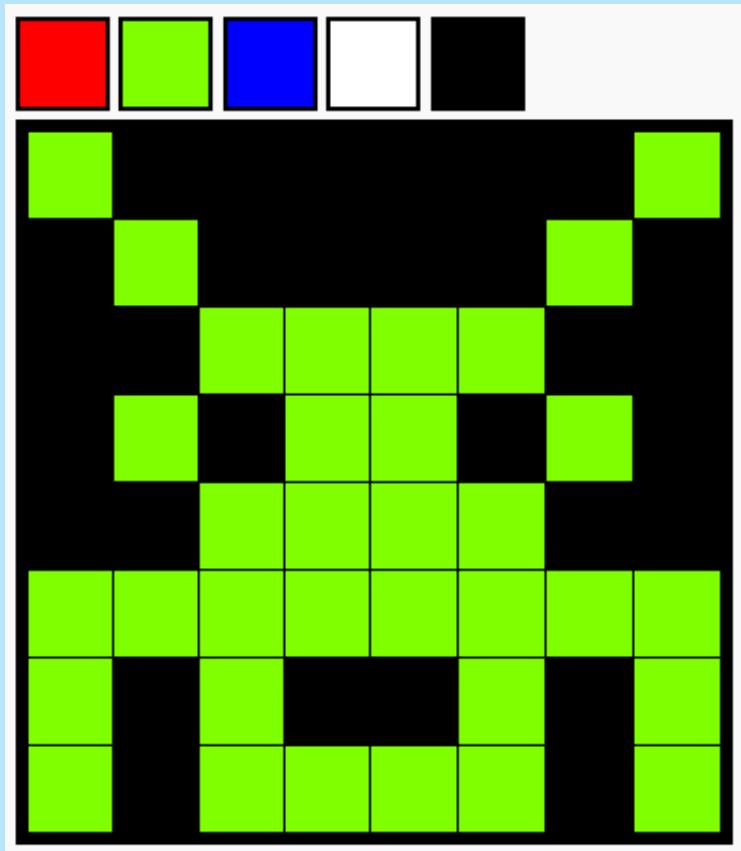
Save seu projeto

Desafio: Adicione mais colunas à paleta

Você consegue adicionar mais cores à paleta? Escolha as cores com as quais você quer criar arte em pixels.

Então, crie arte em pixels.

Dica: A cor verde brilhante se chama **chartreuse**.



Pergunte ao seu instrutor se você pode usar a ferramenta de captura de tela ou uma alternativa para salvar uma cópia da sua arte em pixels como uma imagem.



Save seu projeto