JSON

Front End Web

Msc. Lucas G. F. Alves
e-mail: lucas.g.f.alves@gmail.com

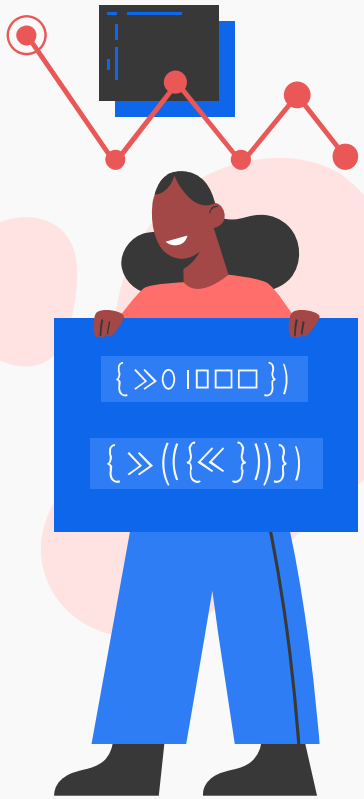
Senac

Planejamento de Aula

Revisão

JSON

Atividade



Revisão

Phaser

O que é?

HTML5 Game Framework, Ferramenta para desenvolver jogos com HTML5.

Utiliza linguagem Javascript ou Typescript.

Está na versão 3.8, foi a última release oficial, desde então a comunidade está fazendo updates.

Recebeu financiamento do mozilla.

Phaser

Funcionalidades (Features)

- Inputs, Canvas ou WebGL

- Gerenciamento Multimídia
  Images, Sprites, Tilemaps, Audio

- Mobile Works

- Handles physics
  Arcade, Ninja, P2.JS, Box2D, Chipmunk, Matter.JS

Phaser

Setting up

Servidores

- Webservers
- Cloud-hosted (Cloud9)
- Local-servers Apps (WAMP, XAMPP ou Cesanta)

- Live-preview do Brackets (NodeJS mascarado)
- NodeJS
- http-server (server local que tem no NPM)
- Express (framework de web applications)

Phaser

Como utilizar?

- phaser.js
- index.html
- new Game(width, height, renderer, HtmlParent, ...)

  Exemplo: const game = new Phaser.Game(config);
  - location
  - tipo de renderer
  - htmlElement (name ou o próprio elemento)
    - onde o canvas vai ser criado no html

Phaser

Estados

O game roda em estados.

Sempre tem um estado ativo.

Conforme o decorrer do tempo ou ações há a troca de estados.

Cada estados tem callbacks:

- preload
- create
- update
- ...

Phaser

Loading resources

- carrega-se tudo no preload dos states
  - Pode ser criado um estado que só carrega tudo antes do jogo começar

- game.load.tipo(nome, caminho, parametros adicionais);
  - game.load.spritesheet('button', 'assets/buttons/button_sprite_sheet.png', 193, 71);
  - this.load.image('background', 'assets/pics/bubble-on.png');
  - game.load.audio('boden', 'assets/audio/bodenstaendig_2000_in_rock_4bit.mp3');

Phaser

Sprite

São imagens que podem sofrer alterações de tamanhos e cortes para mostrar algo

- game.add.sprite(x,y,nomeSpr);
- sprite.tint
- sprite.alpha
- sprite.anchor
- Input
    - sprite.inputEnabled = true;
    - sprite.input.enableDrag(true);
- ...

Phaser

Sound Effects

- Carregar no Preload
  - game.load.audio('sword', 'assets/audio/SoundEffects/sword.mp3');
- Adicionar no jogo
  - sword = game.add.audio('sword');
- Tocar!
  - sword.play();
- Parar...
  - sword.stop();

Phaser

Inputs

- Chamar no manager
  - game.input.keyboard.isDown(Phaser.Keyboard.LEFT)

- Armazenar inputs em variáveis
  - cursors = game.input.keyboard.createCursorKeys();
  - cursors.left.isDown

Phaser

Física

- game.physics.startSystem(Phaser.Physics.ARCADE);
- sprite.enableBody = true;
- sprite.physicsBodyType = Phaser.Physics.ARCADE;
- sprite.body
    - acessar o body

Phaser

Camera

- Posição
  - game.camera.y -= 4;
  - Bounds setados pelo bounds do mundo
    - game.world.setBounds(-1000, -1000, 2000, 2000);
- Rotação
  - game.world.rotation += 0.05;
- flash(color, duration, force)
- fade(color, duration, force) efeito de luz na camera
- focusOn(displayObject)
  - move a camera para focar no objeto

Phaser

Como se começa?

TUTORIALS

https://phaser.io/tutorials/making-your-first-phaser-3-game

EXAMPLES

- https://phaser.io/examples

Phaser

Documentação!!!!!!!!

- Documentação do site
    - pode baixar
    - https://github.com/photonstorm/phaser/tree/v2.6.2/docs

- Phaser chains (mais fácil de procurar)

- https://phaser.io/learn/chains

Exercício

Criem um jogo e postem na página blog.html utilizando o framework Phaser.

Escolham qualquer temática de jogo.

Conta como atividade igual a calculadora.

JSON

JSON

JSON a sigla é derivada de JavaScript Object Notation.

Utilizado em interfaces baseadas em HTML para armazenar dados em memória

Leve para envio/recebimento de informações de serviços remotos.

Os dados definidos em JSON são definidos no mesmo formato que objetos JavaScript, portanto, é de fácil entendimento e manipulação.

Mais simples que um XML que possam prejudicar ou dificultar a leitura das propriedades.

Também não requer nenhum parser sofisticado para converter uma estrutura em variáveis JavaScript.

JSON

Para trabalhar com JSON, ao enviar dados, é necessário formatá-los em uma string no padrão da notação.

Na leitura/recebimento, é necessário converter a string no padrão da notação em uma estrutura JavaScript.

Por outro lado, em casos de dados estruturação por tag, como o XML, é necessário um parser mais sofisticado para converter a string numa estrutura DOM (Document Object Model), interpretar nodos da estrutura e ainda extrair os valores da estrutura para armazená-los em variáveis JavaScript, portanto, requer mais esforço.

JSON

// É comum aplicações JS invocarem serviços remotos e receber uma string JSON
// como resposta. JSON.parse converte esta string em objeto JavaScript.
var respostaServer = JSON.parse(responseText);

Outra vantagem da simplicidade de definição de objetos com JSON é quantidade de bytes utilizada é muito menor que um XML.

Arquivos JSON possuem extensão ".json" e o MIME type (tipo de mídia) definido para texto JSON é "application/json".

Em resumo é mais rápido, mais fácil e produtivo trabalhar com JSON.

JSON

Sintaxe e Definição de Objetos em JSON

São utilizadas estruturas tais como JavaScript, onde:

Dado é definido por um par atributos, nome e valor. Também conhecida como estrutura chave-valor.

O separador de dados é o caractere , (vírgula).

Chaves definem uma estrutura, um objeto, portanto, um conjunto de dados/propriedades.

Colchetes definem lógica de manipulação de coleção de objetos, arrays.

JSON

Definição de propriedades

Uma propriedade é composta de duas partes: a chave e o valor.

Exemplos:

Strings - “Olá mundo”;
Números - 1 ou 56.32;
Arrays - [1, 2, 3] ;
Objetos - {“nome”:”Fulano”};
Dados nulos - null;

Tem uma pequena diferença entre nomes JavaScript e JSON. No JSON é necessário colocar nome de atributo entre aspas-duplas.

JSON

Definição de Objetos

Objetos são estruturas que mantém um conjunto de propriedades sobre um mesmo indivíduo, ou simplesmente objeto. Por exemplo, de uma pessoa, podemos manter atributos de nome, idade e local de residência. Este conjunto de informações relacionadas são as propriedades de um objeto.

Para definir um objeto em JSON devemos fazer uso de caracteres de { } (chaves). As chaves que definem um objeto devem envolver as propriedades deste objeto.

JSON

Definição de Objetos

Exemplo:

```
{
    "nome":"Fulano",
    "idade": 30,
    "local":{
        "pais":"Brasil",
        "uf":"RS",
        "cidade":"Porto Alegre"
        }
}
```

JSON

Definição de Arrays de Objetos

A definição de um array é feita com os caracteres [ ].

```
"pessoas" : [
        {
            "nome":"Fulano",
            "idade": 30,
            "local":{
                "pais":"Brasil",
                "uf":"RS",
            }
        },
        {
            "nome":"Beltrano",
        }]
```

JSON

Exemplo de Javscript + JSON

```
// objeto JSON (um array) armazenado em objeto JavaScript
var pessoas =
// Lógica de array com []
[
    // definição de um objeto com {}
    {
      // proprieades ou atributos são chave-valor
      "nome":"Fulano",
      "idade": 30,
      // é possível associar um outro objeto como um valor de propriedade
      "local":{
         "pais":"Brasil",
         "uf":"RS",
         "cidade":"Porto Alegre"
      }
    },
    {
      "nome":"Beltrano",
      "idade": 32,
      "local":{
         "pais":"Brasil",
         "uf":"RS",
         "cidade":"Montenegro"
      }
    }
];
document.getElementById("demo").innerHTML =
"Primeiro registro do array contém dados do <b>\"" + pessoas[0].nome + "\"</b> de <b>" + pessoas[0].local.cidade + "</b>";
```

JSON

Leitura arquivos JSON

Para carregar um arquivo JSON é necessário utilizar a função fetch. Após é chamado a função response para verificar se o arquivo retornou algo. Se retornou é salvo o dado em uma variável que ao percorrê-la terá os dados do arquivo JSON.

```
Exemplo: fetch("./JSON/data.json")
            .then(response => response.json())
            .then(dados => {
                // Itera sobre os dados JSON
                for (const curso in dados) {
```

Exercício

Exercício

1) Criar um programa que leia um arquivo JSON contendo os Usuários e seus respectivos dados como nome, idade, cpf, telefone (pelo menos 5 usuários) e imprima os usuários cadastrados.

2) Criar um programa que leia um arquivo JSON contendo os alunos e seus respectivos cursos (pelo menos 5 cursos) e imprima os alunos matriculados em cada curso. Utilizem a função Fetch para ler o arquivo.

Exemplo JSON:

```
{
  "Matemática": [
    "João", "Maria", "Pedro"
  ],
  "Português": [
    "Fulano", "Beltrano", "Ciclano"
  ],
}
```

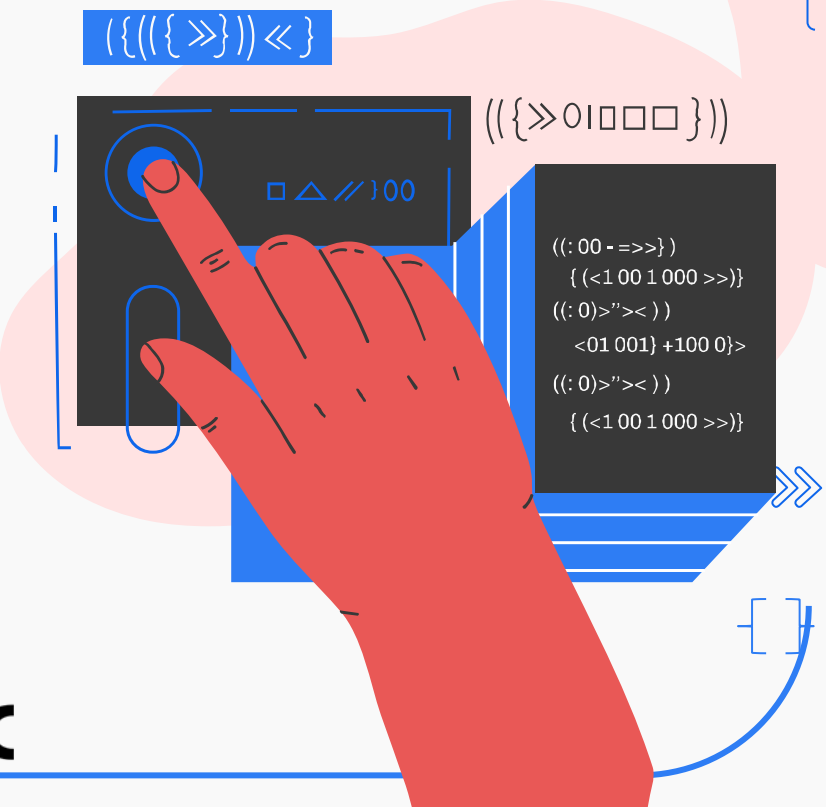
Professor

Lucas G. F. Alves

Senac

Obrigado!

E-mail :lucas.g.f.alves@gmail.com



Senac