

JAVASCRIPT ASSÍNCRONO

Fetch

Fetch API

Permite fazermos requisições HTTP através do método `fetch()`. Este método retorna a resolução de uma Promise. Podemos então utilizar o `then` para interagirmos com a resposta, que é um objeto do tipo `Response`.

```
fetch('./arquivo.txt').then(function(response) {  
  console.log(response); // Response HTTP (Objeto)  
});
```

Response

O objeto Response, possui um corpo com o conteúdo da resposta. Esse corpo pode ser transformado utilizando métodos do protótipo do objeto Response. Estes retornam outras promises.

```
fetch('./arquivo.txt').then(function(response) {  
  return response.text();  
}).then(function(corpo) {  
  console.log(corpo);  
});
```

Servidor Local

O fetch faz uma requisição HTTP/HTTPS. Se você iniciar um site local diretamente pelo index.html, sem um servidor local, o fetch não irá funcionar.

```
fetch('c:/files/arquivo.txt')
  .then((response) => {
    return response.text();
  })
  .then((corpo) => {
    console.log(corpo);
  }); // erro
```

Se dermos um espaço após o objeto ou pularmos linha, o método continua funcionando.

.json()

Um tipo de formato de dados muito utilizado com JavaScript é o **JSON** (JavaScript Object Notation), pelo fato dele possuir basicamente a mesma sintaxe que a de um objeto js. **.json()** transforma um corpo em json em um objeto JavaScript.

```
fetch('https://viacep.com.br/ws/01001000/json/')
  .then(response => response.json())
  .then(cep => {
    console.log(cep.bairro, cep.logradouro);
  });
```

.text()

Podemos utilizar o `.text()` para diferentes formatos como txt, json, html, css, js e mais.

```
const styleElement = document.createElement('style');

fetch('./style.css')
  .then(response => response.text())
  .then(style => {
    styleElement.innerHTML = style;
    document.body.appendChild(styleElement);
  });
```

HTML e .text()

Podemos pegar um arquivo inteiro em HTML, transformar o corpo em texto e inserir em uma div com o innerHTML. A partir daí podemos manipular esses dados como um DOM qualquer.

```
const div = document.createElement('div');

fetch('./sobre.html')
  .then(response => response.text())
  .then(htmlBody => {
    div.innerHTML = htmlBody;
    const titulo = div.querySelector('h1');
    document.querySelector('h1').innerText = titulo.innerText;
  });
```

.blob()

Um blob é um tipo de objeto utilizado para representação de dados de um arquivo. O blob pode ser utilizado para transformarmos requisições de imagens por exemplo. O blob gera um URL único.

```
const div = document.createElement('div');

fetch('./imagem.png')
  .then(response => response.blob())
  .then(imgBlob => {
    const blobUrl = URL.createObjectURL(imgBlob);
    console.log(blobUrl);
  });
```


.clone()

Ao utilizarmos os métodos acima, text, json e blob, a resposta é modificada. Por isso existe o método clone, caso você necessite transformar uma resposta em diferentes valores.

```
const div = document.createElement('div');

fetch('https://viacep.com.br/ws/01001000/json/')
  .then(response => {
    const cloneResponse = response.clone();
    response.json().then(json => {
      console.log(json)
    });
    cloneResponse.text().then(text => {
      console.log(text)
    });
  });
```

.headers

É uma propriedade que possui os cabeçalhos da requisição. É um tipo de dado iterável então podemos utilizar o `forEach` para vermos cada um deles.

```
const div = document.createElement('div');

fetch('https://viacep.com.br/ws/01001000/json/')
  .then(response => {
    response.headers.forEach(console.log);
  });
```

.status e .ok

Retorna o status da requisição. Se foi 404, 200, 202 e mais. ok retorna um valor booleano sendo true para uma requisição de sucesso e false para uma sem sucesso.

```
const div = document.createElement('div');

fetch('https://viacep.com.br/ws/01001000/json/')
  .then(response => {
    console.log(response.status, response.ok);
    if(response.status === 404) {
      console.log('Página não encontrada')
    }
  });
```

.url e .type

`.url` retorna o url da requisição. `.type` retorna o tipo da resposta.

```
const div = document.createElement('div');

fetch('https://viacep.com.br/ws/01001000/json/')
  .then(response => {
    console.log(response.type, response.url);
  });

//types
// basic: feito na mesma origem
// cors: feito em url body pode estar disponível
// error: erro de conexão
// opaque: no-cors, não permite acesso de outros sites
```

Exercícios

```
// Utilizando a API https://viacep.com.br/ws/${CEP}/json/  
// crie um formulário onde o usuário pode digitar o cep  
// e o endereço completo é retornado ao clicar em buscar
```

```
// Utilizando a API https://blockchain.info/ticker  
// retorne no DOM o valor de compra da bitcoin and reais.  
// atualize este valor a cada 30s
```

```
// Utilizando a API https://api.chucknorris.io/jokes/random  
// retorne uma piada randomica do chucknorris, toda vez que  
// clicar em próxima
```