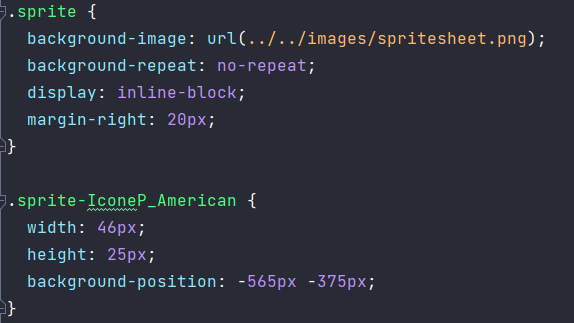
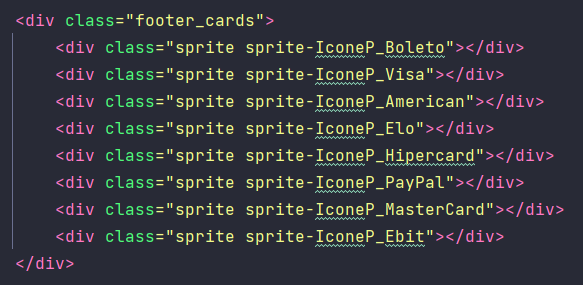
**4-)** A técnica de Sprite serve para você jogar várias imagens a ser utilizada dentro de 1 imagem unificadas.

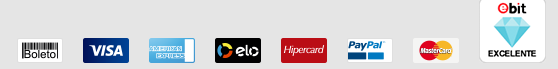
* Faz com várias imagens sejam chamadas por apenas 1 requisição, deixando o site com um **carregamento bem mais rápido**.

**Ex.:** Usei essa técnica no rodapé do exercício 2 na parte dos cartões aceitados.

**sprite.scss**



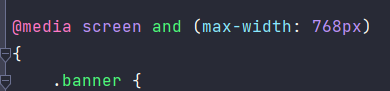
* Ao fazer essa chamada, é necessário criar um elemento HTML e criar um background com o fundo na imagem.



**5-)**  No **HTML**, é necessário informar a metatag viewport para que alguns aparelhos móveis não redimensione o layout por conta própria.

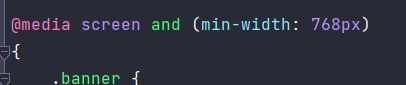


No CSS, o atributo responsável por fazer isso é o media queries



Execute esse código, apenas se a tela for no **máximo** de 768px. ( < 768px)

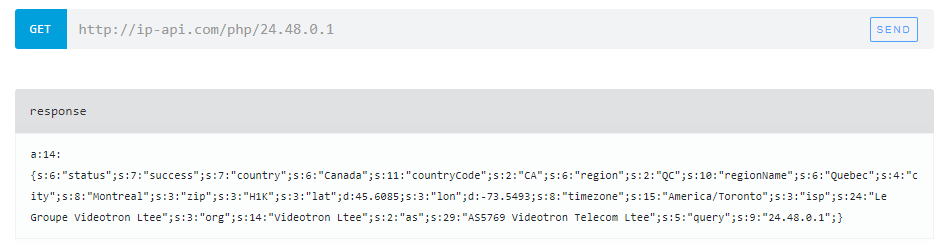
Execute esse código, apenas se a tela for no **mínimo** de 768px. ( > 768px)

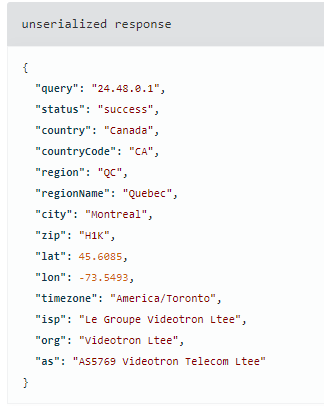


1. **)** A **API** significa (Interface de Programação de Aplicações) e ela é um conceito muito utilizado para que um sistema **front-end converse com o back-end.** Além disso, com uma API, é possível utiliza-la em vários tipos de aplicações front-end como: aplicativos moblie, desktop, web...

**Ex.:** O <https://ip-api.com/> é uma API que permite passar um endereço de IPv4 da máquina que está acessando determinado site e com esse dado, saber a localização da pessoa pelo método **GET**. Isso tudo é informado por URL e o resultado dessa API é em JSON.

A aplicação deve filtrar o JSON e pegar suas respectivas informação.





**8-)** Os verbos apresentam métodos de requisição, cada uma responsavél por uma ação.

**Verbo GET:** O mais simples de todos, solicita um recurso em específico e retorna determinados dados para a aplicação. Serve apenas para leitura.

**Verbo PUT:** utilizada quando criamos ou atualizamos uma informação de uma URL.

PUT /clients/robin

**Verbo DELETE:** ao contrário do PUT, ele deleta recurso identificado pela URL em uma determinada requisição.

curl -v -X DELETE /clients/silvio

**Verbo POST:** usado quando há repetições no processo de eeculção no servidor, elas devem execultar um processo no corpo sobre uma determinada URL.

POST /clients/

**9-)** O HTTP é um protocolo (conjunto de regras) que fornece um modelo de comunicação cliente-servidor entre o navegador e o servidor na internet. Eles devem usar o mesmo canal de comunicação.



**10-)** Imagine que você criasse um login para um sistema qualquer. Na requisição HTTP o pacote de dados a ser enviado para o servidor, vão com texto puro, ou seja, sem uma criptografia.

**Email =** [silvio\_cbsj@hotmail.com](mailto:silvio_cbsj@hotmail.com)

**Password =** silvio12345

Isso é perigoso porque para esse pacote chega até o servidor, ele passa por diversas portas de terceiros em seu caminho (modem, provedor...) e sendo visível para qualquer um.

Por isso foi criado o **HTTPS** que é uma junção do HTTP com uma camada a mais de segurança **(SSL/TLS).** Com isso é criado um pacote todo criptografado, onde sua chave pública fica em seu certificado SSL/TLS para o navegador e uma privada necessária para descriptografar esses dadas no servidor.



Para verificar se um site é seguro, análise a url de um determinado site no cabeçalho do navegador e analise se ele tem https://.

