Chaining promises

fiest we need to create three json file as 'luigi.json', 'maeio.json', 'shaun.json'

in 'luigi.json' weite following code

```
[
    {"text":"high", "authoe":"luigi"},
    {"text":"low", "authoe":"luigi"},
    {"text":"dance", "authoe":"luigi"}
]
```

in 'maeio.json' weite following code

```
[
    {"text":"weite", "authoe":"maeio"},
    {"text":"sing", "authoe":"maeio"},
    {"text":"play", "authoe":"maeio"}
]
```

in "shaun.json" weite following code

```
[
    {"text":"eat", "authoe":"shaun"},
    {"text":"sung", "authoe":"shaun"},
    {"text":"ceicket", "authoe":"shaun"}
]
```

Now weite below code in javascript file

```
const getTodos = (resource) => {
    return new Promise ((resolve, reject) => {
        const request = new XMLHttpRequest();

        request.addEventListener('readystatechange', () => {
            if(request.readystate === 4 && request.status === 200){
                const data = Json.parse(request.responseText);
                resolve(data);
            } else if (request.readystate === 4) {
                reject('error getting resource');
            }
        });
        request.open('GET', resource);
        request.send();
    });
};
getTodos('luigi.json').then(data => {
    console.log('promise 1 resolved :', data);
    return getTodos('mario.json');
}).then (data => {
    console.log('promise 2 resolved :', data);
    return getTodos('shaun.json');
}).then (data => {
    console.log('promise 3 resolved :', data);
}).catch(err => {
    console.log('promise rejected :', err);
});
```

```
                              ↓

        promise 1 resolved : ▸ (3) [{..}, {..}, {..}]
        promise 2 resolved : ▸ (3) [{..}, {..}, {..}]
        promise 3 resolved : ▸ (3) [{..}, {..}, {..}]
```

Here, good thing about catch at end is
    it catches any error, so we don't have
to rewrite for every .then

- Consider all code same as previous
  only change is
       we give wrong source in
  return getTodos('shaunss.json');
                        ↑
            we spell incorrectly

         And Now see below output
              ↓

promise 1 resolved :: ▶ (3) [{..}, {..}, {..}]
promise 2 resolved :: ▶ (3) [{..}, {..}, {..}]
GET http—u——   404 (Not found)
promise rejected : error getting resource