

Promises

Get a hold of Promises

prm1.js

Write a function `compareToTen` that takes a number as an argument and returns a promise. The Promise should resolve to the string 'Valid' if the number ≥ 10 and otherwise rejected with the error: 'Give me more'

```
compareToTen(15)
  .then(result => console.log(result)) // Should Print: 15 is Valid
  .catch(error => console.log(error))

compareToTen(8)
  .then(result => console.log(result))
  .catch(error => console.log(error)) // Should Print: 8 is too small
```

prm2.js

Write two functions: `makeAllCaps`, `sortWords` that gets an array and return a promise if the array contains anything but strings, it should throw an error.

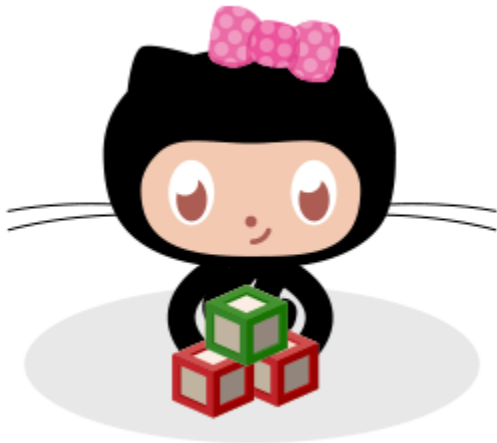
```
makeAllCaps(['cucumber', 'tomatos', 'avocado'])
  .then(sortWords)
  .then((result) => console.log(result))
  .catch(error => console.log(error))

makeAllCaps(['cucumber', 44, true])
  .then(sortWords)
  .then((result) => console.log(result))
  .catch(error => console.log(error))
```

Implement the functions `fetchX`, `fetchY` to make this code work as described

```
function add(prmX, prmY) {  
  return Promise.all([prmX, prmY])  
    .then(values => {  
      return values[0] + values[1]  
    })  
}  
  
// `fetchX()` should return a promise that is resolved to 25 immediately  
// `fetchY()` should return a promise that is resolved after 2 seconds to 17  
add(fetchX(), fetchY())  
  .then(sum => {  
    console.log(sum)  
  })
```

github- repos.html



Use the github [API](#) to get a list of users with images and show it nicely on the page, along with their images

Note: Open APIs will sometimes block you if too many requests are being sent, use a simple caching mechanism as presented in the `city.service`