Intégration continue pour le web

Correction TP n° 2 Javascript avancé

Exercice 1 :

```
import chalk from "chalk";

const colors = ["blue", "red", "green", "yellow", "cyan"];
colors.forEach(color => console.log(chalk[color](color)));
```

Exercice 2 :

En utilisant les promesses

```
import axios from "axios";
const NUMBER_OF_USER_TO_FETCH = 100;

axios
  .get(`https://randomuser.me/api/?results=${NUMBER_OF_USER_TO_FETCH}`)
  .then(response => response.data.results)
  .then(users => console.log(users));
```

En utilisant async/await

```
import axios from "axios";
const NUMBER_OF_USER_TO_FETCH = 100;

async function displayUsers() {
  const response = await axios.get(
    `https://randomuser.me/api/?results=${NUMBER_OF_USER_TO_FETCH}`
  );
  const users = response.data.results;
  console.log(users);
}
displayUsers();
```

Exercice 3 :

En utilisant les promesses

```
import axios from "axios";
const NUMBER_OF_USER_TO_FETCH = 1000;

axios
  .get(`https://randomuser.me/api/?results=${NUMBER_OF_USER_TO_FETCH}`)
  .then(response => response.data.results)
  .then(users =>
    users.filter(user => user.location.timezone.description.includes("Paris"))
  )
  .then(users => console.log(users));
```

En utilisant async/await

```
import axios from "axios";
const NUMBER_OF_USER_TO_FETCH = 1000;

async function displayUsers() {
  const response = await axios.get(
    `https://randomuser.me/api/?results=${NUMBER_OF_USER_TO_FETCH}`
  );
  const users = response.data.results;
  const filteredUsers = users.filter(user =>
    user.location.timezone.description.includes("Paris")
  );
  console.log(filteredUsers);
}
displayUsers();
```

Exercice 4 :

En utilisant les promesses

```
import axios from "axios";
const NUMBER_OF_USER_TO_FETCH = 1000;

axios
  .get(`https://randomuser.me/api/?results=${NUMBER_OF_USER_TO_FETCH}`)
  .then(response => response.data.results)
  .then(users =>
    users.filter(user => user.location.timezone.description.includes("Paris"))
  )
  .then(users =>
    users.map(user => ({
      firstname: user.name.first,
      lastname: user.name.last
    }))
  )
  .then(users => console.log(users));
```

En utilisant async/await

```
import axios from "axios";
const NUMBER_OF_USER_TO_FETCH = 1000;

async function displayUsers() {
  const response = await axios.get(
    `https://randomuser.me/api/?results=${NUMBER_OF_USER_TO_FETCH}`
  );
  const users = response.data.results;
  const filteredUsers = users
    .filter(user => user.location.timezone.description.includes("Paris"))
    .map(user => ({
      firstname: user.name.first,
      lastname: user.name.last
    }));
  console.log(filteredUsers);
}
displayUsers();
```

Exercice 5 :

```
function sleep(ms) {
  return new Promise(resolve => {
    setTimeout(() => {
      resolve();
    }, ms);
  });
}
```

En utilisant setTimeout()

```
console.log("Toc toc");
setTimeout(() => {
  console.log("Qui est là?");
  setTimeout(() => {
    console.log("C'est Internet Explorer");
  }, 10000);
}, 500);
```

En utilisant les promesses et votre nouvelle fonction

```
console.log("Toc toc");
sleep(500)
  .then(() => console.log("Qui est là?"))
  .then(() => sleep(10000))
  .then(() => console.log("C'est Internet Explorer"));
```

En utilisant async/await et votre nouvelle fonction

```
async function joking() {
  console.log("Toc toc");
  await sleep(500);
  console.log("Qui est là?");
  await sleep(10000);
  console.log("C'est Internet Explorer");
}
joking();
```