**Promise, async/await - zadania**

**Zad 1**

Mając do dyspozycji poniższe endpointy API:

const countriesAPI = '<https://restcountries.com/v3.1/all>'

const catsAPI = '<https://api.thecatapi.com/v1/breeds>'

- Pobierz listę wszystkich krajów, a następnie wydrukuj informacje w postaci: Państwo - Stolica - Powierzchnia - Języki Urzędowe - Populacja

- Wyświetl wszystkie imiona kotów, pobierając je spod catsAPI

- Odczytaj nazwy 10 największych państw pod względem

a) powierzchni

b) zaludnienia

- Rozwiązanie zaimplementuj, wykorzystując Promise-y oraz łańcuchowe wywołanie .then()/.catch() oraz z użyciem async/await.

**Zad 2.**

Zmodyfikuj poniższy kod:

| function job() {  return 'hello world'; } |
| --- |

Aby spełniał następujące założenia:

* funkcja job musi zwracać obiekt Promise
* Promise ma się po wywołaniu wykonywać 2 sekundy i dostarczać napis ‘hello world’

**Zad 3.**

Mając do dyspozycji poniższy kod (

Kod wykorzystuje Promise, aby handlować result. W rezultacie zwracana jest informacja z bazy (info.name) lub wyświetlany błąd i rzucany wyjątek.

| function job(result, database, errorManager) {  return result   .then(function(id) {  return database.get(id);  })   .then(function(info) {  return info.name;  })   .catch(function(error) {  errorManager.notify(error);  throw error;  }); } |
| --- |

Przerób przykład tak, aby wykorzystać słowa kluczowe async/await.

**Zad 4.**

Korzystając z API: <https://fakerapi.it/api/v1/persons?_quantity=>**<number\_of\_people>**, stwórz skrypt, który będzie pobierał listę 10 osóbi wyświetlał na stronie informacje o każdej z nich. Informacje te mają być w postaci, np.

Jan Kowalski

<zdjęcie Jana Kowalskiego pobrane z API>

Adam Nowak

<zdjęcie Adama Nowaka pobrane z API>

etc.

**Podpowiedź:**

Aby móc dynamicznie modyfikować zawartość strony, wykorzystaj operacje na DOM.