**1. Dekompozycja obiektu z danych JSON**

Pamiętasz funkcję getJSON, którą stworzyłeś w tygodniu trzecim? Za jej pomocą pobierz dane JSON z tego adresu: http://code.eduweb.pl/bootcamp/json/. Następnie w funkcji callback, gdzie te dane będą już zamienione na obiekt JavaScript, wykorzystaj dekompozycję (*destructuring*), aby utworzyć za pomocą zapisu **ES6** nowe zmienne, które przechowywać będą dane spod kluczy: name, username, email, address.geo[0], address.geo[1], website i company.name. Powyższe dane wstaw do template stringu,dodając odpowiednie etykiety jak np. *Imię*, *Firma* czy *Adres e-mail* wraz z niezbędnym kodem HTML, np. w formie linku dla website. W przypadku współrzędnych geograficznych, wstaw je do takiego linku: <a href=“http://bing.com/maps/ default.aspx?cp=LAT~LON”>Pokaż na mapie</a>, gdzie LAT i LON zastąpisz kolejno przez address.geo[0] i address.geo[1], które na tym etapie powinny być już w zmiennych. Powyższą operację wykonaj oczywiście dla **wszystkich** obiektów z tablicy.

Cały sformatowany ciąg wraz ze wstawkami HTML wstaw na stronę. Sam proces

pobierania danych Ajaxem i dalszego ich formatowania, możesz wywołać za pomocą

kliknięcia jakiegoś przycisku.

**2. Funkcja tagująca do formatowania cen**

Utwórz funkcję tagującą, która użyta na tzw. *template stringu* w **ES6**, sformatuje podane w nim ceny za pomocą kodu: n.toFixed(2).replace(/(\d)(?=(\d{3})+\.)/g, '$1,');

(pożyczamy ciekawe rozwiązanie z tego postu). Zanim jednak dokonasz takiego

formatowania, przelicz cenę przez kurs podanej przy wywołaniu funkcji tagującej waluty. Zakładamy, że ceny bazowe są podane w złotówkach, a nazwa i kurs waluty dostępna będzie pod window.currencies (mogłaby być w ten sposób dodana np. podczas renderowania strony przez system CMS, a dzięki temu mamy dostęp do tych danych w kodzie JavaScript. My jednak wpiszemy to sobie na sztywno). Przykładowe użycie tego kodu powinno wyglądać następująco: https://pastebin.com/6A3WZF6h. Zauważ, że przed samym template stringiem nie jest podana wyłącznie nazwa funkcji formatPrice, ale jest ona wywoływana z argumentem “GBP”. Jak być może się domyślasz, oznacza to, że ta

funkcja ma zwrócić inną funkcję, która zostanie użyta jako *tag function*. Argument jest

przekazywany po to, aby można go było zmienić np. na “USD” i wówczas funkcja powinna przeliczyć cenę po kursie dolara, a także dodać przyrostki USD do ceny w sformatowanym ciągu. Zwracana funkcja, która posłuży jako funkcja tagująca, powinna mieć zatem dostęp do zmiennej przechowującej **kurs** oraz **nazwę waluty** dla podanego argumentu. Zrealizujesz to za pomocą domknięcia.