

Korzystanie z elementów DOM

Nie modyfikuj kodu HTML, chyba że w poleceniu jest napisane inaczej.

Zadanie 1

- Stwórz funkcję `getDatanInfo(elements)`, do której przekaz jako argument zmienną `links` (pamiętaj o tym, że jest to tablica)
- stwórz w funkcji nową tablicę i wypełnij ją wartościami pobranymi z atrybutu **data** każdego elementu `li`
- zwróć tę tablicę.

Zadanie 2

Przeanalizuj kod HTML i JavaScript. W pliku **app.js** jest przygotowane kilka zmiennych, w których zapisane są wyszukane elementy DOM.

Wypisz w konsoli wszystkie te zmienne. Sprawdź, które z nich to tablice. W przypadku tablic przeiteruj (używając np. pętli **for**). Wypisz nazwę tagu oraz klasy dla każdego elementu.

Zadanie 3

Wypisz w konsoli wartości `innerHTML` i `outerHTML` dla elementów zmiennej `blocks`. Napisz w komentarzu, czym się od siebie różnią. Ustaw wartość `innerHTML` na "Nowa wartość diva o klasie blocks". Przeanalizuj, jak zmienił się kod HTML strony.

Zadanie 4

- Znajdź na stronie element o **id** `mainFooter`
- Stwórz funkcję `getId(element)`, do której przekaz jako argument znaleziony element
- pobierz w funkcji nazwę id elementu i zwróć ją

Zadanie 5

- Stwórz funkcję `getTags(elements)`, do której przekaz jako argument zmienną `childElements` (pamiętaj o tym, że jest to tablica)
- stwórz w funkcji nową tablicę i wypełnij ją nazwami tagów pobranymi z elementów znajdujących się w `childElements`
- zwróć tę tablicę.

Zadanie 6

- Stwórz funkcję `getClassInfo(element)`, do której przekaż jako argument zmienną `banner`
- zwróć tablicę z listą klas, pobraną z przekazanego do funkcji argumentu.

Zadanie 7

- Znajdź wszystkie elementy **li**, które są w tagu `nav`.
- Stwórz funkcję `setDataDirection(elements)`, do której przekaż jako argument znalezione elementy.
- W funkcji nadaj każdemu elementowi **li** atrybut `data-direction = "top"`, ale tylko dla tych elementów, które nie mają ustawionego tego atrybutu.