

Programowanie WWW - laboratorium 3

ES6 – nowe możliwości

Cel zajęć:

Celem laboratorium jest przećwiczenie zagadnień związanych z nowymi elementami które pojawiły się w ES6.

Do realizacji zadań pomocne będą następujące strony:

<https://github.com/lukehoban/es6features>

[https://developer.mozilla.org/en-](https://developer.mozilla.org/en-US/docs/Web/JavaScript/New_in_JavaScript/ECMAScript_2015_support_in_Mozilla)

[US/docs/Web/JavaScript/New_in_JavaScript/ECMAScript_2015_support_in_Mozilla](https://developer.mozilla.org/en-US/docs/Web/JavaScript/New_in_JavaScript/ECMAScript_2015_support_in_Mozilla)

Zweryfikuj czy nowe możliwości są wspierane przez Twoją przeglądarkę. Jeśli tak zrób poniższe zadanie, jeśli nie możesz samodzielnie napisać odpowiedniego kodu w ES6 i dokonać transpilacji.

zad. 1

Stwórz 2 tablice:

```
const tab1 = [1, 7, 12, 46, 85];
```

```
const tab2 = [10, 26, 75, 34];
```

Za pomocą operatora rozwijania (**spread syntax**) połącz te dwie tablice tak, by w wyniku uzyskać:

```
const tab3 = [1, 7, 12, 46, 85, 10, 26, 75, 34];
```

Posortuj te tablice odpowiednią metodą.

Następnie pobierz z tej tablicy największą i najmniejszą liczbę wyświetlając je na ekranie/console.

zad. 2

Stwórz funkcję `printUpper()`, która będzie przyjmowała dowolną liczbę parametrów.

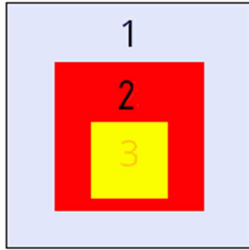
Wykorzystaj do tego operator rozproszenia (**rest parameter**). Wypisz w konsoli jeden tekst, który będzie składał się z przekazanych wartości pisanych dużymi literami i rozdzielonych znakiem " + ":

wywołanie: `printUpper("polska", "norwegia", "australia", "kanada");`

wynik: "POLSKA + NORWEGIA + AUSTRALIA + KANADA"

zad. 3 Dynamiczna obsługa zdarzeń.

Stwórz stronę zawierającą 3 elementy np. `div`/zdjęcia spozycjonowane tak jak na rysunku poniżej.



Niech naciśnięcie któregokolwiek z nich wyświetla odpowiedni komunikat:

Żółtego - „naciśnąłeś żółty o wartości 5”

Czerwonego – „naciśnąłeś czerwony o wartości 2”

Niebieski – „naciśnąłeś niebieski o wartości 1”

Jeśli suma naciśniętych wartości przekroczy 30 należy wyłączyć możliwość dodawania przez obiekt 2. Jeśli 50 usuwamy dodatkowo obsługę zdarzenia przez obiekt 3.

Niech na stronie będą 3 przyciski: StopStratPropagation, RemoveListener, AddListener oraz lista rozwijalna z wartościami pól.

Za pomocą tych przycisków zasymuluj działanie aplikacyjki polegające na czasowym (do kolejnego naciśnięcia przycisku) wyłączeniu obsługi zdarzenia dla obiektu 1.

RemoveListener – usuwa funkcje dla obiektu z aktualnie wybranego numeru obiektu.

AddListener – dodaje nową funkcję kuku() { zwiększającą dodawaną wartość o 3 jednostki).

zad. 4.

Mamy tablicę:

```
const tab = [1,2,3,4,5];
```

Za pomocą destrukuryzacji pobierz pierwszą, drugą i czwartą wartość z tej tablicy i podstaw je pod zmienne a, b, c

Mamy obiekt:

```
const obj = {  
  name : "Marcin",  
  surname : "Kowalski",  
  age : 20  
}
```

Za pomocą destrukuryzacji utwórz 3 zmienne: name, surname i userAge

Niech będzie obiekt o następującej strukturze:

```
const user = {  
  name: 'Grzegorz',  
  type: 'node',  
  info: 'something',  
  id: 21,
```

```
    relatives: {  
      wife: {  
        name: 'Magdalena',  
        id: 22  
      }  
    }  
  }  
};
```

Napisz kod pozwalający na przypisanie imienia zony oraz zawartości pola info oraz id do nowego obiektu o nazwie test. Wykorzystaj operacje destrukuryzacji.

zad. 5.

Stwórz klasę Brick, która będzie opisywać pojedynczy klocek w Arkanoid. Niech ma:

- właściwości x, y
- właściwość graphic
- właściwości width, height
- właściwość type
- właściwość live
- metodę print, która wypisze powyższe detale.
- metodę init, która wypisze w konsoli "Dodano na planszę"

Stwórz klasy BrickRed, BrickBlue, BrickGreen, które będą dziedziczyć po klasie Brick.

Klasy powinny mieć:

BrickBlue powinien mieć:

- graphic ustawiony na "blue.png"
- live ustawione na 10

BrickRed powinien mieć:

- graphic ustawiony na "red.png"
- live ustawione na 15

BrickGreen powinien mieć:

- graphic ustawiony na "green.png"

- live ustawione na 20

Dodatkowo stwórz klasę BrickAnim, która będzie dziedziczyć po klasie Brick. Klasa ta powinna mieć metodę moveHorizontal, która będzie wypisywać w konsoli "poruszam się poziomo z szybkością ...". W miejsce kropek wstaw właściwość speed, którą będzie miała ta klasa.

Stwórz kilka obiektów na bazie powyższych klas. Zainicjuj im metody init() i print().

zad. 6.

Utwórz używając generatora licznik który wyświetlać będzie kolejne liczby od 20 w dół do 1.

zad. 7.

Napisz wyrażenie za pomocą którego zwrócisz informacje o ilości kobiet w poniższej tablicy oraz podasz informacje o ich sumarycznych zarobkach. Wykorzystaj nowe funkcje tablicy – filter oraz reduce.

```
const user = [
  { imie: "Ola", sex: 'F', value: 5000},
  { imie: "Kasia", sex: 'F', value: 650},
  { imie: "Jurek", sex: 'M', value: 10},
  { imie: "Ola", sex: 'F', value: 1200},
  { imie: "Robert", sex: 'M', value: 77},
  { imie: "Wacek", sex: 'M', value: 77},
  { imie: "Ula", sex: 'F', value: 77},
  { imie: "Ola", sex: 'F', value: 77},
]
```

zad. 8.

Napisz kod testujący działanie modularności w JS w tym export i import funkcji z modułów. Napisz 4 przypadkowe funkcje kuku(), ala(), kasia(), fufu() oraz jeszcze raz ala(); Funkcje kuku() i ala() umieść w jednym module i wyeksportuj. Pozostałe funkcje fufu(), kasia() oraz ala() umieść w drugim module eksportując tylko funkcje ala() oraz fufu(). Napisz teraz funkcje test() której zadaniem będzie wywołanie funkcji kuku, ala, fufu i ala w kodzie. Sprawdź również czy uda Ci się wywołać funkcje kasia w kodzie test().

Dodatkowe zadania związane z ES6 możesz znaleźć pod tym adresem
https://marijnhaverbeke.nl/talks/es6_falsyvalues2015/exercises

Zachęcam do przećwiczenia.