

2. Obsługa błędów i debugowanie

Ćwiczenie 2.1.

Materiał do ćwiczenia: <http://www.kurshtml.edu.pl/js/error.html>.

Znajdź i określ rodzaje błędów w podanym niżej kodzie źródłowym JS. Następnie popraw kod tak, by nie było w nim żadnych błędów.

Kod źródłowy:

<https://warsztaty.sealcode.org/TWE/zajecia/2/assets/archives/exercise21.zip>

Jako rozwiązanie wyślij na repozytorium Githuba do katalogu Lesson_02 katalog o nazwie *exercise21*, który będzie zawierał plik **index.html**, plik z poprawionym kodem JS o nazwie **app.js** oraz plik z pierwotnym kodem JS okomentowanym o odpowiednie nazwy rodzajów błędów w odpowiednich liniach kodu **old.js**.

Ćwiczenie 2.2.

Materiał do ćwiczenia: <http://www.obliczeniowo.com.pl/660>

i <https://blog.ragnarson.com/2014/11/03/five-functions-of-the-console-object-you-didnt-know.html>.

Dany jest kod źródłowy JS. Uzupełnij ten kod o następujące instrukcje:

- kiedy będzie wywoływana funkcja o nazwie **main**, należy w konsoli wyświetlić zwykły log o treści „Rozpoczynam zadanie...”;
- kiedy jest dzielenie przez zero, należy wyświetlić w konsoli błąd o nazwie „Dzielenie przez 0!”;
- kiedy wartość zmiennej **x** jest mniejsza od 5, należy wyświetlić w konsoli ostrzeżenie o nazwie „Wartość zmiennej x jest mniejsza od 5!”;

- kiedy tablica o nazwie **tab** jest pusta, należy wyświetlić w konsoli informację o nazwie „*Tablica tab jest pusta.*”;
- na końcu kodu funkcji o nazwie **debugowanie**, należy wyświetlić w konsoli tzw. błąd *assertion failed* o treści „*Błąd zmiennej y!*”, jeśli zmienna **y** jest większa od 3;
- należy wyświetlić w konsoli tablicę o nazwie **tab2** w formie tabeli;
- należy wyświetlić w konsoli informację o czasie wykonywania kodu pętli *for*.

Kod źródłowy:

<https://warsztaty.sealcode.org/TWE/zajecia/2/assets/archives/exercise22.zip>

Jako rozwiązanie wyślij na repozytorium Githuba do katalogu Lesson_02 katalog o nazwie *exercise22*, który będzie zawierał plik **index.html** oraz plik z uzupełnionym kodem JS o nazwie **app.js**.

Ćwiczenie 2.3.

Materiał do ćwiczenia: <https://www.nafrontendzie.pl/obsługa-wyjatkow-javascript>.

Uzupełnij dany kod źródłowy JS o poniższe instrukcje:

- jeśli wywołana zostanie funkcja, która nie jest zdefiniowana, należy wyłapać taki wyjątek i wyświetlić nazwę (*name*) błędu oraz jego treść (*message*);
- kiedy zmienna **x** jest typu string, należy wyrzucić wyjątek o kodzie 200 i treści „*x jest typu string!*”;
- jeśli wystąpi jakikolwiek błąd na stronie, należy wyświetlić w konsoli ostrzeżenie, zawierające treść błędu, adres URL strony, na której występuje błąd oraz numer linii, w której ten błąd powstaje.

Kod źródłowy:

<https://warsztaty.sealcode.org/TWE/zajecia/2/assets/archives/exercise23.zip>

Jako rozwiązanie wyślij na repozytorium Githuba do katalogu Lesson_02 katalog o nazwie *exercise23*, który będzie zawierał plik **index.html** oraz plik z uzupełnionym kodem JS o nazwie **app.js**.

Ćwiczenie 2.4.

*Umieszczając w kodzie źródłowym JS linię z instrukcją **debugger**; ustawiamy w ten sposób punkt kontrolny, aby zatrzymać kod w pewnym miejscu. Jest to przydatne przy debugowaniu kodu.*

Dany jest kod źródłowy JS. Ustaw w nim punkty kontrolne w następujących sytuacjach:

- przed wywołaniem funkcji o nazwie **fun**;
- przed przejściem do kolejnej iteracji w pętli *for*;
- po wykonaniu pętli *for*.

Uruchom tak uzupełniony kod w przeglądarce, uruchom narzędzia programistyczne (F12) i zbadaj jakie wartości mają zmienne przed zatrzymaniem danej części kodu.

Kod źródłowy:

<https://warsztaty.sealcode.org/TWE/zajecia/2/assets/archives/exercise24.zip>

*Jako rozwiązanie wyślij na repozytorium Githuba do katalogu Lesson_02 katalog o nazwie exercise24, który będzie zawierał plik **index.html**, plik z uzupełnionym kodem JS o nazwie **app.js** oraz plik tekstowy ze sprawozdaniem z badania, które przeprowadziłeś/przeprowadziłaś w drugiej części zadania (napisz tam o wartościach wszystkich zmiennych: **i**, **x**, **y** w danych momentach wywoływania kodu, w każdej iteracji pętli osobno). Plik ten nazwij **sprawozdanie.txt**.*