Języki programowania i GUI

Lista 3 - 2022

1.
```
<input>+<input>=<input>
<script>
    let [a, b, c] = document.querySelectorAll('input');
    a.onkeyup = b.onkeyup = function f() { c.value = +a.value + +b.value }
    c.onkeyup = function g() { a.value = b.value = c.value / 2 }
</script>
```

   Wyjaśnij działanie powyższego dokumentu. Sprawdź co stanie się, gdy dane wkleimy do pola `input` (lub z niego wytniemy) za pomocą myszy. Skoryguj zachowanie aplikacji dodając do zmiennych `a, b, c` obsługę zdarzeń `oncut` oraz `onpaste`.
   Uwaga: wywołanie funkcji `f` i `g` należy spowolnić za pomocą `setTimeout` ponieważ `oncut` oraz `onpaste` są uruchamiane **przed** zaktualizowaniem własności `value`.

2. Wykonaj dokument zawierający pole `input` i pewną liczbę przycisków `button`, oznaczonych cyframi i symbolami działań. Zdefiniuj funkcje `onclick` dla przycisków tak, by twój dokument działał jak prosty kalkulator. Wskazówka: można użyć funkcji `eval`.

3. (3pkt) Saper. W dokumencie `html` umieść tabelę o 10 wierszach i 10 kolumnach. Napisz skrypt, który w 10 losowych komórkach tabeli doda atrybut `className="bomba"`. Do każdej komórki dodaj obsługę zdarzenia `onclick`. Kliknięcie komórki o klasie bomba powinno zmieniać kolor tła każdej komórki zawierającej bombę na czerwony. Kliknięcie innej komórki, powinno powodować pokazanie ile w sumie jest bomb w otaczających komórkach. W momencie odsłonięcia ostatniej komórki bez bomby powinien pojawiać się komunikat o wygranej. Do aplikacji dodaj przyciski umożliwiające rozpoczęcie nowej gry na dla różnych poziomach trudności (% bomb) i rozmiarach planszy. Wskazówka: kod generujący planszę i bomby zamknij w funkcji z dwoma parametrami.

4. Sudoku. W dokumencie `html` umieść tabelę o 9 wierszach i 9 kolumnach. Zastosuj `<style>td {height:40px;width:40px;border solid grey 1px }</style>`, by komórki były kwadratowe i widoczne. Napisz następujące funkcje i dodaj do dokumentu guziki, które je uruchomią:

   (a) (2pkt) `function generujPlanszę(n)`, która w `n` przypadkowych komórkach tabeli wpisuje przypadkowe cyfry od `1` do `9` w taki sposób, by nie było jednakowych cyfr w żadnym wierszu, kolumnie ani kwadratowej części 3x3.

   (b) (2pkt) `function rozwiąż()`, która wypełni pozostałe komórki częściowo wypełnionej tabeli, zachowując warunki punktu (a) lub pokaże alert "Brak rozwiązań", jeśli takie wypełnienie jest niemożliwe.

   (c) (2pkt) `function graj()`, która w każde "puste" pole wpisze tabeli listę możliwych cyfr. Kliknięcie cyfry powinno spowodować umieszczenie jej jako jedynej (zatwierdzonej) w tej komórce, dodanie numeru pola do listy ruchów i aktualizację list cyfr w pozostałych "pustych" polach. Po wykonaniu pierwszego ruchu powinien pojawiać się guzik `Cofnij` umożliwiający czyszczeniu pól w kolejności odwrotnej, niż były zatwierdzane i aktualizację list cyfr w "pustych" polach.

   Cyfry stanowiące treść zagadki (np wygenerowane w punkcie (a)), czyli zastane w tabeli (których nie można cofnąć) powinny różnić się wyglądem od cyfr "zatwierdzanych" w trakcie gry (które można cofnąć).

   W nazwie pliku z rozwiązaniem powinna się znaleźć lista liter odpowiadających wykonanym podpunktom (np `zad4ab.html`, `zad4c.html` itp).

5. Napisz skrypt, który po załadowaniu dokumentu, wstawi do elementu `<div id="toc"></div>` spis treści, czyli wszystkich elementów `h1 h2 h3`. Spis powinien być wykonany jako zagnieżdżona lista w postaci:

```
<ol>
<li><a href="#1">Rozdział 1</a>
<ol>
<li><a href="#1-2">Podrozdział 1.1 </a></li>
<li><a href="#1-3">Podrozdział 1.2 </a></li>
</ol>
</li>
<li><a href="#2">Rozdział 2</a>
<ol>
<li><a href="#2-1">Podrozdział 2.1 </a></li>
<li><a href="#2-2">Podrozdział 2.2 </a></li>
</ol>
</li>
</ol>
```

Tytuły rozdziałów i podrozdziałów w spisie treści muszą pokrywać się z zawartością znaczników `h1, h2, h3` w dokumencie i nie muszą być numerowane. Zadbaj o to, by twój skrypt przypisał automatycznie identyfikatory do tych `h1 h2 h3`, które ich nie mają tak, by spis treści być "klikalny". W przypadku gdy element ma już przypisany identyfikator użyj istniejącego. Identyfikatory przypisywane przez twój skrypt mogę mieć inną formę niż ta pokazana na przykładzie, w szczególności mogą być kolejnymi liczbami naturalnymi. Wskazówka: użyj `querySelectorAll("h1,h2,h3")`. Zadbaj o prawidłowe wcięcia w spisie treści.

6. Napisz skrypt, który w każdym słowie treści dokumentu odwróci kolejność liter.
zsipaN tpyrks, yrótk w mydżak eiwoły icśert utnemukod icórwdo ćśonjelok retil.
Wskazówka: poprawne rozwiązanie znajdziesz w przykładach z wykładu.

7. Użyj `new Map`, by policzyć, ile razy każde ze słów występuje w tekście dowolnie złożonego dokumentu `html` zawierającego twój skrypt. Wynik dodaj na końcu dokumentu w ramce, która po kliknięciu znaku `x` w prawym górnym jej rogu zostanie z dokumentu usunięta (`remove`). Wskazówka: zastosuj `textContent` do pobrania treści dokumentu, oraz `x.match(/\p{L}/u)` do sprawdzania, czy `x` jest literą jakiegokolwiek alfabetu.