

Aplikacje WWW

Lista 3/2020

W wykonaniu zadań może być pomocne pobranie i zrozumienie przykładów z wykładów 1-2.

1. W dokumencie wstaw paragraf w zielonej ramce z różowym tłem i napisem „Jeszcze nie kliknięty”. Zdefiniuj atrybut `onclick` tego paragrafu tak, by po jego kliknięciu, treścią paragrafu stawało się zdanie „Kliknięty po raz ..” z odpowiednią liczbą. Wskazówka: w skrypcie zdefiniuj zmienną, która będzie pamiętać dotychczasową ilość kliknięć.

2.

```
<div id="ramka"></div>
<button onclick="wypisz(imiona)">Imiona</button>
<button onclick="wypisz(liczby)">Liczby</button>
<script>
  var imiona=["ola","ala","Ela"];
  var liczby=[2,3,5,7,10,12,23];
  function wypisz(tab) {...}
</script>
```

W miejsce ... wpisz polecenia, które sprawią, że kliknięcie guzika *Imiona/Liczby* spowoduje pojawienie się w ramce imion/liczb pobranych z odpowiedniej tablicy.

Wskazówka: użyj `var ramka=document.getElementById('ramka'), s=""`; oraz `for(let x in tab) s+=...; ramka.innerHTML=s`.

Zadanie wykonaj w czterech wersjach: (a) elementy oddzielone spacjami, (b) elementy wypisane w postaci wypunktowania (``) (c) Jako komórki w tablicy o jednym wierszu, (c) Jako komórki w tablicy o jednej kolumnie. Tak więc docelowo w dokumencie znajdują się guziki *Imiona A,...*, *Imiona D* używające funkcji `wypiszA .. wypiszD` i analogicznie dla liczb. Dodaj do `<head>` taki styl, by granice komórek były widoczne.

3.

```
n=<input name="n" value=55 type="number">
<button onclick="witaj()"> Witaj</button>
<button onclick="liczby100()"> 1 .. 100</button>
<button onclick="liczbyN()"> 1 .. n</button>
....
<div id='ramka'></div>
```

W ciele dokumentu umieść powyższy kod, a pod nim napisz `<script>`, zawierający funkcje o następującym działaniu:

- `witaj()` umieszcza w ramce napis „Dzień dobry” (zamazując poprzednią treść),
 - `liczby100()` umieszcza w ramce liczby od 1 do 100 (zamazując poprzednią treść),
 - `liczbyN()` umieszcza w ramce liczby od 1 do liczby `n`, wpisanej w polu tekstowym (zamazując poprzednią treść),
 - `wieloN()` umieszcza w ramce 100 początkowych wielokrotności liczby `n` (zamazując poprzednią treść),
 - `ukryj()` powoduje ukrycie ramki (`display='none'`),
 - `pokaz()` powoduje pokazanie ramki (`display='block'`),
4. Stwórz dokument zawierający znacznik `` z jakimś ciekawym zdjęciem oraz cztery przyciski 'lewo', 'prawo', 'góra', 'dół' umieszczone na środkach odpowiednich krawędzi okna przeglądarki. Początkowo obrazek powinien znaleźć się na środku strony i mieć wysokość 100px. Przyciśnięcie któregoś z przycisków powinno przesuwając obrazek o 10px w wybranym kierunku. Kliknięcie na obrazek powinno powodować jego zniknięcie. Do wszystkich obiektów zastosuj `position:absolute`.
Wskazówka: np. zadeklaruj i aktualizuj zmienne pamiętające `left` i `top` obrazka.

5. Stwórz dokument, a w nim 15 paragrafów `<p> </p>` zawierających kolejne liczby w wybranym przez ciebie języku obcym (ale nie angielskim) oraz jeden przycisk `<button>OK</button>`.
- (a) stwórz taki styl, by paragrafy o numerach podzielnych przez 5 miały żółtą czcionkę na czarnym tle.
 - (b) napisz funkcję uruchamianą guzikiem, która przywróci wszystkim paragrafom biały kolor tła i czarny tekst.
 - (c) napisz funkcję uruchamianą innym guzikiem, która w co trzecim paragrafie ustawi kolor tła na różowy, a tekstu na niebieski.
 - (d) dodaj kolejny guzik, który na końcu dokumentu doda kopię 15-ego paragrafu (licząc od końca).
 - (e) * dodaj kolejny guzik, na końcu dokumentu doda paragraf zawierający informację „numer tego paragrafu to n ”, gdzie n będzie faktycznie numerem dodanego paragrafu.

Wskazówka: Twój program może zawierać linie podobne do:

```
var p=document.querySelectorAll('p');
for(i=4;i<p.length;i+=5) p[i].style...
document.body.innerHTML += '<p>'+p[5].innerHTML+'</p>';
```

6. (3pkt) Skopiuj swoje rozwiązanie zadania 7 z listy 1. Zwiększ rozmiar tabeli do 10×10 a potem dodaj na początku każdego wiersza komórkę `<th></th>`, a na początku tabeli dodatkowy wiersz zawierający 11 komórek `<th></th>`. Pod tabelą umieść skrypt

```
<script>
var tab=document.getElementsByTagName('table')[0];
for(let i=1;i<=10;i++)
{
    tab.rows[0].cells[i].innerHTML=i;
    tab.rows[i].cells[0].innerHTML=i;
}
</script>
```

Za pomocą reguły w znaczniku `<style>` nadaj komórkom `<th>` jasnoszary (`#aaa`) kolor tła i takie wymiary i brzegi jak w komórkach `<td>`. Dodaj nad skryptem następujące przyciski (a w skrypcie funkcje uruchamiane po ich kliknięciu):

- (a) Przycisk *Mnożenie*, który w lewym górnym rogu tabeli wpisuje `*`, a w komórkach `<td>` wyniki mnożenia numeru wiersza przez numer kolumny.
- (b) Przycisk *Dodawanie*, który w lewym górnym rogu tabeli wpisuje `+`, a w komórkach `<td>` wyniki dodawania numeru wiersza i numeru kolumny.
- (c) Przycisk *Modulo*, który w lewym górnym rogu tabeli wpisuje `%`, a w komórkach `<td>` reszty z dzielenia numeru wiersza przez numer kolumny.
- (d) Przycisk *Zwiększanie*, który zwiększa o 1 wszystkie liczby w komórkach `<td>`.
- (e) Przycisk *Sumy* wpisujący do pierwszej komórki `<th>` każdego wiersza i kolumny sumę liczb wpisanych w komórkach `<td>` tego wiersza/kolumny.
- (f) Przycisk *Parzyste*, który tło komórek `<td>` zawierających liczby parzyste zmienia na zielone, a nieparzystych na żółte (x jest parzyste jeśli $x \% 2 == 0$).
- (g) Przycisk *Szachy*, który “czyści” kolory tła komórek tabeli i zaczyna być widoczne początkowe kolory zadane globalnym stylem.
- (i) Przycisk *Gumkuj*, który usuwa liczby z komórek `<td>` tabeli.
- (j) Przycisk *Liczby*, który wypełnia komórki `<td>` liczbami od 1 to 100.