Technologie Programistyczne Systemy Internetowe Laboratorium 1 Powtórzenie z HTML, Javascript i JQuery

Wprowadzenie

Na zajęciach z przedmiotu Technologie Programistyczne Systemy Internetowe będziemy omawiać różne technologie i biblioteki webowe zarówno front-endowe jak i back-endowe takie jak: jQuery, Angular.js, Bootstrap, Node.js, JSF i inne. Przykłady z technologii związanych z J2EE można będzie wykonywać w dowolnym środowisku ID czyli STS, NetBeans albo InteliJ Idea.

Zadanie 1 – utworzenie ruchomej animacji za pomocą CSS

Utwórz prostą stronę składającą się z 3 kontenerów (elementów): Nagłówek, główna część i stopka. W nagłówku powinien się znajdować tytuł np. „Moja strona domowa", w głównej części dowolny tekst, a w stopce informacje o prawach autorskich (Copyright). Wygląd i wzajemne rozmieszczenie elementów strony zaimplementuj przy pomocy zewnętrznego arkusza CSS.

W kontenerze głównej części strony dodaj za pomocą odpowiednich konstrukcji CSS poruszający się napis" Witaj na mojej stronie". Animacja powinna składać się z co najmniej 4 kroków. W każdym kroku napis powinien się przemieszczać i zmieniać kolor.

wskazówki

Aby zrobić animację trzeba użyć w arkuszu css poleceń:
@keyframes animation-name i animation-duration.

- Najpierw definiujemy animację przy pomocy @keyframe nazwa animacji {
- W definicji animacji umieszczamy zmiany definicji stylów i położenia jakie chcemy osiągnąć w każdym kroku animacji np:

25% {background-color:yellow; left:200px; top:0px;}

Spowoduje to, że po 25 procentach przeznaczonego czasu napis będzie w punkcie 200, 0 i będzie miał kolor tła żółty. Takie definicje trzeba dodać dla każdego kroku animacji.

- Potem definiujemy początkowy styl dla nowego kontenera np o identyfikatorze adwordcontent czyli szerokość, wysokość, kolor tła i pozycjonowanie relatywne.

- Do definicji dla tego kontenera dodajemy definicję animation-name podając nazwę wcześniej utworzonej animacji a potem animation-duration podając czas całej animacji np:

```
#adwordcontent {
 width: 100px;
 ...
 animation-name: animacja;
 animation-duration: 4s;
}
```

- W pliku html dodajemy kontener div o identyfikatorze adwordcontent a w nim napis ”Witaj na mojej stronie”.

Zadanie 2 - migający tekst w 3 kolorach

W kontenerze przeznaczonym na nagłówek strony z poprzedniego zadania umieść napis ”Strona domowa”, który będzie migał w 3 różnych kolorach.

Wskazówki:

Do zaimplementowania tej funkcji można wykorzystać funkcję standardową JavaScriptu setTimeOut.

1. Do elementu span z tekstem można dodać ID a potem pobierać go zapomocą funkcji getElementBy ID i zmieniać jego kolor.
2. Zmienianie koloru można zaprogramować ustawiając właściwość style.colorelementu span z tekstem.
3. Funkcja zapalająca tekst na dany kolor może działać rekurencyjnie tzn wniej samej ma być wywołanie tej samej funkcji tylko że z innymi parametrami. np.

```
function migaj(id, kolor1, czas1, kolor2, czas2, kolor3, czas3)
{
//zmieniamy kolor elementu id na 1 po czasie 1 //uruchamiamy migaj z inną
kolejnością parametrów.
setTimeout('migaj(
   "' + id + '","' + kolor2 + '",' + czas2
   + ' ,"' + kolor3 + '",' + czas3 + ',"'
   + kolor + '",' + czas + ')', czas);
}
```

Zadanie 3 - implementacja stopera za pomocą javascript

W głównej części swojej strony z poprzednich zadań umieść stoper, który będzie zaimplementowany za pomocą javaScript. Stoper powinien mieć przyciski ”Start”,

”Stop”, i ”Reset” powiązane z odpowiednimi funkcjami Javascript. Aby nie spowodować ”zawieszenia się” przeglądarki stoper może się odświeżać np. co 0,5 sekundy.

Wskazówki:

1. Konstrukcja

   teraz = new Date().getTime(); zwróci nam aktualny czas wmilisekundach.

2. Aby uzyskać na wyświetlaczu stopera informacje ile upłynęło sekund iminut trzeba różnicę czasu pomiędzy momentem teraz a momentem startu stopera odpowiednio podzielić całkowitoliczbowo. Do tego można użyć funkcji Math.floor(), która zwraca całkowitą wartość z liczby zmiennoprzecinkowej np.:

   sekund =Math.abs((teraz -kiedyStart)/1000); minut =Math.floor(sekund/60);

3. Do odświeżania zawartości elementu np div z wynikiem stopera można użyć właściwości InnerHTML oraz funkcji getElementById np.

   Document.getElementById(’wynikStopera’).InnerHTML =wynik;

Zadanie 4 rysowanie za pomocą Javascript

W nowym dokumencie html dodaj rysunek człowieczka. Rysunek powinien być zrobiony z prostokątów, owali, kół, okręgów, trójkątów itd. Figury narysuj za pomocą obiektów i metod dostępnych w Javascript na tablicy Canvas.

Wskazówki:

- Do rysowania za pomocą Javascript można wykorzystać obiekty canvas oraz context. Obiekt canvas trzeba najpierw zdefiniować w html podając jego długość, szerokość i styl, w którym można określić np kolor tła. np.

```
<canvas id="canvas" width="640" height="480" style="position:
    absolute; background: #fff;">
</canvas>
```

- Potem ten element można w skrypcie Javascript pobrać z drzewa Dom dokumentu, utworzyć z niego kontekst na którym następnie można rysować figury. Kontekst też można odświeżać, czyścić itd. Np.:

```
var canvas = document.getElementById('canvas');  var
ctx = canvas.getContext('2d');
```

- Korzystając z metod obiektu kontekstu (w powyższym przykładzie ctx) można rysować np ustalić kolor prostokąta a potem go narysować podając współrzędne jego lewego górnego rogu i wymiary np.

```
ctx.fillStyle = "red";
ctx.fillRect(10, 10, 100, 50);
```

- Figury inne niż prostokąt możemy rysować jako ścieżki. Służą do tego metody:

  - beginPath(); - określa początek odcinka tzn zaczyna rysowanie

  - moveTo(x,y); - przenosi kursor do punktu bez rysowania,

  - lineTo(x, y); - rysuje odcinek od aktualnego miejsca kursora do podanego punktu,

  - stroke(); - powoduje wyświetlenie całej narysowanej łamanej;

  - fill(); - wypełnia łamaną. łamana musi być zamknięta ale powinno działać również jeśli narysujemy 2 boki trójkąta i wykonamy fill() wtedy trzeci bok zostanie obliczony i cała figura będzie wypełniona.

  - closePath(); - zamyka łamaną.

  - arcTo(x1, y1, x2, y2, r); - narysuje łuk o początku x1 y1 końcu x2 y2 i promieniu r.

- arc(x, y, r, alfa, beta); - narysuje koło o środku x y, promieniu r kącie początkowym alfa i kącie końcowym beta. Kąty są podawane w radianach więc do narysowania koła trzeba podać kąty 0 i 2*Math.pi();

Wszystkie te metody działają na obiekcie kontekstu.

Zadanie 5 - pokazywanie i ukrywanie elementów w JQuery

W nowo utworzonym pliku html umieść dwie listy zadań (prac). W jednej powinny być zadania, które planujesz wykonać a w drugiej takie, które już wykonałeś przykładowo: Zrobić zakupy, umyć samochód, przygotować się do zajęć, itp. Do każdej z list dodaj po 3 przykładowe zadania. Koło każdej listy umieść przyciski" Pokaż" i "ukryj". Powinny one pokazywać i ukrywać daną listę, przy której zostały umieszczone. Dodatkowo, gdy dana lista jest wyświetlona powinien być wyświetlony tylko przycisk" Ukryj listę" a gdy jest ukryta - powinien być wyświetlony tylko przycisk" Pokaż listę". Do zaimplementowania tego mechanizmu wykorzystaj bibliotekę JQuery. Możesz ją pobrać z:
https://jquery.com/download/

Wskazówki

- Na początku dokumentu html trzeba zadeklarować użycie skryptu z biblioteką JQuery korzystając z lokalnego pliku lub używając adresu zdalnego.
- Do zrealizowania operacji ukrywania i pokazywania elementów można użyć funkcji JQuery toggle lub hide i show, albo atrybutu hidden dla elementu.
- Do skojarzenia zdarzenia z przyciskiem można użyć funkcji z biblioteki JQuery click() np.

```
$(document).ready(function() {
    $("#mojprzycisk").click(function() {
        //tu robimy coś
});
});
```

Zadanie 6 - przenoszenie elementów

Za pomocą biblioteki JQuery zaimplementuj funkcjonalność przenoszenia elementów (zadań) pomiędzy listami na stronie z poprzedniego zadania. Przy każdym elemencie z listy zadania do wykonania umieść przycisk "Przenieś do wykonanych" a przy każdym zadaniu z listy wykonanych umieść przycisk "Cofnij do planowanych". Po naciśnięciu takiego przycisku element powinien się przenieść do odpowiedniej listy.

Wskazówki

- Do pobierania i wstawiania elementów można użyć funkcji JQuery takich jak prev(), next(), remove(), clone(), find(), first(), last(), appendChild(), append(), prepend(), insertBefore(), children(), parent(), firstchild itd. a także różnych selektorów. insertAfter(), before(), after(),

Zadanie 7 - obsługa zdarzenia keypress

W nowym pliku html umieść pole edycyjne przeznaczone do wpisania wiadomości. Pole edycyjne powinno zachowywać się w taki sposób, że wpisywanie w nim nowych znaków będzie powodowało ich pojawienie się w osobnym elemencie pod tym polem. Dodatkowo zaimplementuj mechanizm który będzie powodował wyczyszczenie całego pola po naciśnięciu kombinacji klawiszy ctrl średnik (jeśli w firefox lub chrome)

wskazówki

Do oprogramowania zdarzenia naciśnięcia klawisza Wykorzystaj funkcje keypress a do pobrania wartości klawisza funkcję which z biblioteki JQuery. Przykładowy fragment kodu

```
$('input').keypress(function(e){      var
which = e.which; //....
});
```

Obiekt e.which zawiera odczytany klawisz.

- Aby zaimplementować obsługę klawisza kontrol i jednoczesnego naciśnięcia innego klawisza trzeba użyć właściwości controlKey obiektu zdarzenia. Przykładowy fragment kodu poniżej:

```
function klawiszDown(e) {
e = e || this.event;   var key
= e.which || e.keyCode;   if
(e.ctrlKey)
 {
   if (key ==65) //naciśnięto ctrl a  wszystko
 }
}
```

- Kod znaku średnik to 62

Zadanie 8 - obsługa zdarzeń myszki

Za pomocą biblioteki JQuery na stronie z listami prac z poprzednich zadań zaimplementuj funkcje powodujące zmianę tła danego elementu menu po najechaniu na niego myszką i powracające do oryginalnego tła po zjechaniu wskaźnika myszki z elementu.

Wskazówki

- Do zaimplementowania tego mechanizmu można wykorzystać zdarzenia mouseEnter i MouseLeave albo jedno zdarzenie Hover.

Zadanie 9 - obsługa zdarzenia change

Utwórz prostą stronę z formularzem do akceptacji regulaminu oraz warunków użytkowania np. sklepu Internetowego. Strona powinna zawierać takie elementy jak: tekst regulaminu, pole wyboru" Zaakceptuj regulamin sklepu", tekst warunków użytkowania sklepu, pole wyboru „Zaakceptuj warunki użytkowania" oraz przycisk „Zarejestruj". Obszary przeznaczone na teksty regulaminu i warunki użytkowania mogą być puste lub zawierać dowolne teksty. Formularz na stronie powinien działać w ten sposób, że przycisk „Zarejestruj" jest niedostępny a uaktywni się on dopiero wtedy, gdy użytkownik zaznaczy oba pola wyboru. Podobnie, jeśli użytkownik odznaczy choćby jedno pole wyboru - wówczas przycisk „Zarejestruj" stanie się znowu nieaktywny.

Po naciśnięciu przycisku, użytkownik powinien otrzymać komunikat „Dziękujemy za rejestrację."

Działanie formularza zaprogramuj przy pomocy biblioteki JQuery.

Wskazówki

- Do sprawdzania zaznaczenia pól można wykorzystać funkcję change(...)
- Do uaktywnienia przycisku można wykorzystać selektor atrybutu [...] i atrybut disabled. Przykładowy fragment kodu:

```
$("input[type='checkbox']").change(function() {
   //robimy coś gdy zmieni się zaznaczenie pola wyboru
});
```

Zadanie 10 do samodzielnego wykonania

Na stronie z listą prac z zadań 5 i 6 zaimplementuj przy pomocy jQuery funkcjonalności dodawania nowego zadania i usuwania zadań z list. Nowe zadania użytkownik powinien móc dopisać w polu edycyjnym z którego po naciśnięciu przycisku przeniesie się ono do odpowiedniej listy. Możesz zaimplementować też mechanizm potwierdzania przez użytkownika chęci usunięcia zadania oraz zabezpieczenia list przed dodawaniem do nich identycznych zadań.