

# Podstawy animacji Flash

---

Zajęcia 3 – projekt galerii, dynamiczne wczytanie pliku SWF

dr Joanna Sekulska-Nalewajko  
Instytut Informatyki Stosowanej Politechniki Łódzkiej



*Łódź, 2018*

---

## Spis treści

1	Galeria zdjęć.....	3
2	Wczytanie pliku SWF z galerią do pliku SWF tzw. kontenera .....	7
3	(zadanie dodatkowe) Opcja pełnego ekranu (full screen).....	14

## 1 Galeria zdjęć

Ćwiczenie prezentuje sposób tworzenia galerii zdjęć z możliwością powiększania miniatur; wystarczy kliknąć na małe zdjęcia a zostanie ono powiększone dzięki ciekawej animacji.

### ETAPY PRACY:

- Krok 1: Ustawienie pola roboczego**
- Krok 2: Importowanie zdjęć**
- Krok 3: Tworzenie pierwszego klipu ze zdjęciem**
- Krok 4: Przygotowanie kolejnych klipów**
- Krok 5: Dodanie kodu ActionScript**

#### Krok 1 – Ustawienie pola roboczego

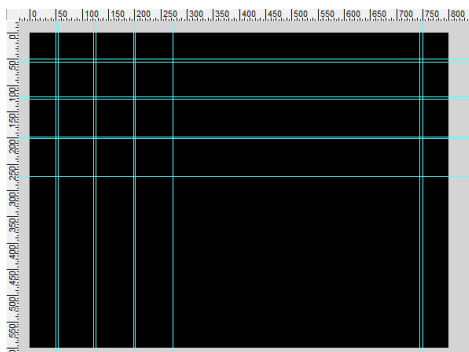
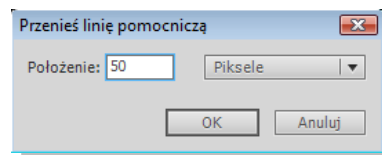
W nowym pliku stawiamy wielkość pola roboczego na 800 x 600 pikseli i ustawiamy ciemny kolor tła.

Następnie umieszczamy siatkę prowadnic, która wyznaczy położenie „miniatur”, czyli po prostu klipów, które zostaną utworzone w dalszej części ćwiczenia. Zamiast prowadnic można użyć siatki.

**Położenie prowadnice pionowych:** 50, 56, 122, 126, 198, 202, 274, 744 i 750

**Położenie prowadnice poziomych:** 50, 56, 122, 126, 198, 202 i 274

Ustawienie prowadnic jest dość proste: musimy mieć widoczne linijki (dostęp przez menu Widok), z których ściągamy linie pomocnicze. Klikając na wprowadzoną linię możemy w oknie ustawić jej dokładne położenie, jak pokazane jest na rysunku obok.

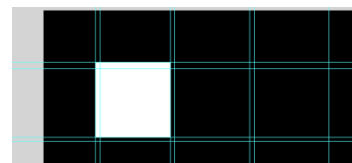


#### Krok 2 – Importowanie zdjęć

Pobierz zdjęcia do wykorzystania w galerii z lokalizacji podanej przez prowadzącego. Z menu **Plik** wybierz polecenie **Importuj > Importuj do biblioteki...** i zaimportuj zdjęcia do biblioteki projektu. Zdjęcia zostały wcześniej przygotowane w programie do obróbki zdjęć. Źródło do pobrania zdjęć poda prowadzący w trakcie zajęć.

#### Krok 3 – Tworzenie pierwszego klipu ze zdjęciem

Na początek umieszczamy w pierwszym „oczku” siatki biały kwadrat o wielkości boku 72 px. Konwertujemy go do symbolu Grafika o nazwie „kwadrat” (skrót w Windows: **F8**), a następnie do klipu filmowego, który możemy nazwać np. *foto1* (...a kolejne klipy *foto2*, *foto3*...). Będzie to pierwszy klip z całej serii klipów, które będą zawierały zdjęcia.

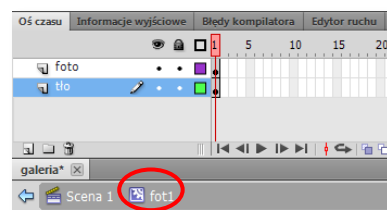


Edytujemy klip w miejscu – kliknij na klipie dwa razy lub wybierz z menu podręcznego polecenie **Edycja w miejscu**.

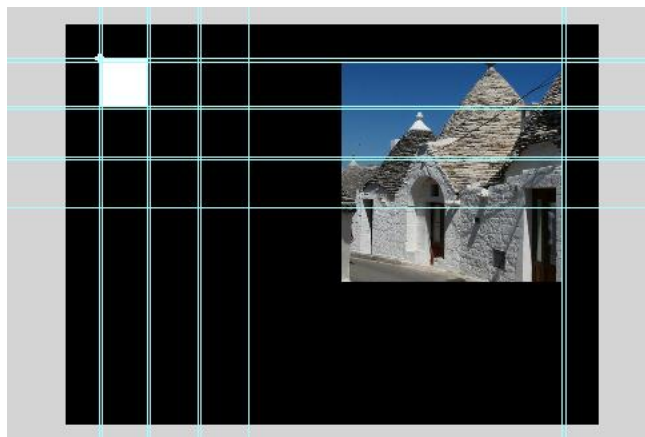
*Dzięki temu będziemy mieli widoczną siatkę prowadnic ustawioną na stole montażowym.*

**Wewnątrz klipu** dodajemy warstwę. Warstwy w klipie możemy nazwać „foto” i „tło” – warstwa „tło” leży na dole i zawiera już biały kwadrat (symbol typu grafika). Na warstwie „foto” umieszczamy pierwsze ze zdjęć – pobieramy je z biblioteki.

Umieszczamy zdjęcie z prawej strony pola roboczego przy pionowej prowadnicy „744” i poziomej „56”.

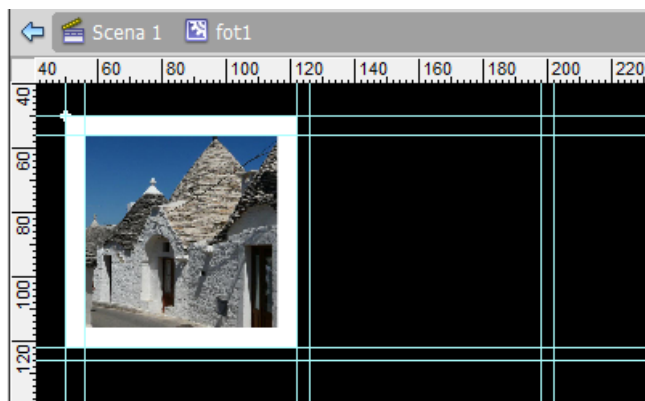


Zdjęcie jest duże (330 x 330 px) i taki wymiar będzie zachowany gdy zdjęcie będzie w powiększeniu (klatka 15), natomiast na początku animacji musi wystartować jako miniatura (klatka 1 - 60 x 60 px).



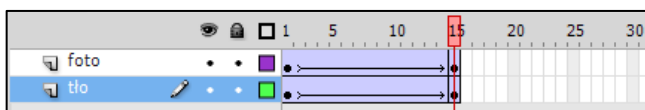
Wobec tego na obu warstwach stosujemy animację ruchu (zaznacz klatki i użyj polecenia **Utwórz klasyczną animację** z menu podręcznego). Animacja ta powinna przebiegać do klatki nr 15, więc tam dodaj na każdej warstwie klatkę kluczową (**F6**).

Powróć do klatki nr 1 w warstwie „foto” (zaznacz ją), w której zdjęcie zmniejsz do wielkości 60 x 60 px i umieść je dokładnie pośrodku białego kwadratu.

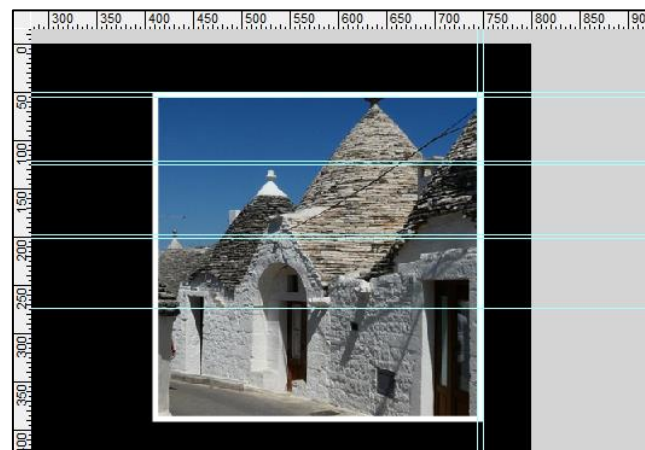


Podobną animację zastosujemy dla białego kwadratu – od klatki 1 do 15. W klatce nr 1 zachowujemy rozmiar 72 x 72 px, natomiast w klatce nr 15 powiększamy kwadrat do rozmiaru 342 x 342 px i umieszczamy go przy pionowej prowadnicy „750” i poziomej „50”.

Oto wygląd osi czasu klipu na tym etapie animacji:

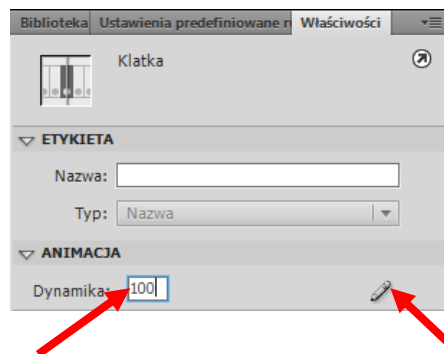
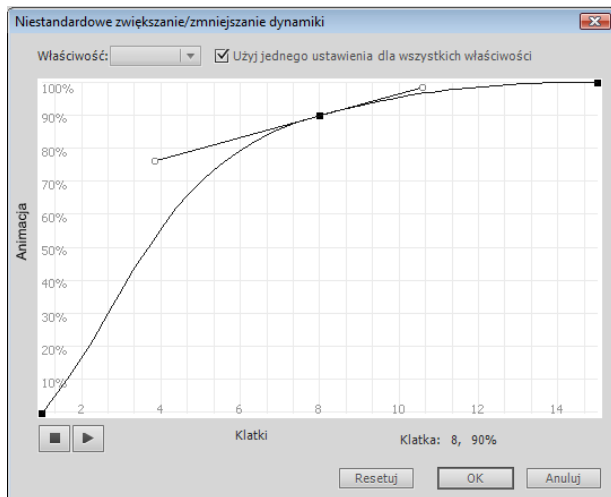


A to położenie elementów w klatce 15 – obiekty (zdjęcie i tło) są powiększone:



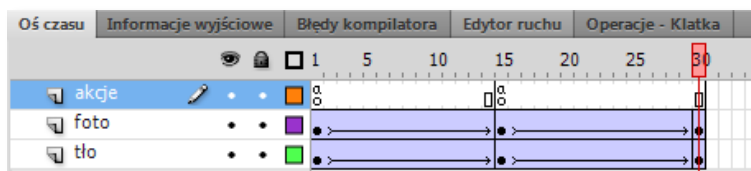
Do animacji zastosujemy własne ustawienia **Dynamiki**.

Zaznaczamy obie klatki nr 1 i w panelu **Właściwości** klatki ustawiamy wartość **Dynamiki** na **100**, a następnie zmieniamy te ustawienia w oknie **Niestandardowe zwiększanie / zmniejszanie dynamiki** – okno to otworzy się po kliknięciu ołówka:



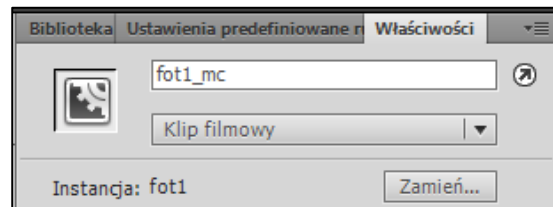
Następnie w obu warstwach kopiujemy klatki nr 1 (**Kopiuj klatki**) i wklejamy ją w klatce nr 30 (**Wklej klatki**) i tym samym odwracamy animację. Dynamikę możemy zmienić też w klatkach nr 15.

Dodajemy warstwę „*akcje*”, w której w klatce 1 i 15 wstawiamy akcje **stop( )**.



To koniec pracy z tym klipem. **Wracamy więc do głównej sceny.**

Zaznaczamy umieszczony na scenie klip i w panelu **Właściwości** wprowadzamy nazwę jego instancji: *fot1\_mc*.



#### Krok 4 – Przygotowanie kolejnych klipów

W podobny sposób przygotowujemy kolejnych 8 klipów ze zdjęciami. Oczywiście dbając o odpowiednie rozmieszczenie zdjęć w siatce prowadnic. Instancje klipów nazywamy odpowiednio *fot2\_mc*, *fot3\_mc*, *fot4\_mc* itd. Jako biały podkład do zdjęć wykorzystujemy symbol graficzny o nazwie *kwadrat*.

#### Krok 5 – Dodanie kodu ActionScript

Na głównej scenie dodaj warstwę i nazwij ją „*akcje*”. W klatce tej warstwy umieść I część skryptu:

```
import flash.events.MouseEvent;
import flash.display.MovieClip;

var currentImg:MovieClip;
var oldImg:MovieClip;

function onPhotoRelease(myevent:MouseEvent):void {
    currentImg = MovieClip(myevent.target);
    if(oldImg == null){
        showFoto(currentImg);
    }
}
```

```
        oldImg = currentImg;}
    else{
        if(currentImg == oldImg){
            hideFoto(currentImg);
            oldImg = null;}
        else{
            hideFoto(oldImg);
            showFoto(currentImg);
            oldImg = currentImg;}
    }
}
function showFoto(photo_):void {
    photo_.gotoAndPlay(2);
}
function hideFoto(photo_):void {
    photo_.gotoAndPlay(21);
}
```

---

Należy teraz określić, że klipy ze zdjęciami umieszczone na scenie głównej i oznaczone nazwami instancji, (po to aby były „widoczne” dla skryptów AS) będą odbiornikami zdarzenia myszki - *MouseEvent*.

Tuż po poprzednim skrypcie dodajemy więc kod nasłuchujący zdarzenia klikania myszką na klipy. Za nasłuch odpowiedzialna jest metody **addEventListener**, która ma dwa parametry – pierwszym z nich jest typ zdarzenia, które chcemy wykryć (w obecnym ćwiczeniu jest to **CLICK**), a drugim jest funkcja wykonywana gdy wykryte jest zdarzenie (w naszym przypadku jest to nasza funkcja główna z poprzedniej części **onPhotoRelease**).

Na początek II części kodu wstawiamy tablicę, która w pętli (While) zostanie uzupełniona naszymi klipami. W drugiej pętli (for), która iteracyjnie przegląda elementy tablicy, do każdego elementu tablicy zostanie dodana metoda nasłuchiwanie:

---

```
var klipy:Array = new Array();
var i:Number = 0;

while(1){
    i++;
    var nazwa:String = "fot" + i + "_mc";
    var obj:DisplayObject = getChildByName(nazwa);

    if (obj == null){
        break;
    }
    klipy.push(obj);
}

for each (var element in klipy){
    element.addEventListener(MouseEvent.CLICK,onPhotoRelease);
}
```

---

Przetestuj galerię. Opublikuj ostateczny plik SWF o nazwie galeria.swf.

## 2 Wczytanie pliku SWF z galerią do pliku SWF tzw. kontenera

Do aplikacji Flash można załadować dynamicznie (za pomocą ActionScript-u lub gotowych komponentów – kontenerów dostępnych w panelu *Składniki*) poniższe obiekty graficzne:

- Pliki SWF utworzone w języku ActionScript 3.0
- Pliki SWF zapisane w języku ActionScript 1.0 lub 2.0 (ale pod pewnymi warunkami).
- Pliki obrazu: JPG, PNG i GIF.

**Plik SWF który, zawiera kod AS ładujący zewnętrzny plik SWF**

```
var movieLoader:Loader = new Loader();
var movieURL:String = "galeria.swf";
var movieURLReq:URLRequest = new URLRequest(movieURL);
movieLoader.load(movieURLReq);
this.addChild(movieLoader);
```





**Zewnętrzny plik galeria.swf  
ładowany do kontenera  
(kontenerem jest obiekt Loader  
zawarty  
w docelowym pliku SWF)**

Do ładowania tych zasobów służy klasa Loader.

Można także załadować dynamicznie teksty, filmy i dźwięki, do czego służą inne klasy i metody.

Aby załadować plik SWF lub plik graficzny należy za pomocą kodu ActionScript 3.0 utworzyć obiekt **Loader**, następnie wskazać plik do wyświetlenia, a na końcu za pomocą metody **AddChild** dodać obiekt Loader do listy wyświetlania (uwidocznisz go). Wszystkie te etapy zawierają się w krótkim kodzie podanym w etapie 2 ćwiczenia.

### ETAPY PRACY:

**Krok 1: Ustawienia nowego pliku**

**Krok 2: Dodanie kodu ActionScript dla Loadera**

**Krok 3: Tworzenie paska postępu**

**Krok 4: Wstawienie pola tekstowego**

**Krok 5: Kod ActionScript do nasłuchiwanie ładowania pliku z galerią**

**Krok 6: Modyfikacje na osi czasu**

**Krok 7: Praca w klipie – efekt falującej wody**

**Krok 8: Przycisk**

**Krok 9: Akcje do przycisku**

### Krok 1 – Ustawienie nowego pliku

Otwórz nowy plik. Ustaw wielkość pola roboczego 800 x 600 pikseli, ciemne tło i zapisz plik pod nazwą „kontener” w lokalizacji tej samej co plik galeria.swf.

### Krok 2 – Dodanie kodu ActionScript dla Loadera

Zaznacz klatkę 1 i w panelu **Operacje** wpisz poniższy kod. Zmień nazwę warstwy na „akcje”.

```
var movieLoader:Loader = new Loader();
var movieURLReq:URLRequest = new URLRequest("galeria.swf");
movieLoader.load(movieURLReq);
this.addChild(movieLoader);
```

Można teraz przetestować aplikację.

W zasadzie powyższy kod wystarcza, aby załadować plik .swf. Jednak wykonamy ćwiczenie, które zademonstruje także śledzenie procesu ładowania za pomocą tak zwanego paska postępu i pola tekstowego z wyświetlaną wartością w % odnośnie ilości załadowanej informacji.

Dodamy także wstępną animację, która będzie stanowić tak zwany preloader.

Jest to wstęp do tworzenia wielowątkowych i modułowych aplikacji.

Jak widać kontener od razu wczytuje plik z galerii. Dzieje się to bardzo szybko, gdyż korzystamy z lokalnych zasobów komputera. Gdyby jednak odtwarzanie naszego pliku odbywało się poprzez połączenie sieciowe, to moglibyśmy dłużej oczekiwać na wyświetlenie zawartości, dlatego niezbędne jest wyświetlenie jakiejś informacji, że plik jest właśnie pobierany. Dobrą praktyką jest informowanie odbiorcy w postaci informacji graficznej lub tekstowej, że za jakiś czas nastąpi poprawne wyświetlenie oczekiwanej treści. Przygotujemy teraz elementy, które spełnią taką rolę informacyjną.

### Krok 3 – Tworzenie paska postępu

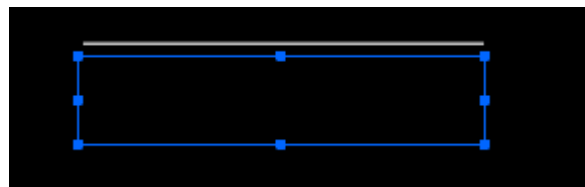
Utwórz nowy klip filmowy o nazwie „pasek”. Wewnątrz klipu narysuj wąski prostokąt o wypełnieniu białym lub szarym i długości ok. 600 pikseli. Zwróć uwagę, aby punkt zaczepienia paska (środek klipu) znajdował się z lewej strony.



Wróć do głównej sceny, wstaw nową warstwę o nazwie „pasek” i umieść w niej stworzony klip z paskiem w położeniu x około 100 i y około 300; i nadaj mu nazwę instancji *bar\_mc*.

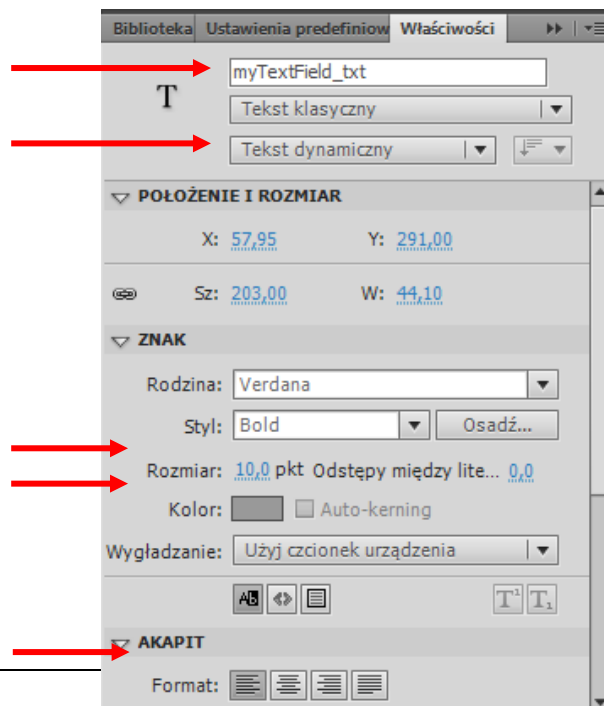
### Krok 4 – Wstawienie pola tekstowego

Wstaw nową warstwę o nazwie „pole\_txt” i na tej warstwie za pomocą narzędzia **Tekst** wstaw poniżej paska pole tekstowe.



Ustaw właściwości pola tekstowego:

- nazwa instancji: **myTextField\_txt**
- tekst dynamiczny
- rozmiar: 10 pkt
- kolor szary (lub biały)
- użyj czcionek urządzenia
- wyrównanie do lewej





### Krok 5 – Kod ActionScript do nasłuchiwania ładowania pliku z galerią

Do klatki ze skryptami (warstwa „*akcje*”) wpiszemy kod służący do nasłuchu ładowania pliku .swf.

```
movieLoader.contentLoaderInfo.addEventListener(ProgressEvent.PROGRESS,
progressHandler);

function progressHandler(myevent:ProgressEvent):void{
    var myprogress:Number = myevent.target.bytesLoaded/myevent.target.bytesTotal;
    bar_mc.scaleX = myprogress;
    myTextField_txt.text=Math.round(myprogress*100)+"%";
}
```

Dodajemy jeszcze kod do nasłuchiwania zakończenia pobierania danych:

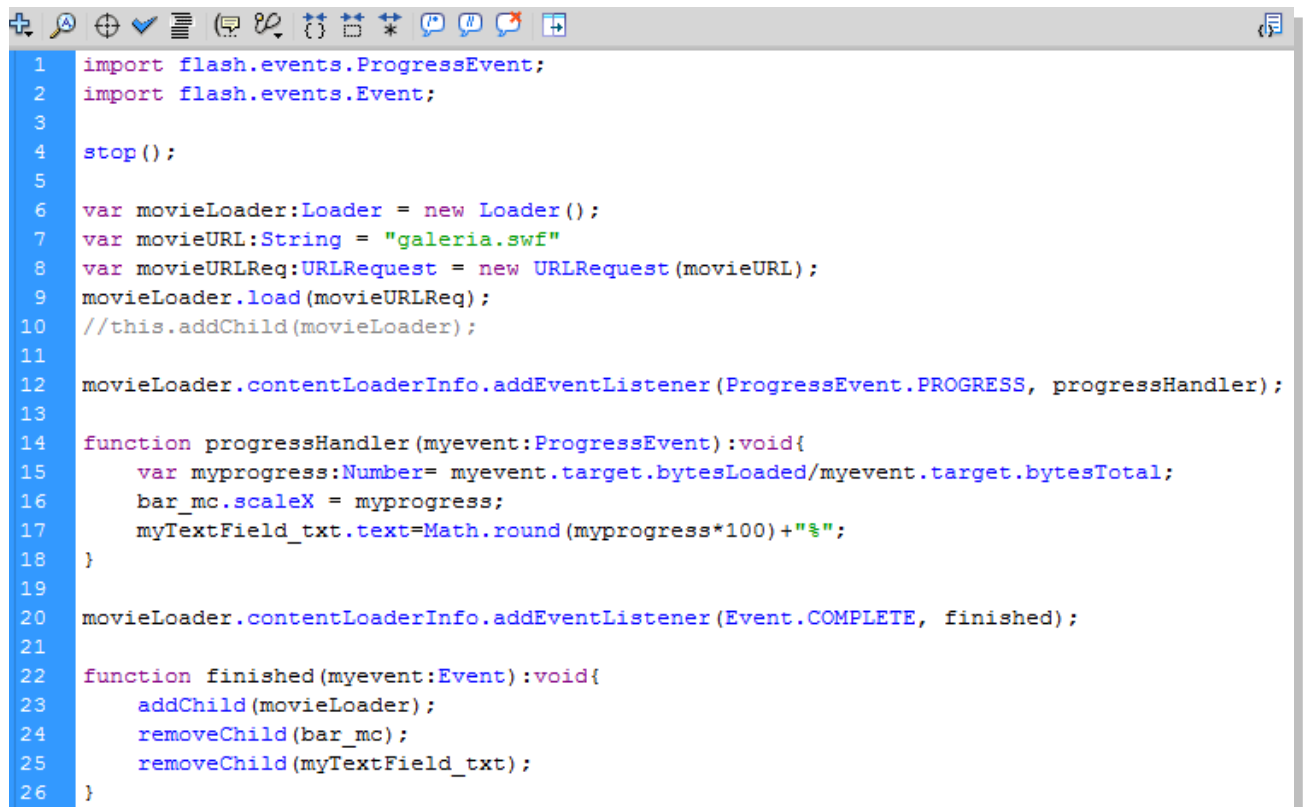
```
movieLoader.contentLoaderInfo.addEventListener(Event.COMPLETE, finished);

function finished(myevent:Event):void{
    addChild(movieLoader);
    removeChild(bar_mc);
    removeChild(myTextField_txt);
}
```

Na górze skryptu dodajemy jeszcze akcję `stop()` ;

**!!! Z pierwszej zaś części skryptu, którą pisaliśmy na początku ćwiczenia usuwamy linijkę `addChild(movieLoader)` ;** gdyż pojawiła się ona w funkcji `finished`.

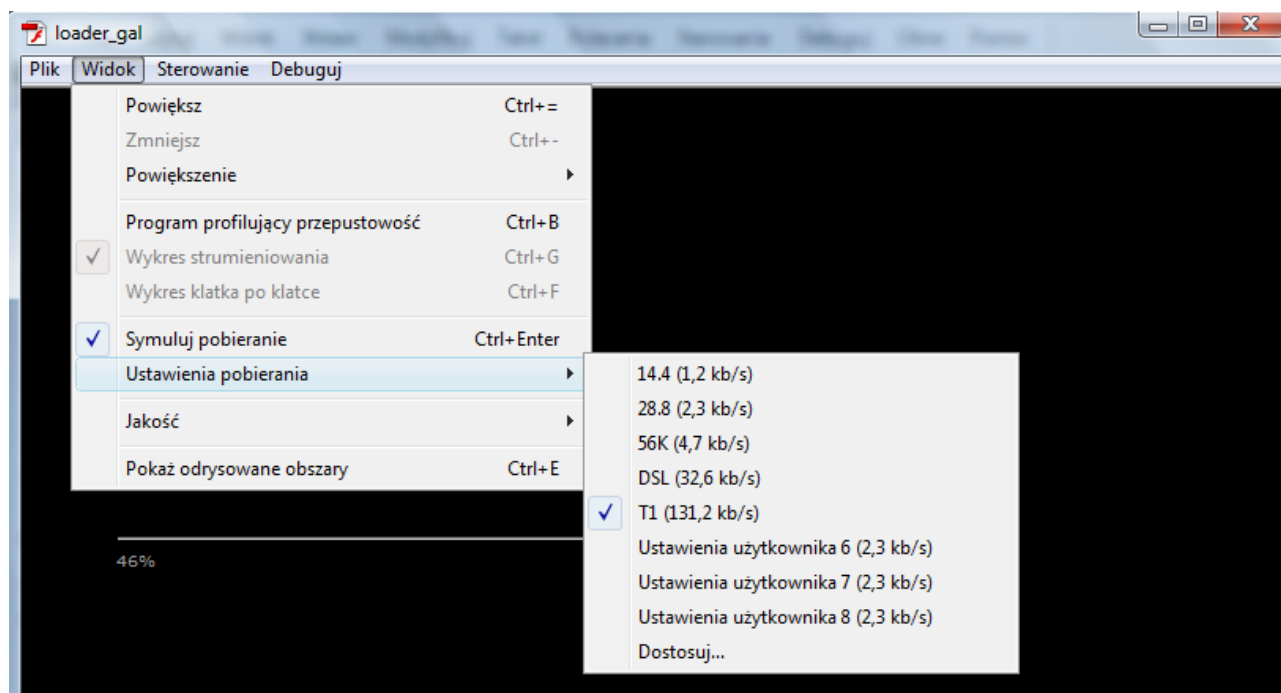
Skrypt jest już w całości gotowy. Powinien on wyglądać następująco:



```
1 import flash.events.ProgressEvent;
2 import flash.events.Event;
3
4 stop();
5
6 var movieLoader:Loader = new Loader();
7 var movieURL:String = "galeria.swf"
8 var movieURLRequest:URLRequest = new URLRequest(movieURL);
9 movieLoader.load(movieURLRequest);
10 //this.addChild(movieLoader);
11
12 movieLoader.contentLoaderInfo.addEventListener(ProgressEvent.PROGRESS, progressHandler);
13
14 function progressHandler(myevent:ProgressEvent):void{
15     var myprogress:Number= myevent.target.bytesLoaded/myevent.target.bytesTotal;
16     bar_mc.scaleX = myprogress;
17     myTextField_txt.text=Math.round(myprogress*100)+"%";
18 }
19
20 movieLoader.contentLoaderInfo.addEventListener(Event.COMPLETE, finished);
21
22 function finished(myevent:Event):void{
23     addChild(movieLoader);
24     removeChild(bar_mc);
25     removeChild(myTextField_txt);
26 }
```

Przetestuj film. Jeśli nie widzisz paska postępu ładowania a galeria pojawia się natychmiast to nic złego się nie dzieje. Plik .swf po prostu ładuje się szybko i sprawnie. Możemy podejrzeć jednak jak działa nasz efekt postępu ładowania i czy pojawiają się wszystkie elementy. Śledzenie takie wykonuje się za

pomocą narzędzi **Ustawienia pobierania** i **Symuluj pobieranie**, które dostępne są w oknie Flash Playera. W nowszych wersjach Flash/Animate oraz w aplikacji działającej na systemie OS niestety lokalizacja tych narzędzi może być zmieniona ☹.

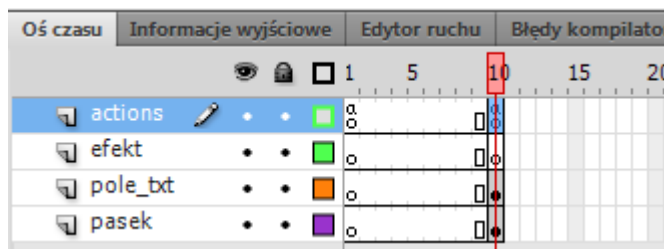


### Krok 6 – Modyfikacje na osi czasu

Teraz wykonany drugą część ćwiczenia – animację, która pojawi się zanim jakkolwiek treść pochodząca ze źródeł zewnętrznych zostanie załadowana. Wprowadzimy także element interaktywności – użytkownik sam zadecyduje kiedy przejść do etapu ładowania zawartości z zewnątrz.

Zaczynamy od dodania warstwy, której nadajemy nazwę *efekt*. W każdej warstwie wstawiamy klatkę kluczową w klatce nr 10 i przenosimy tam dotychczasową zawartość filmu (przeciągamy z klatki 1 do 10). Do klatki nr 1 w warstwie *action* wstawiamy akcję `stop()` ;

Listwa czasowa powinna teraz wyglądać następująco:



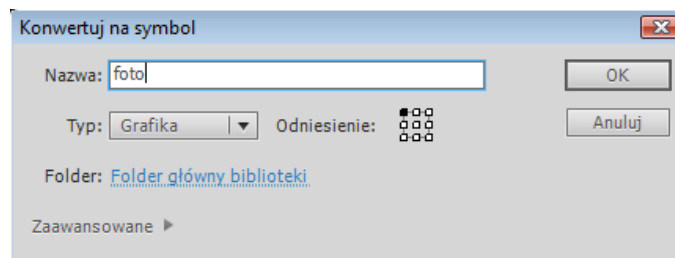
Zaznaczamy klatkę nr 1 warstwy *efekt* i importujemy na stół montażowy zdjęcie. Ustawiamy zdjęcie na środku stołu montażowego, najlepiej z pomocą panelu *Wyrównaj (align)*.

Zdjęcie konwertujemy do klipu (F8 lub „Konwertuj na symbol” – polecenie z menu podręcznego) i nadajemy mu nazwę „*efekt*”.

### Krok 7 – Praca w klipie – efekt falującej wody

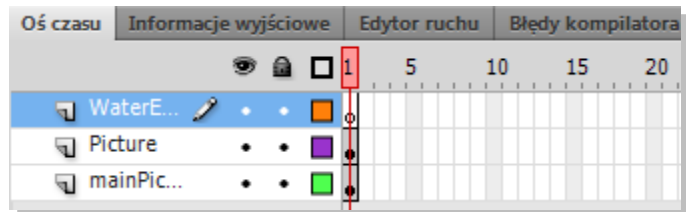
Klikamy dwa razy na utworzony klip w celu edycji go w miejscu.

Wewnątrz klipu zaznaczamy zdjęcie i konwertujemy je do grafiki (klawisz F8 lub „Konwertuj na symbol” – polecenie z menu podręcznego). Możemy grafikę nazwać „*foto*”.



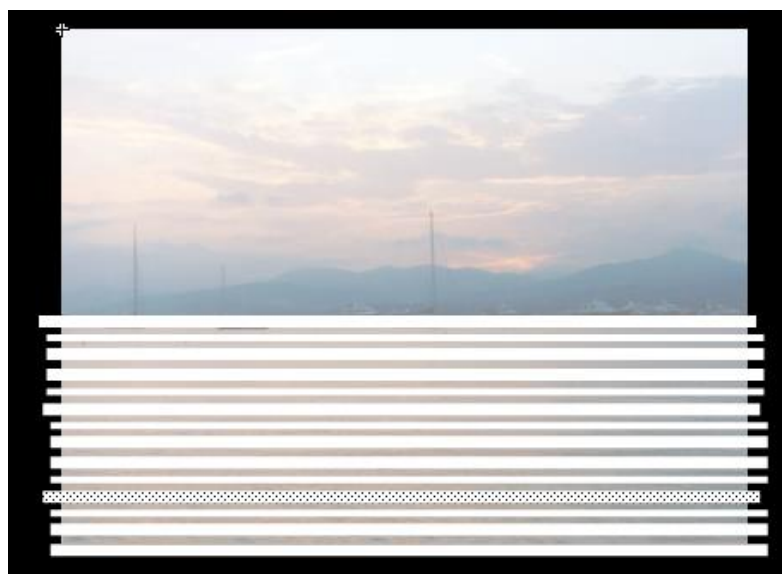
Zmieniamy nazwę warstwy na **mainPicture**. Klikamy na klatkę nr 1 i z menu podręcznego wybieramy polecenie **Kopiuj klatki**.

Wstawiamy nową warstwę i nazywamy ją **Picture**. Klikamy na klatkę nr 1 tej warstwy i wybieramy z menu podręcznego polecenie **Wklej klatki**.

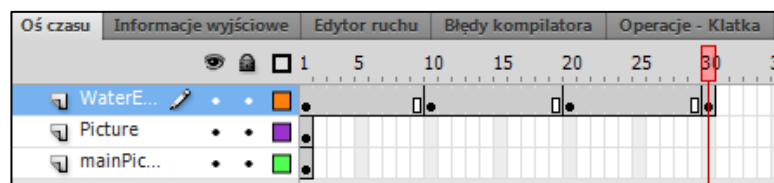


Wstawiamy kolejną nową warstwę i nazywamy ją **WaterEffect**.

Wybieramy narzędzie prostokąt (R) i na warstwie **WaterEffect** rysujemy na dolnej połowie zdjęcia bardzo wąskie prostokąty bez obramowania. Prostokąty muszą być bardzo gęsto położone:

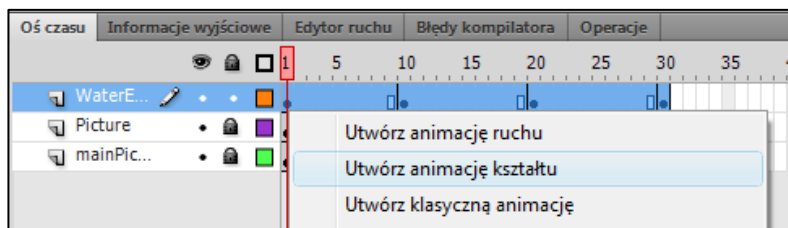


Klikamy kolejno na klatkę 10, 20 i 30 na warstwie **WaterEffect** i za każdym razem wstawiamy klatki kluczowe (F6).

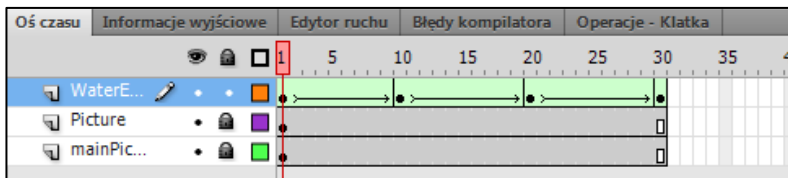


Wracamy do klatki 10, zaznaczamy ją (a tym samym zaznaczamy jej zawartość czyli paski) i używając strzałki na klawiaturze przenosimy nasz "efekt" nieznacznie w dół. Potem klikamy na klatkę 30, przenosimy nasz "efekt" nieznacznie w górę.

Zaznaczamy warstwę **WaterEffect**, i będąc na klatce prawym klawiszem myszy uruchamiamy menu podręczne, w którym wybieramy opcję **Utwórz animacje kształtu**.



W dwóch niższych warstwach wstawiamy klatki zwykłe (F5) w klatkach nr 30.



Zaznaczamy warstwę *WaterEffect* i nadajemy jej właściwość maska (opcja dostępna w menu podręcznym).

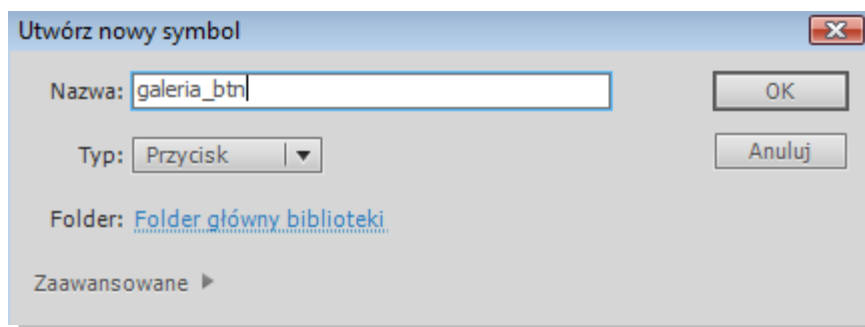
Na koniec pracy z klipem zablokuj i ukryj warstwy *WaterEffect* i *Picture*, a zdjęcie które znajduje się na warstwie *mainPicture* przesuń nieznacznie w dół (używając np. strzałki na klawiaturze). Przetestuj film.

#### Krok 8 – Przycisk

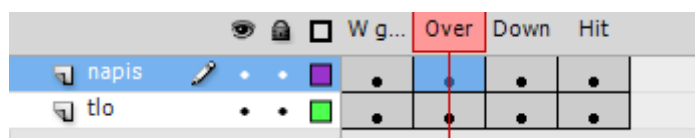
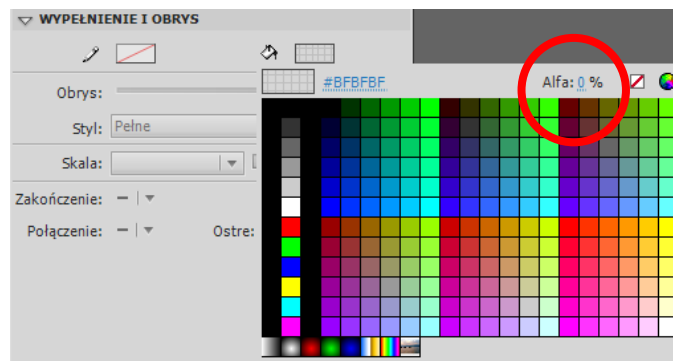
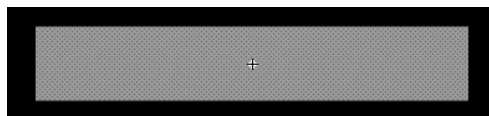
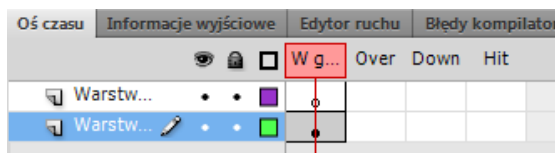
Wracamy do głównej sceny. Na początek dodajemy warstwę „*napis*”, w której na początek można umieścić ozdobny napis GALERIA.



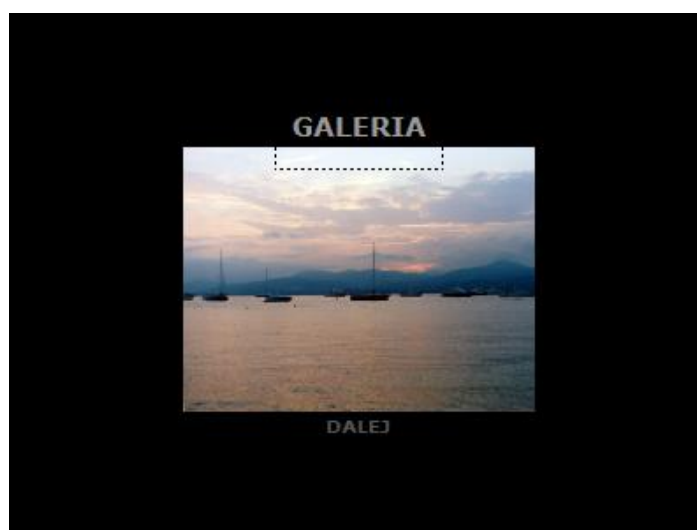
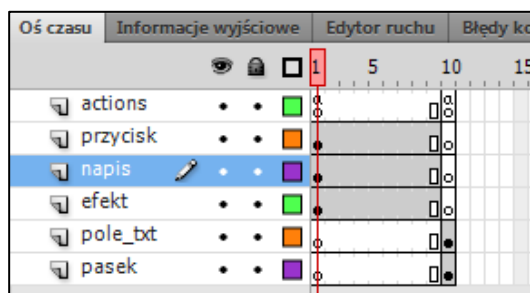
Teraz przystępujemy do projektowania przycisku. Wstawiamy nowy symbol typu przycisk:



I projektujemy go według własnego pomysłu. Ja użyłam do jego wykonania prostokąta z niewidocznym wypełnieniem (alfa=0) i bez obramowania. Widoczny będzie tylko sam napis „DALEJ”.



Przycisk ten będzie przełączał użytkownika do klatki nr 10, w której następuje ładowanie pliku z galerią. Umieszczamy przycisk na głównej scenie. Możemy wstawić dla niego osobną warstwę ale nie jest to konieczne. Może znaleźć się w tej samej warstwie co napis galeria. Instancji przycisku nadajemy nazwę „dalej\_btn”.



### Krok 8 – Akcje do przycisku

W warstwie „actions” w klatce nr 1 otwieramy panel **Operacji**. Poniżej akcji `stop()`; wpisujemy kod:

```
dalej_btn.addEventListener(MouseEvent.CLICK, goGallery);  
  
function goGallery(event:MouseEvent):void{  
    gotoAndPlay(10);  
}
```

I to koniec pracy. Przetestuj aplikację.

### 3 (zadanie dodatkowe) Opcja pełnego ekranu (full screen)

Wykonaj przycisk obsługujący opcję Full Screen, umieść go w projekcie galerii i nadaj mu nazwę instancji.

Następnie do klatki z przyciskiem/osobnej klatki nowej warstwy lub klatki z pozostałymi skryptami pisz kod obsługujący ten efekt:

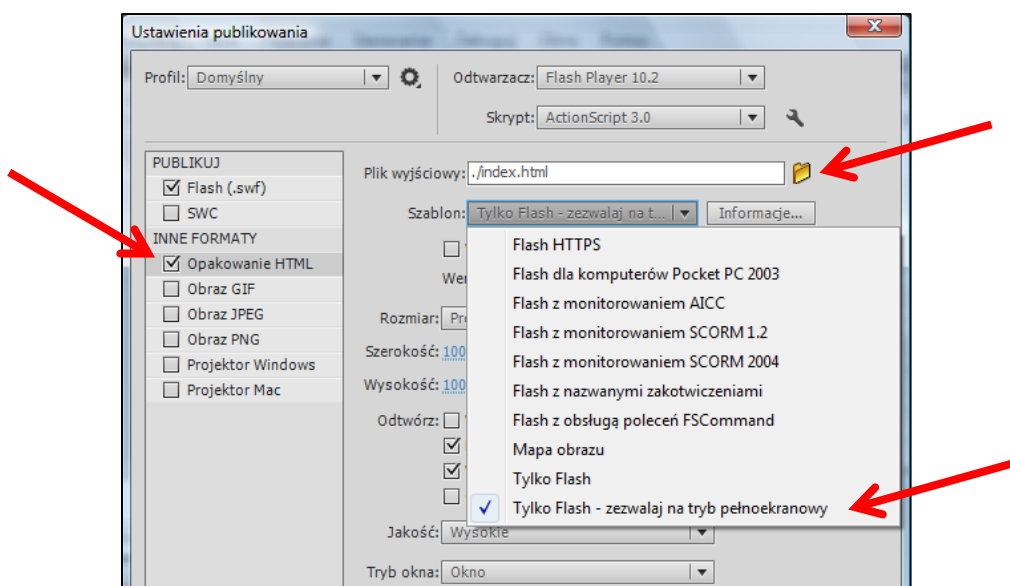
```
import flash.events.MouseEvent;

fullScreen_btn.addEventListener(MouseEvent.CLICK, fsMode);

var statusFS:Boolean = false;

function fsMode(event:MouseEvent):void{
    if(statusFS==false){
        stage.displayState = StageDisplayState.FULL_SCREEN;
        statusFS=true;
    }
    else
    { stage.displayState = StageDisplayState.NORMAL;
      statusFS=false;
    }
}
```

Użyj sekcji **Ustawienia Publikowania** (karta właściwości pliku) i w oknie **Ustawienia publikowania** dla formatu HTML użyj szablonu *Tylko Flash - zezwalaj na tryb pełnoekranowy*. Następnie opublikuj projekt. Publikowany jest zarówno plik SWF jak i HTML, domyślnie w lokalizacji w której zapisany jest plik FLA. Plik HTML wczytuje plik SWF, dlatego pliki te muszą być w jednym folderze. Jeśli chcemy zmienić lokalizację gotowych plików to musimy wskazać inną ścieżkę do zapisu. Możemy też nadać nazwę pliku wynikowego inną, niż nazw pliku roboczego.



Sprawdzamy czy działa tryb pełnoekranowy otwierając plik HTML w dowolnej przeglądarce internetowej.