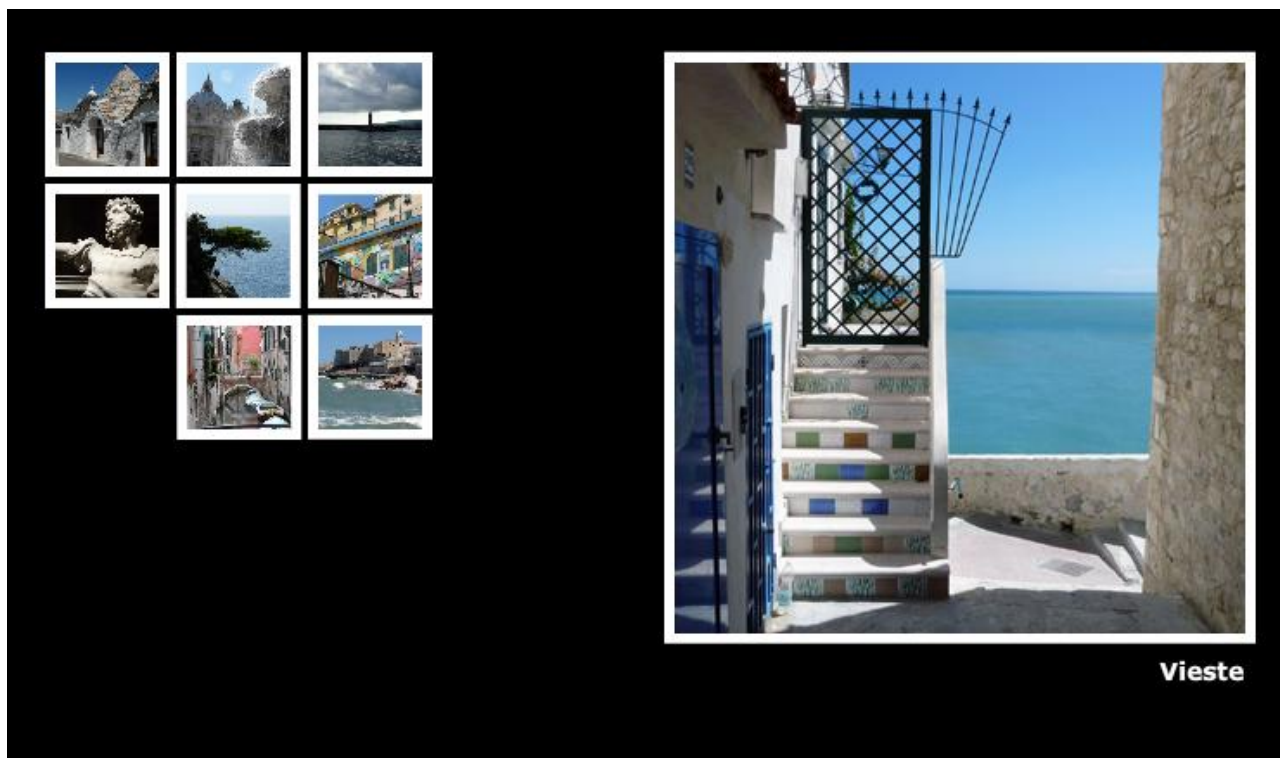


Ćwiczenie 1 – Galeria zdjęć



Pobierz przykład (<http://jsekulska.kis.p.lodz.pl/studia.htm>).

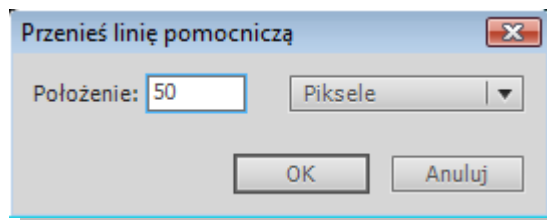
Krok 1

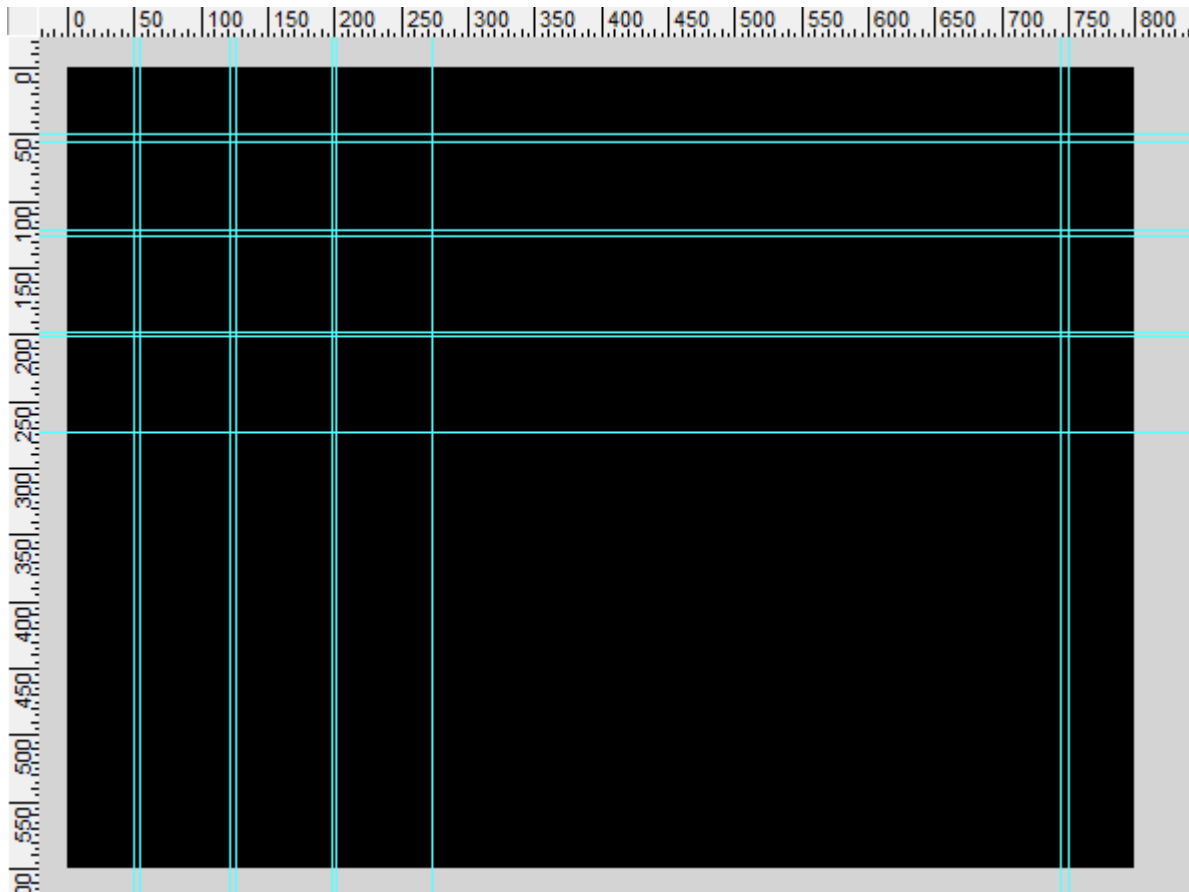
Ustawienie stołu montażowego

Otwieramy nowy plik i nazywamy go (np. gallery fla). Ustawiamy wielkość pola roboczego na 800 x 600 pikseli i ustawiamy ciemny kolor tła.

Umieszczamy siatkę prowadnic, która wyznaczy położenie „małych zdjęć”, czyli po prostu klipów, które zostaną utworzone w dalszej części ćwiczenia. Prowadnice pionowe i poziome umieszczone są w położeniu: 50, 122, 126, 198, 202 i 274. Dodajemy także prowadnice pionowe w położeniu 744 i 750 oraz dodatkową poziomą i pionową prowadnicę 56, które to wyznaczają nam krawędzie, do których będą przylegać duże zdjęcia wyświetlane po kliknięciu miniatur.

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.



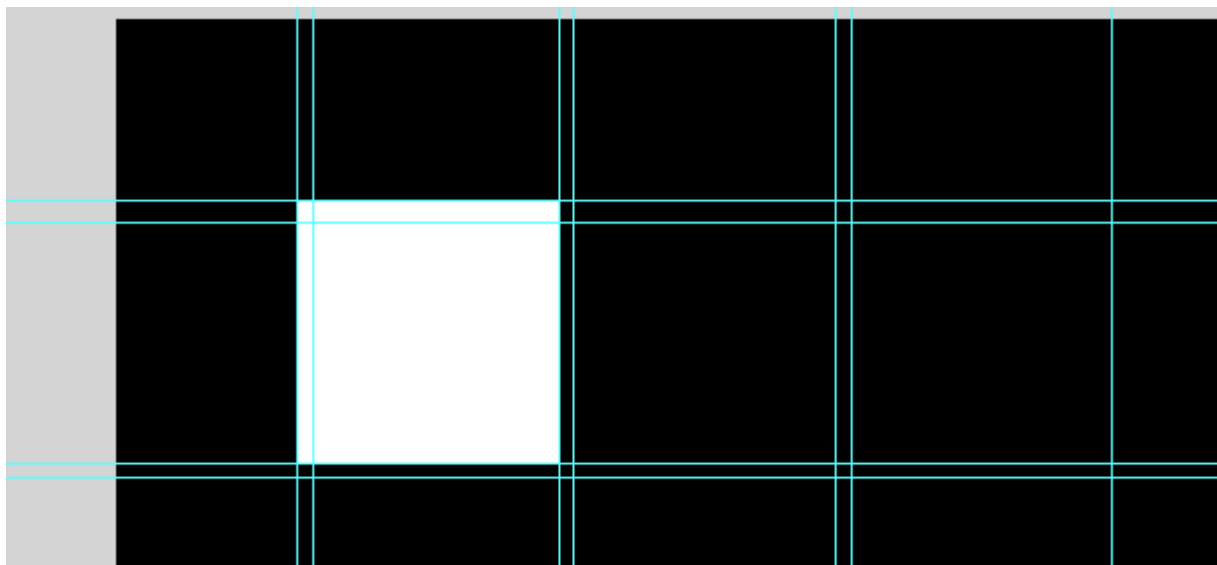


Importujemy zdjęcia do biblioteki - z menu **Plik** wybieramy polecenie **Importuj > Importuj do biblioteki...**

Krok 2

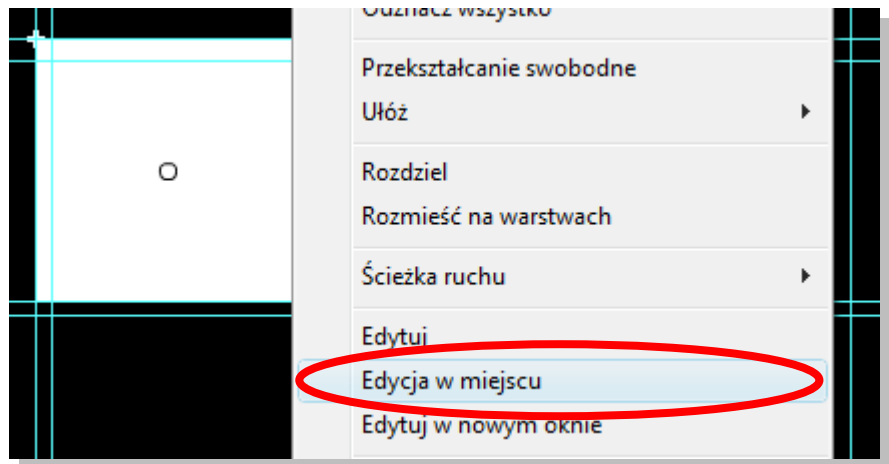
Tworzenie klipów

Na początek umieszczamy w pierwszym „oczku” siatki biały kwadrat o wielkości boku 72 px. Konwertujemy go do klipa filmowego (**F8**). Możemy go nazwać np. fot1 (a kolejne klipy fot2, fot3....). Będzie to pierwszy klip z całej serii klipów, które będą powiększały zdjęcia.

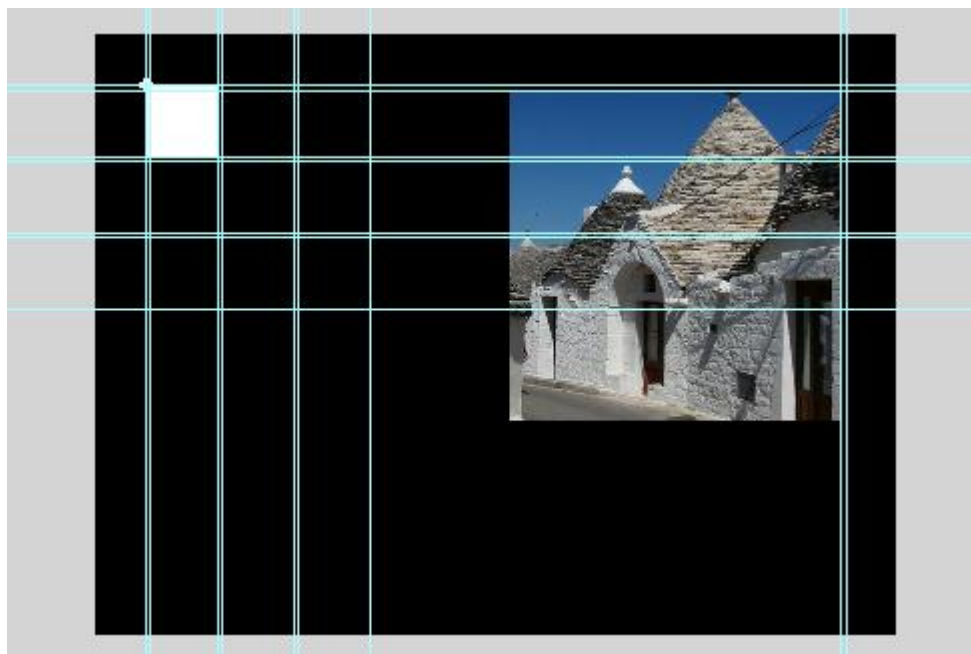
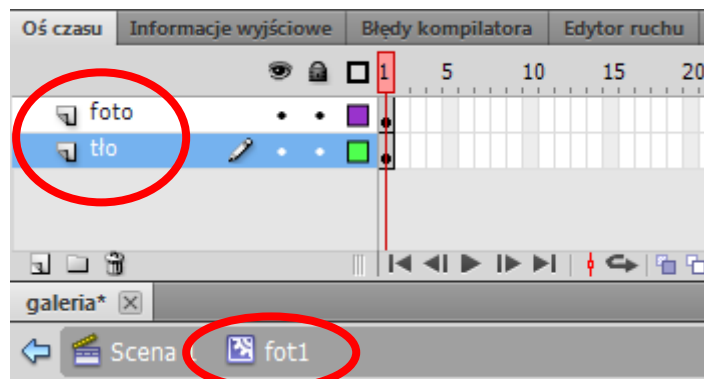


Edytujemy klip w miejscu – kliknij na klipie dwa razy lub wybierz z menu podręcznego polecenie **Edycja w miejscu**:

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

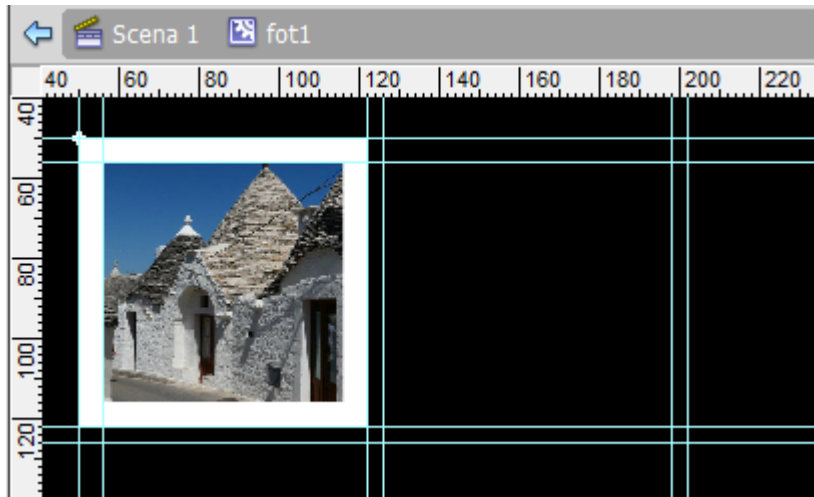


W klipie dodajemy warstwę, na której umieszczamy pierwsze zdjęcie (pobieramy je z biblioteki). Umieszczamy je z prawej strony pola roboczego przy pionowej prowadnicy „744” i poziomej „56”. Warstwy w klipie możemy nazwać „foto” i „tło”:



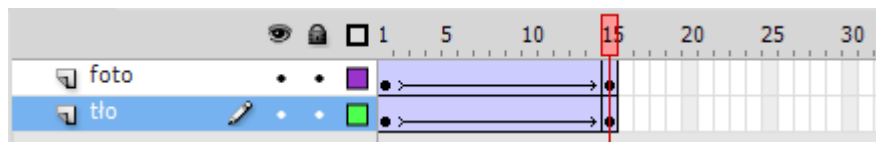
Zdjęcie jest duże (330x330 px) i tak będzie wyglądać po powiększeniu, natomiast musi wystartować jako miniatura (60x60 px). Wobec tego stosujemy animację ruchu (polecenie **Utwórz klasyczną animację**). Animacja ta powinna przebiegać do klatki nr 15, więc tam dodajemy klatkę kluczową (**F6**).

Wracamy do klatki nr 1, w której zdjęcie zmniejszamy do wielkości 60x60 pikseli i umieszczamy je dokładnie pośrodku białego kwadratu.

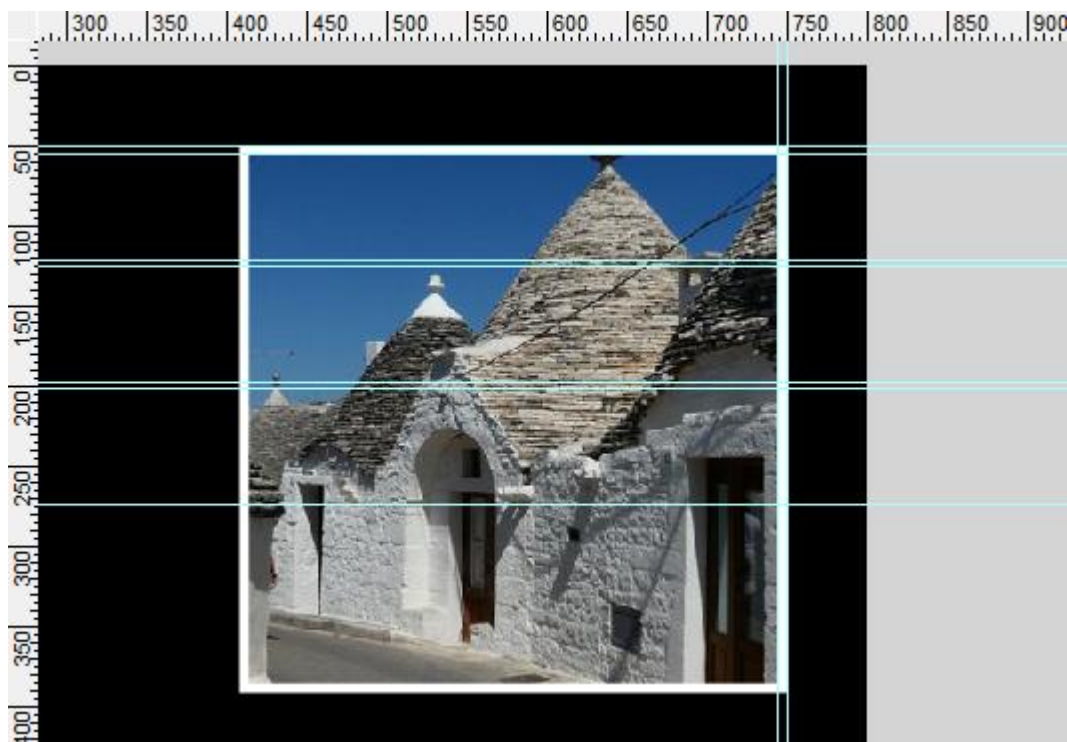


Podobną animację zastosujemy dla białego kwadratu - od klatki 1 do 15. W klatce nr 1 zachowujemy rozmiar 72x72 px, natomiast w klatce nr 15 powiększamy kwadrat do rozmiaru 342x342 px i umieszczamy go przy pionowej prowadnicy „750” i poziomej „50”.

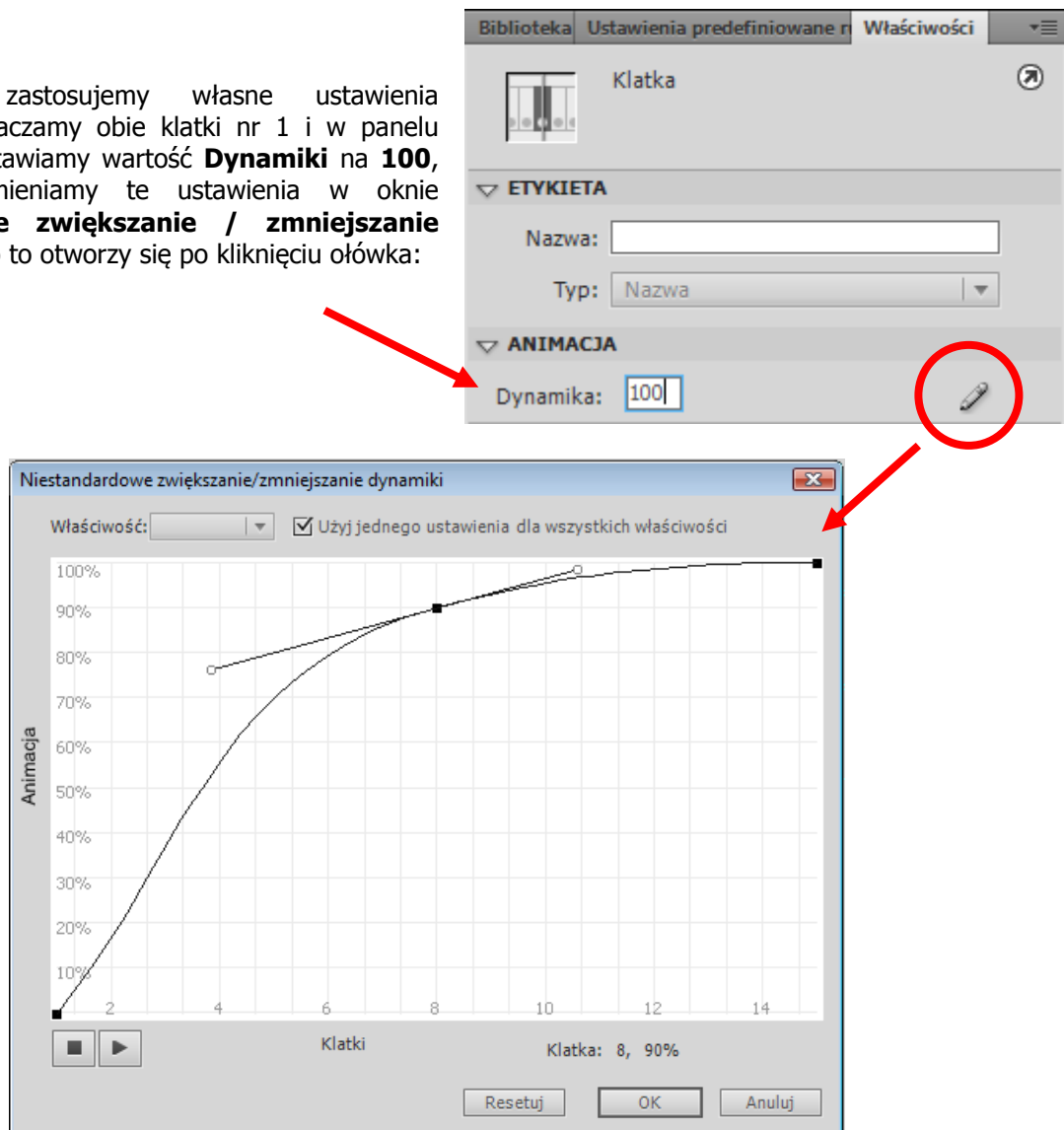
Oto obecny wygląd osi czasu klipu:



A to położenie elementów w klatce 15:



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

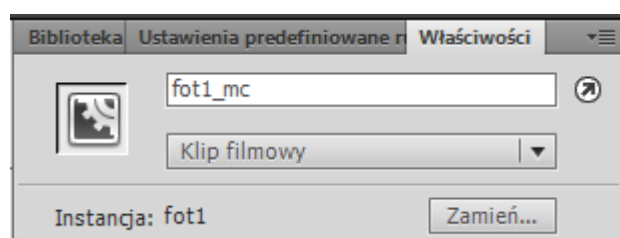


Teraz w obu warstwach kopiujemy klatkę nr 1 (**Kopiuj klatki**) i wklejamy ją w klatce nr 30 (**Wklej klatki**) i tym samym odwracamy animację. Dynamikę możemy zmienić też w klatkach nr 15.

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



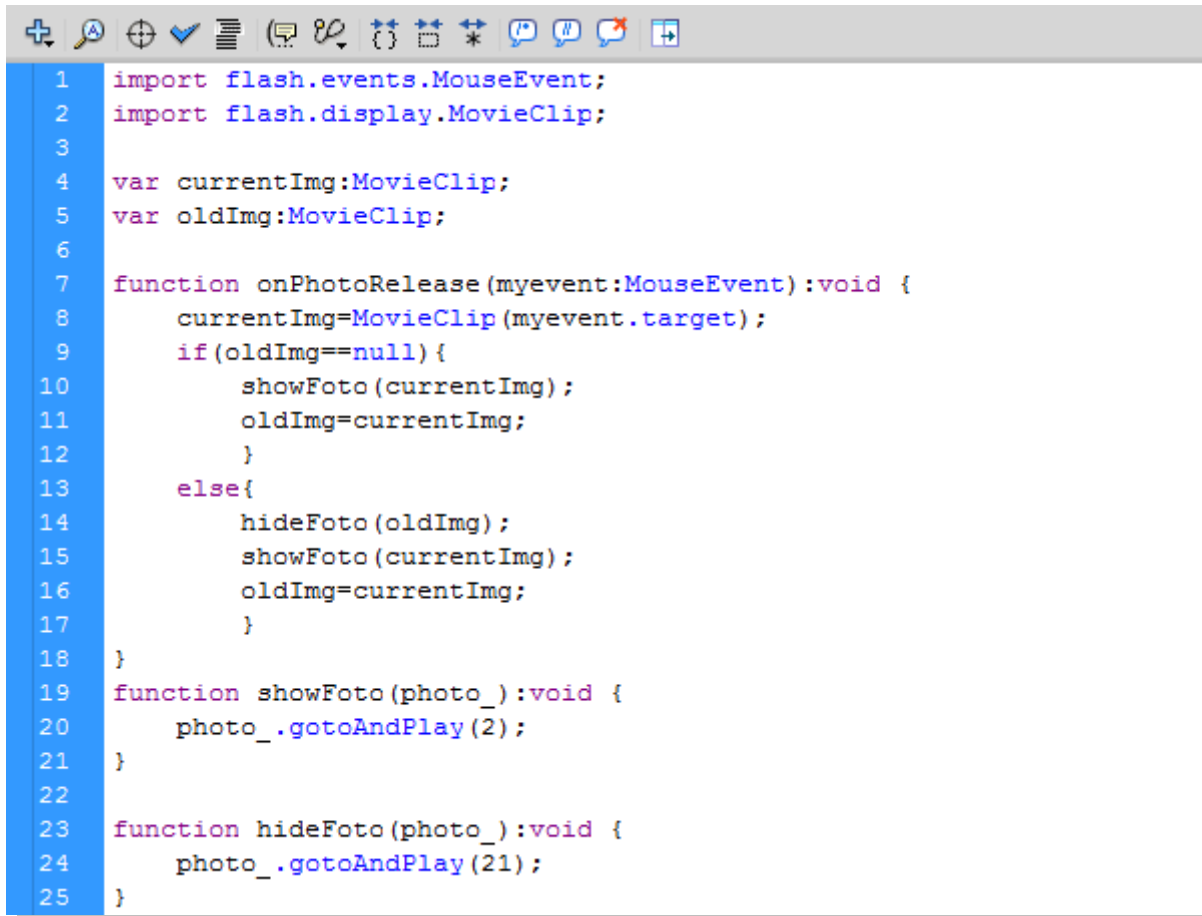
To koniec pracy z tym klipem. Wracamy więc do głównej sceny. Zaznaczamy umieszczony w niej klip i wprowadzamy nazwę jego instancji: **fot1_mc**.



W podobny sposób przygotowujemy kolejnych 8 klipów ze zdjęciami. Oczywiście dbając o odpowiednie rozmieszczenie zdjęć w siatce. Instancje klipów nazywamy odpowiednio fot2_mc, fot3_mc, fot4_mc itd.

Krok 3**Dodanie akcji**

Na głównej scenie dodajemy warstwę i nazywamy ją „akcje”. W klatce tej warstwy umieszczamy skrypt:



```
1 import flash.events.MouseEvent;
2 import flash.display.MovieClip;
3
4 var currentImg:MovieClip;
5 var oldImg:MovieClip;
6
7 function onPhotoRelease (myevent:MouseEvent) :void {
8     currentImg=MovieClip (myevent.target) ;
9     if (oldImg==null) {
10         showFoto (currentImg) ;
11         oldImg=currentImg;
12     }
13     else{
14         hideFoto (oldImg) ;
15         showFoto (currentImg) ;
16         oldImg=currentImg;
17     }
18 }
19 function showFoto(photo_) :void {
20     photo_.gotoAndPlay (2) ;
21 }
22
23 function hideFoto (photo_) :void {
24     photo_.gotoAndPlay (21) ;
25 }
```

Kod do skopiowania:

```
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{
        hideFoto (oldImg) ;
        showFoto (currentImg) ;
        oldImg=currentImg;
    }
}

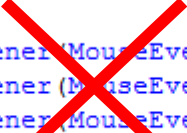
function showFoto(photo_) :void {
    photo_.gotoAndPlay (2) ;
}

function hideFoto (photo_) :void {
    photo_.gotoAndPlay (21) ;
}
```

Należy jeszcze określić, które obiekty będą reagowały i wykonywały nasze funkcje (będą odbiornikami zdarzenia myszki - *MouseEvent*). Są to oczywiście klipy ze zdjęciami umieszczone na scenie głównej i oznaczone nazwami instancji, po to aby były „widoczne” dla skryptów AS.

Na koniec prawie gotowego skryptu, który mamy w klatce 1, musimy jeszcze dodać obiekt nasłuchujący zdarzenia myszki (klikanie na klipy). Robimy to za pomocą metody **addEventListener**. Metoda ta ma dwa parametry – pierwszym z nich jest typ zdarzenia, które chcemy wykryć (dla nas jest to CLICK), a drugim jest funkcja reagująca na zdarzenie (w naszym przypadku jest to nasza funkcja główna **onPhotoRelease**). Kod, mógłby wyglądać następująco:

```
26
27 fot1_mc.addEventListener(MouseEvent.CLICK, onPhotoRelease);
28 fot2_mc.addEventListener(MouseEvent.CLICK, onPhotoRelease);
29 fot3_mc.addEventListener(MouseEvent.CLICK, onPhotoRelease);
```



Jest to kod dla tylko trzech klipów, a my mamy ich już 9, a w przyszłości mogłoby być ich więcej! Przy wielu liniach prawie identycznych instrukcji nie ma sensu pisać ich aż tylu. Powielenie instrukcji zrobimy więc w inny, dynamiczny sposób - za pomocą pętli.

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

```
34 var klipy:Array = new Array();
35 var i:Number=0;
36
37 while(1){
38     i++;
39     var nazwa:String = "fot"+i+"_mc";
40     var obj:DisplayObject = getChildByName(nazwa);
41
42     if (obj == null){
43         break;
44     }
45     klipy.push(obj);
46 }
47
48 for each (var element in klipy){
49     element.addEventListener(MouseEvent.CLICK, onPhotoRelease);
50 }
```

Kod do skopiowania:

```
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);
}
```

Krok 4

Modyfikacja akcji

Jeśli przetestujesz teraz film (Ctrl+Enter) to zobaczysz, że wszystko działa. Jednak kliknięcie na duże zdjęcie powoduje jakiś dziwny efekt – ono startuje znów od początku. Zróbmy modyfikację kodu po to aby kliknięte duże zdjęcie wracało na swoje miejsce wśród miniaturk. Modyfikujemy linie 14-16 poprzedniego kodu zagnieżdżając kolejną instrukcję warunkową (modyfikowany fragment obejmuje teraz linie 14-22):

```
7 function onPhotoRelease(myevent:MouseEvent):void {
8     currentImg=MovieClip(myevent.target);
9     if(oldImg==null){
10        showFoto(currentImg);
11        oldImg=currentImg;
12    }
13    else{
14        if(currentImg==oldImg){
15            hideFoto(currentImg);
16            oldImg=null;
17        }
18        else{
19            hideFoto(oldImg);
20            showFoto(currentImg);
21            oldImg=currentImg;
22        }
23    }
24 }
```

Kod do skopiowania:

```
else{
    if(currentImg==oldImg){
        hideFoto(currentImg);
        oldImg=null;
    }
    else{
        hideFoto(oldImg);
        showFoto(currentImg);
        oldImg=currentImg;
    }
}
```

Przetestuj film. Teraz galeria działa o wiele lepiej.
Zachowaj plik .swf do jednego z kolejnych ćwiczeń.