



Specyfikacja techniczna kreacji HTML5

www.gemius.pl

Parametry przekazywane do kreacji

Kreacje HTML osadzone są na stronie za pomocą znacznika <iframe>. W adresie do pliku kreacji przekazywane są parametry po znaku krzyżyka (#) w postaci NAZWA_PARAMETRU=WARTOŚĆ.

W celu odczytania parametrów przekazywanych do kreacji (m. in. skryptów do pomiaru kliknięć), należy dołączyć poniższy kod do kodu kreacji na początku <body>:

```
<script type="text/javascript">
    var parsed = (document.location.href.split('#')[1]||'').split('&');
    var params = parsed.reduce(function (params, param) {
        var param = param.split('=');
        params[param[0]] = decodeURIComponent(param.slice(1).join('='));
        return params;
    }, {});
</script>
```

clickTag'i

Kreacja, w głównym pliku HTML, powinna udostępniać zmienną, do której można przekazać adres docelowy kreacji. Do tej zmiennej należy przypisać wartość odczytaną z parametrów przekazanych automatycznie do kreacji, znajdującą się w zmiennej params.clickTag (po wykonaniu kodu odczytującego parametry).

Poniżej znajduje się przykład, w którym adres docelowy należy przypisać zmiennej globalnej clickTag:

```
<script>
    ...
    //umieść adres strony docelowej w zmiennej clickTag, np.:
    //clickTag='http://www.gemius.pl';
    clickTag=params.clickTag; //przekazuje wartość odczytaną z parametrów
    ...
</script>
```

Interakcje

Kreacje typu Expand lub Toplayer wymagają interakcji z kodem szablonu, który odpowiada za zmianę rozmiarów lub ukrycie kreacji. Do kreacji HTML5 przekazywane są za pomocą dodatkowych parametrów (analogicznie jak clickTag) nazwy funkcji, które należy wykonać w momencie odpowiedniego zdarzenia. W przypadku kreacji Expand są to parametry **doexpand** oraz **dolittle**, a w przypadku kreacji Toplayer **onCrossClick**.

Instrukcja odpowiedzialna za wykonanie tych funkcji ma postać:

```
window.parent.postMessage(params.NAZWA_PARAMETRU, '*');
```

Przykładowy fragment kodu kreacji typu Toplayer, w której po kliknięciu w element o id równym closeButton powinno nastąpić zamknięcie kreacji:

```
<script type="text/javascript">
  ...
  //dodanie wykonania funkcji zamykającej kreację po kliknięciu
  //w element o id="closeButton"
  document.getElementById('closeButton').
    addEventListener('click', function() {
      window.parent.postMessage(params.onCrossClick, '*');
    });
  ...
</script>
```

Przykładowy fragment kodu kreacji typu Expand, w której po najechaniu kursorem myszy na element o id równym expandButton powinno nastąpić rozwinięcie kreacji, a po opuszczeniu tego elementu przez kursor myszy powinien nastąpić powrót do rozmiaru wyjściowego:

```
<script type="text/javascript">
  ...
  //dodanie wykonania funkcji rozwijającej kreację po najechaniu
  //na element o id="expandButton"
  document.getElementById('expandButton').
    addEventListener('mouseenter', function() {
      window.parent.postMessage(params.doexpand, '*');
    });
  ...
</script>
```

```
//dodanie wykonania funkcji zwijającej kreację po zjechaniu
//z elementu o id="expandButton"
document.getElementById('expandButton').
    addEventListener('mouseleave', function() {
        window.parent.postMessage(params.dolittle, '*');
    });
...
</script>
```

UWAGA: Powyższe funkcje odpowiadają za sterowanie kontenera nadrzędnego, w którym osadzona jest na stronie kreacja. Dodatkowe akcje związane ze zmianą rozmiaru (np. zmiana układu graficznego) powinny być wykonane w ramach kreacji.

Elementy składowe kreacji

Kreacja HTML powinna składać się z jak najmniejszej liczby elementów. Im mniejsza liczba dodatkowych elementów, tym szybciej kreacja zostanie załadowana i wyświetlona końcowemu użytkownikowi.

Wszystkie odwołania do elementów składowych powinny być wykonywane za pomocą adresów względnych (wyjątkiem są elementy umieszczone na zewnętrznych serwerach).

Arkusze stylów i biblioteki JS mogą zostać umieszczone bezpośrednio w treści głównego pliku html, co zmniejszy liczbę elementów składowych.

Elementy graficzne również mogą zostać umieszczone w głównym pliku (poprzez zakodowanie za pomocą Base64), jednak może to w znacznym stopniu zwiększyć rozmiar pliku oraz czas ładowania kreacji w przeglądarce użytkownika.

UWAGA: Poniższe ograniczenie jest tymczasowe, do czasu umożliwienia uploadowania wszystkich elementów kreacji jako archiwum ZIP.

Wszystkie elementy składowe muszą znajdować się w jednym katalogu (tym samym co główny plik html).

Szablony kreacji

Szablony wspierające tego typu kreacje zawierają w nazwie **[HTML5]**. Jeżeli na koncie nie ma szablonu odpowiedniego do zdefiniowania konkretnej formy kreacji, prosimy o kontakt z Działem Wsparcia Technicznego (help@gemius.com).

Przykładowe kreacje

Poniżej znajdują się kody przykładowych kreacji z poprawną obsługą przekierowania na stronę docelową.

Kreacja wykorzystująca element <a>

Najprostszym sposobem wykonania przekierowania jest opakowanie całej kreacji w element <a>

```
<html>
  <head></head>
  <body>
    <a href="#" target="_blank" id="creativelink">
      
    </a>

    <script type="text/javascript">
      var parsed = (document.location.href.split('#')[1]||'').split('&');
      var params = parsed.reduce(function (params, param) {
        var param = param.split('=');
        params[param[0]] = decodeURIComponent(param.slice(1).join('='));
        return params;
      }, {});

      //ustawienie adresu docelowego w linku HTML
      document.getElementById('creativelink').href = params.clickTag;
    </script>
  </body>
</html>
```

Konwerter Swiffy

Swiffy umożliwia konwersję kreacji wykonanych w technologii Flash do postaci HTML. Konwersja uwzględnia parametry przekazywane do plików Flash'owych (tzw. flashVars'y), więc wystarczy przekazać parametry odczytane w kodzie strony.

Poniżej znajduje się przykład kreacji typu Expand (plik swf był zgodny ze specyfikacją kreacji typu Expand):

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>Swiffy Output</title>
    <script type="text/javascript"
src="https://www.gstatic.com/swiffy/v7.3.0/runtime.js"></script>
    <script>
      swiffyobject = {"as3":true,"frameRate":12,...};
</script>
    <style>html, body {width: 100%; height: 100%;}</style>
  </head>
  <body style="margin: 0; overflow: hidden;">
    <div id="swiffycontainer" style="width: 750px; height: 300px">
</div>
    <script>
//odczytanie parametrów
var parsed = (document.location.href.split('#')[1]||'').split('&');
  var params = parsed.reduce(function (params, param) {
    var param = param.split('=');
    params[param[0]] = decodeURIComponent(param.slice(1).join('='));
    return params;
  }, {});

//oryginalny kod z konwertera
var stage = new swiffy.Stage(document.getElementById('swiffycontainer'),
  swiffyobject, {});

//ustawienie przezroczystego koloru tła
stage.setBackground(null);
//przekazanie clickTag'a
stage.setFlashVars('clickTag='+params.clickTag)
//przekazanie nazw funkcji odpowiedzialnych za zmianę rozmiaru
```

```
stage.setFlashVars('doexpand=html_doexpand')
stage.setFlashVars('dolittle=html_dolittle')

//oryginalny kod z konwertera
stage.start();

//zdefiniowanie funkcji sterujących zmianą rozmiaru
html_doexpand=function (){
    window.parent.postMessage(params.doexpand, '*');
};
html_dolittle=function (){
    window.parent.postMessage(params.dolittle, '*');
};
</script>
</body>
</html>
```



Gemius Polska Sp. z o.o.

ul. Postępu 18 B
02-676 Warszawa
Tel: + 48 22 390 90 90
+ 48 22 378 30 50
Fax: + 48 22 874 41 01

ok@gemius.com
www.gemius.com