

## 1) Zdarzenia w JavaScript

Zdarzenia są sygnałami generowanymi w chwili wykonywania ściśle określonych czynności. JavaScript dysponuje narzędziami do informowania o zdarzeniach oraz pozwala skryptom na obsługę konkretnych zdarzeń. Przykładowo, zdarzenia mogą się pojawić, gdy użytkownik postawi wskaźnik myszy na tekście hiperpołączenia, zmieni dane w formularzu albo gdy zostanie zakończone ładowanie strony.

Nazwa Zdarzenia	Opis
abort	Zachodzi, gdy użytkownik przerywa ładowanie obrazka.
blur	Zachodzi, gdy <a href="#">miejsce wprowadzania</a> zostaje przesunięte do innego pola formularza lub okna.
Click	Zachodzi, gdy użytkownik kliknie na połączeniu lub elemencie formularza.
Change	Zachodzi, gdy zawartość pola formularza ulegnie zmianie.
Error	Zachodzi, gdy podczas ładowania strony lub obrazka wystąpi błąd.
Focus	Zachodzi, gdy miejsce wprowadzania zostaje umieszczone w polu.
Load	Zachodzi, gdy ładowanie strony zostanie zakończone.
Mouseout	Zachodzi, gdy użytkownik przesunie wskaźnik myszy poza tekst połączenia lub aktywny obszar mapy graficznej.
Mouseover	Zachodzi, gdy użytkownik przesunie wskaźnik myszy poprzez tekst połączenia lub aktywny obszar mapy graficznej.
Reset	Zachodzi, gdy użytkownik zażąda wyczyszczenia pól formularza poprzez kliknięcie na przycisku Reset.

---

Select                    Zachodzi, gdy użytkownik wybierze jedno z pól formularza.

---

Submit                    Zachodzi, gdy dane z formularza zostają wysłane (zazwyczaj po kliknięciu przycisku Submit).

---

Unload                    Zachodzi, gdy użytkownik zmienia wyświetlaną stronę.

Kiedy coś dzieje się na stronie WWW, przeglądarka generuje zdarzenie (z ang. event). Jako skutek, powstaje obiekt zdarzenia, który to zdarzenie opisuje. Jednocześnie, badane jest, czy istnieje procedura obsługi zdarzenia – jeśli tak, jest wykonywana. Można zatem powiedzieć, że procedury obsługi zdarzeń są to pisane przez programistę skrypty, będące dodatkowymi argumentami określonych etykiet HTML. Te, tworzone przez programistę procedury obsługi wykonywane są w chwili wystąpienia odpowiedniego zdarzenia. Oto podstawowa składnia JavaScriptu, używana do definiowania procedur obsługi zdarzeń.

**<Znacznik Atrybuty ProceduraObsługi="JS\_Program" >**

Przykład:

**<div onclick = "alert(Przecież było napisane: nie klikać!);">**

---

<b>Nazwa zdarzenia</b>	<b>Opis</b>	<b>Elementy HTML obsługujące zdarzenie</b>
<b>onabort</b>	Procedura zostanie uruchomiona, gdy ładowanie obrazu zostanie przerwane.	<b>img</b>
<b>onblur</b>	Procedura zostanie uruchomiona, gdy element utraci fokus, np. gdy miejsce wprowadzania zostaje przesunięte do innego pola formularza lub okna.	Większość elementów strony.
<b>onchange</b>	Procedura zostanie uruchomiona, gdy element utraci fokus i jednocześnie zmieni się zawartość tego elementu (np. pola tekstowego).	<b>input, select, textarea</b>
<b>onclick</b>	Procedura zostanie uruchomiona, gdy element zostanie kliknięty.	Większość elementów

---

strony.

---

<b>ondblclick</b>	Procedura zostanie uruchomiona, gdy element zostanie dwukrotnie kliknięty.	Większość elementów strony.
<b>onerror</b>	Procedura zostanie uruchomiona, gdy przy ładowaniu obrazu wystąpi błąd.	<b>img</b>
<b>onfocus</b>	Procedura zostanie uruchomiona, gdy element otrzyma fokus.	Większość elementów strony.
<b>onkeydown</b>	Procedura zostanie uruchomiona, gdy zostanie naciśnięty klawisz klawiatury.	Większość elementów strony.
<b>onkeypress</b>	Procedura zostanie uruchomiona, gdy klawisz klawiatury zostanie naciśnięty i puszczony.	Większość elementów strony.
<b>onkeyup</b>	Procedura zostanie uruchomiona, gdy klawisz klawiatury zostanie puszczony.	Większość elementów strony.
<b>onload</b>	Procedura zostanie uruchomiona, gdy przeglądarka zakończy ładowanie strony lub ramki.	<b>body,</b> <b>frameset</b>
<b>onmousedown</b>	Procedura zostanie uruchomiona, gdy klawisz myszy zostanie naciśnięty nad elementem.	Większość elementów strony.
<b>onmousemove</b>	Procedura zostanie uruchomiona, gdy kursor myszy jest przesuwany nad elementem.	Większość elementów strony.
<b>onmouseover</b>	Procedura zostanie uruchomiona, gdy kursor myszy wejdzie w obszar elementu.	Większość elementów strony.

---

<b>onmouseup</b>	Procedura zostanie uruchomiona, gdy klawisz myszy zostanie zwolniony nad elementem.	Większość elementów strony.
<b>onreset</b>	Procedura zostanie uruchomiona, gdy formularz zostanie zresetowany (np. przez kliknięcie przycisku <i>reset</i> ).	<b>form</b>
<b>onresize</b>	Procedura zostanie uruchomiona, gdy zmieni się rozmiar okna.	<b>body, frameset</b>
<b>onselect</b>	Procedura zostanie uruchomiona, gdy fragment tekstu zostanie zaznaczony.	<b>input (text, textarea)</b>
<b>onsubmit</b>	Procedura zostanie uruchomiona, gdy formularz zostanie wysłany (np. przez kliknięcie przycisku <i>submit</i> ).	<b>form</b>
<b>onunload</b>	Procedura zostanie uruchomiona, gdy przeglądarka usunie bieżący dokument.	<b>body, frameset</b>

Przykład zdarzenia "onchange":

```
<INPUT TYPE="text" SIZE="8" value="0" onChange="
    if(parseInt(this.value)<=5) {
        alert('Proszę podać liczbę większą od 5!');
    }
">
```

Słowo kluczowe **this** odnosi się tutaj do aktualnego obiektu. W rozważanym przykładzie tym obiektem jest bieżące pole formularza (które to pole jest jednocześnie obiektem i właściwością obiektu wyższego rzędu – formularza).

Przykład zdarzenia "onclick" na elemencie <div>:

```
<div id="dataDiv"
    style="background-color:silver;width:200px;heighth:30px;
        text-align:center;"
    onclick="alert('Kliknąłeś w element strony!');">
```

Ten sam przykład tylko z rozbudowaną funkcjonalnością:

```
<script>
function dataDivClick()
{
    var dataDiv = document.getElementById("dataDiv");
    dataDiv.innerHTML = "Warstwa została kliknięta.";
}
</script>
<div id="dataDiv"
    style="background-color:silver;width:200px;heigth:30px;
        text-align:center;"
    onclick="dataDivClick();"
>
```

Przykład utworzenia linku na zdarzeniu "onclick":

```
<p style="cursor:pointer;"
    onclick="location.href='http://www.fais.uj.edu.pl'">
Witryna Wydziału Fizyki, Astronomii i Informatyki Stosowanej
</p>
```

## 2) Emulowanie zdarzeń:

Emulowanie zdarzeń może się okazać przydatne, gdy np. trzeba wysłać formularz bez konieczności prośbienia użytkownika o kliknięcie na przycisku Submit lub gdy trzeba np. zmienić miejsce wprowadzania informacji w zależności od czynności podejmowanych przez użytkownika. I tak, po wypełnieniu formularza, użytkownik mógłby kliknąć na przycisk Zamów, co powinno wywołać odpowiedni skrypt sprawdzający, czy formularz jest poprawnie wypełniony. Dopiero pozytywny wynik tego sprawdzenia generowałby (emulowałby) zdarzenie submit w celu ostatecznego wysłania danych z formularza do serwera.

Poniżej zestawiono listę najczęściej wykorzystywanych funkcji, dostępnych w JavaScriptcie, emulujących zdarzenia:

- blur()
- click()
- focus()
- reset()
- select()
- submit()

Przykład emulowania zdarzenia "click":

```
<form>
  <input type="checkbox" id="myCheck" onmouseover="myFunction()" onclick="
alert('kliknąłeś mnie')">
</form>
<script>
function myFunction() {
    document.getElementById("myCheck").click();
}
</script>
```

**Zadanie:**

*Proszę napisać skrypt, który będzie za pomocą zdarzenia "onload" elementu <body> losowo otwierał stronę internetową o określonym adresie URL (proszę zdefiniować przynajmniej 4 takie adresy). Adresy proszę trzymać w tablicy, a realizację otwarcia strony wykonać za pomocą metody "window.open(array[], string);".*