

## Funkcje anonimowe

Wpisany przez Patryk yarpo Jar  
poniedziałek, 01 lutego 2010 12:38

---

JS jako jeden z najbardziej ekspresyjnych języków posiada ciekawy i często wygodny mechanizm funkcji anonimowych.

Polega to na tworzeniu funkcji, które nie mają nazwy i są jakby przypisane do konkretnego miejsca w kodzie.

Spójźmy na przykład (wykorzystanie zwykłych funkcji):

```
{codecitation class='brush: js'}function example1() {
```

```
    alert("&quot;example 1&quot;");
```

```
}
```

```
function example2() {
```

```
    // tu cos robimy
```

## Funkcje anonimowe

Wpisany przez Patryk yarpo Jar  
poniedziałek, 01 lutego 2010 12:38

---

```
return example1();  
  
}
```

```
example2();{/codecitation}
```

Założmy teraz, że funkcja `example1` jest wywoływana tylko w jednym miejscu, co więcej jej funkcjonalność jest ściśle związana z `example2`.

Możemy zatem zrobić coś takiego:

```
{codecitation class='brush: js'}function example2() {  
  
    // tu cos robimy  
  
    return function() { //2  
  
        alert("&quot;example 1&quot;"); // 3
```

## Funkcje anonimowe

Wpisany przez Patryk yarpo Jar  
poniedziałek, 01 lutego 2010 12:38

---

```
}
```

```
}
```

```
example2(); // 1{/codecitation}
```

Cóż to spowoduje? Nic. Dlaczego?! (Nic, to nie jest najlepsze określenie. Coś się stanie, jednak nie zobaczysz niczego).

Przypatrzmy się, co tu się dzieje:

1. wywołujemy funkcję `example2()`

2. funkcja ta zwraca funkcję. **Nie wynik** działania funkcji, a funkcję. Czyli, jak zrobimy tak:

```
{codecitation class='brush: js'}function example2() {
```

```
// tu cos robimy
```

```
return function() { //2
```

## Funkcje anonimowe

Wpisany przez Patryk yarpo Jar  
poniedziałek, 01 lutego 2010 12:38

---

```
    alert("example 1"); // 3

}

}
```

```
var e = example2(); // 1
```

```
e();
```

```
e();{/codecitation}
```

to uzyskamy dwukrotnie alert "example 1". Co zrobiliśmy? Przypisaliśmy do zmiennej e referencję na funkcję (teraz ta zmienna ma w sobie wartość adresu, gdzie ta funkcja się znajduje w pamięci. Jeśli wywołasz funkcję - przez dodanie `()` za nazwą zmiennej `e` - zostanie wywołana funkcja przypisana pod zmienną `e`).

W poprzednim przypadku (tym z funkcją jawnie zdefiniowaną i nazwaną example1), moglibyśmy to otrzymać tak:

```
{/codecitation class='brush: js'}
```

## Funkcje anonimowe

Wpisany przez Patryk yarpo Jar  
poniedziałek, 01 lutego 2010 12:38

---

```
function example2() {
```

```
    // tu cos robimy
```

```
    return example1; // zwracamy referencję na tę funkcję, nie wynik jej działania
```

```
}
```

Innymi słowy operator `()` "aktywizuje" funkcję.

Aby od razu działało jak przykład 1:

```
{codecitation class='brush: js'} function example2() {
```

```
    return (function() { // 1
```

```
        alert("example 1");
```

```
    })(); // 2
```

```
}
```

## Funkcje anonimowe

Wpisany przez Patryk yarpo Jar  
poniedziałek, 01 lutego 2010 12:38

---

```
example2();{/codecitation}
```

1. całe ciało anonimowej funkcji objęliśmy w nawiasy.
2. dodaliśmy za ciałem funkcji operator wywołania `()'`.

Wynik powyższego kodu będzie identyczny do działania pierwszego listingu z tego artykułu.

To oczywiście nie wszystko na temat funkcji anonimowych. Ale o innych rzeczach (rekurencja, sprytny triki) napiszę kiedy indziej :)

Warto pobawić się samemu. Wujek google też z pewnością pomoże :)