

Technologie internetowe

Javascript

Paweł Rajba

pawel@ii.uni.wroc.pl

<http://www.kursy24.eu/>

Plan wykładu

- Obiektość w Javascript
 - Wprowadzenie
 - Tworzenie obiektu
 - Prototypy
- Przykłady
- CoffeeScript
- Literatura

Wprowadzenie

- Implementacja nie jest tak obszerna jak w np. Javie
- Różnice wynikają też z innego modelu (język interpretowany)
- Najpierw przyjrzymy się podstawom
- ... a potem zobaczymy, jak tego używać

Tworzenie obiektu

- Typem podstawowym jest Object
- Ponieważ JavaScript jest interpretowany, właściwości można dodawać dynamicznie.
- Utworzenie prostego obiektu:

```
obj = new Object;  
obj.x = 1;  
obj.y = 2;
```
- Znaczenie słowa `this` jest standardowe
- Obiekt tworzymy operatorem `new`

Tworzenie obiektu

- Konstruktor jest dowolna funkcja
 - Możemy przekazywać parametry
- Przykład

```
function Point( nx, ny )
{
    this.x = nx;
    this.y = ny;
}
var p = new Point( 1, 2 );
```

Tworzenie obiektu

- Możemy także użyć literałów:

```
person = {  
  imie : "Jan",  
  nazwisko : "Kowalski",  
  stan : ["wolny", "niewolny"],  
  pokaz : function() { alert(imie+" "+nazwisko); }  
};
```

```
var rectangle = {  
  upperLeft : { x : 2, y : 2 },  
  lowerRight : { x : 4, y : 4 }  
}  
alert(rectangle.upperLeft.x)
```

Prototypy

- Obiekt reprezentujący strukturę obiektu
 - Odpowiednik klasy (w pewnym sensie)
- Możemy definiować pola i metody związane z klasą
 - Pola zdefiniowane są wspólne dla wszystkich klas (odpowiednik statyczności)
- Przykłady:

```
Punkt.prototype.liczba = 1;  
Punkt.prototype.odleglosc = Punkt_odleglosc;  
Punkt.prototype.odleglosc = function( p ) { ... }
```

Przykłady

- `Prototypes.html`
- `Singleton.html`
- `ModulePattern.html`
- `RevealingModulePattern.html`
- `RevealingPrototypePattern.html`

CoffeeScript

- Język składniowo bardziej obiektywy
- W pełni kompilowany do JavaScript
- Ciekawa propozycja
- Oglądamy *<http://coffeescript.org/>*

Do przeczytania

- Ciekawe omówienie przedstawionych wzorców:

<http://www.klauskomenda.com/code/javascript-programming-patterns/>

- Obszerny podręcznik nt. wzorców w JavaScript:

<http://www.addyosmani.com/resources/essentialjsdesignpatterns/book/>

- Mniej obszerny przegląd implementacji wzorców w JavaScript:

<http://javascriptpatterns.org/>

- Wszelchnica różnych aspektów JavaScript:

<http://www.crockford.com/javascript/>

- Stara, dobra dokumentacja do ECMAScript:

<http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf>