

Javascript posiada wiele sposobów na tworzenie obiektów. Jednym z nich jest wzorzec fabryki obiektów (czasem zwany wzorcem modułu). Chciałbym pokazać, jak można wykorzystując ten wzorzec stworzyć IMO wygodniejsze obiekty.

### Kod prostego obiektu

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

    var yPerson = function()
    fName( name ) {
    }
    function fAge( age )
    {
    }
    return {
    {
        var sName,
        sName = name || sName;
        {
            nAge = age || nAge;
            name : fName,
            nAge;
            return sName;
            return nAge;
            age : fAge
        };
    };
    };

{/codecitation}
```

Jak skorzystać z takiego kodu? Całkiem przyjemnie:

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

    var oPerson = yPerson(); oPerson.name('Jan Kowalski'); oPerson.age(26);
    alert(oPerson.name() + ' ' + oPerson.age())

{/codecitation}
```

### Do czego dążymy?

Chcemy uzyskać ostatecznie obiekt, którego będziemy mogli używać w ten sposób:

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

var oPerson = yPerson(); oPerson.name('Jan Kowalski').age(26); alert(oPerson.name() + ' '
+ oPerson.age())

{/codecitation}
```

Co prawda, przy tak prostym obiekcie zyskaliśmy tylko jedną linię. Jednak w przypadku bardziej złożonych struktur można naprawdę oszczędzić sobie spoor kodu. Wzorzec łańcuchowy jest powszechnie wykorzystywany w jQuery.

## Jak to osiągnąć?

Aby osiągnąć taki efekt należy po prostu w każdej metodzie zwracać referencję na obiekt, którego używamy.

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

var yPerson = function() { var sName, nAge; // tworz obiekt var that = {
name : fName, age : fAge }; function fIsSetter( data ) { return
('undefined' !== typeof data); } function fName( name ) { if (fIsSetter(name))
{ sName = name || sName; return that; // zwroc referencje na samego siebie
} return sName; // metoda w roli gettera } function fAge( age ) { if
(fIsSetter(age)) { nAge = age || nAge; return that; // zwroc referencje na
samego siebie } return nAge; } return that; // zwroc obiekt };

{/codecitation}
```