

# Obiekty wprowadzenie

**Obiekt** to konstrukcja programistyczna mająca swoje cechy charakterystyczne (właściwości), którymi mogą być zmienne lub inne obiekty. Dodatkowo obiekt ma możliwość wykonywania różnych funkcji, które nazywamy metodami. Obiekt definiujemy, tworząc specjalną funkcję zwaną **konstruktorem**, a następnie tworzymy, stosując operator **new**.

Najłatwiej przedstawić to na przykładzie list. 10.1. W skrypcie zdefiniowano za pomocą konstruktora obiekt **osoba** zawierający właściwości: **imie** i **nazwisko** oraz obiekt **szpital** zawierający właściwości: **lekarz**, **specjalizacja** oraz **pacjent** (obiekt). Następnie utworzono oba obiekty za pomocą operatora **new**. Ostatnim krokiem jest wyświetlenie informacji poprzez odwołanie się do konkretnej właściwości obiektu (**nazwa\_objektu.nazwa\_właściwości**).

Na przykład **window**, **document**, **array**, **string** itp. to typowe obiekty, które posiadają swoje metody i właściwości :).

Oczywiście jak i inne języki, tak i Javascript umożliwia tworzenie własnych obiektów.

## Tworzenie pojedynczego obiektu

Aby utworzyć nowy pojedynczy obiekt możemy skorzystamy z poniższej konstrukcji:

```
var myObject = {
  name: "Marcin",
  height: 184,
  print : function() {
    console.log(this.name)
  }
}
```

Nasz obiekt już na starcie posiada dwie ustalone właściwości - name i height, oraz jedną metodę, która wypisuje jego imię.

```
myObject.print(); //wypisze w konsoli "Marcin"  
console.log(myObject.height); //wypisze 184
```

Aby odwołać się do danego obiektu z jego wnętrza stosujemy instrukcję **this**. Dzięki temu możemy w łatwy sposób wywoływać z wnętrza inne metody danego obiektu lub korzystać z jego właściwości:

```
var myObject = {  
  number: 100,  
  square : function() {  
    return this.number * this.number  
  }  
}  
  
myObject.number = 200;  
console.log(myObject.square())
```

Gdy nasz obiekt już istnieje, możemy do niego dodawać nowe metody lub właściwości, tak samo jak to robimy normalnie dla obiektu window:

```
var x = 10;  
  
//jest równoznaczne z  
window.x = 10;  
  
var printSomething = function() {...}  
  
//jest równoznaczne z  
window.printSomething = function() {...}
```

Tak naprawdę deklarując zwykłe zmienne globalne i funkcje definiujemy po prostu właściwości i metody dla obiektu window. JS nie wymaga stosowania odwołania do window, więc praktycznie zawsze pomijamy to odwołanie.

Podobnie właściwości i metody możemy deklarować dla nowych obiektów:

```
var myObject = {  
  name: "Marcin",  
  height: 184,  
  print : function() {  
    alert(this.name)  
  }  
}
```

```

    }
}

myObject.weight = 73; //dodaliśmy nową właściwość
myObject.printDetail = function() {
    return {
        height : this.height,
        name :
    }
}

myObject.printDetail()

```

## Ćwiczenie

Zapisz poniższy skrypt i sprawdź jego działanie

```

<script type="text/javascript">
function szpital(lekarz,specjalizacja,pacjent) {
this.lekarz=lekarz;
this.specjalizacja=specjalizacja;
this.pacjent=pacjent;
}
function osoba(imie,nazwisko) {
this.nazwisko=nazwisko;
this.imie=imie;
}
pacjent=new osoba("Jan","Kowalski");
oddzial=new szpital("P.Nowak","neurolog",pacjent);
document.write("<b>lekarz:</b> "+oddzial.lekarz+"<br>");
document.write("<b>specjalizacja:</b> "+oddzial.specjalizacja-
+"<br>");
document.write("<b>pacjent:</b> "+ oddzial.pacjent.imie +"
"+oddzial.pacjent.nazwisko);
</script>

```