

JS - obiekty cz. 2

Skrócona notacja: właściwości

```
1  const name = "Krzysiek";  
2  const surname = "Dąbrowski";  
3  
4  
5  const person = {  
6      name,      // zamiast name: name  
7      surname,   // zamiast surname: surname  
8  };
```

warto stosować, jeśli właściwość ma taką samą nazwę jak zmienna/stała

Skrócona notacja: metody

```
1  const person = {  
2    name: "Krzysiek",  
3    sayHello() {  
4      console.log(`Cześć! Jestem ${this.name}`);  
5    }  
6  };
```

słowo **function** niepotrzebne

Computed Property Names

```
1  const currency = "EUR";  
2  
3  const exchangeData = {  
4    PLN: 45,  
5    [currency]: 200,  
6  };
```

Destrukturyzacja

```
1  const person = {
2    name: "Krzysiek",
3    surname: "Dąbrowski",
4    age: 29,
5    gender: "male",
6  };
7
8  const { name, surname, ...rest } = person;
9  // rest - obiekt z pozostałymi właściwościami
10
11  const { city = "N/A" } = person; // domyślna wartość
12  const { surname: lastName } = person; // inna nazwa zmiennej/stałej
13
14  // obiekt jako argument funkcji
15  const getPersonFullName = ({ name, surname }) => `${name} ${surname}`;
```

Destrukturyzacja w zagnieżdżonym obiekcie

```
1  const person = {  
2    name: "Krzysiek",  
3    surname: "Dąbrowski",  
4    father: {  
5      name: "Hieronim",  
6      surname: "Hieronimski",  
7    },  
8  };  
9  
10 const { father: { name: fatherName } } = person;  
11  
12 console.log(`Imię ojca: ${fatherName}`);
```

Łączenie obiektów

```
1  const personBasicData = {  
2    name: "Krzysiek",  
3    surname: "Dąbrowski",  
4  };  
5  
6  const personAdditionalData = {  
7    age: 29,  
8  };  
9  
10 const allPersonData = {  
11   ...personBasicData,  
12   ...personAdditionalData,  
13 };
```

Klonowanie

```
1  const person = {  
2    name: "Krzysiek",  
3    surname: "Dąbrowski",  
4    father: {  
5      name: "Hieronim",  
6      surname: "Hieronimski",  
7    },  
8  };  
9  
10  
11  const personCopy = { ...person };  
12  
13  
14  console.log(personCopy === person);           // false  
15  console.log(personCopy.father === person.father); // true
```

jest to tzw. shallow copy (płytki kopia)

for...in

```
1  const person = {  
2    name: "Krzysiek",  
3    surname: "Dąbrowski",  
4  };  
5  
6  
7  for (const propertyName in person) {  
8    console.log(`${propertyName}: ${person[propertyName]}`);  
9  }
```

- iteruje po właściwościach obiektu
- kolejność może być różna

Linki

- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Object_initializer
- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Destructuring_assignment
- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Spread_syntax