

Об'єкти - вступ

Об'єктоорієнтоване програмування^[1] (**ООП**, іноді *об'єктозорієнтоване програмування*^{[2][1]}, раніше *об'єктно-орієнтоване програмування*^[3]; [англ.](#) *Object-oriented programming*, OOP) — одна з [парадигм програмування](#), яка розглядає програму як множину [«об'єктів»](#), що взаємодіють між собою.

Об'єкти в JavaScript поєднують в собі два важливих функціонали:

- це асоціативний масив: структура, придатна для зберігання будь-яких даних;
- можливості для об'єктно-орієнтованого програмування.

Для чого потрібні об'єкти?

Дано дані про автомобіль : модель, рік випуску, колір, вага, номер, власник. Потрібно визначити «вік» авто

Для чого потрібні об'єкти?

Дано дані про автомобіль : модель, рік випуску, вага, власник. Потрібно визначити «вік» авто

Крок 0. Позначення змінних

model	– модель,
productionYear	– рік випуску,
weight	– вага,
owner	– власник

*Можемо описати **4** змінних для збереження інформації про автомобілі*

Для чого потрібні об'єкти?

Дано дані про 2 автомобілі : модель, рік випуску, вага, власник. Потрібно визначити загальну вагу

Крок 0. Позначення змінних

model1 – модель,
productionYear1 – рік випуску,
weight1 – вага,
owner1 – власник,

model2 – модель,
productionYear2 – рік випуску,
weight2 – вага,
owner2 – власник

Можемо описати 8 змінних для збереження інформації про автомобілі

Для чого потрібні об'єкти?

Дано дані про 3 автомобілі : модель, рік випуску, вага, власник. Потрібно визначити загальну вагу

Крок 0. Позначення змінних

model1 – модель,
productionYear1 – рік випуску,
weight1 – вага,
owner1 – власник,

model2 – модель,
productionYear2 – рік випуску,
weight2 – вага,
owner2 – власник

model3 – модель,
productionYear3 – рік випуску,
weight3 – вага,
owner3 – власник

*Можемо описати **12** змінних для збереження інформації про автомобілі*

Дуже погано:

- інформація про кожне авто знаходиться у різних змінних*
- неможна використати цикли*

Для чого потрібні об'єкти?

Дано дані про 20 автомобілів : модель, рік випуску, вага, власник. Потрібно визначити загальну вагу

Крок 0. Позначення змінних

Для чого потрібні об'єкти?

Дано дані про 20 автомобілів : модель, рік випуску, вага, власник. Потрібно визначити загальну вагу

Крок 0. Позначення змінних

models	– масив моделей,
productionYears	– масив років випуску,
weights	– масив ваг,
owners	– масив власників

Можемо описати 4 окремих масиви для збереження окремих категорій даних (не дуже добре, бо треба інформацію про один об'єкт зберігати у різних масивах)

```
let models = ['Opel', 'Audi', 'BMW', 'Fiat', 'Nissan', ...]
let productionYears = [2000, 1978, 1999, 2010, 2021, ...]
let weights = [1700, 2100, 1500, 1450, 1870, ...]
let owners = ['John', 'Olga', 'Ivan', 'Petro', 'Sara', ...]
```

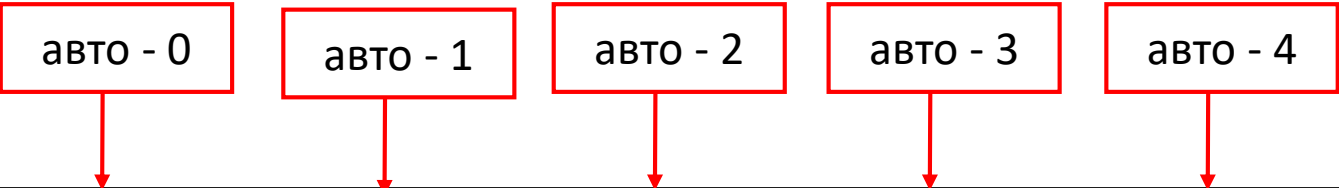

Для чого потрібні об'єкти?

Дано дані про 20 автомобілів : модель, рік випуску, вага, власник. Потрібно визначити загальну вагу

Крок 0. Позначення змінних

- models** – масив моделей,
- productionYears** – масив років випуску,
- weights** – масив ваг,
- owners** – масив власників

Можемо описати 4 окремих масиви для збереження окремих категорій даних (не дуже добре, бо треба інформацію про один об'єкт зберігати у різних масивах, важко виконувати операції додавання, редагування, ...)



```
let models = ['Opel', 'Audi', 'BMW', 'Fiat', 'Nissan', ...]
let productionYears = [2000, 1978, 1999, 2010, 2021, ...]
let weights = [1700, 2100, 1500, 1450, 1870, ...]
let owners = ['John', 'Olga', 'Ivan', 'Petro', 'Sara', ...]
```

Масив – нумерована послідовність елементів деякого типу. Іншими словами кожному значенню відповідає деякий номер (індекс або ключ доступу)

Індекс, ключ доступу (номер) – це ціле число ≥ 0 , яке використовуємо для звертання до елемента

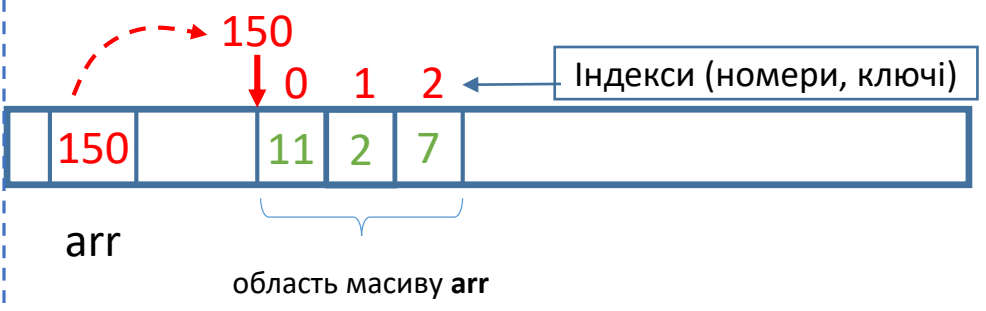
Значення – величина довільного типу

Опис масивів

```
let масив = [ ел0, ел1, ел2, ..., елN ]
```

Масив

```
let arr = [ 11, 2, 7 ]
```



Об'єкт може бути використаний як **асоціативний масив**: структура, що представляє собою набори пар «ключ»: «значення», або ж (що те ж саме) «назва_властивості»: «значення» і дозволяють описувати об'єкти даних зберігаючи при цьому усі дані як єдину структуру

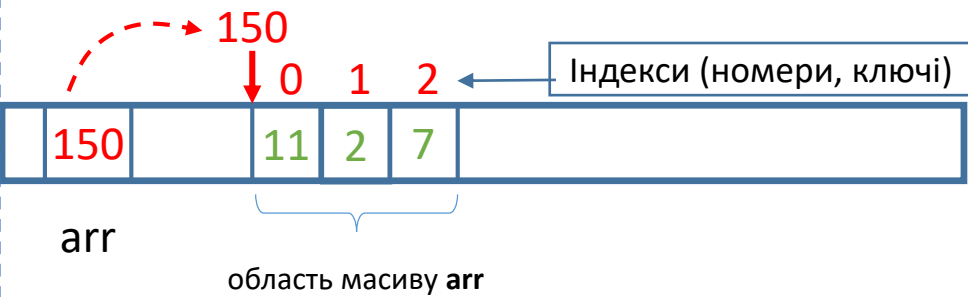
Ключ (назва_властивості) – це рядок тексту (string) або символ (Symbol), що використовуємо для звертання до значення
Значення – величина довільного типу

Опис масивів

```
let масив = [ ел0, ел1, ел2, ..., елN ]
```

Масив

```
let arr = [ 11, 2, 7 ]
```

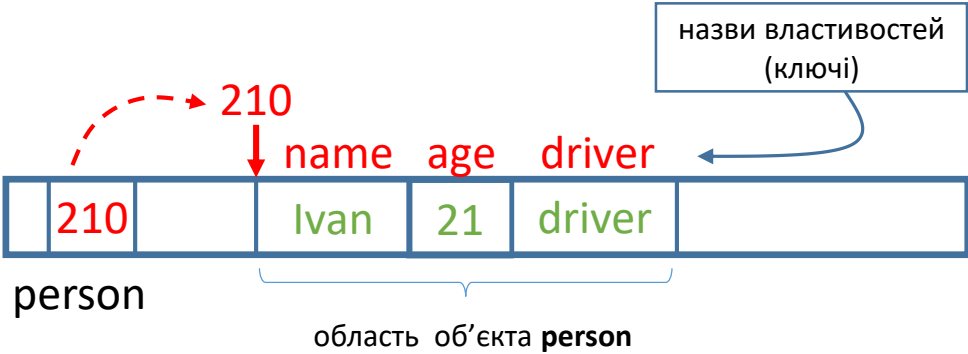


Опис об'єктів
(як асоціативних масивів)

```
{  
  властивість1 : значення1,  
  властивість2 : значення2,  
  ...  
  властивістьN : значенняN  
}
```

Асоціативний масив

```
let person={  
  name : 'Ivan',  
  age : 21,  
  position: 'driver'  
}
```



Об'єкт може бути використаний як **асоціативний масив**: структура, що представляє собою набори пар «ключ»: «значення», або ж (що те ж саме) «властивість»: «значення» і дозволяють описувати об'єкти даних зберігаючи при цьому усі дані як єдину структуру

Ключ (властивість) – це рядок (string) або символ (Symbol)

Значення – величина довільного типу

Опис масивів

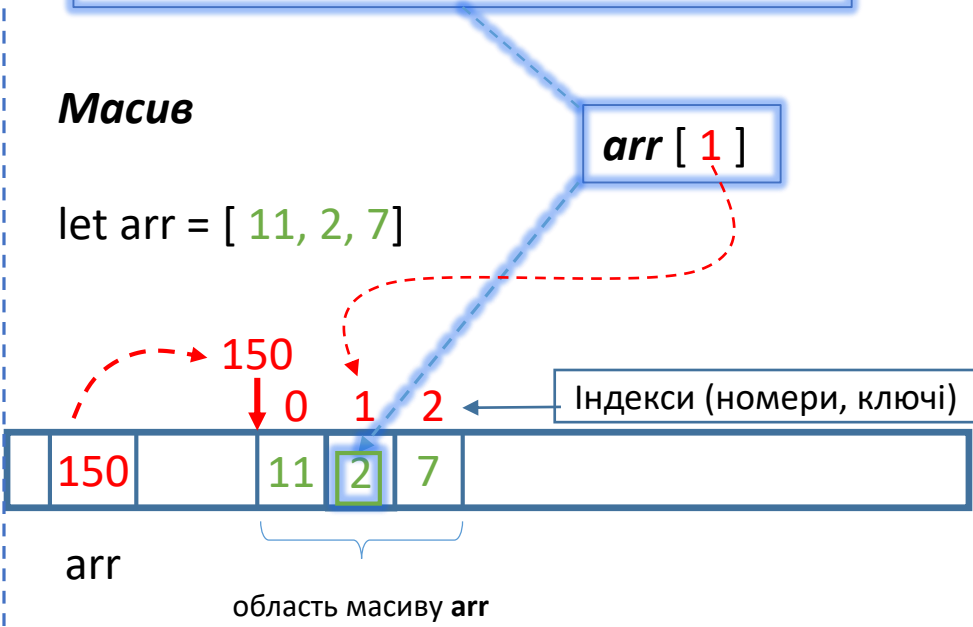
```
let масив = [ ел0, ел1, ел2, ..., елN ]
```

Звертання до елементів

назва_масиву [індекс_елемента]

Масив

```
let arr = [ 11, 2, 7 ]
```

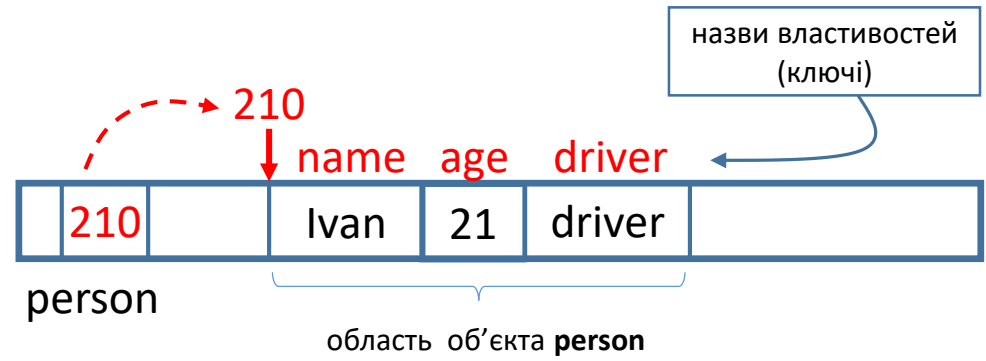


Опис об'єктів
(як асоціативних масивів)

```
{  
  властивість1 : значення1,  
  властивість2 : значення2,  
  ...  
  властивістьN : значенняN  
}
```

Асоціативний масив

```
let person={  
  name : 'Ivan',  
  age : 21,  
  position: 'driver'  
}
```



Об'єкт може бути використаний як **асоціативний масив**: структура, що представляє собою набори пар «ключ»: «значення», або ж (що те ж саме) «властивість»: «значення» і дозволяють описувати об'єкти даних зберігаючи при цьому усі дані як єдину структуру

Ключ (властивість) – це рядок (string) або символ (Symbol)

Значення – величина довільного типу

Опис масивів

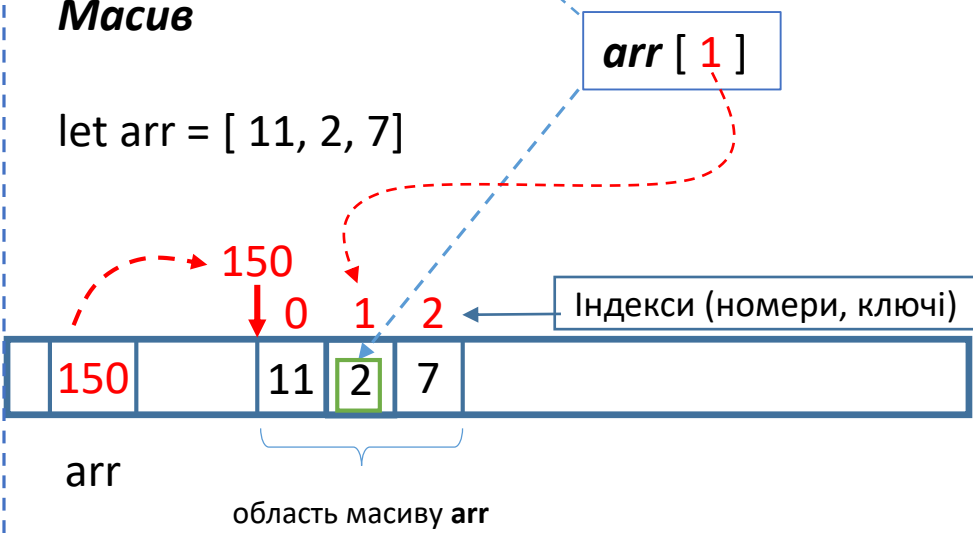
```
let масив = [ ел0, ел1, ел2, ..., елN ]
```

Звертання до елементів

```
назва_масиву [ індекс_елемента ]
```

Масив

```
let arr = [ 11, 2, 7 ]
```



Опис об'єктів
(як асоціативних масивів)

назва у лапках!!!

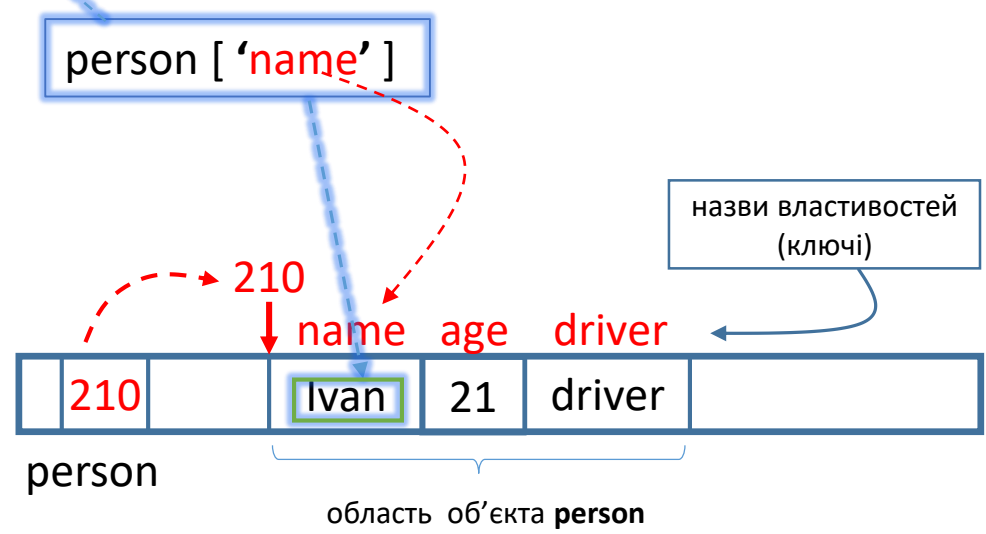
```
об'єкт [ 'назва властивості' ]
```

Асоціативний масив

```
let person={  
  name : 'Ivan',  
  age  : 21,  
  position: 'driver'  
}
```

```
{  
  властивість1 : значення1,  
  властивість2 : значення2,  
  ...  
  властивістьN : значенняN  
}
```

Звертання до елементів



Об'єкт може бути використаний як **асоціативний масив**: структура, що представляє собою набори пар «ключ»: «значення», або ж (що те ж саме) «властивість»: «значення» і дозволяють описувати об'єкти даних зберігаючи при цьому усі дані як єдину структуру

Ключ (властивість) – це рядок (string) або символ (Symbol)

Значення – величина довільного типу

Опис масивів

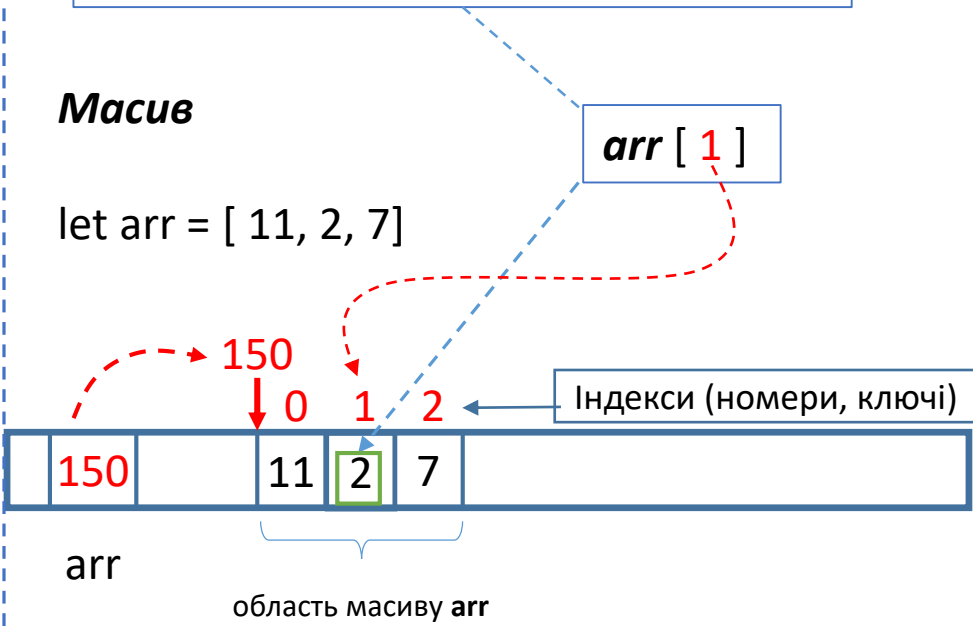
```
let масив = [ ел0, ел1, ел2, ..., елN ]
```

Звертання до елементів

```
назва_масиву [ індекс_елемента ]
```

Масив

```
let arr = [ 11, 2, 7 ]
```



Опис об'єктів (як асоціативних масивів)

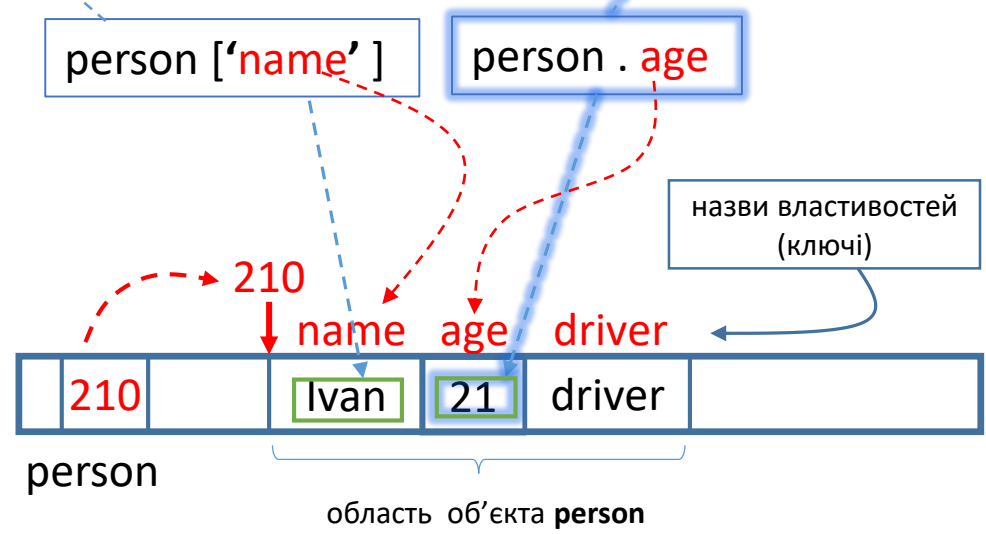
Звертання до елементів

```
об'єкт [ 'назва властивості' ] або об'єкт .назва_властивості
```

Асоціативний масив

```
let person={  
  name : 'Ivan',  
  age  : 21,  
  position: 'driver'  
}
```

```
{  
  властивість1 : значення1,  
  властивість2 : значення2,  
  ...  
  властивістьN : значенняN  
}
```



Об'єкт може бути використаний як **асоціативний масив**: структура, що представляє собою набори пар «ключ»: «значення», або ж (що те ж саме) «властивість»: «значення» і дозволяють описувати об'єкти даних зберігаючи при цьому усі дані як єдину структуру

Ключ (властивість) – це рядок (string) або символ (Symbol)

Значення – величина довільного типу

Звертання до елементів

об'єкт . назва_властивості

Можна використовувати тоді, коли **назва_властивості** є коректним ідентифікатором (як назва змінної)

```
person . name = 'Olga'    //ok
person . age = 18         //ok
person . second name = 'Kein'  //error
//second name - некоректний ідентифікатор
```

об'єкт [' назва_властивості ']

Можна використовувати при довільному значенні **назви_властивості**

```
person [ 'name' ] = 'Olga'    //ok
person [ 'age' ] = 18         //ok
person [ 'second name' ] = 'Kein'  //ok
```

Опис об'єктів
(як асоціативних масивів)

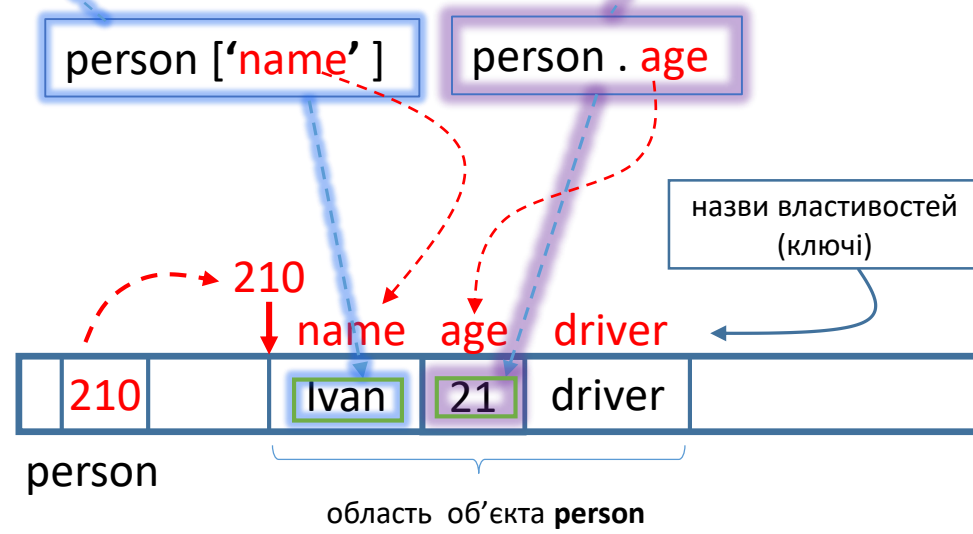
Звертання до елементів

об'єкт [' назва_властивості '] або **об'єкт** . назва_властивості

```
{
  властивість1 : значення1,
  властивість2 : значення2,
  ...
  властивістьN : значенняN
}
```

Асоціативний масив

```
let person={
  name : 'Ivan',
  age  : 21,
  position: 'driver'
}
```



Задача. Описати учня з такими властивостями: ім'я, прізвище, номер класу, оцінки з 3 предметів

Задача. Описати учня з такими властивостями: ім'я, прізвище, номер класу, оцінки з 3 предметів

Крок 0. Позначення величин

ім'я - '***first name***' або '***first_name***', або ***firstName***

прізвище – '***second name***' або '***second_name***', або ***secondName***

номер класу – '***class number***' або '***class_number***', або ***classNumber***

оцінки - ***scores***

Задача. Описати учня з такими властивостями: ім'я, прізвище, номер класу, оцінки з 3 предметів

Крок 0. Позначення

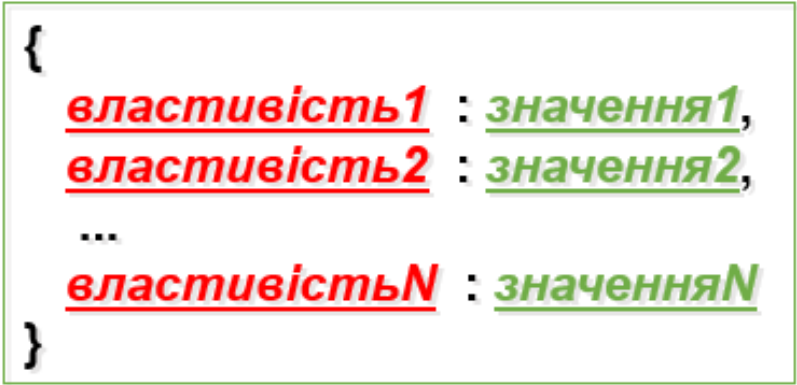
ім'я - *'first name'* або *'first_name'*, або *firstName*

прізвище – *'second name'* або *'second_name'*, або *secondName*

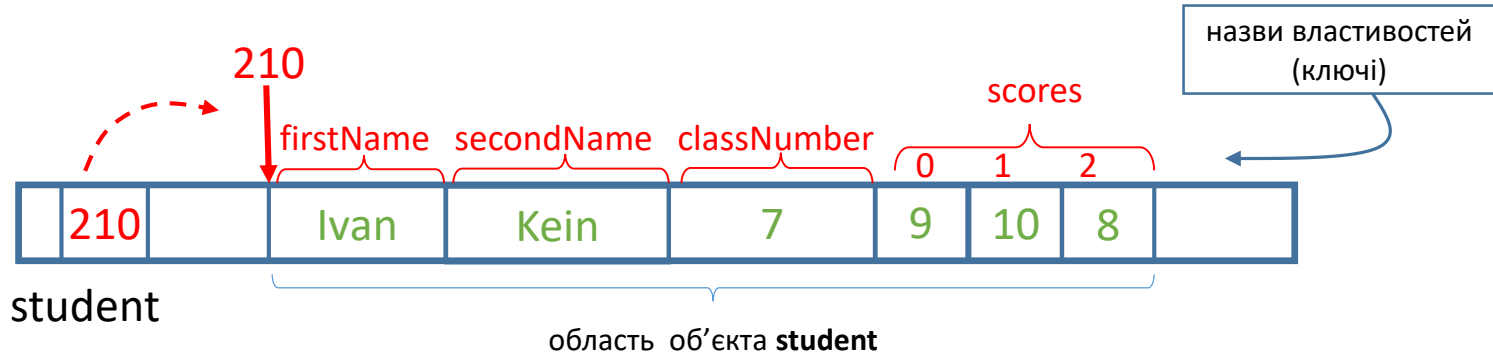
номер класу – *'class number'* або *'class_number'*, або *classNumber*

оцінки - *scores*

```
let student = {
  firstName: 'Ivan',
  secondName: 'Kein',
  classNumber: 7,
  scores: [9, 10, 8],
}
```



Схематичне представлення
объекта *student*



Задача. Описати учня з такими властивостями: ім'я, прізвище, номер класу, оцінки з 3 предметів

Крок 0. Позначення

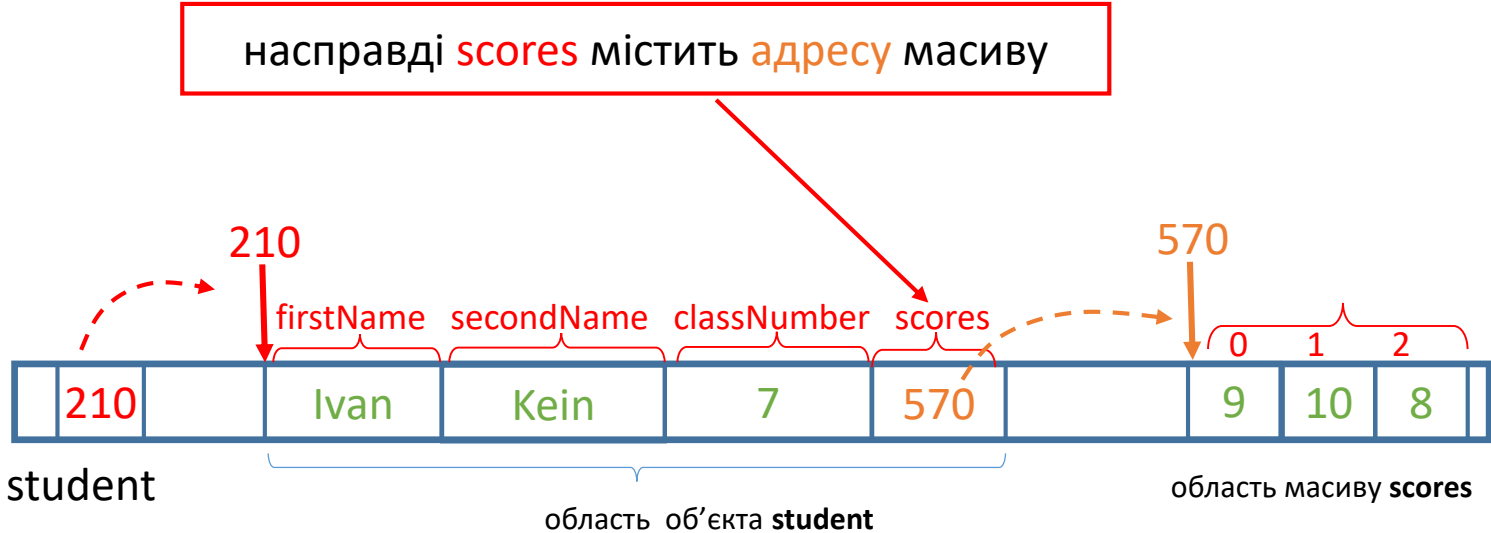
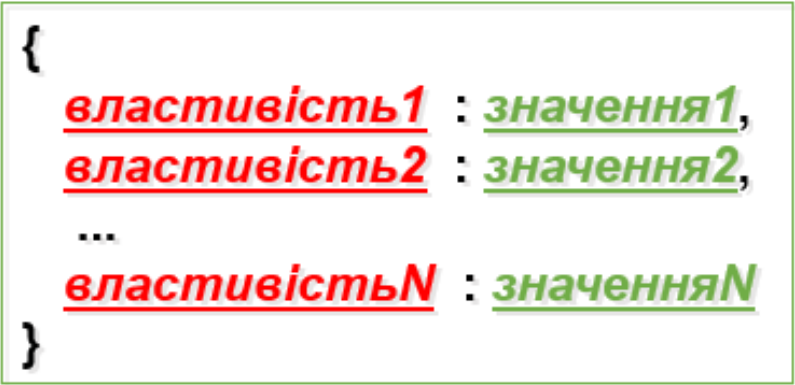
ім'я - *'first name'* або *'first_name'*, або *firstName*

прізвище – *'second name'* або *'second_name'*, або *secondName*

номер класу – *'class number'* або *'class_number'*, або *classNumber*

оцінки - *scores*

```
let student = {
  firstName: 'Ivan',
  secondName: 'Kein',
  classNumber: 7,
  scores: [9, 10, 8],
}
```



Задача. Описати учня з такими властивостями: ім'я, прізвище, номер класу, оцінки з 3 предметів

Крок 0. Позначення

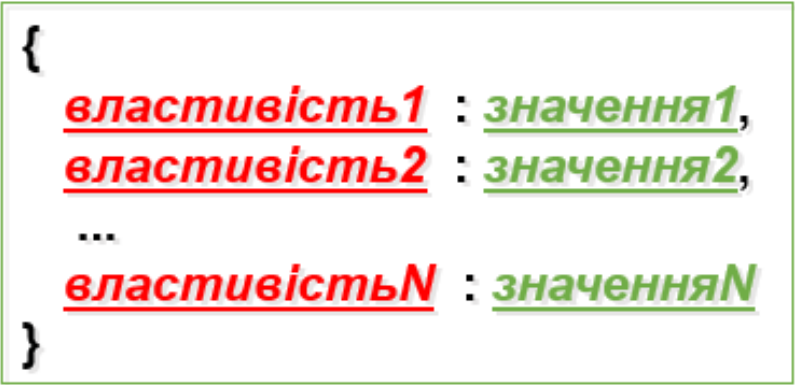
ім'я - *'first name'* або *'first_name'*, або *firstName*

прізвище – *'second name'* або *'second_name'*, або *secondName*

номер класу – *'class number'* або *'class_number'*, або *classNumber*

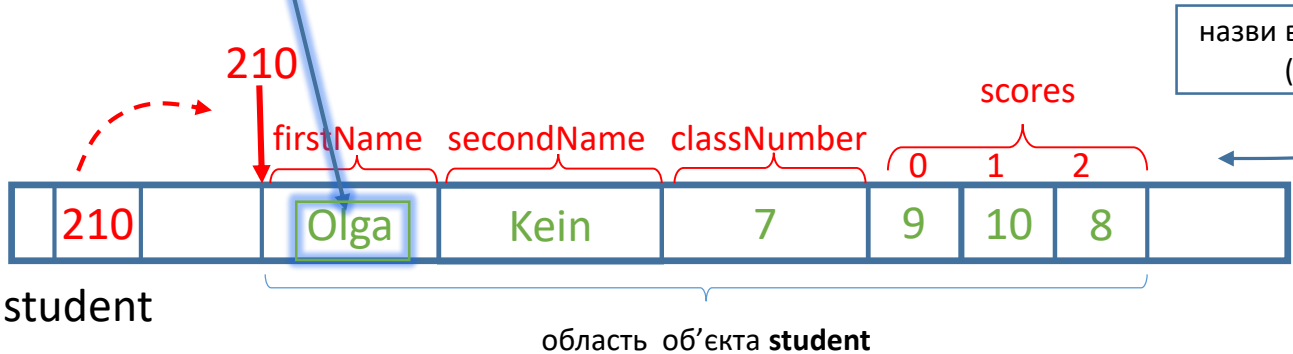
оцінки - *scores*

```
let student = {
  firstName: 'Ivan',
  secondName: 'Kein',
  classNumber: 7,
  scores: [9, 10, 8],
}
```



Доступ до властивостей
объекта *student*

student . firstName = 'Olga'



Задача. Описати учня з такими властивостями: ім'я, прізвище, номер класу, оцінки з 3 предметів

Крок 0. Позначення

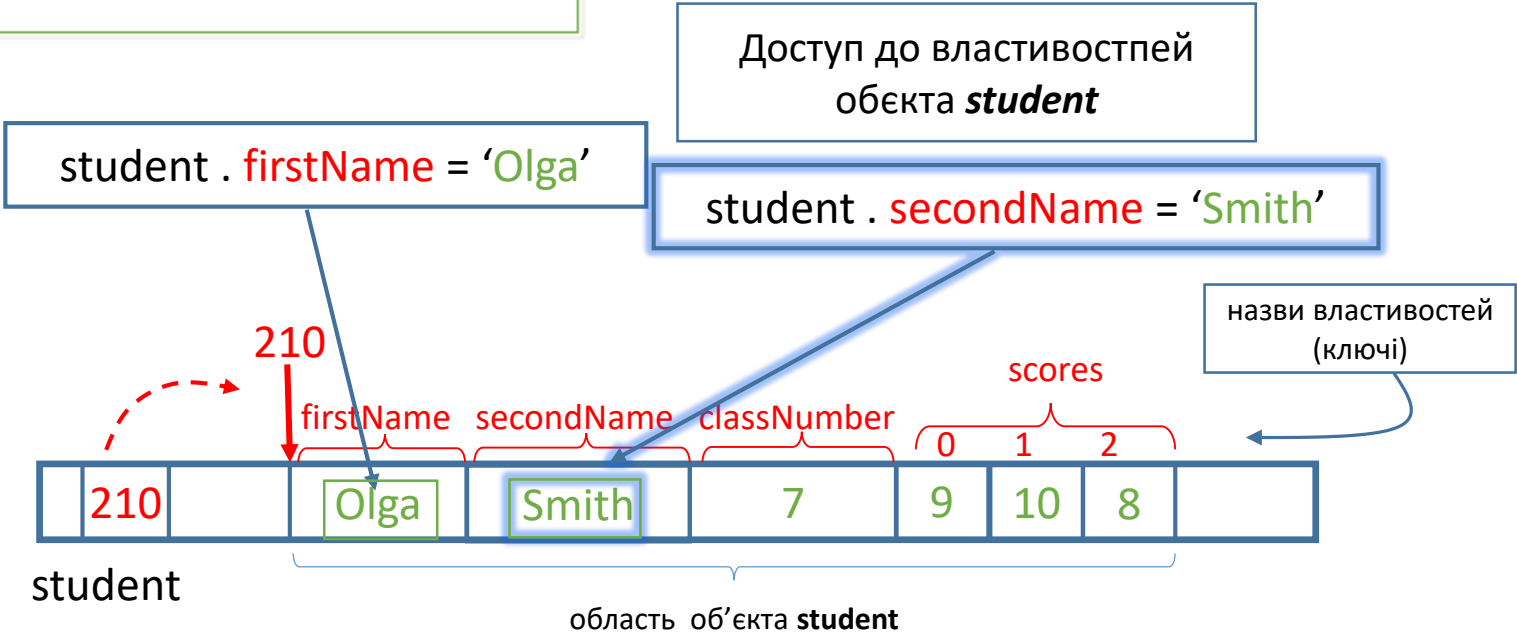
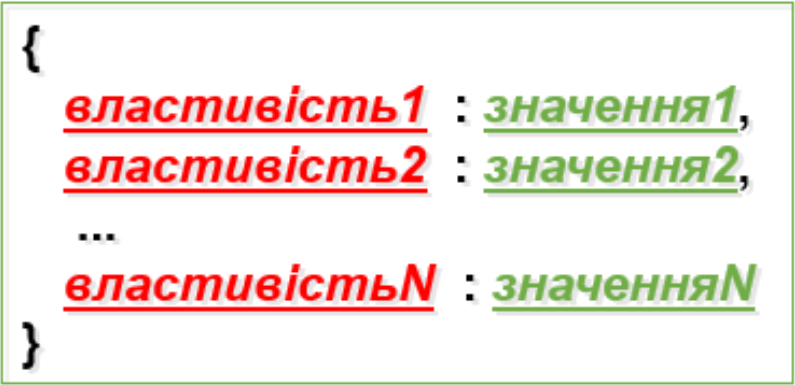
ім'я - *'first name'* або *'first_name'*, або *firstName*

прізвище – *'second name'* або *'second_name'*, або *secondName*

номер класу – *'class number'* або *'class_number'*, або *classNumber*

оцінки - *scores*

```
let student = {
  firstName: 'Ivan',
  secondName: 'Kein',
  classNumber: 7,
  scores: [9, 10, 8],
}
```

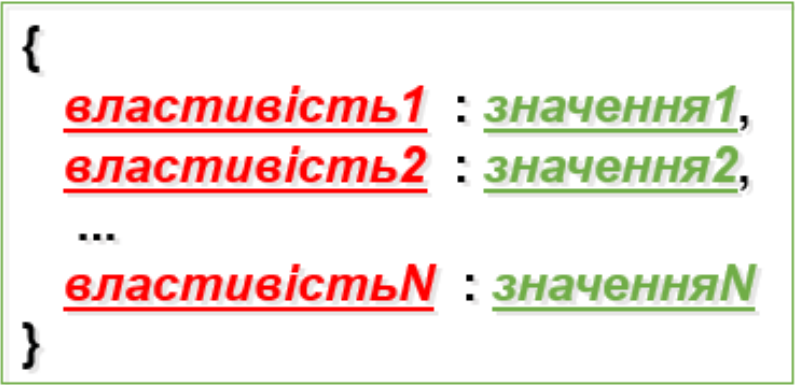


Задача. Описати учня з такими властивостями: ім'я, прізвище, номер класу, оцінки з 3 предметів

Крок 0. Позначення

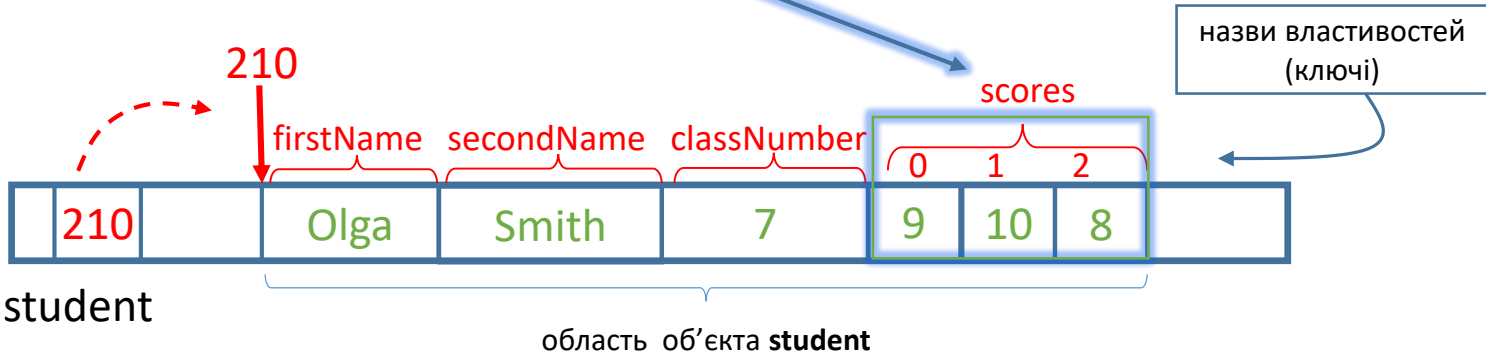
- ім'я - *'first name'* або *'first_name'*, або *firstName*
- прізвище – *'second name'* або *'second_name'*, або *secondName*
- номер класу – *'class number'* або *'class_number'*, або *classNumber*
- оцінки - *scores*

```
let student = {
  firstName: 'Ivan',
  secondName: 'Kein',
  classNumber: 7,
  scores: [9, 10, 8],
}
```



Доступ до властивостей
объекта *student*

student . scores

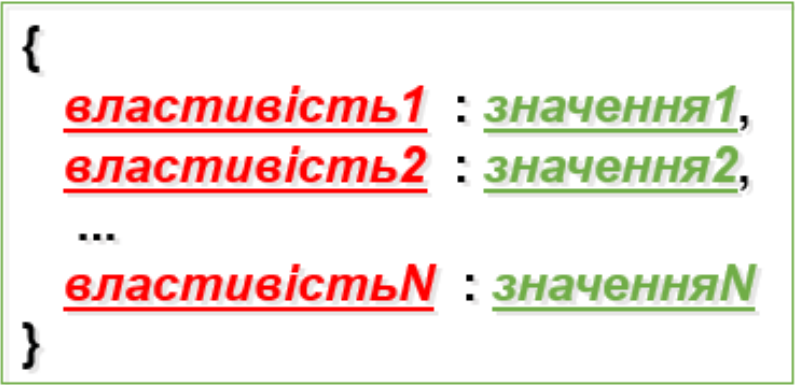


Задача. Описати учня з такими властивостями: ім'я, прізвище, номер класу, оцінки з 3 предметів

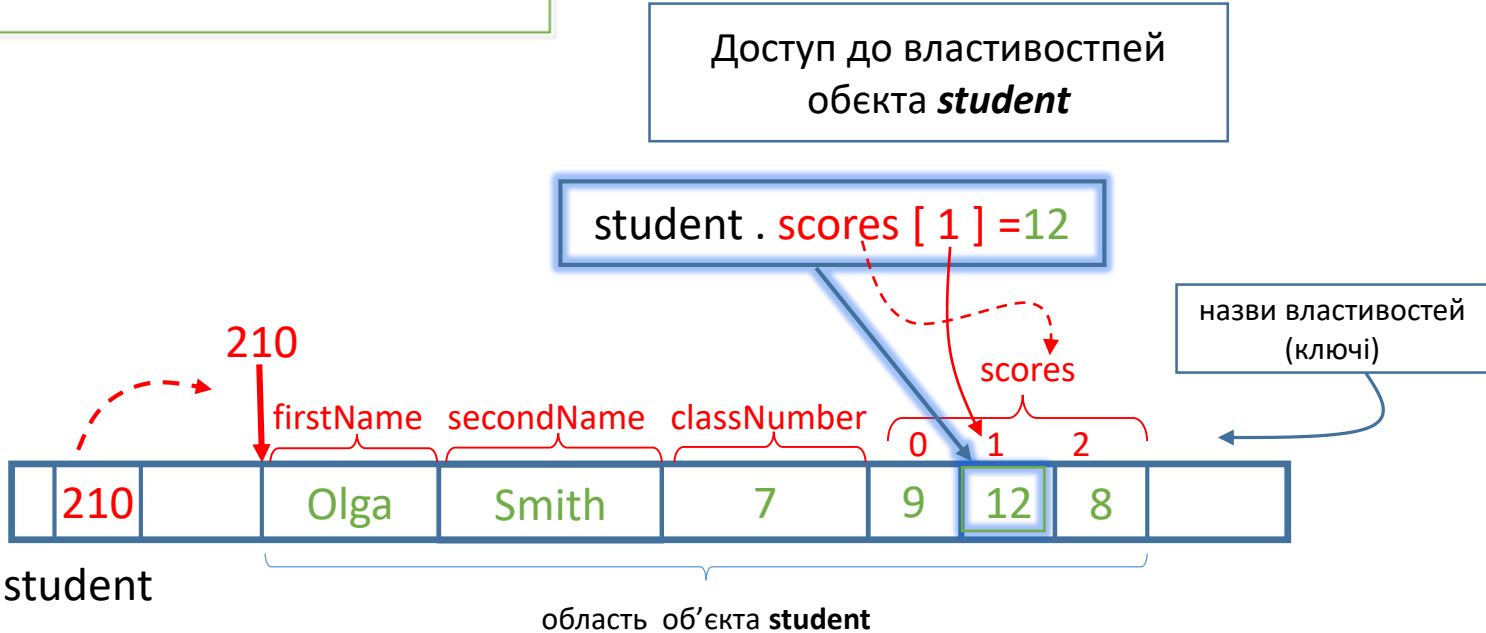
Крок 0. Позначення

- ім'я - *'first name'* або *'first_name'*, або *firstName*
- прізвище – *'second name'* або *'second_name'*, або *secondName*
- номер класу – *'class number'* або *'class_number'*, або *classNumber*
- оцінки - *scores*

```
let student = {
  firstName: 'Ivan',
  secondName: 'Kein',
  classNumber: 7,
  scores: [9, 10, 8],
}
```



```
alert(student.firstName)
student.firstName = 'Petro'
alert(student.classNumber)
student.classNumber = 8
alert(student.scores[1])
student.scores[1] = 12
```



Задача. Описати учня з такими властивостями: ім'я, прізвище, номер класу, оцінки з 3 предметів

Крок 0. Позначення

ім'я - *'first name'* або *'first_name'*, або *firstName*

прізвище – *'second name'* або *'second_name'*, або *secondName*

номер класу – *'class number'* або *'class_number'*, або *classNumber*

оцінки - *scores*

```
let student = {  
  'first name': 'Ivan',  
  'second name': 'Kein',  
  'class number': 7,  
  scores: [9, 10, 8],  
}
```

```
{  
  властивість1 : значення1,  
  властивість2 : значення2,  
  ...  
  властивістьN : значенняN  
}
```

```
alert(student['first name'])  
student['first name'] = 'Petro'  
alert(student['class number'])  
student['class number'] = 8  
alert(student['scores'][1])  
student['scores'][1] = 12
```

Можна і так, але незручно потім звертатися до властивостей

Задача. Описати учня з такими властивостями: ім'я, прізвище, номер класу, оцінки з 3 предметів

Крок 0. Позначення

ім'я - *'first name'* або *'first_name'*, або *firstName*

прізвище – *'second name'* або *'second_name'*, або *secondName*

номер класу – *'class number'* або *'class_number'*, або *classNumber*

оцінки - *scores*

Опис

```
let student = {  
  firstName: 'Ivan',  
  secondName: 'Kein',  
  classNumber: 7,  
  scores: [9, 10, 8],  
}
```

Так краще

Звертання до властивостей

```
alert(student.firstName)  
student.firstName = 'Petro'  
alert(student.classNumber)  
student.classNumber = 8  
alert(student.scores[1])  
student.scores[1] = 12
```

Опис

```
let student = {  
  'first name': 'Ivan',  
  'second name': 'Kein',  
  'class number': 7,  
  scores: [9, 10, 8],  
}
```

Звертання до властивостей

```
alert(student['first name'])  
student['first name'] = 'Petro'  
alert(student['class number'])  
student['class number'] = 8  
alert(student['scores'][1])  
student['scores'][1] = 12
```

Вкладені об'єкти

Задача. Описати клієнта банку

----- Властивості -----

- ПІБ
 - * прізвище
 - * ім'я
- адреса
 - * код
 - * місто
 - * вулиця
 - * номер будинку
- номер рахунку
- кількість грошей

Вкладені об'єкти

Задача. Описати клієнта банку

----- Властивості -----

- ПІБ
 - * прізвище
 - * ім'я
- адреса
 - * код
 - * місто
 - * вулиця
 - * номер будинку
- номер рахунку
- кількість грошей

Крок 0. Позначення величин

----- Властивості -----

- ПІБ
 - * прізвище
 - * ім'я
- адреса
 - * код
 - * місто
 - * вулиця
 - * номер будинку
- номер рахунку
- кількість грошей

--- позначення ---

- title
 - * secondName
 - * firstName
- address
 - * zipCode
 - * city
 - * street
 - * num
- accountNumber
- balance

Вкладені об'єкти

Задача. Описати клієнта банку

----- Властивості -----

- ПІБ
 - * прізвище
 - * ім'я
- адреса
 - * код
 - * місто
 - * вулиця
 - * номер будинку
- номер рахунку
- кількість грошей

Крок 0. Позначення величин

----- Властивості -----

- ПІБ
 - * прізвище
 - * ім'я
- адреса
 - * код
 - * місто
 - * вулиця
 - * номер будинку
- номер рахунку
- кількість грошей

--- позначення ---

- title
 - * secondName
 - * firstName
- address
 - * zipCode
 - * city
 - * street
 - * num
- accountNumber
- balance

```
let client = {  
  title: {  
    secondName: 'Smith',  
    firstame: 'John',  
  },  
  address: {  
    zipCode: '88000',  
    city: 'Uzhhorod',  
    street: 'Svobody',  
    num: 15,  
  },  
  accountNumber: 'ac2341fsr2',  
  balance: 2587000,  
}
```

Вкладені об'єкти

Задача. Описати клієнта банку

----- Властивості -----

- ПІБ
 - * прізвище
 - * ім'я
- адреса
 - * код
 - * місто
 - * вулиця
 - * номер будинку
- номер рахунку
- кількість грошей

Крок 0. Позначення величин

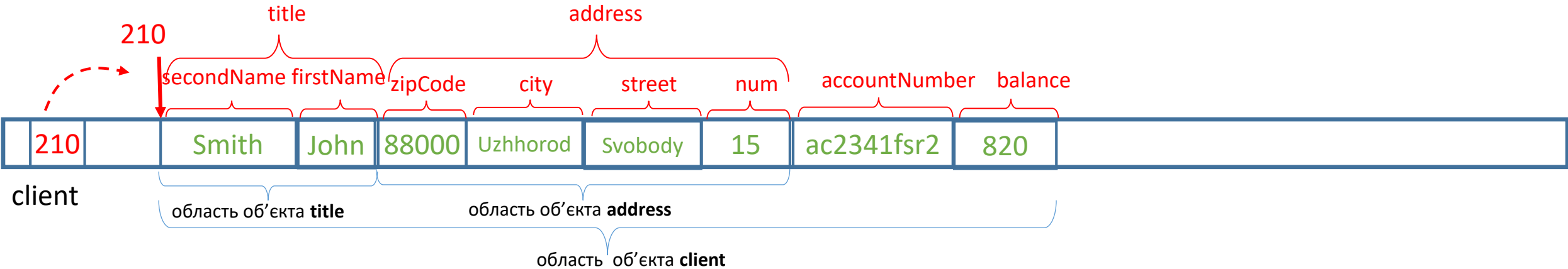
----- Властивості -----

- ПІБ
 - * прізвище
 - * ім'я
- адреса
 - * код
 - * місто
 - * вулиця
 - * номер будинку
- номер рахунку
- кількість грошей

--- позначення ---

- title
 - * secondName
 - * firstName
- address
 - * zipCode
 - * city
 - * street
 - * num
- accountNumber
- balance

```
let client = {
  title: {
    secondName: 'Smith',
    firstame: 'John',
  },
  address: {
    zipCode: '88000',
    city: 'Uzhhorod',
    street: 'Svobody',
    num: 15,
  },
  accountNumber: 'ac2341fsr2',
  balance: 2587000,
}
```



Вкладені об'єкти

Задача. Описати клієнта банку

----- Властивості -----

- ПІБ
 - * прізвище
 - * ім'я
- адреса
 - * код
 - * місто
 - * вулиця
 - * номер будинку
- номер рахунку
- кількість грошей

Крок 0. Позначення величин

----- Властивості -----

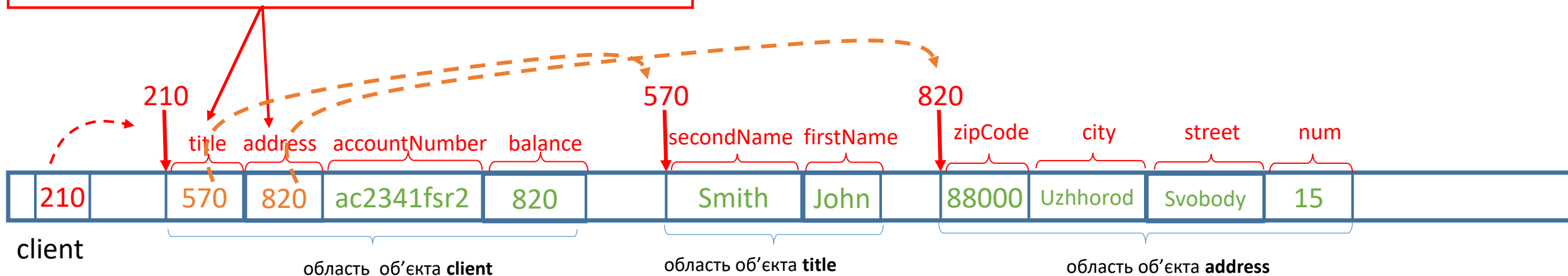
- ПІБ
 - * прізвище
 - * ім'я
- адреса
 - * код
 - * місто
 - * вулиця
 - * номер будинку
- номер рахунку
- кількість грошей

--- позначення ---

- title
 - * secondName
 - * firstame
- address
 - * zipCode
 - * city
 - * street
 - * num
- accountNumber
- balance

```
let client = {  
  title: {  
    secondName: 'Smith',  
    firstame: 'John',  
  },  
  address: {  
    zipCode: '88000',  
    city: 'Uzhhorod',  
    street: 'Svobody',  
    num: 15,  
  },  
  accountNumber: 'ac2341fsr2',  
  balance: 2587000,  
}
```

насправді **title**, **address** містять **адреси** об'єктів



Вкладені об'єкти

Задача. Описати клієнта банку

----- Властивості -----

- ПІБ
 - * прізвище
 - * ім'я
- адреса
 - * код
 - * місто
 - * вулиця
 - * номер будинку
- номер рахунку
- кількість грошей

Крок 0. Позначення величин

----- Властивості -----

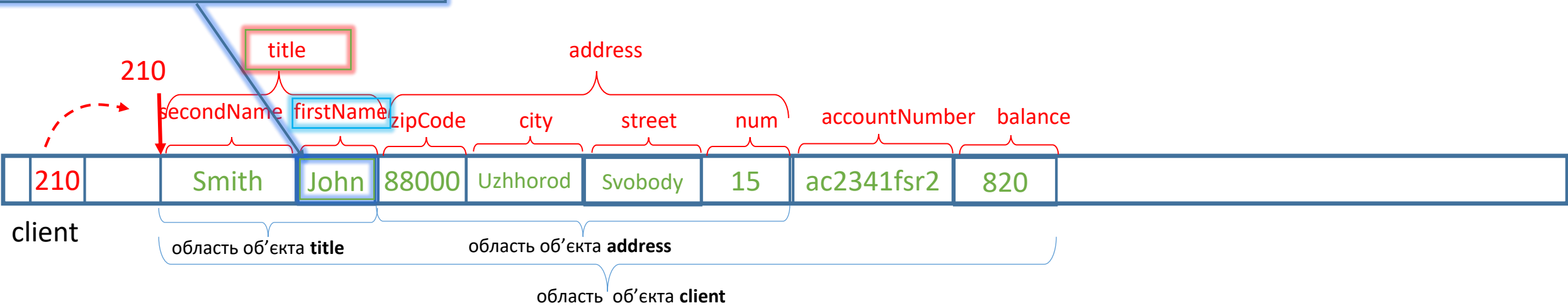
- ПІБ
 - * прізвище
 - * ім'я
- адреса
 - * код
 - * місто
 - * вулиця
 - * номер будинку
- номер рахунку
- кількість грошей

--- позначення ---

- title
 - * secondName
 - * firstName
- address
 - * zipCode
 - * city
 - * street
 - * num
- accountNumber
- balance

```
let client = {
  title: {
    secondName: 'Smith',
    firstame: 'John',
  },
  address: {
    zipCode: '88000',
    city: 'Uzhhorod',
    street: 'Svobody',
    num: 15,
  },
  accountNumber: 'ac2341fsr2',
  balance: 2587000,
}
```

client . title . firstName = 'John'



Вкладені об'єкти

Задача. Описати клієнта банку

----- Властивості -----

- ПІБ
 - * прізвище
 - * ім'я
- адреса
 - * код
 - * місто
 - * вулиця
 - * номер будинку
- номер рахунку
- кількість грошей

Крок 0. Позначення величин

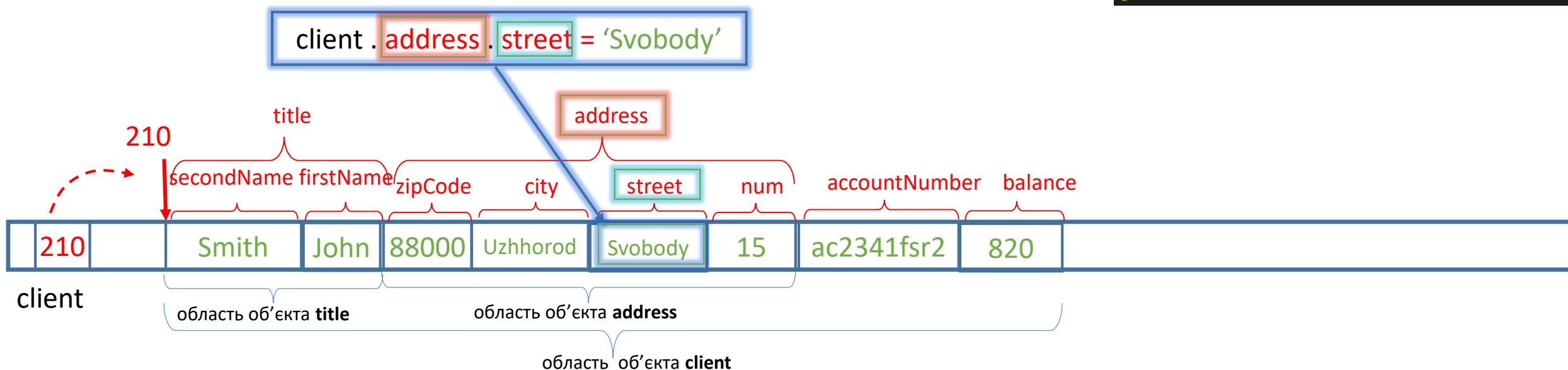
----- Властивості -----

- ПІБ
 - * прізвище
 - * ім'я
- адреса
 - * код
 - * місто
 - * вулиця
 - * номер будинку
- номер рахунку
- кількість грошей

--- ПОЗНАЧЕННЯ ---

- title
 - * secondName
 - * firstame
- address
 - * zipCode
 - * city
 - * street
 - * num
- accountNumber
- balance

```
let client = {  
  title: {  
    secondName: 'Smith',  
    firstame: 'John',  
  },  
  address: {  
    zipCode: '88000',  
    city: 'Uzhhorod',  
    street: 'Svobody',  
    num: 15,  
  },  
  accountNumber: 'ac2341fsr2',  
  balance: 2587000,  
}
```



Вкладені об'єкт

Задача. Описати інформацію про товар

----- Властивості -----

- назва товару
- виробник
 - назва підприємства
 - адреса
 - * код
 - * місто
 - * вулиця
 - * номер будинку
- кінцевий строк споживання
 - * рік
 - * місяць
 - * день
- ціна

Повернемось до задач з автомобілями

Дано дані про автомобіль : модель (Audi), рік випуску (1978), вага (2100), власник (Olga). Потрібно визначити «вік» авто

Дано дані про автомобіль : модель (Audi), рік випуску (1978), вага (2100), власник (Olga). Потрібно визначити «вік» авто

Крок 0. Позначення змінних

----- Влатисвості об'єкта «авто» -----

model	– модель,
productionYear	– рік випуску,
weight	– вага,
owner	– власник

Дано дані про автомобіль : модель (Audi), рік випуску (1978), вага (2100), власник (Olga). Потрібно визначити «вік» авто

Крок 0. Позначення змінних

----- Влатисвості об'єкта «авто» -----

model — модель,
productionYear — рік випуску,
weight — вага,
owner — власник

```
{  
  властивість1 : значення1,  
  властивість2 : значення2,  
  ...  
  властивістьN : значенняN  
}
```

Дано дані про автомобіль : модель (**Audi**), рік випуску (**1978**), вага (**2100**), власник (**Olga**). Потрібно визначити «вік» авто

Крок 0. Позначення змінних

----- Влатисвості об'єкта «авто» -----

model — модель,
productionYear — рік випуску,
weight — вага,
owner — власник

```
{  
  властивість1 : значення1,  
  властивість2 : значення2,  
  ...  
  властивістьN : значенняN  
}
```

```
let auto = {  
  model : 'Audi',  
  productionYear : 1978,  
  weight : 2100,  
  owner : 'Olga'  
}
```

Дано дані про автомобіль : модель (**Audi**), рік випуску (**1978**), вага (**2100**), власник (**Olga**). Потрібно визначити «вік» авто

Крок 0. Позначення змінних

----- Влатисвості об'єкта «авто» -----

model — модель,
productionYear — рік випуску,
weight — вага,
owner — власник

```
{  
  властивість1 : значення1,  
  властивість2 : значення2,  
  ...  
  властивістьN : значенняN  
}
```

```
let auto = {  
  model : 'Audi',  
  productionYear : 1978,  
  weight : 2100,  
  owner : 'Olga'  
}
```

```
let auto = {  
  model: 'Audi',  
  productionYear: 1978,  
  weight: 2100,  
  owner: 'Olga',  
}  
  
const usedYearsNumber = 2023 - auto.productionYear  
alert(  
  `Користувач ${auto.owner} використовує ${auto.model} ${usedYearsNumber} років`  
)
```

Задача. Дано дані про 2 автомобілі : модель, рік випуску, вага, власник. Потрібно визначити загальну вагу

Задача. Дано дані про 2 автомобілі : модель, рік випуску, вага, власник. Потрібно визначити загальну вагу

Крок 0. Позначення змінних

----- Влатисвості об'єкта «авто» -----

model – модель,
productionYear – рік випуску,
weight – вага,
owner – власник

Задача. Дано дані про 2 автомобілі : модель, рік випуску, вага, власник. Потрібно визначити загальну вагу

Крок 0. Позначення змінних

----- Влатисвості об'єкта «авто» -----

model – модель,
productionYear – рік випуску,
weight – вага,
owner – власник

```
{  
  властивість1 : значення1,  
  властивість2 : значення2,  
  ...  
  властивістьN : значенняN  
}
```

Задача. Дано дані про 2 автомобілі : модель, рік випуску, вага, власник. Потрібно визначити загальну вагу

Крок 0. Позначення змінних

----- Влатисвості об'єкта «авто» -----

model — модель,
productionYear — рік випуску,
weight — вага,
owner — власник

```
{  
  властивість1 : значення1,  
  властивість2 : значення2,  
  ...  
  властивістьN : значенняN  
}
```

```
let auto1 = {  
  model      : 'Audi',  
  productionYear : 1978,  
  weight     : 2100,  
  owner      : 'Olga'  
}
```

```
let auto2 = {  
  model      : 'Opel',  
  productionYear : 2000,  
  weight     : 1700,  
  owner      : 'John'  
}
```

Задача. Дано дані про 2 автомобілі : модель, рік випуску, вага, власник. Потрібно визначити загальну вагу

Крок 0. Позначення змінних

----- Влатисвості об'єкта «авто» -----

model — модель,
productionYear — рік випуску,
weight — вага,
owner — власник

```
let auto1 = {  
  model: 'Audi',  
  productionYear: 1978,  
  weight: 2100,  
  owner: 'Olga',  
}  
  
let auto2 = {  
  model: 'Opel',  
  productionYear: 2000,  
  weight: 1700,  
  owner: 'John',  
}  
  
const totalAutosWeight = auto1.weight + auto2.weight  
alert(`Загальна вага автомобілів ${totalAutosWeight} кг`)
```

```
let auto1 = {  
  model      : 'Audi',  
  productionYear : 1978,  
  weight      : 2100,  
  owner       : 'Olga'  
}
```

```
let auto2 = {  
  model      : 'Opel',  
  productionYear : 2000,  
  weight      : 1700,  
  owner       : 'John'  
}
```

Задача. Дано дані про 20 автомобілів : модель, рік випуску, вага, власник. Потрібно визначити загальну вагу

Задача. Дано дані про 20 автомобілів : модель, рік випуску, вага, власник. Потрібно визначити загальну вагу

Крок 0. Позначення змінних

----- Влатисвості об’єкта «авто» -----

model – модель,

productionYear – рік випуску,

weight – вага,

owner – власник

Задача. Дано дані про 20 автомобілів : модель, рік випуску, вага, власник. Потрібно визначити загальну вагу

Крок 0. Позначення змінних

----- Влатисвості об'єкта «авто» -----

- model** – модель,
- productionYear** – рік випуску,
- weight** – вага,
- owner** – власник

```
{  
  властивість1 : значення1,  
  властивість2 : значення2,  
  ...  
  властивістьN : значенняN  
}
```

Задача. Дано дані про 20 автомобілів : модель, рік випуску, вага, власник. Потрібно визначити загальну вагу

Крок 0. Позначення змінних

Масив об'єктів «авто»

----- Влатисвості об'єкта «авто» -----

- model** – модель,
- productionYear** – рік випуску,
- weight** – вага,
- owner** – власник

```
{  
  властивість1 : значення1,  
  властивість2 : значення2,  
  ...  
  властивістьN : значенняN  
}
```

let autoList = [

(авто № 0)
autoList [0]

(авто № 1)
autoList [1]

(авто № 2)
autoList [2]

model

productionYear

weight

owner

: 'Opel',

: 2000,

: 1700,

: 'John'

model

productionYear

weight

owner

: 'Audi',

: 1978,

: 2100,

: 'Olga'

model

productionYear

weight

owner

: 'BMW',

: 1999,

: 1500,

: 'Ivan'

.....

]

Так значно краще:

- уся інформація про об'єкт «авто» мітиться в одному місці
- можна використати методи роботи з масивами

Задача. Дано дані про 20 автомобілів : модель, рік випуску, вага, власник. Потрібно визначити загальну вагу

Масив об'єктів «авто»

```
let autoList = [  
  {  
    model: 'Opel',  
    productionYear: 2000,  
    weight: 1700,  
    owner: 'John',  
  },  
  {  
    model: 'Audi',  
    productionYear: 1978,  
    weight: 2100,  
    owner: 'Olga',  
  },  
  {  
    model: 'BMW',  
    productionYear: 1999,  
    weight: 1500,  
    owner: 'Ivan',  
  },  
]  
  
const totalWeight = autoList.reduce(  
  (prevSum, auto) => prevSum + auto.weight,  
  0  
)  
  
alert(`Загальна вага = ${totalWeight}`)
```

let autoList = [

(авто № 0)
autoList [0]

{
 model : 'Opel',
 productionYear : 2000,
 weight : 1700,
 owner : 'John'
},

(авто № 1)
autoList [1]

{
 model : 'Audi',
 productionYear : 1978,
 weight : 2100,
 owner : 'Olga'
},

(авто № 2)
autoList [2]

{
 model : 'BMW',
 productionYear : 1999,
 weight : 1500,
 owner : 'Ivan'
},

]

Задача. Дано дані про 20 автомобілів : модель, рік випуску, вага, власник. Потрібно визначити загальну вагу

Масив об'єктів «авто»

```
let autoList = [  
  {  
    model: 'Opel',  
    productionYear: 2000,  
    weight: 1700,  
    owner: 'John',  
  },  
  {  
    model: 'Audi',  
    productionYear: 1978,  
    weight: 2100,  
    owner: 'Olga',  
  },  
  {  
    model: 'BMW',  
    productionYear: 1999,  
    weight: 1500,  
    owner: 'Ivan',  
  },  
]
```

let autoList = [

(авто № 0)
autoList [0]

(авто № 1)
autoList [1]

(авто № 2)
autoList [2]

```
{  
  model      : 'Opel',  
  productionYear : 2000,  
  weight     : 1700,  
  owner      : 'John'  
},  
{  
  model      : 'Audi',  
  productionYear : 1978,  
  weight     : 2100,  
  owner      : 'Olga'  
},  
{  
  model      : 'BMW',  
  productionYear : 1999,  
  weight     : 1500,  
  owner      : 'Ivan'  
},  
  .....  
]
```

З використанням циклу for..of

```
let totalWeight = 0  
for (const auto of autoList) {  
  totalWeight += auto.weight  
}
```

З використанням циклу for

```
let totalWeight = 0  
for (let autoNum = 0; autoNum < autoList.length; autoNum++) {  
  totalWeight += autoList[autoNum].weight  
}
```

```
const totalWeight = autoList.reduce(  
  (prevSum, auto) => prevSum + auto.weight,  
  0  
)
```

```
alert(`Загальна вага = ${totalWeight}`)
```

Задача. Дано список книг (назва, автор, рік видання, вартість, кількість одиниць).

Знайти:

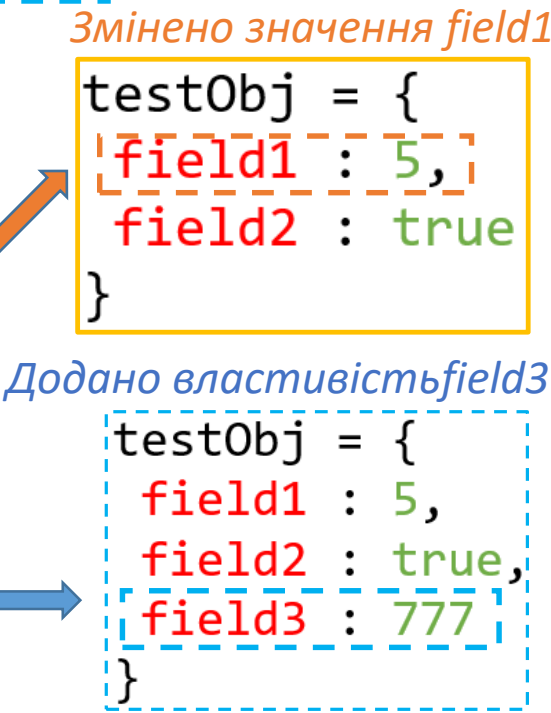
1. кількість книг, що було видано після 2000 року
2. знайти загальну вартість кожної із книг
3. знайти загальну вартість усіх книг
4. визначити, чи є книга, що була видана у 2000 році
5. визначити, чи усі книги були визаді після 2005 року
6. сформувати список книг, ціна яких менше за 1000 грн
7. знайти загальну вартість книг, які було видано у 2000 році
8. знайти першу книгу, вартість якої більше за 3000грн
9. знайти номер першої книги, вартість якої більше за 3000грн

Динамічне додавання властивостей

Якщо присвоювати значення властивості об'єкта:

- Якщо властивість з такою назвою існує, то її значення буде змінено.
- Якщо властивості з такою назвою немає, то вона буде створена з вказаним значенням

Загальна форма	Приклад
об'єкт . властивість = значення	<pre>let testObj = { field1 : 123, field2 : true } testObj . field1 = 5 testObj . field3 = 777</pre>

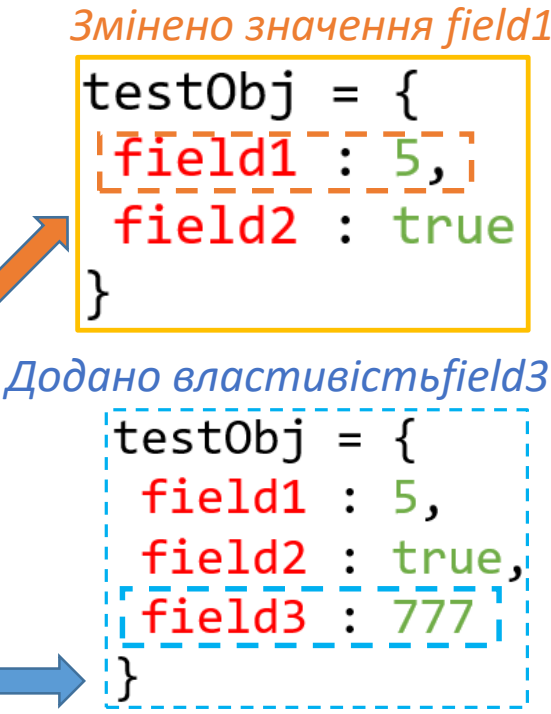


Динамічне додавання властивостей

Якщо присвоювати значення властивості об'єкта:

- Якщо властивість з такою назвою існує, то її значення буде змінено.
- Якщо властивості з такою назвою немає, то вона буде створена з вказаним значенням


Загальна форма	Приклад
об'єкт . властивість = значення	<pre>let testObj = { field1 : 123, field2 : true } testObj . field1 = 5 testObj . field3 = 777</pre>
об'єкт ['властивість'] = значення	<pre>let testObj = { field1 : 123, field2 : true } testObj['field1'] = 5 testObj['field3'] = 777</pre>



Динамічне видалення властивостей

Видалення властивостей за іменем здійснюється за допомогою спеціального оператора **delete**

Загальна форма	Приклад
<code>delete</code> об'єкт . властивість	<pre>let testObj = { field1 : 123, field2 : true } delete testObj.field1</pre>




```
testObj = {  
  field2 : true  
}
```

Динамічне видалення властивостей

Видалення властивостей за іменем здійснюється за допомогою спеціального оператора **delete**

Загальна форма	Приклад
<code>delete</code> об'єкт . властивість	<pre>let testObj = { field1 : 123, field2 : true } delete testObj.field1</pre>
<code>delete</code> об'єкт ['властивість']	<pre>let testObj = { field1 : 123, field2 : true } delete testObj['field1']</pre>



```
testObj = {  
  field2 : true  
}
```

Перевірка існування властивості в об'єкті

Перевірка існування властивості з певним іменем здійснюється за допомогою оператора « ***in*** »

Загальна форма	Приклад
<div data-bbox="25 535 726 621">'властивість' in об'єкт</div> <p data-bbox="25 749 726 949">Результат: <i>true</i> - властивість існує; <i>false</i> - властивість не існує</p>	<pre data-bbox="802 342 2497 985">let testObj = { field1 : 123, field2 : true } alert('field1' in testObj) //true //(властивість 'field1' існує) alert('age' in testObj) //false //(властивість 'age' не існує)</pre>

Перевір властивостей об'єкта

Перевір властивостей об'єкта здійснюється з використанням циклу *for..in*

Загальна форма

```
for (param_властивість in об'єкт) {  
    . . . . що робити з . . . . .  
    . . об'єкт [param_властивість] . .  
    . . . . .  
}
```

Лапок не треба!

param_властивість – змінна/константа, у яку поступово копіюються назви властивостей

Приклад

```
var user = {  
    name: "Іван",  
    surname: "Петров",  
    age: 25  
}  
  
for (const propertyName in user) {  
    document.write(`  
        ${propertyName}-${user[propertyName]}  
        <br>  
    `)  
}
```

name - Іван
surname - Петров
age - 25

Задача. Дано масив імен. Підрахувати кількість входжень кожного імені за зразком.


```
{  
  'Ivan' : 12,  
  'Olga' : 12,  
  'Petro' : 12,  
  'John' : 12,  
}
```

Задача. Для рядка тексту вивести на екран кількість входжень кожної літери (побудувати об'єкт, у якому поля – літери, а значення – кількість входжень).

Наприклад, для слова «HELLO»

H	:	1
E	:	1
L	:	2
O	:	1

Обчислювані назви властивостей

Загальна форма	Приклад
<pre>{ властивість1 : значення1, властивість2 : значення2, ... [вираз_для_назви] : значення, ... властивістьN : значенняN }</pre>	<div><pre>let someName = 'test' let v = 11 let testObj = { field1 : 123, field2 : true, ['my_'+v] : 888, [someName] : 777, }</pre></div> <div></div> <div><pre>testObj = { field1 : 123, field2 : true, my_11: 888, test: 777, }</pre></div>

Створення об'єктів з змінних і констант

Якщо назва властивості у об'єкті і значення цієї властивості формуються з на основі деякої змінної чи константи, то можна викоирстовувати спрощений синтаксис

Окремі змінні/константи, що містять властивості об'єкта

```
let model = 'Audi'  
let weight = 2100  
const owner = 'Ivan'
```

Стандартний опис

```
----Загальна форма ----  
let об'єкт = {  
  змінна1 : змінна1,  
  змінна2 : змінна1,  
  . . . . .  
  зміннаN: зміннаN,  
}  
(змінна або константа)
```

----- Приклад -----

```
let auto= {  
  model : model,  
  weight : weight,  
  owner : owner  
}
```

Спрощений опис

```
----Загальна форма ----  
let об'єкт = {  
  змінна1,  
  змінна2,  
  . . . . .  
  зміннаN,  
}  
(змінна або константа)
```

----- Приклад -----

```
let auto= {  
  model,  
  weight,  
  owner  
}
```



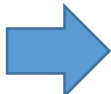
```
auto= {  
  model : 'Audi',  
  weight : 2100,  
  owner : 'Ivan'  
}
```

Створення об'єктів з масивів пар

[[«ключ1», «значення1»], [«ключ2», «значення2»], . . . , [«ключN», «значенняN»]]

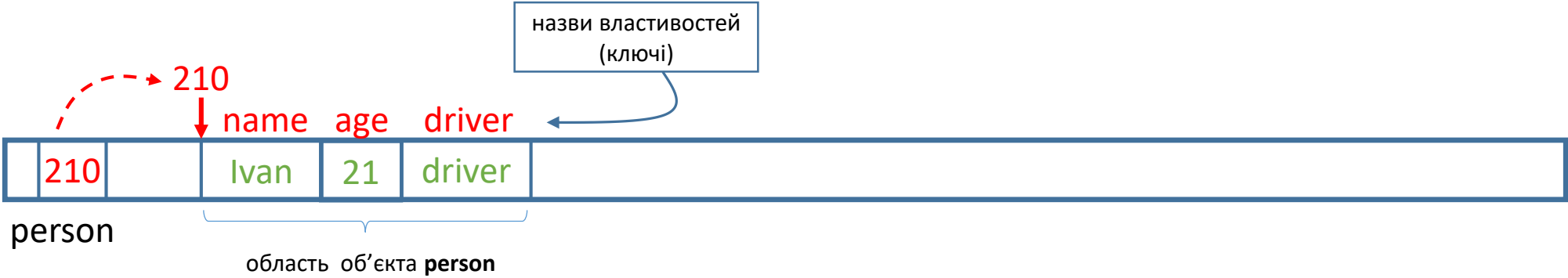
Якщо маємо набір назв і властивостей (ключів) і відповідних значень у формі масивів, то об'єкт може бути створено з використанням *Object.fromEntries()*

Загальна форма	<pre>let колекція_пар_ключ_значення = [[«ключ1» , «значення1»], [«ключ2» , «значення2»], . . . [«ключN» , «значенняN»]] let об'єкт = Object.fromEntries(колекція_пар_ключ_значення)</pre>
----------------	--

Приклад	<pre>let autoData = [['model' , 'Audi'], ['weight' , 2100], ['owner' , 'Ivan']] let auto = Object.fromEntries(autoData)</pre> 	<pre>auto= { model : 'Audi', weight : 2100, owner : 'Ivan' }</pre>
---------	---	--

Копіювання об'єктів

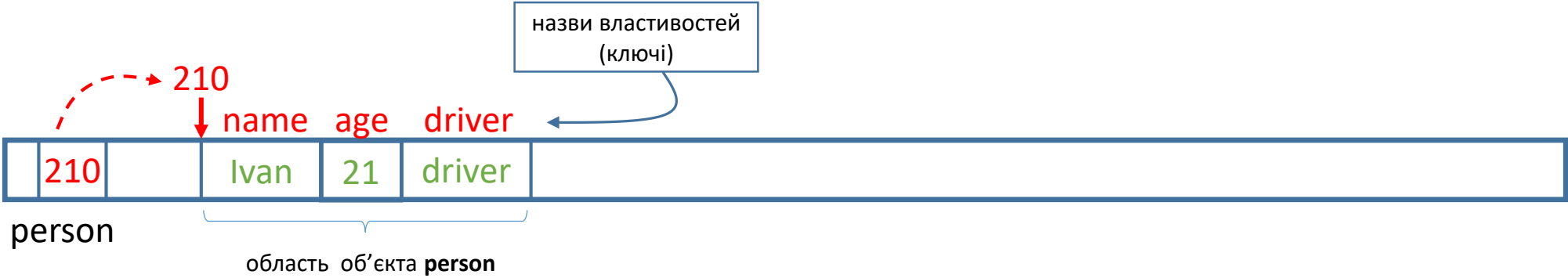
```
let person={  
  name  : 'ivan',  
  age    : 21,  
  position: 'driver'  
}
```



Копіювання об'єктів

```
let person={
  name  : 'ivan' ,
  age    : 21 ,
  position: 'driver'
}
```

let person2 = person

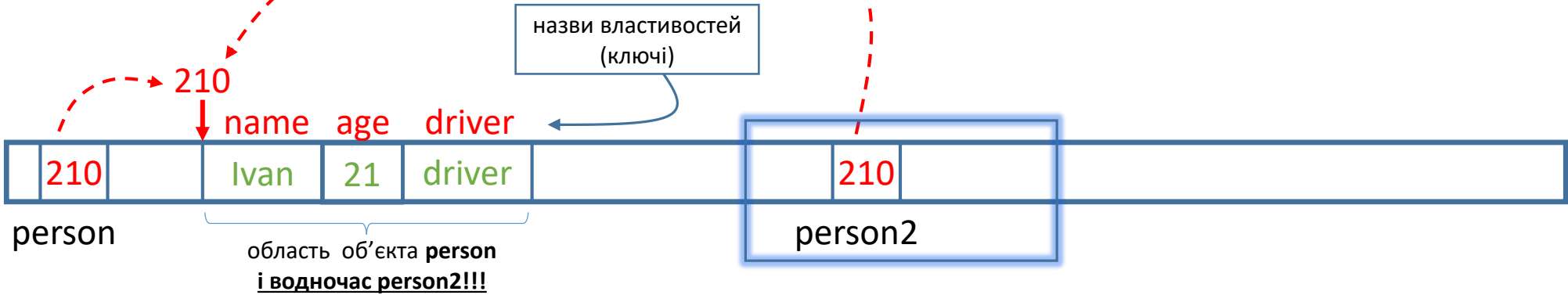


Копіювання об'єктів

```
let person={
  name  : 'ivan',
  age    : 21,
  position: 'driver'
}
```

```
let person2 = person
```

ТАК КОПІЮВАТИ ОБ'ЄКТИ НЕМОЖНА !!!
Буде скопійовано адресу!!!!



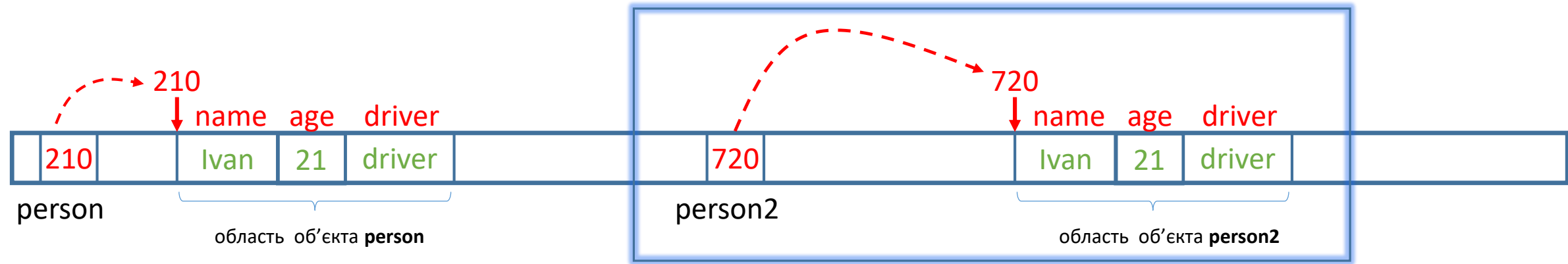
Копіювання об'єктів

```
let person={
  name  : 'Ivan' ,
  age    : 21 ,
  position: 'driver'
}
```

```
let person2 = { ... person }
```

*Поверхнєве копіювання об'єктів
з використанням спред оператора*

{ ... об'єкт_для_копіювання }



```
let student = {  
  firstName: 'Ivan',  
  secondName: 'Kein',  
  classNumber: 7,  
  scores: [9, 10, 8],  
}
```

```
let student2 = { ... student }
```

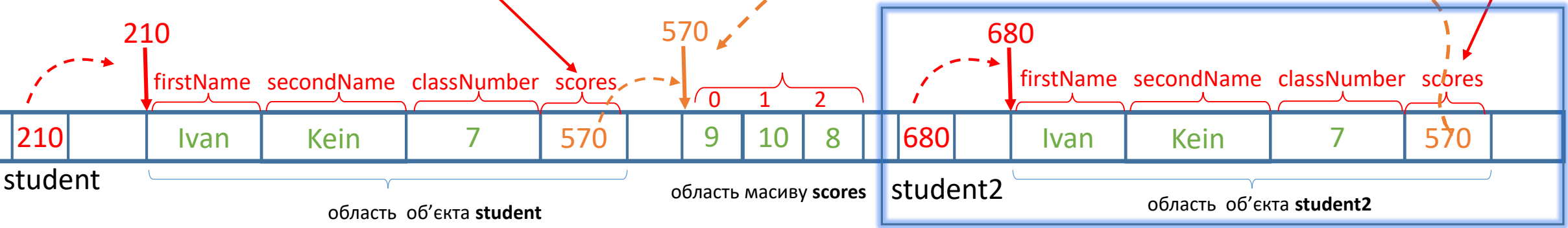
*Поверхнєве копіювання об'єктів
з використанням спред оператора*

{ ... об'єкт_для_копіювання }

**ОБЕРЕЖНО КОЛИ Є ПОСИЛАННЯ НА ВКЛАДЕНІ
МАСИВИ ЧИ ОБ'ЄКТИ !!!!!**

насправді **scores** містить адресу масиву

scores містить адресу масиву з об'єкта **student !!!**



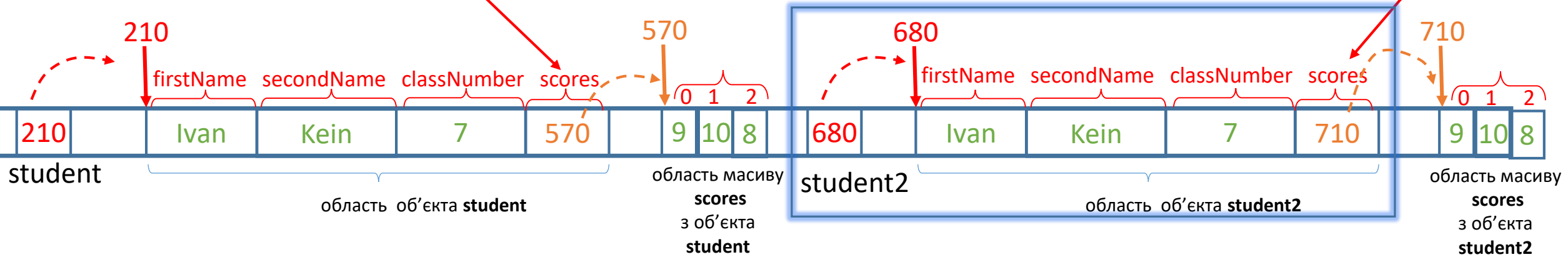
```
let student = {  
  firstName: 'Ivan',  
  secondName: 'Kein',  
  classNumber: 7,  
  scores: [9, 10, 8],  
}
```

```
let student2 = JSON.parse(JSON.stringify(student))
```

Повне копіювання з використанням
`JSON.parse(JSON.stringify(об'єкт_для_копіювання))`

насправді **scores** містить **адресу** масиву

scores містить **адресу** нового масиву



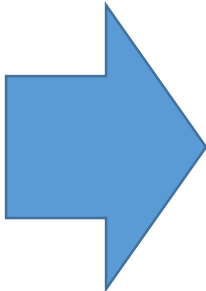
Копіювання властивостей об'єктів

Копіювання властивостей з одного об'єкта в інший об'єкт з використанням *Object.assign*

Загальна форма	<code>Object.assign(об'єкт_куди_копівювати , список_об'єктів_звідки_копівювати)</code>
----------------	--

Приклад	<pre>let obj1 = { model : 'Audi', weight : 2100 , owner : 'Ivan' } let obj2 = { price : 35000, fuel : 'diesel', }</pre>
---------	---

	<pre>let obj3 = { designType : 'coupe', author : 'Gucci Sara', }</pre>
	----- з використанням <i>Object.assign</i> -----
	<pre>Object.assign(obj1 , obj2 , obj3)</pre>
	----- з використанням спред оператора -----
	<pre>obj1 = { ... obj1 , ...obj2 , ...obj3 }</pre>



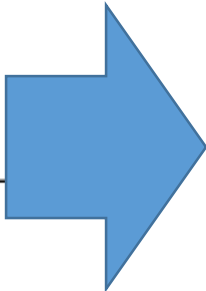
<pre>obj1 = { model : 'Audi', weight : 2100 , owner : 'Ivan' price : 35000, fuel : 'diesel', designType : 'coupe', author : 'Gucci Sara', }</pre>

Копіювання властивостей об'єктів

Копіювання властивостей з одного об'єкта в інший об'єкт з використанням *Object.assign*

Загальна форма	<code>Object.assign(об'єкт_куди_копіювати , список_об'єктів_звідки_копіювати)</code>
----------------	--

Приклад	<pre>Let obj0 = { } //Порожній об'єкт, можна і так: <u>Let obj = new Object()</u> Let obj1 = { model : 'Audi', weight : 2100 , price : 20000, } Let obj2 = { model : 'Opel', price : 35000, fuel : 'diesel', } ----- з використанням Object.assign ----- Object.assign(obj0, obj1 , obj2) ----- з використанням спред оператора ----- obj0 = { ... obj1 , ...obj2 }</pre>
---------	---



obj0 = {

 model : 'Opel',
 weight : 2100 ,
 price : 35000,
 fuel : 'diesel',

}

Копіювання властивостей об'єктів

Копіювання властивостей з одного об'єкта в інший об'єкт з використанням *Object.assign*

Загальна форма	<code>Object.assign(об'єкт_куди_копіювати , список_об'єктів_звідки_копіювати)</code>
----------------	--

Приклад	<pre>Let obj0 = { } //Порожній об'єкт, можна і так: <u>Let obj = new Object()</u> Let obj1 = { model : 'Audi', weight : 2100 , price : 20000, } Let obj2 = { model : 'Opel', price : 35000, fuel : 'diesel', }</pre> <p>Будуть перевизначені, бо у наступному об'єкті також визначено властивості з таким іменем</p> <p>----- з використанням <i>Object.assign</i> -----</p> <pre>Object.assign(obj0, obj1 , obj2)</pre> <p>----- з використанням спред оператора -----</p> <pre>obj0 = { ... obj1 , ...obj2 }</pre>
---------	--

obj0 = {
 model : 'Opel',
 weight : 2100 ,
 price : 35000,
 fuel : 'diesel',
}

Задача. Розробити функцію, яка буде аналізувати дані користувача, і у якій будуть значення параметрів за замовчуванням (використати спред оператор)

Деструктуризація об'єктів

Якщо необхідно у формі змінних або констант отримати деякі значення з об'єкта, то можна використати деструктуризацію

Загальна форма	<code>let {«ключ1», «ключ2», ..., «ключN» } = об'єкт</code>
Приклад	<pre>let objAuto = { model : 'Audi', weight : 2100 , price : 20000, } let { model, price } = objAuto // model='Audi', price=20000</pre>

Деструктуризація об'єктів

Якщо необхідно у функції у формі змінних отримати деякі значення з об'єкта, то можна використати деструктуризацію

Загальна форма	<pre>function назва_функції({«ключ1», «ключ2», ..., «ключN» }) { деякі операції з ключами як з змінними }</pre>
Приклад	<pre>function getTotalPrice({price, productNumber} { return price * productNumber //буде передано price=20000, productNumber=10 } Let objAuto = { model : 'Audi', weight : 2100 , price : 20000, productNumber: 10 } Let totalPrice = getTotalPrice (objAuto)</pre>

Задача. Розробити функцію, у яку передають об'єкт (день, місяць, рік). Визначити, який буде місяць через N місяців.

Задача. Розробити функцію, у яку передають об'єкт (день, місяць, рік). Визначити, який буде рік через N місяців.

Задача. Розробити функцію, яка б генерувала випадковим чином покази годинника (години, хвилини, секунди)