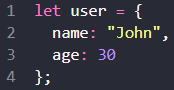
**OBJECT BASICS**

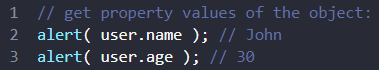
Seven data types called “primitive”, because their values contain only a single thing. object sama seperti array, yang menyimpan banyak nilai dalam 1 variabel, bedanya array diakses dengan index sedangkan object diakses dengan key.

An object can be created with figure brackets {…} with an optional list of properties. A property is a “key: value” pair, where key is a string/symbols (also called a “property name”), and value can be anything.

1. **Syntax (check c. summary part 1)**

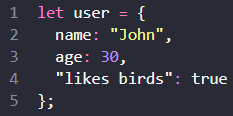
An empty object can be created using one of two syntaxes:

Property values are accessible using the dot notation:

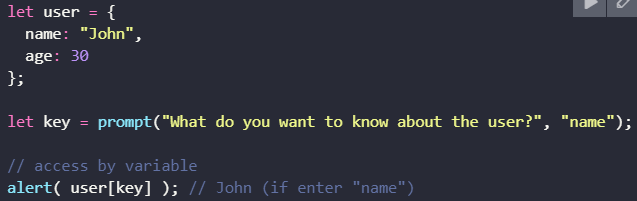




To remove a property, we can use the delete operator:

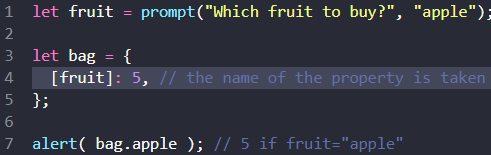
multiword key must be quoted:

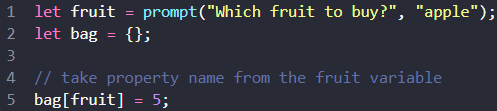
multiword key are only accessible using square bracket notation:

Square brackets allow to taking key from variable:

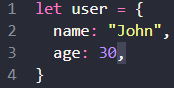
The dot notation cannot be used in a similar way^^:

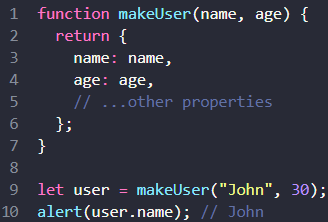
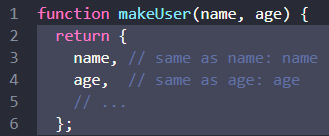
most of the time, when property names are known and simple, the dot is used. And if we need something more complex, then we switch to square brackets.

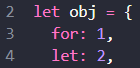
* Computed Properties: use square brackets in an object literal

^^That same as:

* The last property in the list may end with a comma:

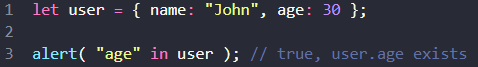
That is called a “trailing” or “hanging” comma. Makes it easier to add/remove/move around properties, because all lines become alike.

* In real code, we often use existing variables as values for property names
* Property names limitations

There are no limitations on property names, like “for”, “let”, “return” etc.

* ‘in’ operator (property existence test)

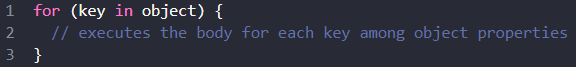
Syntax:

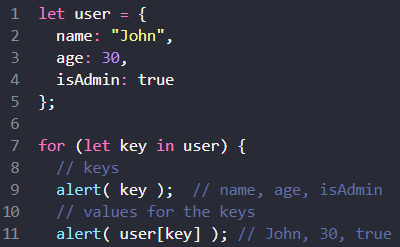


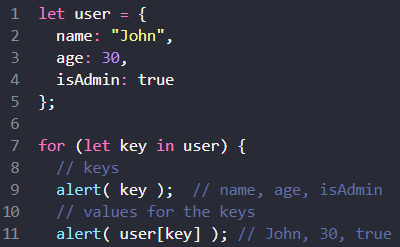
Example:

1. **“for..in” loop**

To walk over all keys of an object, there exists a special form of the loop: for..in.

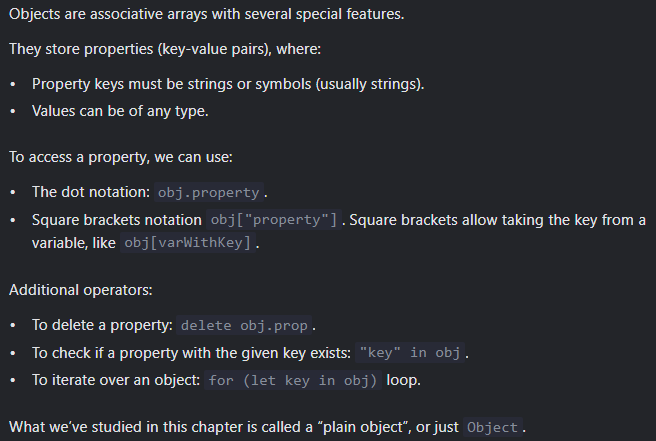
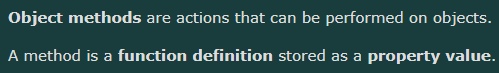
Syntax:

Example:

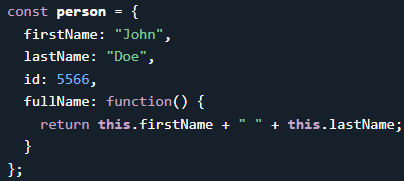
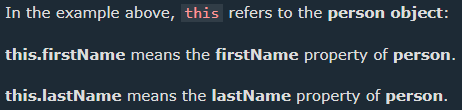


* Integer properties are sorted (automatically)

Adding a plus "+" sign before each code to fix the issue

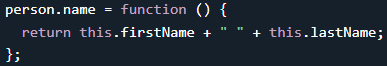
1. **Summary part 1**
2. **Object Methods**

(hanya?) Object literal mendukung method.

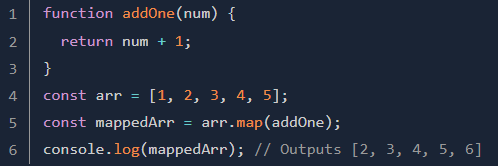
* **This**
* Access Object Methods

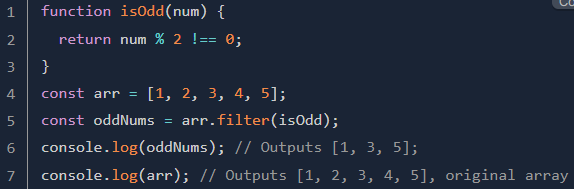
****Example:

If you access the fullName property without (), it will return the function definition:

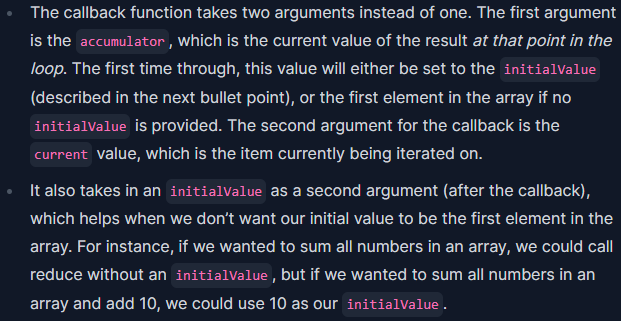
* Adding a method (same with adding property)

1. **Intermediate/Advanced Array Method**
2. Map(function(currentValue, index, arr), thisValue) : (creates new array)performing function to every array element. Example: (more in 5. Example)

better if :

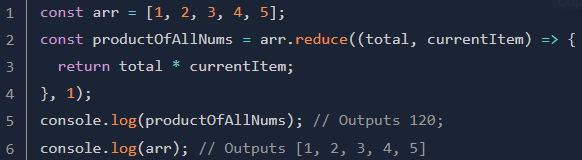
1. Filter() : (creates new array)return (all) element that match the function.
2. Reduce() : (creates new array) returns a single value(the function's accumulated result). This method is very useful for various operations such as adding values, multiplying values, combining objects, and so on.

Syntax:



**Cara kerja reduce** adalah loop through every element until the last element

* sebelum loop: accumulator merupakan parameter kedua dari reduce() ATAU elemen pertama dari array.
* setelah loop: accumulator merupakan hasil return dari loop sebelumnya ATAU hasil return loop saat ini(jika saat ini adalah loop terakhir).

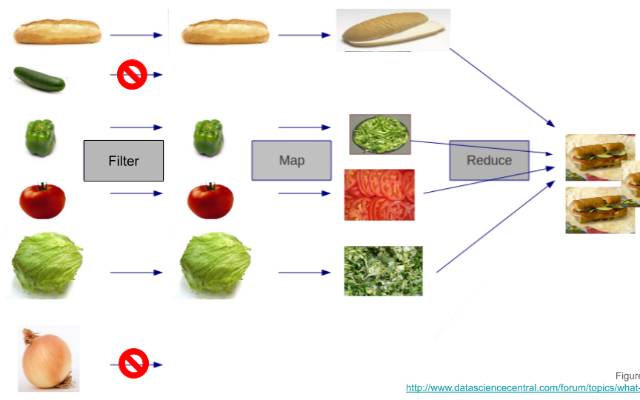
Example: (more in 5. Example)

Penjelasan:

1. loop ke-1: total adalah 1 (karena 1 merupakan elemen pertama dari arr) & currentItem adalah 1.
2. return 1 (karena 1 \* 1)
3. loop ke-2: total adalah 1 (karena loop sebelumnya me-return 1) & currentItem adalah 2
4. return 2 (karena 1 \* 2)
5. loop ke-3: total adalah 2 (karena loop sebelumnya me-return 2) & currentItem adalah 3
6. return 6 (karena 2 \* 4)
7. loop ke-4: total adalah 6 & currentItem adalah 4
8. return 24 (karena 6 \* 4)
9. loop ke-5 total adalah 24 & currentItem adalah 5
10. return 120

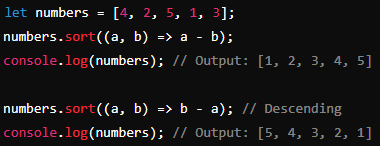
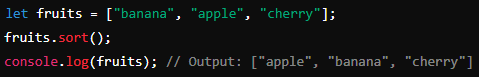
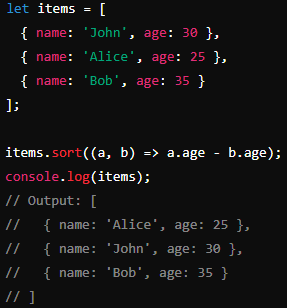
Output: 120 karena loop terakhir me-return 120

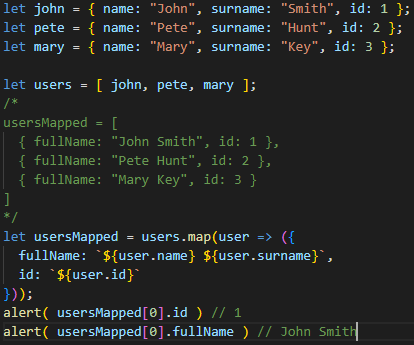
note: callback function is a function in reduce().

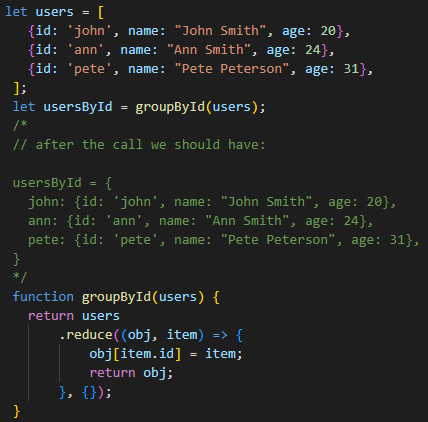
* Picture

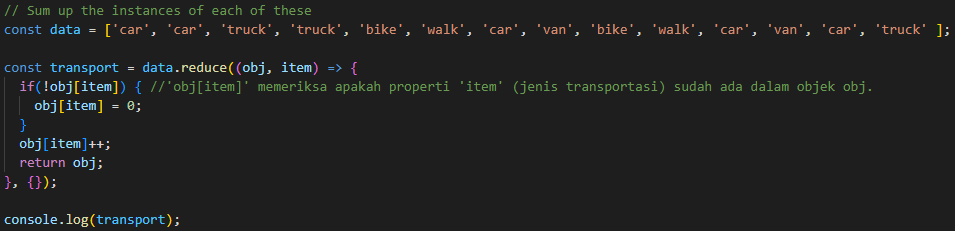
1. Sort() : mengurutkan elemen-elemen dalam sebuah array(sesuai urutan karakter Unicode)

Syntax :

* Mengurutkan Angka
* Mengurutkan String
* Mengurutkan Objek berdasarkan Properti

1. Example
2. Map() : mapping array of obj to array of ob
3. Reduce()

Array of object to object

Sum the same array element

Object property processing:

