



Object



Object

Object, merupakan satu dari delapan tipe data pada *Javascript*. Tujuh type data lainnya merupakan data *primitive* karena hanya bisa berisikan satu *value*(*string*, *number*, atau yang lain).

Object digunakan untuk menyimpan satu ataupun banyak data. *Object* berisikan daftar *properties*. Sebuah *property* adalah sebuah pasangan dari [*key* : *value*].



Pembuatan object

Dapat dilakukan dengan dua cara;

1. *Object constructor syntax.*
2. *Object literal syntax.*

```
// Object constructor syntax
let objectPertama = {};
// Object literal syntax
let objectKedua = new Object();

console.log(objectPertama);  {}
console.log(objectKedua);   {}
```



Menambahkan property pada object

Bisa dilakukan dengan berbagai cara;

1. Inisialisasi langsung pada *object*.
2. Menggunakan *dot notation*;
 - a. Digunakan pada *key* yang memiliki hanya satu kata.
3. Menggunakan *bracket notation*;
 - a. Digunakan pada *key* yang memiliki satu kata atau lebih.

```
let student = {  
  'first name': 'John',  
};  
  
student.age = 30;  
student['last name'] = 'Doe';
```

```
{ 'first name': 'John', age: 30, 'last name': 'Doe' }
```



Method pada object

Kita juga dapat menambahkan *method/function* ke dalam sebuah *object*.

```
let student = {  
  'first name': 'John',  
  grade: 'A',  
  sayHi() {  
    return this['first name'];  
  },  
  sayGrade() {  
    return this.grade;  
  },  
};
```



Mengakses value dari property

Mengakses value dari sebuah *property*, kita juga dapat mengakses dengan *dot* maupun *bracket notation*.

Syntax-nya seperti pada gambar disamping.

```
console.log(student.age); 30  
console.log(student['first name']); 'John'
```



Menghapus property

Dilakukan dengan menggunakan *keyword delete*. Sama seperti menambahkan, menghapus dapat menggunakan *dot* maupun *bracket notation*.

```
delete student.age;  
delete student['first name'];  
  
{ 'last name': 'Doe' }
```



Optional Chaining

Merupakan sebuah cara “aman” untuk mengakses *nested object property* dalam *objects*. Yang dimaksud aman adalah, tidak menimbulkan *error*/menghentikan program ketika *property* yang ingin diakses tidak ada(*null/undefined*).

```
console.log(student.address); undefined
```

```
console.log(student.address.street); Cannot read properties of undefined (reading 'street')
```

```
console.log(student.address?.street); undefined
```




For in loop

Merupakan metode untuk meng-*iterasi* sebuah *object*.

```
for (let key in student) {  
  console.log(key);    'first name', 'age', 'last name'  
  console.log(student[key]);  'John', 30, 'Doe'  
}
```



Object Method “this”

Biasanya, *method* pada *object* memerlukan akses ke *property* yang disimpan dalam *object* tersebut.

```
console.log(student.sayHi());  'John'  
console.log(student.sayGrade());  'A'
```

```
let student = {  
  'first name': 'John',  
  grade: 'A',  
  sayHi() {  
    return this['first name'];  
  },  
  sayGrade() {  
    return this.grade;  
  },  
};
```



Referensi Belajar

Dalam *programming*, kita harus siap “bosan” untuk belajar.

- [Javascript.info](https://javascript.info)