

Object Methods (Lengkap)

assign - Object.assign()

Penjelasan: Menyalin properti enumerable dari satu atau lebih objek sumber ke objek target.

Syntax: `Object.assign(target, ...sources)`

Contoh:

```
Object.assign({a: 1}, {b: 2});
```

Output:

```
{a: 1, b: 2}
```

create - Object.create()

Penjelasan: Membuat objek baru dengan prototipe dan properti tertentu.

Syntax: `Object.create(prototype, propertiesObject)`

Contoh:

```
const obj = Object.create({greet() { return 'hi'; }});  
obj.greet();
```

Output:

```
'hi'
```

defineProperty - Object.defineProperty()

Penjelasan: Menambahkan atau memodifikasi properti langsung pada objek.

Syntax: `Object.defineProperty(obj, prop, descriptor)`

Contoh:

```
const obj = {};  
Object.defineProperty(obj, 'x', { value: 42 });  
obj.x;
```

Output:

```
42
```

entries - Object.entries()

Penjelasan: Mengembalikan array pasangan key-value dari properti enumerable objek.

Syntax: `Object.entries(obj)`

Contoh:

```
Object.entries({a: 1, b: 2});
```

Output:

```
[[ 'a', 1 ], [ 'b', 2 ]]
```

freeze - Object.freeze()

Penjelasan: Membekukan objek, mencegah modifikasi terhadap properti yang ada.

Syntax: `Object.freeze(obj)`

Contoh:

```
const obj = {a: 1};  
Object.freeze(obj);  
obj.a = 2;  
obj.a;
```

Output:

```
1
```

Object Methods (Lengkap)

defineProperties - Object.defineProperties()

Penjelasan: Menambahkan beberapa properti ke objek sekaligus.

Syntax: `Object.defineProperties(obj, props)`

Contoh:

```
const obj = {};  
Object.defineProperties(obj, {  
  x: { value: 1 },  
  y: { value: 2 }  
});  
[obj.x, obj.y];
```

Output:

```
[1, 2]
```

fromEntries - Object.fromEntries()

Penjelasan: Mengubah array pasangan [key, value] menjadi objek.

Syntax: `Object.fromEntries(iterable)`

Contoh:

```
Object.fromEntries([['a', 1], ['b', 2]]);
```

Output:

```
{a: 1, b: 2}
```

getOwnPropertyDescriptor - Object.getOwnPropertyDescriptor()

Penjelasan: Mengembalikan deskripsi properti langsung dari objek.

Syntax: `Object.getOwnPropertyDescriptor(obj, prop)`

Contoh:

```
Object.getOwnPropertyDescriptor({x: 42}, 'x');
```

Output:

```
{ value: 42, writable: true, enumerable: true, configurable: true }
```

getOwnPropertyDescriptors - Object.getOwnPropertyDescriptors()

Penjelasan: Mengembalikan semua deskriptor properti dari objek.

Syntax: `Object.getOwnPropertyDescriptors(obj)`

Contoh:

```
Object.getOwnPropertyDescriptors({x: 1, y: 2});
```

Output:

```
{ x: {...}, y: {...} }
```

getOwnPropertyNames - Object.getOwnPropertyNames()

Penjelasan: Mengembalikan array nama properti milik langsung objek (termasuk non-enumerable).

Syntax: `Object.getOwnPropertyNames(obj)`

Contoh:

```
Object.getOwnPropertyNames({a: 1, b: 2});
```

Output:

```
['a', 'b']
```

Object Methods (Lengkap)

getOwnPropertySymbols - Object.getOwnPropertySymbols()

Penjelasan: Mengembalikan array simbol milik langsung objek.

Syntax: Object.getOwnPropertySymbols(obj)

Contoh:

```
const sym = Symbol('s');  
const obj = {[sym]: 123};  
Object.getOwnPropertySymbols(obj);
```

Output:

```
[Symbol(s)]
```

getPrototypeOf - Object.getPrototypeOf()

Penjelasan: Mengembalikan prototipe dari objek.

Syntax: Object.getPrototypeOf(obj)

Contoh:

```
Object.getPrototypeOf({});
```

Output:

```
Object.prototype
```

hasOwn - Object.hasOwn()

Penjelasan: Memeriksa apakah objek memiliki properti sendiri (bukan dari prototype).

Syntax: Object.hasOwn(obj, prop)

Contoh:

```
Object.hasOwn({a: 1}, 'a');
```

Output:

```
true
```

is - Object.is()

Penjelasan: Membandingkan dua nilai untuk kesamaan (seperti === tapi lebih tepat).

Syntax: Object.is(value1, value2)

Contoh:

```
Object.is(NaN, NaN);
```

Output:

```
true
```

isExtensible - Object.isExtensible()

Penjelasan: Memeriksa apakah properti baru bisa ditambahkan ke objek.

Syntax: Object.isExtensible(obj)

Contoh:

```
Object.isExtensible({});
```

Output:

```
true
```

isFrozen - Object.isFrozen()

Penjelasan: Memeriksa apakah objek dibekukan.

Syntax: Object.isFrozen(obj)

Contoh:

```
Object.isFrozen(Object.freeze({}));
```

Output:

```
true
```

Object Methods (Lengkap)

isSealed - Object.isSealed()

Penjelasan: Memeriksa apakah objek disegel.

Syntax: Object.isSealed(obj)

Contoh:

```
Object.isSealed(Object.seal({}));
```

Output:

```
true
```

keys - Object.keys()

Penjelasan: Mengembalikan array nama properti enumerable.

Syntax: Object.keys(obj)

Contoh:

```
Object.keys({a: 1, b: 2});
```

Output:

```
['a', 'b']
```

preventExtensions - Object.preventExtensions()

Penjelasan: Mencegah penambahan properti baru ke objek.

Syntax: Object.preventExtensions(obj)

Contoh:

```
const obj = {};  
Object.preventExtensions(obj);  
Object.isExtensible(obj);
```

Output:

```
false
```

seal - Object.seal()

Penjelasan: Menyegel objek agar tidak bisa ditambah atau hapus propertinya.

Syntax: Object.seal(obj)

Contoh:

```
const obj = {a: 1};  
Object.seal(obj);  
delete obj.a;  
obj.a;
```

Output:

```
1
```

setPrototypeOf - Object.setPrototypeOf()

Penjelasan: Mengatur prototipe dari objek.

Syntax: Object.setPrototypeOf(obj, prototype)

Contoh:

```
const proto = {greet() { return 'hi'; }};  
const obj = {};  
Object.setPrototypeOf(obj, proto);  
obj.greet();
```

Output:

```
'hi'
```

Object Methods (Lengkap)

values - Object.values()

Penjelasan: Mengembalikan array dari nilai properti enumerable.

Syntax: `Object.values(obj)`

Contoh:

```
Object.values({a: 1, b: 2});
```

Output:

```
[1, 2]
```