

OBJECT	
Khái báo	<pre> var myDog = { name: "Lợn", isAlive: false } </pre>
Tuy xuất	<pre> console.log(myDog); console.log(myDog.name); </pre> <p>Tuy xuất tất cả object // { ... } Tuy xuất phần tử trong object</p>
Thay đổi	<pre> myDog.name = "chó"; </pre> <p>// chó</p>
Cách khai thống đại diện()	<pre> myDog['name'] </pre> <p>// Phải có ngoặc ""</p>
ARRAY	<pre> console.log(newDog[0].name); </pre> <p>// "chó". tên array myDog là name.</p>
Phương thức của object	<p>tức là function sử dụng trong Object.</p> <pre> var myDog = { bark: function() { console.log("meo meo"); } } </pre>
this	<p>Đại diện cho myDog (object). dùng trong method (phương thức). // tóm lại this luôn là cái object có function.</p> <p>myDog.name. => this.name.</p>
Gán function có có tên như có biến	<pre> myDog = { weight: 5, eat: function() { this.weight = this.weight + 1; return this; } } </pre>
for in	<pre> var myDog = { weight: 5, age: 10 } for (var key in myDog) { console.log(key); } </pre> <p>// Lấy ra các phần tử weight, age => console.log(myDog[key])</p>
for of	<pre> var employees = [{ name: "thinh", age: 28 }, { name: "tam", age: 24 }]; for (var a of employees) { console.log(a.name); } </pre> <p>// thính thăm // a đại diện cho từng phần tử</p>

Tạo đối tượng từ lớp:

Sử dụng từ khóa new, xem tạo 1 vùng nhớ cho OOP và tạo vị trí tham chiếu OOP.
vd `carObject = new Car("Hyundai", "i10", 1100, "White");`
`carObject.start();`

Phương thức khởi tạo (đặc biệt do)

- + Tên phương thức trùng với tên lớp
- + Không cần khởi tạo biến địa chỉ.
- + dùng để tạo đối tượng, khởi tạo giá trị ban đầu cho lớp
- + Được gọi sử dụng ngay khi được tạo ra bằng từ khóa new
vd `new carObject (name, model...)`

3 cách Sử dụng OOP

① Sử dụng Object literals.

```
var apple = {  
  type: "A"  
  color: "red"  
  getInfo: function() {  
    return this.color + this.type;  
  }  
};
```

Thay cấp thuộc tính
vào phương thức.
`apple.color = "B";`
// đổi tên lại.
`alert(apple.getInfo());`
↑ xuất kết quả

② Sử dụng hàm hàm tạo

+ tạo function.

```
function Apple (type) {  
  this.type = type;  
  this.getInfo = function() {  
    ...  
  }  
};
```

Sử dụng từ khóa new để tạo OOP.
`var apple = new Apple('A');`

③ Sử dụng Singleton.

```
var apple = new function() {  
  this.type = 'A';  
  ...  
};
```

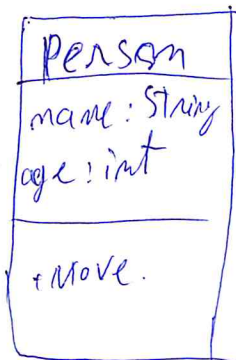

Ngôn ngữ lập trình
đối tượng
OOP

- Mã lệnh và dữ liệu. Kết hợp với nhau
- Đối tượng: + Thuộc tính: các dữ liệu tính chất của Oop
+ Hành vi: các hoạt động, hành động của OOP thực hiện
+ Các đối tượng có thể quan hệ với nhau.

Các khái niệm

Đối tượng (Object)
Lớp (class)
Thuộc tính (Property)
Phương thức (Method)

Attributes
Fields



Object

Tính bao gói
tính kế thừa

Tách xuất nhúng thành phần nhưng vẫn tồn tại toàn thể.
đối tượng có thể chia sẻ hay mở rộng mà không cần cần tìm định nghĩa.

Khi tạo
đối tượng.

```
Var person = {
  firstName: "John",
  age: 50
};
```

Định nghĩa Lớp

dùng function. hàm khai báo đối tượng gọi là lớp
function. person (firstName, age) {
 this.firstName = firstName;
 this.age = age;
}

Tách xuất sử dụng
thuộc tính

Cú pháp objectName.property.

vd: person.age. → myFather.age.

với lớp: objectName["property"]

vd: person["age"] → myFather["age"]