

```
class Person{
  constructor(name){
    console.log(`${name}`);
  }
}
let p = new Person("json")
```

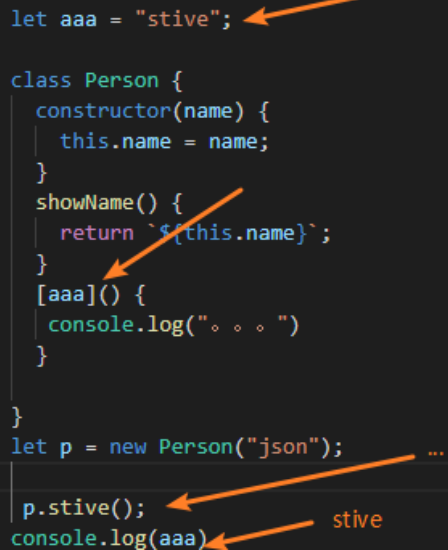
```
function Person(name) {
  this.name= name;
}
Person.prototype.showName =function() {
  return `name:${this.name}`;
}
let p1 = new Person("json")

console.log(p1.showName())
```

```
32
33
34 function Person(name) {
35   this.name= name;
36   }
37   Person.prototype.showName =function() {
38     return `name:${this.name}`;
39   }
40   let p1 = new Person("json")
41
42   console.log(p1.showName())
```

```
94 class Person{
95   constructor(name){
96     this.name = name;
97   }
98   showName(){
99     return `${this.name}` ;
100   }
101 }
102 let p = new Person("json")
```

```
let aaa = "stive";  
  
class Person {  
  constructor(name) {  
    this.name = name;  
  }  
  showName() {  
    return `${this.name}`;  
  }  
  [aaa]() {  
    console.log("。。。")  
  }  
}  
let p = new Person("json");  
  
p.stive();  
console.log(aaa)
```



The diagram illustrates the relationship between the variable `aaa` and the method `stive`. An orange arrow points from the string `"stive"` in the `let aaa = "stive";` line to the `[aaa]()` property access in the `Person` class. Another orange arrow points from the `stive` property access in `p.stive();` to the `[aaa]()` property access in the class. A third orange arrow points from the `aaa` argument in `console.log(aaa)` to the `[aaa]()` property access in the class. The text `stive` is written next to the `p.stive();` line.

1: es6里面class没有提升功能，在es5种函数会有提升

2: this的问题 好点

class新增了 取值函数 getter /存值 setter 封装底层代码用的多

set 设置属性

get 获取属性

```

class Person {
  constructor(name) {
    this.name = name;
  }
  showName() {
    return `${this.name}`;
  }
  [aaa]() {
    console.log("。 。 。 ")
  }
  set age(age){
    console.log(age)
  }
  get age(){
    return `...`
  }
}

let p = new Person("json");
p.stive();
console.log(aaa)
p.age =10;
console.log(p.age);

```

设置属性的值

获取已经设置的值

```

class Person {
  constructor(name) {
    this.name = name;
  }
  static showName() {
    return `${this.name}`;
  }
  [aaa]() {
    console.log("。 。 。 ")
  }
  set age(age){
    console.log(age)
  }
  get age(){
    return `...`
  }
}

```

类直接调用