La méthode bind()

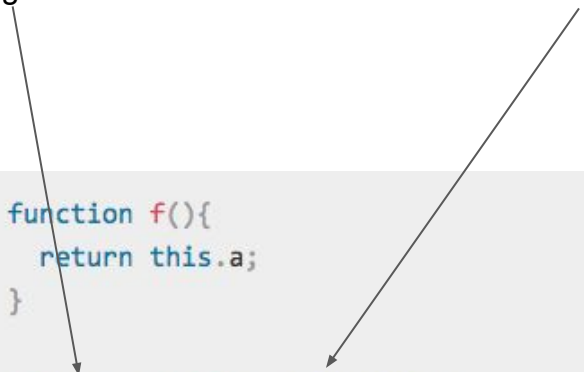
info :

Les méthodes bind,call et apply permette également de pouvoir fixé un argument a la fonction que l'on crée.

```
function presentMe(onomatopée){
  return console.log(onomatopée +" Hello it's me "+ this.name, this);
}

const person1 = {
  name: "Matt",
  present : presentMe
}

const person2 = {
  name: "Mike",
  present : presentMe
}

const myFunctionBind = presentMe.bind(person1,"ouch");
myFunctionBind();

presentMe.call(person1,"Hey");

presentMe.apply(person1,["Youhou"])
```

Avec ECMAScript 5, une nouvelle fonction fut introduite: Function.prototype.bind, Lorsqu'on appel f.bind(unObjet), on crée une nouvelle fonction qui possède le même corps et la même la même portée que f, mais ou 'this' sera lié, de façon permanente, au premier argument passé à bind(), quelle que soit la façon dont la méthode est utilisée.

'var g ' devient une nouvelle fonction et est lié de façon permanente au premier argument passé à bind().



```
function f(){
  return this.a;
}

var g = f.bind({a:"azerty"});
console.log(g()); // azerty

var h = g.bind({a:"coucou"}); // bind ne fonctionne qu'une seule fois
console.log(h()); // azerty

var o = {a:37, f:f, g:g, h:h};
console.log(o.f(), o.g(), o.h()); // 37, azerty, azerty
```