

```

0 </html>
1 <script>
2   // inheritance
3
4   let grandfather = {
5     name: "goku",
6     lastName: "sharma",
7     DNA: "123",
8   };
9
10  // let Father = {
11  //   name: "gohan",
12  //   lastName: "sharma",
13  //   DNA: "123",
14  // };
15
16  let Father = Object.create(grandfather); // it is
17  // creating a object with grandfather properties it has
18  // inherited;
19
20  Father.name = "gohan";
21
22  console.log(Father);
23 </script>

```

Elements Console Sources >>

1

top Filter Default levels

1 Issue: 1

2 messages

2 user mes...

No errors

No warnings

2 info

No verbose

Live reload enabled. index3.html:37

Object index3.html:68

name: "gohan" → further

[[Prototype]]: Object

DNA: "123"

lastName: "sharma"

name: "goku"

[[Prototype]]: Object

g grandfather

```

12 // Inheritance
13
14 let grandfather = {
15     name: "goku",
16     lastName: "sharma",
17     DNA: "123",
18 };
19
20 // let Father = {
21 //     name: "gohan",
22 //     lastName: "sharma",
23 //     DNA: "123",
24 // };
25
26 let Father = Object.create(grandfather); // it is
    creating a object with grandfather properties it has
    inherited;
27
28 Father.name = "gohan";
29 Father.houseName = "Dragon ball";
30 Father.bloodGroup = "o+";
31
32 let son = Object.create(Father);
33
34 son.name = "picalo";
35
36 console.log(son);

```

Elements Console Sources >>

1 Issue: 1

1 message

1 user mes...

No errors

No warnings

1 info

No verbose

index3.html:74

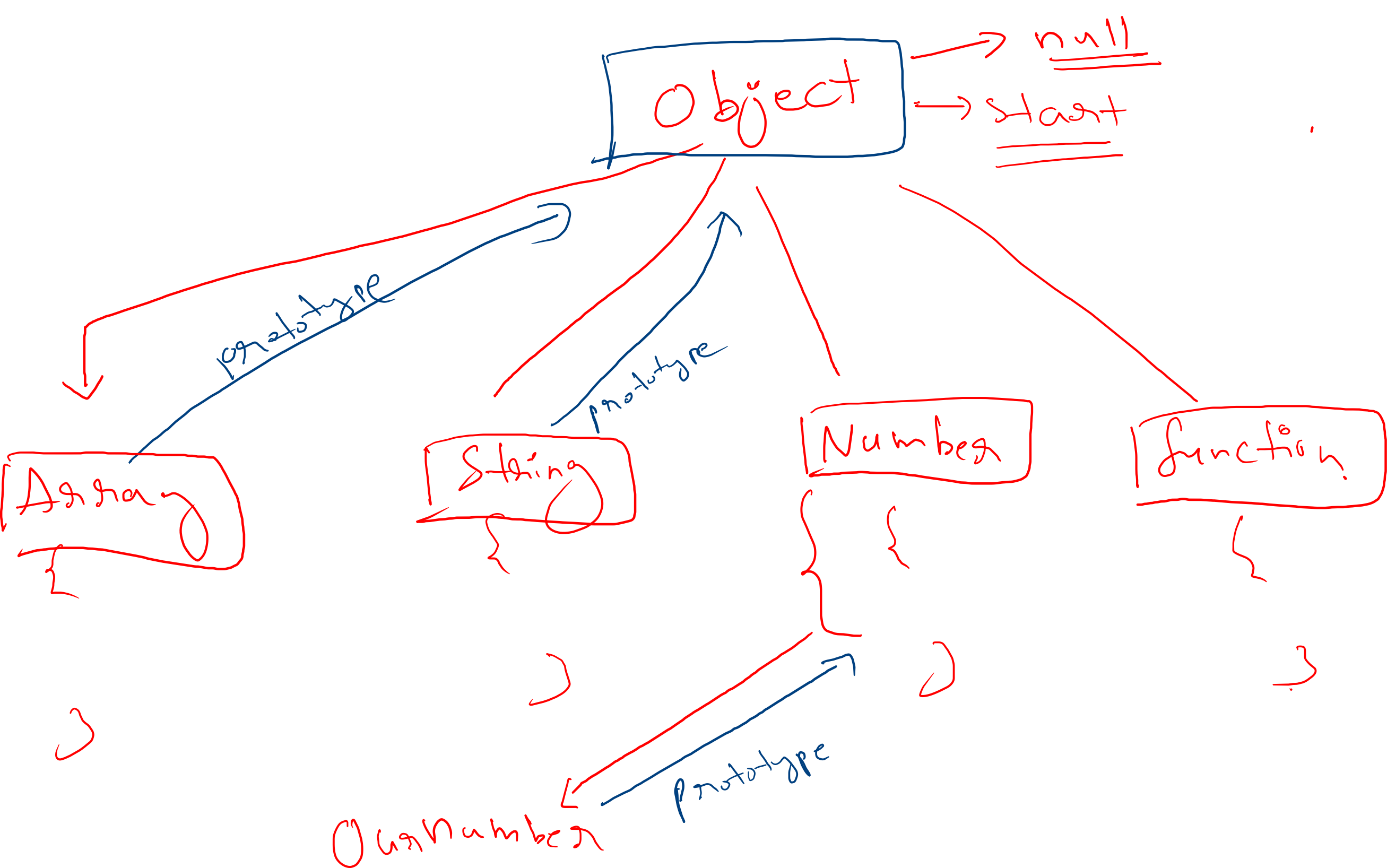
{name: 'picalo'}
 name: "picalo"
 [[Prototype]]: Object
 bloodGroup: "o+"
 houseName: "Dragon ball"
 name: "gohan"
 [[Prototype]]: Object
 DNA: "123"
 lastName: "sharma"
 name: "goku"
 [[Prototype]]: Object

Handwritten notes:
 } Father
 } Grandfather

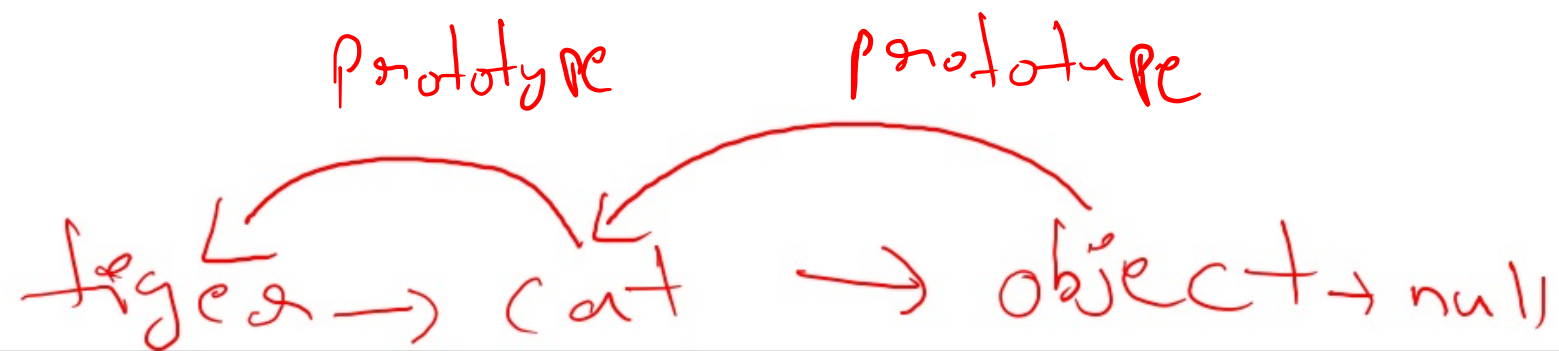
```
▼ {name: 'masai'} ⓘ
  name: "masai"
  ▼ [[Prototype]]: Object
    ► constructor: f Object()
    ► hasOwnProperty: f hasOwnProperty()
    ► isPrototypeOf: f isPrototypeOf()
    ► propertyIsEnumerable: f propertyIsEnumerable()
    ► toLocaleString: f toLocaleString()
    ► toString: f toString()
    ► valueOf: f valueOf()
    ► __defineGetter__: f __defineGetter__()
    ► __defineSetter__: f __defineSetter__()
    ► __lookupGetter__: f __lookupGetter__()
    ► __lookupSetter__: f __lookupSetter__()
    ▼ __proto__: Object
      ► constructor: f Object()
      ► hasOwnProperty: f hasOwnProperty()
      ► isPrototypeOf: f isPrototypeOf()
      ► propertyIsEnumerable: f propertyIsEnumerable()
      ► toLocaleString: f toLocaleString()
      ► toString: f toString()
      ► valueOf: f valueOf()
      ► __defineGetter__: f __defineGetter__()
      ► __defineSetter__: f __defineSetter__()
      ► __lookupGetter__: f __lookupGetter__()
      ► __lookupSetter__: f __lookupSetter__()
      ► __proto__: null
        ► get __proto__: f __proto__()
        ► set __proto__: f __proto__()
        ► get __proto__: f __proto__()
        ► set __proto__: f __proto__()
```

obj →

object → null



Prototypal Chain



Constructor inheritance

```
29
30 function person(firstName, lastName) {
31   this.firstName = firstName;
32   this.lastName = lastName;
33 }
34
35 person.prototype.displayFullName = function () {
36   console.log(this.firstName + " " + this.lastName);
37 };
38 person.prototype.city = "hyderabad";
39
40 let person1 = new person("ashish", "roul");
41 console.log(person1);
42
43 function student(firstName, lastName, batchNo, techStack) {
44   // person.call(this, firstName, lastName);
45   person.apply(this, [firstName, lastName]);
46   this.batchNo = batchNo;
47   this.techStack = techStack;
48 }
49
50 // Object.setPrototypeOf(student.prototype, person.prototype);
51
52 student.prototype = Object.create(person.prototype);
53
54 let student1 = new student("vishal", "sharma", 15, "MERN");
55
56 console.log(student1.displayFullName());
57 </script>
```

) → object

) — Prototype

) — object 2.

attach.