```
function Foo(){
}
const f1 = new Foo();
const o1 = new Object();
```

```
Foo
Input: console.log('Foo',Foo, typeof(Foo));
Output: 'Foo', function Foo(){}, 'function'
```

```
Foo()
Input: console.log('Foo()',new Foo(), typeof(new Foo()));
Output: 'Foo()',Foo {}, 'object'
```

```
Foo.prototype
Input: console.log('f1.__proto__',f1.__proto__, typeof(f1.__proto__))
Output: 'f1.__proto__', (0) {}, 'object'
```

```
Foo.prototype
Input: console.log('Foo.prototype',Foo.prototype, typeof(Foo.prototype))
Output: 'Foo.prototype', (0) {}, 'object'
```

```
Foo
Input: console.log('Foo.prototype.constructor',Foo.prototype.constructor,
typeof(Foo.prototype.constructor))
Output: 'Foo.prototype.constructor', function Foo(){}, 'function'
```

```
Function.prototype
Input: console.log('Foo.__proto__',Foo.__proto__, typeof(Foo.__proto__))
Output: 'Foo.__proto__', function () { [native code] }, 'function'
```

```
Oject()
Input: console.log('o1',o1, typeof(o1))
Output: 'o1' ,(0) {} ,'object'
```

```
Object.prototype
Input: console.log('o1.__proto__',o1.__proto__, typeof(o1.__proto__))
Output: 'o1.__proto__', (0) {} ,'object'
```

```
Object.prototype
Input: console.log('Object.prototype',Object.prototype, typeof(Object.prototype))
Output: 'Object.prototype', (0) {}, 'object'
```

```
Object
Input: console.log('Object.prototype.constructor',Object.prototype.constructor,
typeof(Object.prototype.constructor))
Output: 'Object.prototype.constructor', function Object() { [native code] }, 'function'
```

```
Function.prototype
Input: console.log('Object.__proto__',Object.__proto__, typeof(Object.__proto__))
Output: 'Object.__proto__', function () { [native code] }, 'function'
```

```
Function
Input: console.log('Function',Function,typeof(Function))
Output: 'Function', function Function() { [native code] }, 'function'
```

```
Function.prototype
Input: console.log('Function.prototype',Function.prototype,typeof(Function.prototype))
Output: 'Function.prototype', function () { [native code] }, 'function'
```

```
Function.prototype
Input: console.log('Function.__proto',Function.__proto__,typeof(Function.__proto__))
Output: 'Function.__proto', function () { [native code] }, 'function'
```

```
Function
Input:
console.log('Function.__proto.construtor',Function.__proto__.constructor,typeof(Function.__proto__.const
ructor))
```

```
Output: 'Function.__proto.construtor', function Function() { [native code] }, 'function'
```

```
Object.prototype
Input:
console.log('Function.__proto.__proto__',Function.__proto__.__proto__,typeof(Function.__proto__.__proto__))
Output: 'Function.__proto.__proto__', (0) {}, 'object'
```

```
Object.prototype
Input: console.log('Foo.__proto.__proto__',Foo.__proto__.__proto__,typeof(Foo.__proto__.__proto__))
Output: 'Foo.__proto.__proto__', (0) {}, 'object'
```

```
null
Input:
console.log('Object.prototype.__proto__',Object.prototype.__proto__,typeof(Object.prototype.__proto__))
Output: 'Object.prototype.__proto__', null, 'object'
```

```
null
Input:
console.log('Function.__proto.__proto__.__proto__',Function.__proto__.__proto__.__proto__,typeof(Function.__proto__.__proto__.__proto__))
Output: 'Function.__proto.__proto__.__proto__', null, 'object'
```

```
null
Input:
console.log('Foo.__proto.__proto__.__proto__',Foo.__proto__.__proto__.__proto__,typeof(Foo.__proto__.__proto__.__proto__))
Output: 'Foo.__proto.__proto__.__proto__', null, 'object'
```

```
constructors of all functions are always Function
__proto__ of all functions are always Function.prototype
```