Object

Um objeto é uma coleção dinâmica de chaves e valores, que podem ser de qualquer tipo, .

```
1.  {};
2.  new Object();
3.  Object.create(null);
```

Existem 3 formas de criar um objeto

```
1.  let book = {
2.    title: "Harry Potter",
3.    pages: 700,
4.    [Math.random()]: "Anything"
5.  };
6.  book["category"] = "Fiction";
7.  book[Symbol("publisher")] = "Bloomsbury";
8.  Object.assign(book, {
9.    price: 39.90,
10.   isbn: "9788700631625"
11. });
```

Atribuindo propriedades em um objeto

```
1.  let title = "Harry Potter";
2.  let pages = 700;
3.  let isbn = "9788700631625";
4.  let book = {title, pages, isbn};
```

Atribuindo propriedades em um objeto

```
1.  Object.keys(book);
2.  Object.entries(book);
3.  Object.values(book);
4.  for(let property in book) {
5.    book[property];
6.  }
7.  ('isbn' in book);
8.  book.hasOwnProperty('title');
```

Listando as propriedades de um objeto

```
1.  delete book.category;
2.  book.category = undefined;
3.  book.category = null;
```

Apagando as propriedades de um objeto

```
1.  Object.defineProperty(book, "author", {
2.    value: "J. K. Rowling",
3.    enumerable: false,
4.    writable: false,
5.    configurable: false
6.  };
```

Configurando restrições nas propriedades de um objeto

Propriedades

value – É o valor da propriedade
enumerable – Determina se a propriedade é visível e pode ser exibida em toString(), for/in, in, Object.keys
writable – Atua sobre o value, permitindo ou não que ele seja modificado
configurable – Está relacionado com todas as propriedades, menos o value
get – É uma função, invocada no momento em que a propriedade é consultada
set - É uma função, invocada no momento em que a propriedade é definida

```
1.    JSON.stringify(book);
2.    JSON.parse("{\"title\": \"Harry Potter\"}");
```

JSON - JavaScript Object Notation

```
1.  let book = {
2.    title: "Harry Potter"
3.  };
4.  book == book;
5.  book === book;
6.  Object.is(book, book);
7.  let movie = {
8.    title: "Harry Potter"
9.  };
10. book == movie;
11. book === movie;
12. Object.is(book, movie);
```

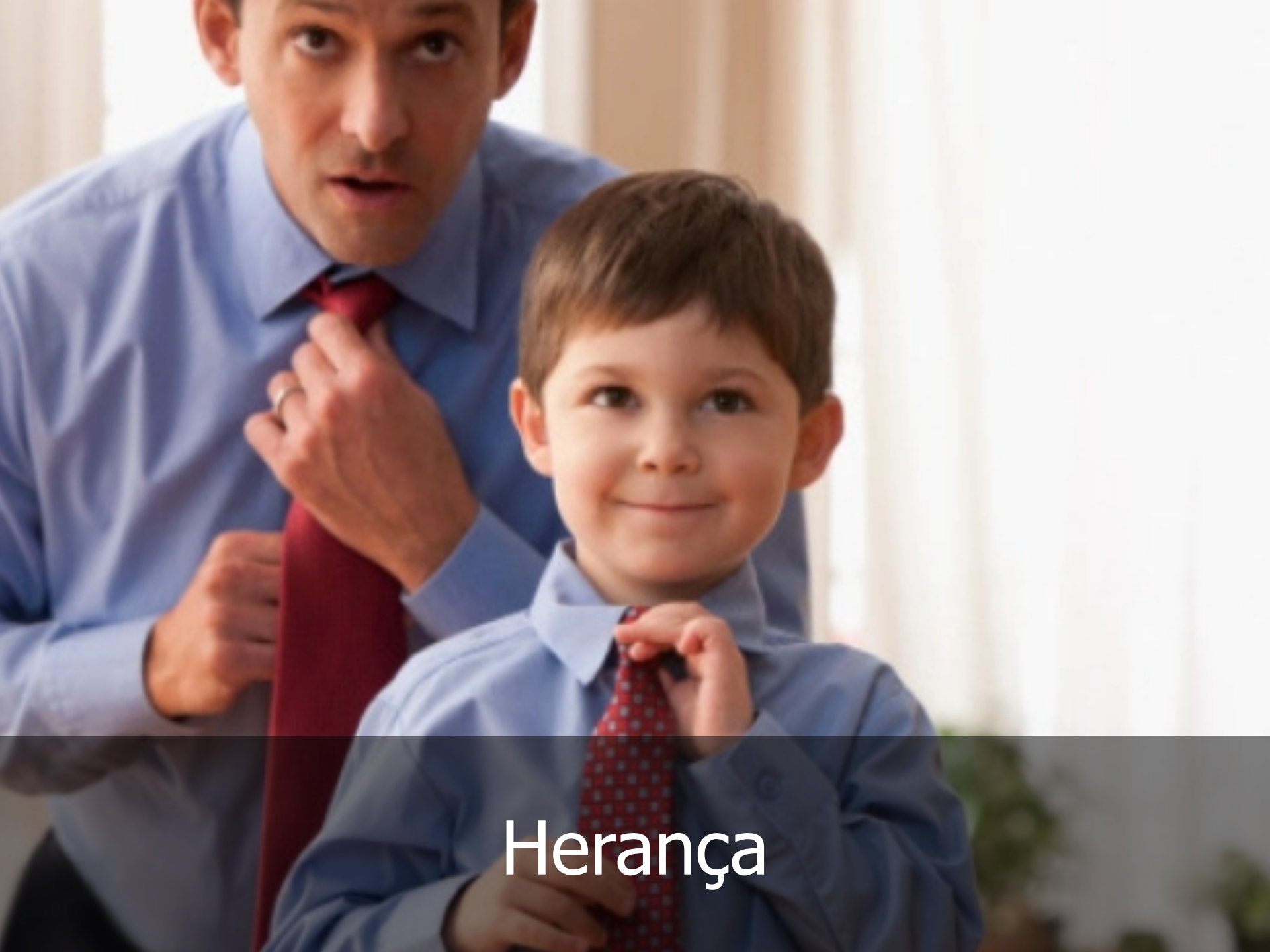
Comparando dois objetos

```
1.    JSON.stringify(book) == JSON.stringify(movie);
2.    JSON.stringify(book) === JSON.stringify(movie);
3.    Object.is(JSON.stringify(book), JSON.stringify(movie);
```

Comparando dois objetos utilizando JSON

```
1.    JSON.parse(JSON.stringify(book));
```

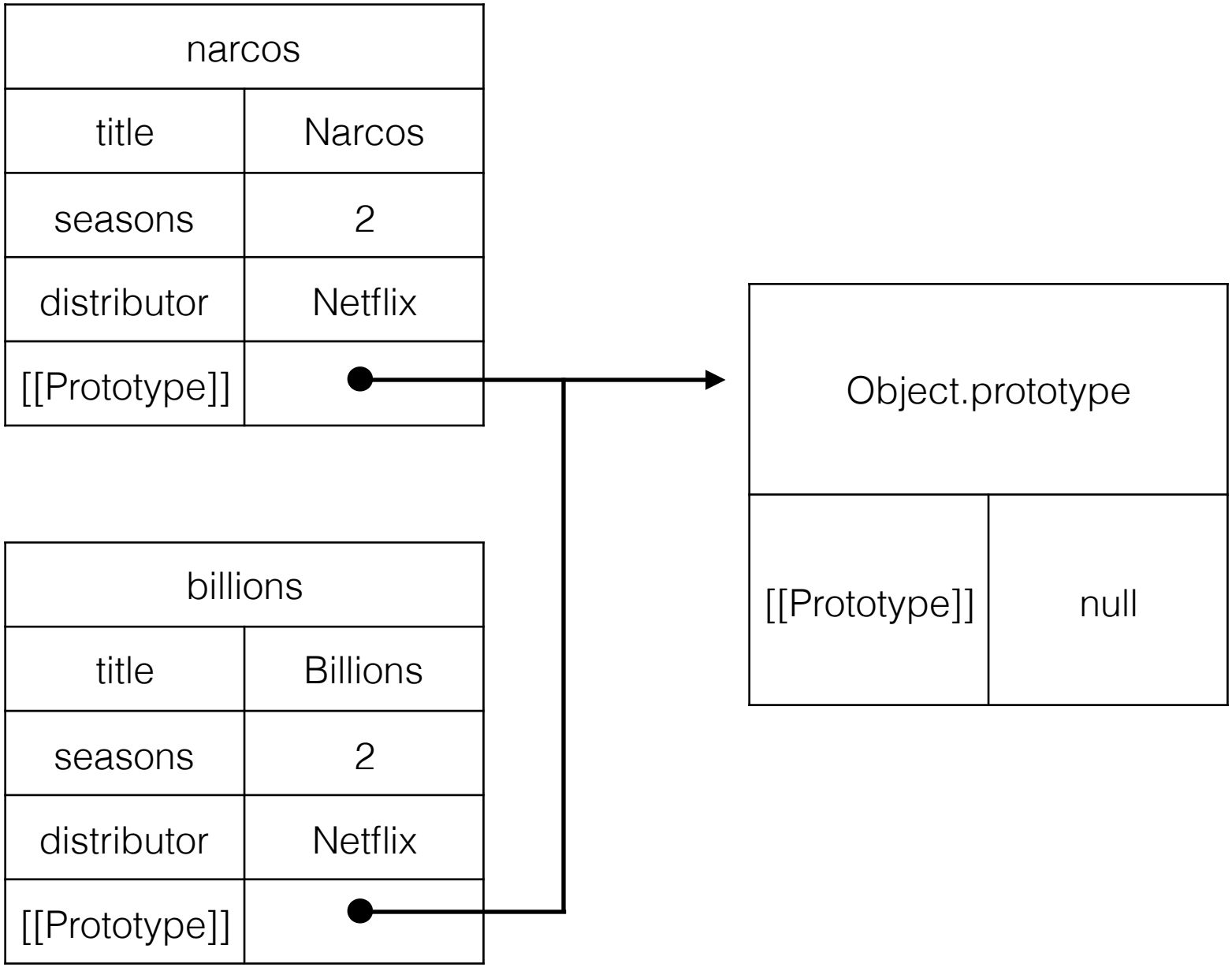
Clonando objetos utilizando JSON



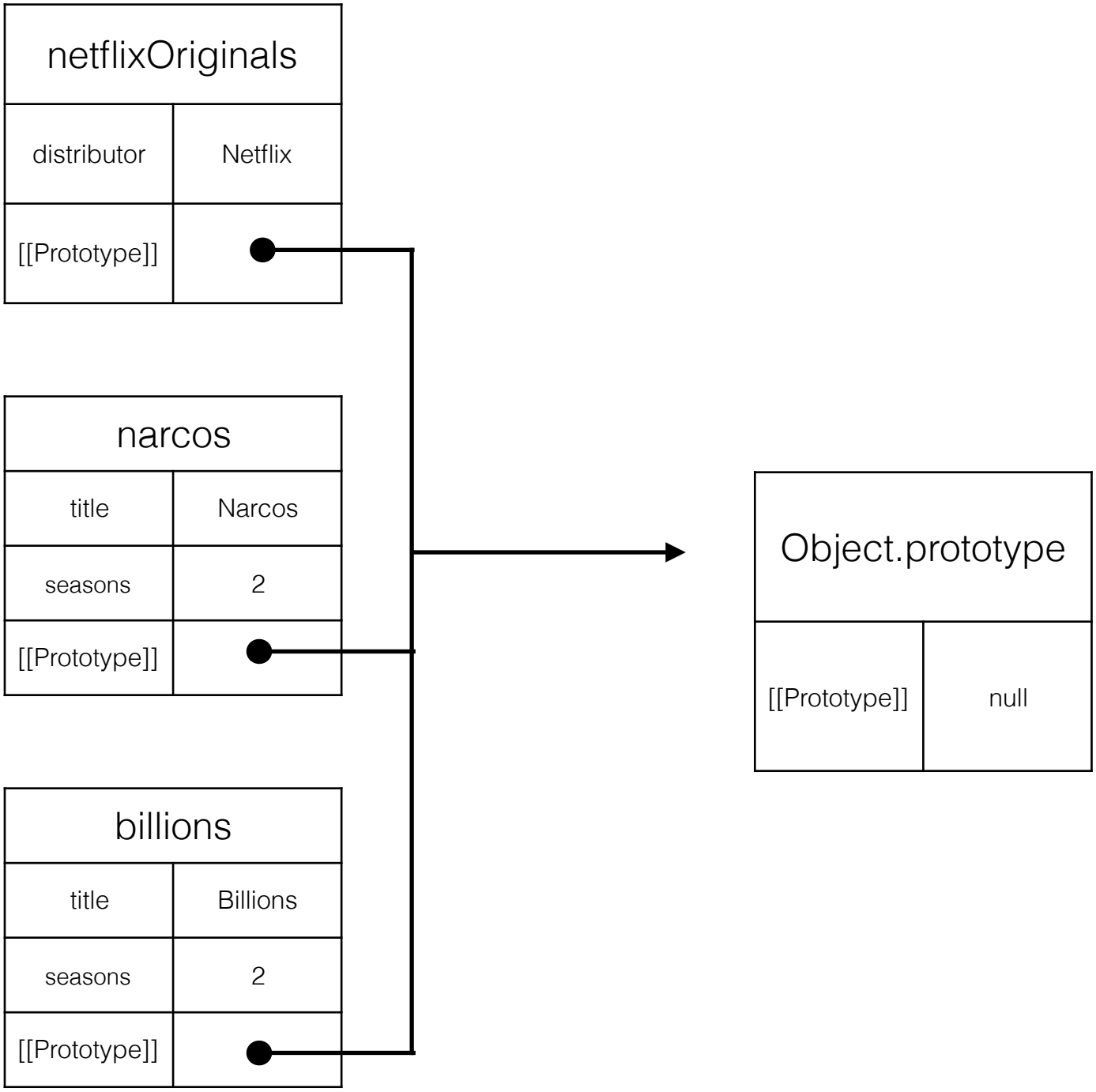
Herança

Um objeto é uma coleção dinâmica de chaves e valores, que podem ser de qualquer tipo, e um protótipo que pode ser um objeto ou null.

```
let narcos = {
  title: "Narcos",
  seasons: 2,
  distributor: "Netflix"
};

let billions = {
  title: "Billions",
  seasons: 2
  distributor: "Netflix"
};
```

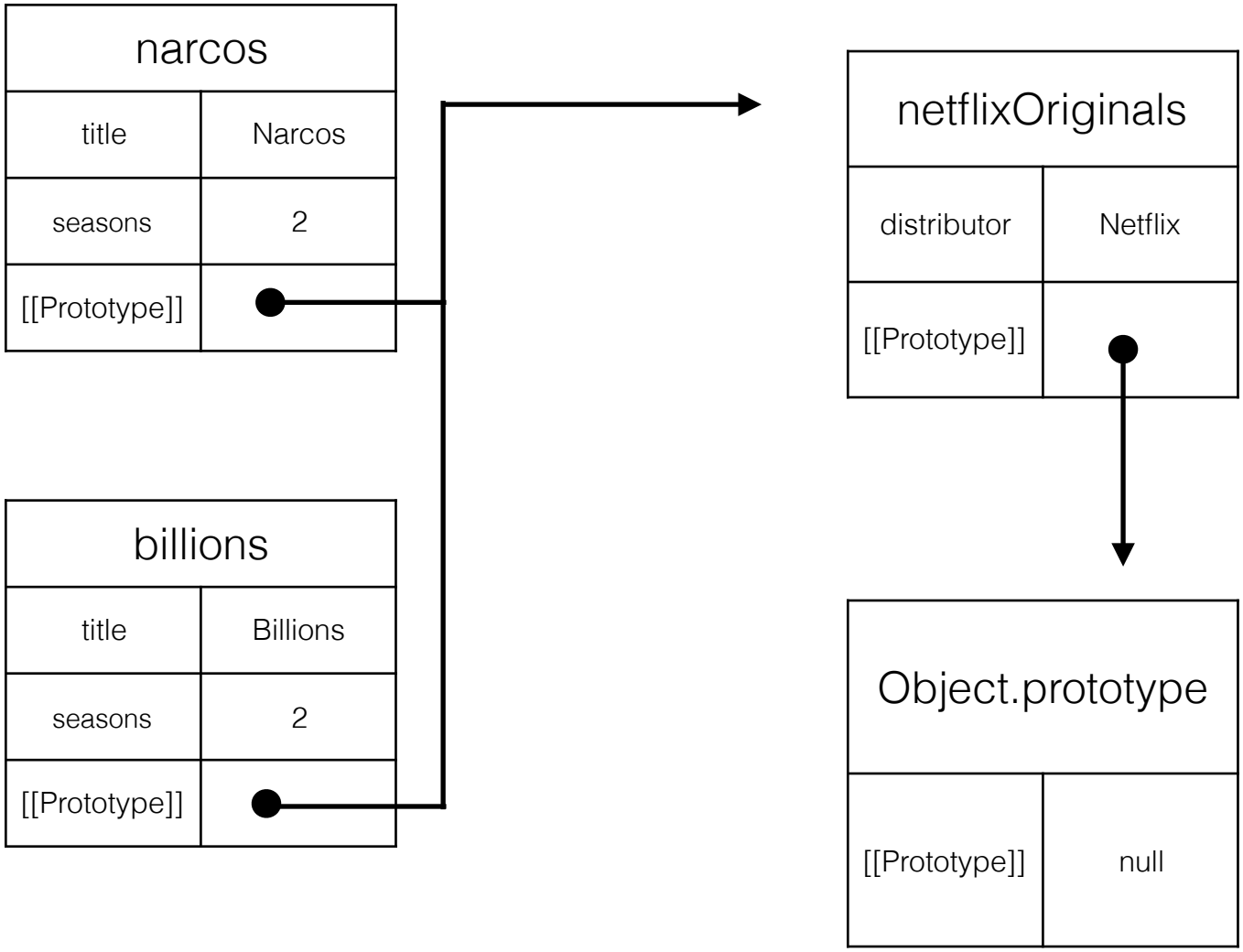

narcos
title
Narcos
seasons
2
distributor
Netflix
[[Prototype]]
billions
title
Billions
seasons
2
distributor
Netflix
[[Prototype]]
Object.prototype
[[Prototype]]
null

```
let netflixOriginals = {
  distributor: "Netflix"
};

let narcos = {
  title: "Narcos",
  seasons: 2,
};

let billions = {
  title: "Billions",
  seasons: 2
};
```

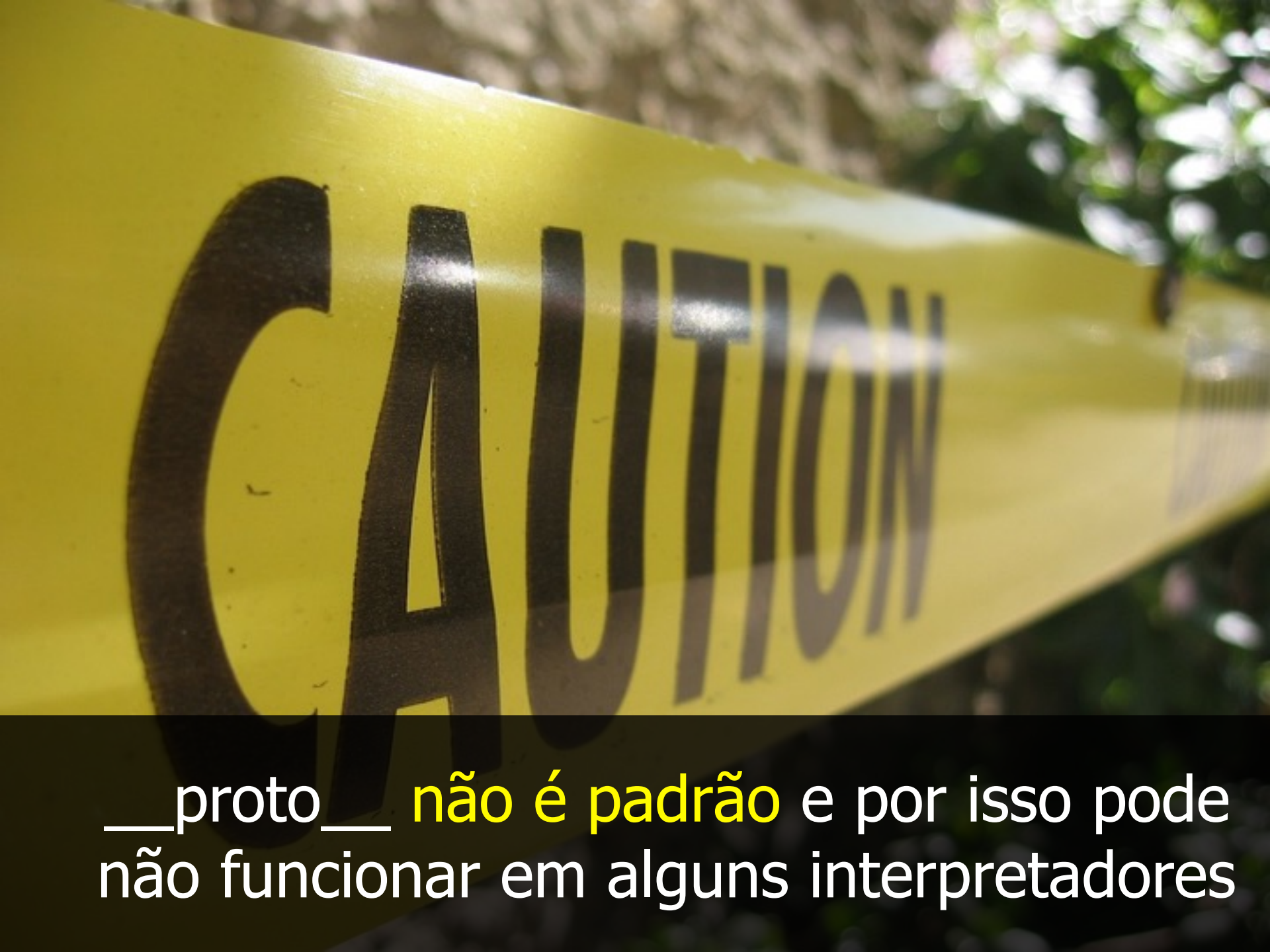

netflixOriginals
distributor
Netflix
[[Prototype]]
narcos
title
Narcos
seasons
2
[[Prototype]]
billions
title
Billions
seasons
2
[[Prototype]]
Object.prototype
[[Prototype]]
null

A propriedade __proto__ é uma referência para o protótipo do objeto.

```
let netflixOriginals = {
  distributor: "Netflix"
};

let narcos = {
  title: "Narcos",
  seasons: 2,
  __proto__: netflixOriginals
};

let billions = {
  title: "Billions",
  seasons: 2,
  __proto__: netflixOriginals
};
```


narcos
title
Narcos
seasons
2
[[Prototype]]
billions
title
Billions
seasons
2
[[Prototype]]
netflixOriginals
distributor
Netflix
[[Prototype]]
Object.prototype
[[Prototype]]
null


Shadowing



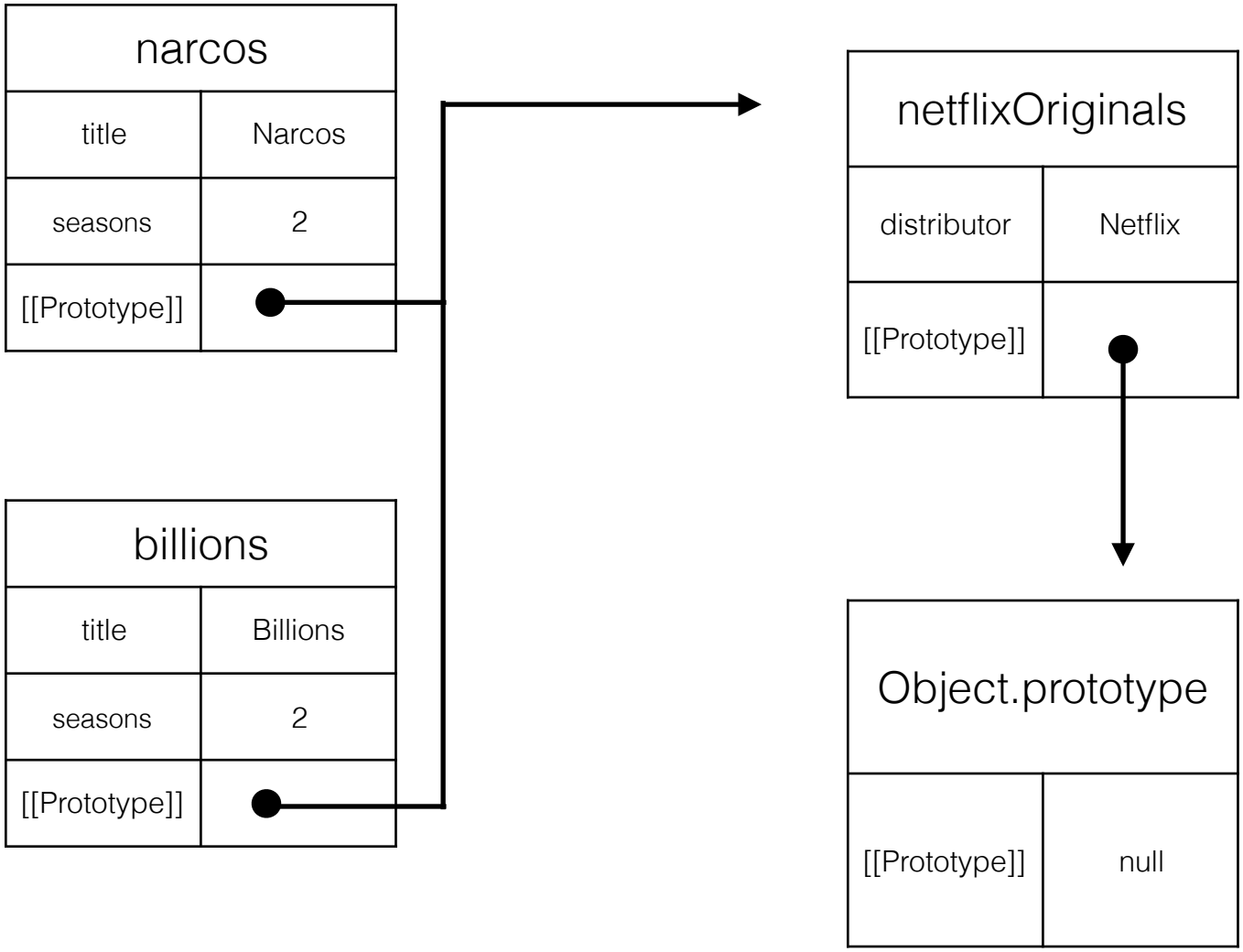
__proto__ não é padrão e por isso pode não funcionar em alguns interpretadores

Prefira a utilização de Object.getPrototypeOf e Object.setPrototypeOf para interagir com o protótipo do objeto.

```
let netflixOriginals = {
  distributor: "Netflix"
};

let narcos = {
  title: "Narcos",
  seasons: 2
};
Object.setPrototypeOf(narcos, netflixOriginals);

let billions = {
  title: "Billions",
  seasons: 2
};
Object.setPrototypeOf(billions, netflixOriginals);
```

Também é possível utilizar Object.create para determinar o protótipo do objeto.

```
let netflixOriginals = {
  distributor: "Netflix"
};

let narcos = Object.create(netflixOriginals);
Object.assign(narcos, {
  title: "Narcos",
  seasons: 2
});

let billions = Object.create(netflixOriginals);
Object.assign(billions, {
  title: "Billions",
  seasons: 2
});
```


narcos
title
Narcos
seasons
2
[[Prototype]]
billions
title
Billions
seasons
2
[[Prototype]]
netflixOriginals
distributor
Netflix
[[Prototype]]
Object.prototype
[[Prototype]]
null