

How new works?

Every time we call "new", it does the following 4 steps procedure:

- (i) It creates a brand new empty object.
- (ii) It calls the constructor of the class and passes the newly created object (not as a parameter) but inside a keyword called as 'this'.
 So constructor automatically has access to ~~the~~ the 'this' keyword and when we call 'new', then "the 'this' keyword has access to the plain object created in step 1 and constructor now can use the 'this' keyword inside it. And then whatever is logic of constructor it is executed.

Plain object



{ }

constructor () {

// "this"

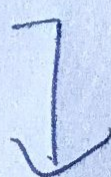
}

Note: → Every function has access to "this" keyword, not only "constructors".

- (iii) "new" triggers everything need to be done for prototypes to work.
- (iv) Now, if from a constructor an object is manually returned then this manually returned object is stored in the called variable, otherwise in any other case i.e., either we don't return anything or return something about from object, constructor doesn't care about it and returns the value inside ~~the~~ the 'this' keyword.

Note ⇒ objects like → object, array, function.

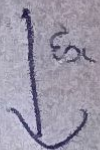
Flowchart
Step 4



Are we returning an object manually from constructor?

YES

Then this manual object only will be given back to the caller



```
constructor () {  
    return { x: 25 }  
}
```

No

Did we return anything else ~~from~~ object?

yes

```
constructor () {  
    return 10;  
}
```

NO, i.e., we did not return anything.

```
constructor {  
    // nothing returned  
}
```

Constructor will automatically return the value of 'this' keyword.

Code: ↴

```
class Product {
```

```
  name;  
  price;
```

```
  constructor(name, price, ...) {
```

```
    this.name = name;  
    this.price = price;
```

```
  }  
}
```

only if we do like this

```
  addToCart() {
```

```
  }  
}
```

```
let iphone = new Product('a', 2, ...);
```

to define data members

either use this way or this way.

otherwise it'll return empty object.

Steps ↴

(i)



new object

(ii)

this



PO

```
{  
  name: 'iphone 11'  
}
```

(iii)

(iv)

return this*