

Prototypes?

Prototypes are the mechanism by which JavaScript objects inherit features (properties & methods) from other objects.

```
const instructor = {  
  name: 'Max',  
  age: 31,  
  greet() {  
    console.log('Hi there!')  
  }  
};
```

person is the **prototype** of instructor.

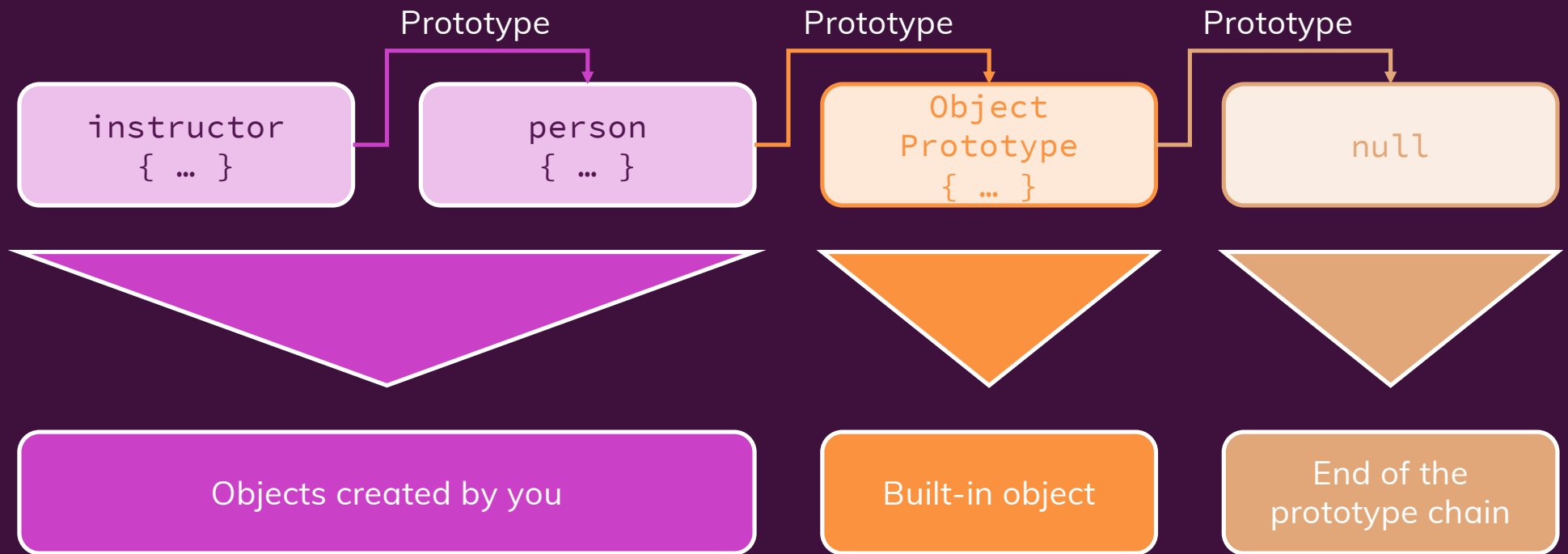
```
const person = {  
  setName(newName) {  
    this.name = newName;  
  }  
};
```

instructor has
no **setName()**
method!

```
instructor.setName('Maximilian');
```

instructor and person are
“linked” and JS automatically looks
up missing properties or methods.

The Prototype Chain



Object vs Object.prototype vs Object.__proto__

This applies to ALL objects/ constructor functions!

Object

The “Object” function (which itself is also a JS object technically) which can be called to create new objects.

Used for new object creation

Object.prototype

Points **at another object** (!) which will act as the prototype (`__proto__`) of objects **created via new Object()**.

`Object.create()` is a way of using a different prototype for new objects!

Object.__proto__

The **prototype of the Object function object** itself, **NOT** the prototype that will be used for newly created objects.

Summary – Prototypes

Prototypes are the mechanism by which **JavaScript objects inherit features** (properties & methods) from other objects.

Prototypes are **NOT blueprints** for objects (the name might imply that they are) but can be thought of as “**fallback objects**”.

New objects receive a **default prototype** but you can **change or replace** that prototype (both **during as well as after object creation**).

JavaScript uses prototypes **to look up missing functionalities** (properties or methods).

Lookup is done by **going up the prototype chain** (i.e. from an object to its prototype, to that prototype's prototype etc.)

The non-standard **__proto__** property can be used to **view (and replace) the prototype** of an object.

The **prototype** property of a function can be used to configure the **to-be-assigned prototype** for objects created by that function.