Object
null

Object.prototype.__proto__

Function
Object()

Object.prototype

Object
Object.protoype

Object.prototype.constructor

Object.__proto__

Both the Object and Function constructor function prototypes point to the Function.prototype Object

Function.prototype.__proto__

Function.__proto__

Function
Function()

Function.prototype

Object
Function.protoype

Function.prototype.constructor

Person.constructor

Object.getPrototypeOf(Person)
or
Person.__proto__

Person.prototype.__proto__
or
Object.getPrototypeOf(Person.prototype)

Constructor
Function
Person()

prototype: Person.prototype

Hidden [[Prototype]]

Person()'s Function Prototype
Person.protoype

Object
Person.prototype

constructor: Person()

Hidden [[Prototype]]

Person.protoype.constructor
'Objects' constructor property points to the Person() constructor function

The Person() constructor function `Object`'s object prototype

"The `Object`'s object prototype"

person1.constructor

let person = new Person()

person.__proto__

person.__proto__
or
Object.getPrototypeOf(person)

Object
person

Hidden [[Prototype]]

"The `person` `Object`'s object prototype"

Diagram key

◯ Function

▢ Object

Access hidden [[Prototype]] property with Object.getprototypeof(objectName)
or with .__proto__ (dunder proto)

The `person` object is "technically" an instantiation of the Person.prototype object, not the `Person()` constructor function itself. `new` calls the 'constructor' method, which is the `Person()` function, which executes the Person() constructor function with `this` context set to the new object being created.