

JavaScript Parte III

IIC2513 - Tecnologías y Aplicaciones Web

Sebastián Vicencio R. 2do Semestre 2020

JavaScript moderno

Rest parameters

Spread syntax
...myVariable

Podemos utilizarlo con arrays en tres situaciones

- 1. En parámetros de una función
- 2. Al llamar a una función
- 3. Crear nuevos arrays

Rest parameters - Parámetros función

```
Spread syntax
...myVariable
```

```
function getRestParams(firstArg, ...rest) {
  console.log(rest);
}
getRestParams(1, 'test', true, 20);
```

Rest parameters - Llamado funciones

```
Spread syntax
...myVariable
```

```
const numbers = [1, 4, 5, 2];
Math.max(...numbers);
```

Rest parameters - Nuevos arrays

```
Spread syntax
...myVariable
```

```
// Create new arrays
const clonedArray = [...numbers];
const newArray = [...numbers, 10, 13];
```

Destructuring

Spread syntax
...myVariable

Forma de manejar datos de un objeto o array

- 1. Asignar a variables
- 2. Crear nuevos objetos

Destructuring

```
Spread syntax
                      ...myVariable
const numbers = [1, 4, 5, 2];
const person = { name: 'Student', age: 22 };
const [firstNumber, secondNumber, ...restNumbers] = numbers;
const { name, age } = person;
// Create new object
const newObj = { ...person };
```

Y herencia en JavaScript

Preguntémonos lo siguiente

```
const newArray = [];
```

newArray.length newArray.map newArray.reduce

¿Cómo un array obtiene propiedades y métodos?

Recordemos ahora los constructores

```
function Person(name, age, city, isStudent = false) {
   this.name = name;
   this.age = age;
   this.city = city;
   this.isStudent = isStudent;
   this.talk = function() {
      console.log('My name is ' + this.name + '. I live in ' + this.city);
   }
}
```

Recordemos ahora los constructores

```
const person1 = new Person('John', 26, 'New York');
const person2 = new Person('Helen', 32, 'Berlin');
person1.talk === person2.talk;
```

¿Qué pasa con esta comparación?

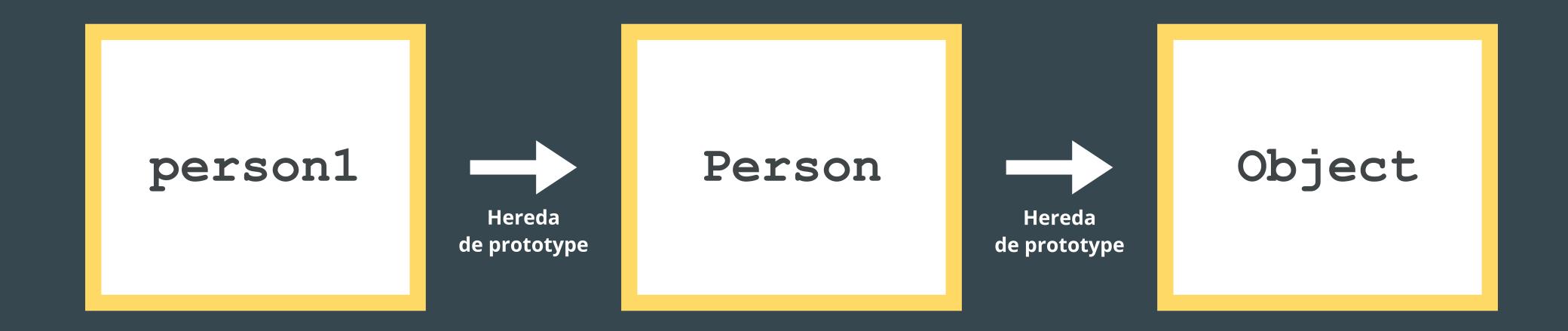
IIC2513 - Tecnologías y Aplicaciones Web - DCC UC

Prototypes Definición

JavaScript es un lenguaje basado en prototipos

Todo objeto tiene un objeto llamado prototype Hereda propiedades y métodos de este prototype

Prototype chain



El prototype también tiene su propio prototype

Eslabón final es Object, cuyo prototype es null

Prototypes Prototype chain

¿Cómo se accede al prototipo?

```
person1.toString();

person1.__proto__;

Object.getPrototypeOf(person1);
```

Prototypes Propiedad prototype

Propiedades y métodos heredables en objeto bajo key prototype

Person prototype

// And one of the base object
Object.prototype

Array prototype

Función constructora - Propiedad constructor

Toda función constructora tiene en su prototype la key constructor

person1.constructor

person1.constructor.name

Prototypes Modificación prototype

Agregamos propiedad a objeto prototype

```
Person.prototype.introduce = function() {
  console.log(`I am ${this.isStudent ? '' : 'not '}a student`)
}
person1.introduce();
```

Modificación prototype

En general se suelen definir propiedades en constructor y métodos en prototype

```
function OtherPerson(name) {
   this.name = name;
}
OtherPerson.prototype.talk = function() {
   console.log('My name is ' + this.name);
}
```

Referencias

- MDN Spread syntax (...)
- MDN Object prototypes
- Eloquent JavaScript The Secret Life of Objects