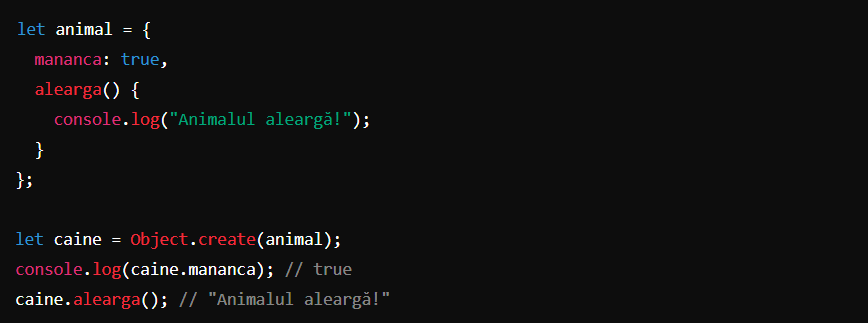
**Prototipuri și Moștenire în JavaScript**

**1.1 Conceptul de prototip în JavaScript**

În JavaScript, fiecare obiect are un prototip. Un prototip este un obiect din care obiectul curent moștenește proprietăți și metode. Astfel, când se încearcă accesarea unei proprietăți a unui obiect, motorul JavaScript va căuta mai întâi în obiectul însuși și, dacă nu găsește, va căuta în prototipul său.

Exemplu:



**1.2 Crearea de obiecte cu constructori**

Constructorii sunt funcții speciale utilizate pentru a crea obiecte. Aceștia definesc o structură de obiect, iar obiectele noi sunt instanțiate prin utilizarea operatorului new.

Exemplu:



**1.3 Moștenirea prototipală**

În JavaScript, moștenirea funcționează prin prototipuri. Orice obiect poate moșteni proprietăți și metode de la prototipul altui obiect.

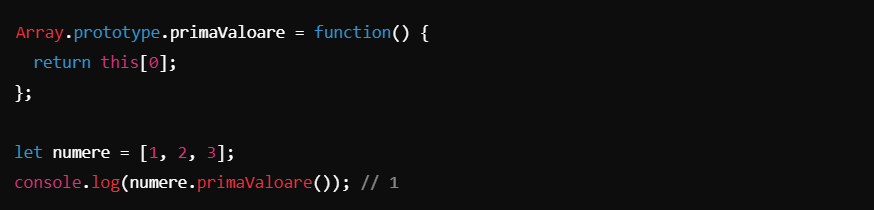
Exemplu:



**1.4 Extinderea prototipurilor**

Prototipurile pot fi extinse cu noi metode sau proprietăți, ceea ce permite modificarea tuturor obiectelor care moștenesc prototipul respectiv.

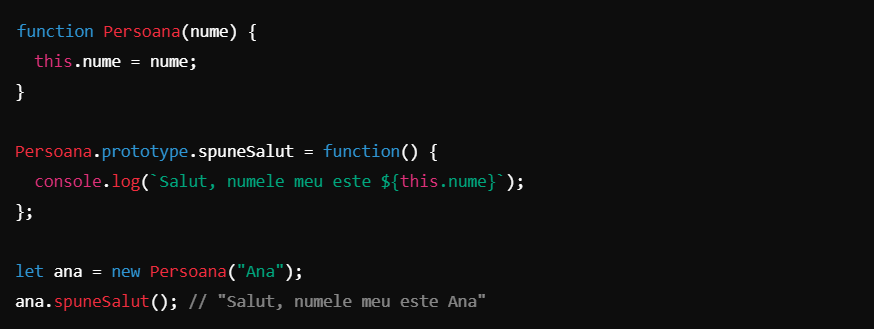
Exemplu:



**1.5 Obiectul "prototype"**

Orice funcție în JavaScript are proprietatea prototype, care este utilizată pentru a stoca metode ce pot fi moștenite de toate instanțele acelei funcții.

Exemplu:



**2. Clase în JavaScript**

**2.1 Introducere în clase**

Clasele sunt o abstracție a sistemului de prototipuri din JavaScript. Ele simplifică crearea de obiecte și moștenirea.

**2.2 Definirea unei clase**

Clasele sunt definite folosind cuvântul cheie class, iar constructorii sunt creați prin metoda constructor.

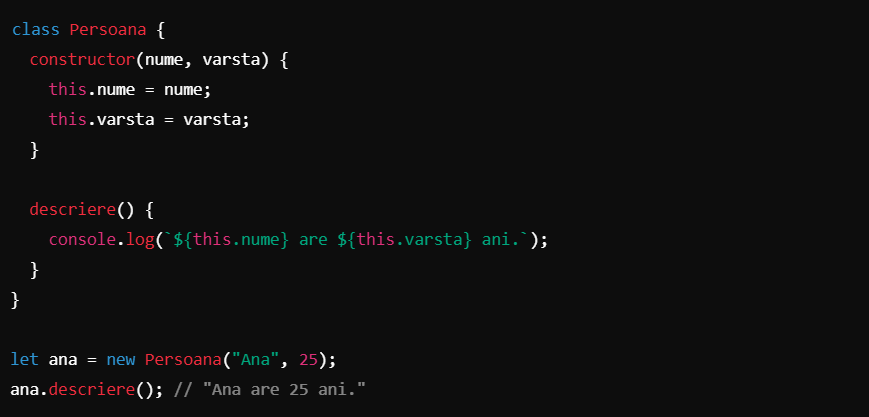
Exemplu:



**2.3 Constructori și metode în clase**

Constructorii sunt utilizați pentru a inițializa datele unei clase. Metodele pot fi definite fără cuvântul cheie function.

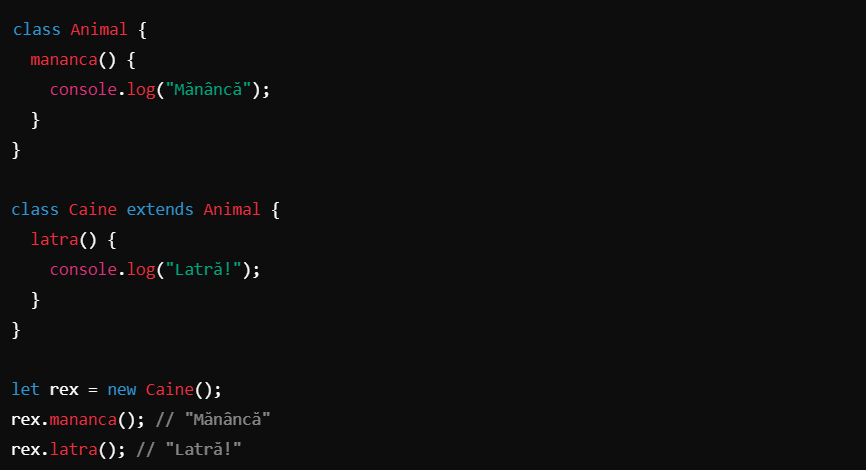
Exemplu:



**2.4 Moștenire cu clase**

Moștenirea se realizează cu ajutorul cuvântului extends.

Exemplu:



**2.5 Utilizarea "extends"**

extends permite unei clase să moștenească proprietăți și metode de la altă clasă.

Exemplu:



**2.6 Clase și prototipuri**

Chiar și cu clase, obiectele din spate folosesc prototipuri.

Exemplu:



**Exerciții practice**

1. **Exercițiul 1**: Creează o funcție constructor numită Car care să aibă proprietățile marca și model, și o metodă care să afișeze aceste proprietăți.
2. **Exercițiul 2**: Creează o clasă Student care să aibă proprietatea nume și metoda studiaza. Extinde această clasă pentru a crea o clasă Elev, care să adauge o metodă mergeLaScoala.
3. **Exercițiul 3**: Extinde prototipul obiectului Array cu o metodă care returnează suma tuturor elementelor numerice dintr-un array.
4. **Exercițiul 4**: Creează un lanț de moștenire utilizând atât funcții constructor, cât și clase pentru a defini o ierarhie de elemente de formular. Structura ar trebui să includă ierarhii precum (Form, Block, Input).