1 classical Inheritance
objects are created from classes
Relationships are hierarchical

Prototypal Inheritance
objects inherits directly from other object
There is no rigid class structure
It is dynamic allowing objects to inherit

2 object.create

```
const parent = { greet: function () {
                 console.log ("Hello"); } };

const child = object.create (parent);
    child.greet ();
```

2 Prototype Inheritance

```
function Animal (name) {
    this.name = name;
}
  Animal.prototype.makeSound = function () {
     console.log ("Some sound");
  };

const dog = new Animal ("Dog");
  dog.makeSound ();
```

3 ES6 class

```
class Animal {
    constructor (name) {
       this.name = name;
   }
    makeSound () {
      console.log ("some sound");
   }
}
 class Dog extends Animal {
    make Sound () {
     console.log ("Back");
     }
  }

   const dog = new Dog("Dog");
    dog.makesound ();
```

3 Advantages of prototypal

① Flexibility
② Runtime Extension
③ Simpler Object Composition
④ Memory Efficiency

4 Dis advantages

① complexity
② performance concern
③ Dynamic Nature Risk
④ Overhead of Simulating classes

5 challenges

- Accidental Overwriting
- Inconsistent Behaviour

use cases

① object composition
② performance optimization
③ Dynamic prototyping