

```
//constructor function
function Person() {
  this.name = "John",
  this.age = 20;
}

//create an object
const person = new Person();
console.log(person.name, person.age);

function Person(name, age) {
  this.name = name,
  this.age = age,
  this.greet = () => console.log(`Hello ${this.name}`);
}

//create objects
const person1 = new Person("John", 20);
const person2 = new Person("Jack", 24);

//access properties
console.log(person1.name, person1.age);
person1.greet();
console.log(person2.name, person2.age);
person2.greet();

// creating a class
class Person {
  constructor(name, age) {
    this.name = name;
    this.age = age;
  }
}
```

```
//creating an object
const person1 = new Person("John", 20);
const person2 = new Person("Jack", 24);

console.log(person1.name, person1.age);
console.log(person2.name, person2.age);

class Person {
  constructor(name, age) {
    this.name = name;
    this.age = age;
  }
  //defining method
  greet() {
    console.log(`Hello ${this.name}`);
  }
}

const person1 = new Person("John", 30);

//accessing property
console.log(`Name: ${person1.name}`);
console.log(`Age: ${person1.age}`);

//accessing method
person1.greet();

const person2 = new Person("Jack", 20);

//accessing property
console.log(`Name: ${person2.name}`);
console.log(`Age: ${person2.age}`);
```

```
//accessing method
```

```
person2.greet();
```

```
//parent class
```

```
class Person {
```

```
  constructor(name) {
```

```
    this.name = name;
```

```
  }
```

```
  greet() {
```

```
    console.log(`Hello ${this.name}!`);
```

```
  }
```

```
}
```

```
//inheriting parent class
```

```
class Student extends Person {
```

```
}
```

```
let student1 = new Student("Jack");
```

```
student1.greet();
```

```
//super() keyword
```

```
//parent class
```

```
class Person {
```

```
  constructor(name) {
```

```
    this.name = name;
```

```
  }
```

```
  greet() {
```

```
    console.log(`Hello ${this.name}!`);
```

```
  }
```

```
}
```

```
//child class
class Student extends Person {
  constructor(name) {
    console.log("Creating child class");
    //call super class constructor and pass in the name parameter
    super(name);
  }
}
let student1 = new Student("John");
student1.greet();
```

```
//Method overriding
```

```
//parent class
```

```
class Person {
  constructor(name) {
    this.name = name;
    this.occupation = "unemployed";
  }
  greet() {
    console.log(`Hello ${this.name}!`);
  }
}
```

```
//child class
```

```
class Student extends Person {
  constructor(name) {
    // call super class constructor and pass in the name parameter
    super(name);

    //overriding an occupation's property
    this.occupation = "student";
  }
}
```

```
//overriding person's method
greet() {
  console.log(`Hello student: ${this.name}!`);
  console.log(`Occupation: ${this.occupation}!`);
}
}
let student1 = new Student("Jack");
student1.greet();
```