```
class Animal {
    walk() {

    }
}
```

```javascript
"use strict";

var _createClass = function () { function defineProperties(target, props) { for (var i = 0; i
< props.length; i++) { var descriptor = props[i]; descriptor.enumerable =
descriptor.enumerable || false; descriptor.configurable = true; if ("value" in descriptor)
descriptor.writable = true; Object.defineProperty(target, descriptor.key, descriptor); } }
return function (Constructor, protoProps, staticProps) { if (protoProps)
defineProperties(Constructor.prototype, protoProps); if (staticProps)
defineProperties(Constructor, staticProps); return Constructor; }; }();

function _classCallCheck(instance, Constructor) { if (!(instance instanceof Constructor))
{ throw new TypeError("Cannot call a class as a function"); } }

var Animal = function () {
  function Animal() {
    _classCallCheck(this, Animal);
  }

  _createClass(Animal, [{
    key: "walk",
    value: function walk() {}
  }]);

  return Animal;
}();
```

```javascript
class Animal {
    walk() {

    }
}
```

```javascript
"use strict";

var _createClass = function () { function defineProperties(target, props) { for
< props.length; i++) { var descriptor = props[i]; descriptor.enumerable =
descriptor.enumerable || false; descriptor.configurable = true; if ("value" in
descriptor.writable = true; Object.defineProperty(target, descriptor.key, descr
return function (Constructor, protoProps, staticProps) { if (protoProps)
defineProperties(Constructor.prototype, protoProps); if (staticProps)
defineProperties(Constructor, staticProps); return Constructor; }; }();

function _classCallCheck(instance, Constructor) { if (!(instance instanceof Cor
{ throw new TypeError("Cannot call a class as a function"); } }

var Animal = function () {
  function Animal() {
    _classCallCheck(this, Animal);
  }

  _createClass(Animal, [{
    key: "walk",
    value: function walk() {}
  }]);

  return Animal;
}();
```

```
async function loadData() {
    const response = await this.http.load(
        '/some-data'
    );
}
```