

Model objet

Les objets Javascript ne supportant pas l'observation des changements de leurs propriétés, une application Ember utilisera des objets [Ember.Object](#) pour mettre en place le binding des propriétés en lieu et place des objets javascript standards.

De la même façon, Ember étend l'objet Array javascript par la classe [Ember.Enumerable](#).

Création de classes

La création de classe se fait en utilisant la méthode `extend` sur la classe `Ember.Object` :

```
import EmberObject from '@ember/object';

const Person = EmberObject.extend({
  say(thing) {
    alert(thing);
  }
});
```

Héritage/surdéfinition

```
import EmberObject from '@ember/object';

const Person = EmberObject.extend({
  say(thing) {
    alert(`${this.get('name')} says: ${thing}`);
  }
});

const Soldier = Person.extend({
  say(thing) {
    // this will call the method in the parent class (Person#say), appending
    // the string ', sir!' to the variable `thing` passed in
    this._super(`${thing}, sir!`);
  }
});

let yehuda = Soldier.create({
  name: 'Yehuda Katz'
});

yehuda.say('Yes'); // alerts "Yehuda Katz says: Yes, sir!"
```

Initialisation/construction

Quand un objet est instancié, la méthode **init()** de sa classe est invoquée automatiquement. La méthode `init` permet d'initialiser chaque instance :

Instanciation

```
let person = Person.create();  
person.say('Hello'); // alerts " says: Hello"
```

From:

<http://slamwiki2.kobject.net/> - **Broken SlamWiki 2.0**

Permanent link:

<http://slamwiki2.kobject.net/richclient/emberjs/modelobjet?rev=1516462887>

Last update: **2019/08/31 14:38**

