Decorators

Status Update


BABEL


TypeScript

Quick Recap

Intercept "defineProperty"

7.3.5 CreateMethodProperty (O, P, V)

The abstract operation CreateMethodProperty is used to create a new own property of an object. The operation is called with arguments *O*, *P*, and *V* where *O* is the object, *P* is the property key, and *V* is the value for the property. This abstract operation performs the following steps:

1. Assert: Type(*O*) is Object.
2. Assert: IsPropertyKey(*P*) is **true**.
3. Let *newDesc* be the PropertyDescriptor{[[Value]]: *V*, [[Writable]]: **true**, [[Enumerable]]: **false**, [[Configurable]]: **true**}.
4. Return *O*.[[DefineOwnProperty]](*P*, *newDesc*).

NOTE This abstract operation creates a property whose attributes are set to the same defaults used for built-in methods and methods defined using class declaration syntax. Normally, the property will not already exist. If it does exist and is not configurable or if *O* is not extensible, [[DefineOwnProperty]] will return **false**.

```
class Person {
  name() { }
}

// ->
Object.defineProperty(Person.prototype, 'name', {
  enumerable: false,
  configurable: true,
  writable: true,
  value: <closure>
});
```

```
// How do you express this declaratively:

Object.defineProperty(Person.prototype, 'name', {
  enumerable: false,
  configurable: true,
  writable: false,
  value: <closure>
});
```

```
class Person {
  @readonly name() { }
}

function readonly(prototype, name, desc) {
  desc.writable = false;
  return desc;
}

// ->
Object.defineProperty(Person.prototype, 'name', readonly(Person.prototype,
'name', {
  enumerable: false,
  configurable: true,
  writable: true,
  value: <closure>
}));
```

bit.ly/ember-js-classes

bit.ly/js-decorators-with-props

With Property Initializers

```
function concat(...args) {
  let sep = args.pop();

  return function(target, key, descriptor) {
    descriptor.initializer = function() {
      return args.map(arg => this[arg]).join(sep);
    }
  }
}

class Person {
  firstName = "Sebastian";
  lastName = "McKenzie";

  @concat('firstName', 'lastName', ' ') fullName;
  @concat('lastName', 'firstName', ', ') formalName;
}

console.log(new Person().fullName);
console.log(new Person().formalName);
```