

Aller plus loin

JS





LE PROTOTYPAGE

2



3

- Tout type est “wrappé” par un objet
- Tout objet contient une propriété **prototype**
- **prototype** contient l'ensemble des méthodes et propriétés accessibles de l'objet hérité
- Héritage

`var Object2 = Object.create(Object1);`

Déclaration d'un objet

- Double accolade (A)

```
var myObject = {  
  p1: "foo"  
};
```

- Constructeur (C)

```
function MyObject() {  
  this.p1 = "foo"  
}  
  
var myObject = new MyObject()
```

5

Publique

Propriétés

- (A)

```
{  
  p1: "foo"  
};
```

- (C)

```
function MyObject() {  
  this.p1 = "foo";  
}
```

Méthodes

- (A)

```
{  
  m1: function() {}  
};
```

- (C)

```
function MyObject() {  
  this.m1 = function() {};  
}
```

Prototype

```
myObject.prototype.p1 = "foo";  
myObject.prototype.m1 = function() {};
```


6

Privé

Propriétés

- (A)

Impossible

- (C)

```
function MyObject() {  
  var pr1 = "foo";  
}
```

Méthodes

- (A)

Impossible

- (C)

```
function MyObject() {  
  var mr1 =function() {};  
}
```




Exercice 3 - Prototypons

8

Reprendre l'exercice 1 - Manipulation des chaînes

- Rendre toutes les fonctions accessibles pour chaque String, excepté `prop_access`

Exemples:

```
ucfirst("ma chaine") => "ma chaine".ucfirst()
```

```
vig("ma chaine", "ma clé") => "ma chaine".vig("ma clé")
```

- Rendre `prop_access` accessible pour chaque Object

Exemple:

```
prop_access(object, "animal.type.name") => object.prop_access("animal.type.name")
```