

Chapitre 1 : Programmation par objets

- 2 parties :
 - ❶ Qu'est-ce que la programmation par objets
 - notion d'objet
 - notion de classe
 - ❷ Les objets en Javascript

Rappels (1/3) :

- Algorithme :
suite d'instructions à donner à un ordinateur pour résoudre un problème
- Un algorithme est composé de deux types d'éléments :
les **instructions**
organisées à l'aide de **structures de contrôle** et les **données**
- Programme :
algorithme traduit dans un langage donné

Rappels (2/3) :

- **Structure de contrôle :**
séquence, sélection, répétition
- **Données :**
stockées dans des variables
- **Variable :**
nom + type
- **Type :**
simple (nombre, caractère, booléen)
structuré : **tableau**, objet

Rappels (3/3) :

- Instructions :
 - saisie / affichage de données
 - affectation** d'une variable
 - appel d'un sous-programme**
- **Sous-programme :**
 - fonction (return) / procédure
 - définition :**
 - signature = nom + **paramètres**
 - corps = {...}
 - utilisation :**
 - appel

Questions ???

Un peu de code Javascript...

`window`

comment appelle-t-on cet élément ?

`window.innerHeight`

`console`

`console.log("blabla")`

comment appelle-t-on un élément comme `log()` ?

→ **objets** de base de Javascript

D'après vos connaissances en programmation

`window` : identificateur de variable

`log()` : sous-programme (fonction ou procédure)

`innerHeight`

attribut

méthode applicable sur un objet

référence à un objet

Les objets du DOM

```
<html>
```

```
  <head>
```

```
    <title>cours : prog objet</title>
```

```
  </head>
```

```
  <body>
```

```
    <p> bonjour </p>
```

```
    <p id="dateAff">    </p>
```

```
</body>
```

```
</html>
```

→ chaque élément du DOM est un objet de type **Node**



Définition d'un objet

Un objet est une entité concrète ou abstraite. Un objet a un état, un comportement et une identité.

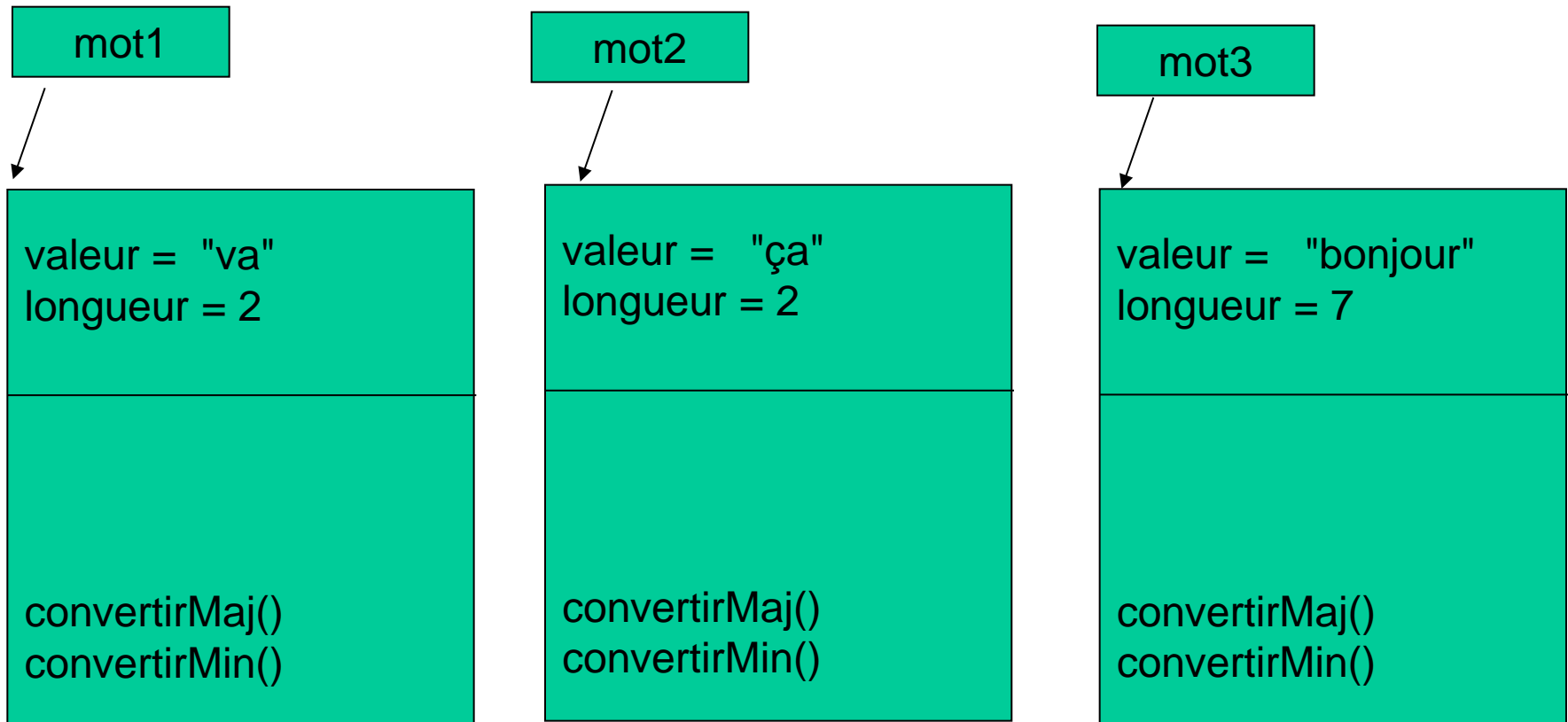
L'état d'un objet englobe les propriétés de l'objet et la valeur courante de chacune de ces propriétés.

Le comportement est la façon dont un objet agit et réagit, en termes de changement d'état et de transmission de messages.

Qu'est-ce qu'un objet ?

- entité + ou – complexe
 - un mot
 - une page
 - un livre
- décrite par des propriétés et des comportements
 - propriétés d'un mot : nombre de caractères, valeur...
 - comportements d'un mot : passer en majuscule, ...

Représentation des objets



l'ensemble des objets « mot »

Plusieurs mots « construits » selon le même modèle

→ notion de classe



Définition d'une classe

Une classe est un ensemble d'objets qui partagent une structure commune (attributs) et un comportement commun (méthodes).

Les termes instance et objet sont similaires.



- classe = commence par une majuscule
- objet = commence par une minuscule
- attribut = commence par une minuscule
- méthode = commence par une minuscule (sauf la méthode de type constructeur) – a **toujours** des parenthèses.
- pas de caractères *espace* dans un nom

Si correct, indiquez ce que peut être :

Nom	classe
nom	attribut objet
nom-prénom	
nom ()	méthode
NOM	constante

- toutes les entités sont des objets :
 - objet prédéfini (1) : console, window, document...
 - objet défini par le programmeur
 - À partir d'une classe prédéfinie (2)
 - Avec un initialiseur d'objet (3)
 - En définissant une classe (4)

objet prédéfini (1) : utilisation

```
objet.methode(...)
```

Exemple :

```
console.log("essai");
```

Objet défini par le programmeur à partir d'une classe prédéfinie (2)

- Utiliser une classe prédéfinie : **Date**, **String**...
- Deux actions

1 - Déclarer l'identificateur d'objet

2 – Construire l'objet avec l'opérateur new et le constructeur de la classe



Constructeur d'objet

- Un constructeur est une méthode particulière
- Il a 0 ou plusieurs paramètres formels
- Il porte toujours le nom de la classe d'objets qu'il construit
- Son rôle est de créer un objet et d'initialiser son état (valeurs aux attributs de l'objet)

Application

- Définir un objet de la classe **Array**
- Définir un objet de la classe **String** avec la valeur "bonjour MMI "
- Définir un objet de la classe **Date**

```
let tab= new Array();
```

```
let tab=[];
```

```
let mot=
```

```
new String("bonjour MMI" );
```

```
let mot="bonjour MMI" ;
```

```
let today= new Date();
```

Objet défini par le programmeur avec un initialiseur d'objet (3)

- Initialisateur d'objet

```
nomObjet = { }
```

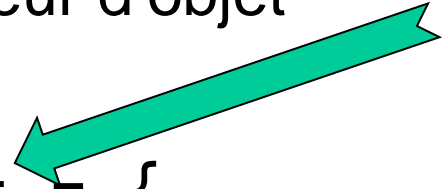
Exemple :

```
let unMot = { valeur:"bonjour",  
              longueur:7  
            };
```

Objet défini par le programmeur avec un initialiseur d'objet (3)

- Initialisateur d'objet

objet littéral



```
let unMot = {  
    valeur: "bonjour",  
    longueur: 7,  
    convertirMin: function()  
    {  
        let min;  
        min=this.valeur.toLowerCase();  
        return min;  
    }  
}; // fin objet
```

Objet défini par le programmeur en définissant une classe (4)

- Définir un modèle d'objet avec une classe
- Modèle de quel objet ? un mot
- Quels attributs ?
chaîne de caractères
longueur
type
- Quelles méthodes ?
mettre en minuscule
mettre la première lettre
en majuscule

Objet défini par le programmeur en définissant une classe (4)

- Coder le constructeur

mettre une valeur aux 3
attributs

- Coder les autres méthodes

mettre en minuscule
mettre la première lettre
en majuscule

codage du constructeur d'objet de type Mot (1/2)

Nom de l'attribut	rôle	Peut être trouvé par le programmeur	Doit être donné
valeur	chaîne représentant le mot		
longueur	longueur de la chaîne		
type	type du mot (nom, adverbe, ...)		

codage du constructeur d'objets de type Mot (1/2)

Nom de l'attribut	rôle	Peut être trouvé par le programmeur	Doit être donné
valeur	chaîne représentant le mot		x
longueur	longueur de la chaîne		
type	type du mot (nom, adverbe, ...)		

codage du constructeur d'objets de type Mot (1/2)

Nom de l'attribut	rôle	Peut être trouvé par le programmeur	Doit être donné
valeur	chaîne représentant le mot		x
longueur	longueur de la chaîne	x	
type	type du mot (nom, adverbe, ...)		

codage du constructeur d'objets de type Mot (1/2)

Nom de l'attribut	rôle	Peut être trouvé par le programmeur	Doit être donné
valeur	chaîne représentant le mot		x
longueur	longueur de la chaîne	x	
type	type du mot (nom, adverbe, ...)		x

codage du constructeur d'objets de type Mot (2/2)

```
function Mot(valeur, type) {  
  
    this.valeur=valeur           ;  
    this.longueur= valeur.length ;  
    this.type=type               ;  
  
}
```

this ?

→ this représente l'**objet courant**

- *Dans un constructeur = l'objet que l'on est en train de créer*
`this.valeur; // attribut valeur de l'objet en cours de construction`
- *Dans une autre méthode que le constructeur = l'objet sur lequel on applique la méthode*
`console.log(...)` → DANS la méthode `log()`, `this` représente l'objet `console`

Définir soi-même un objet de type « Mot »

- Utiliser la classe définie **Mot**
- Deux actions

1 - Déclarer l'identificateur d'objet

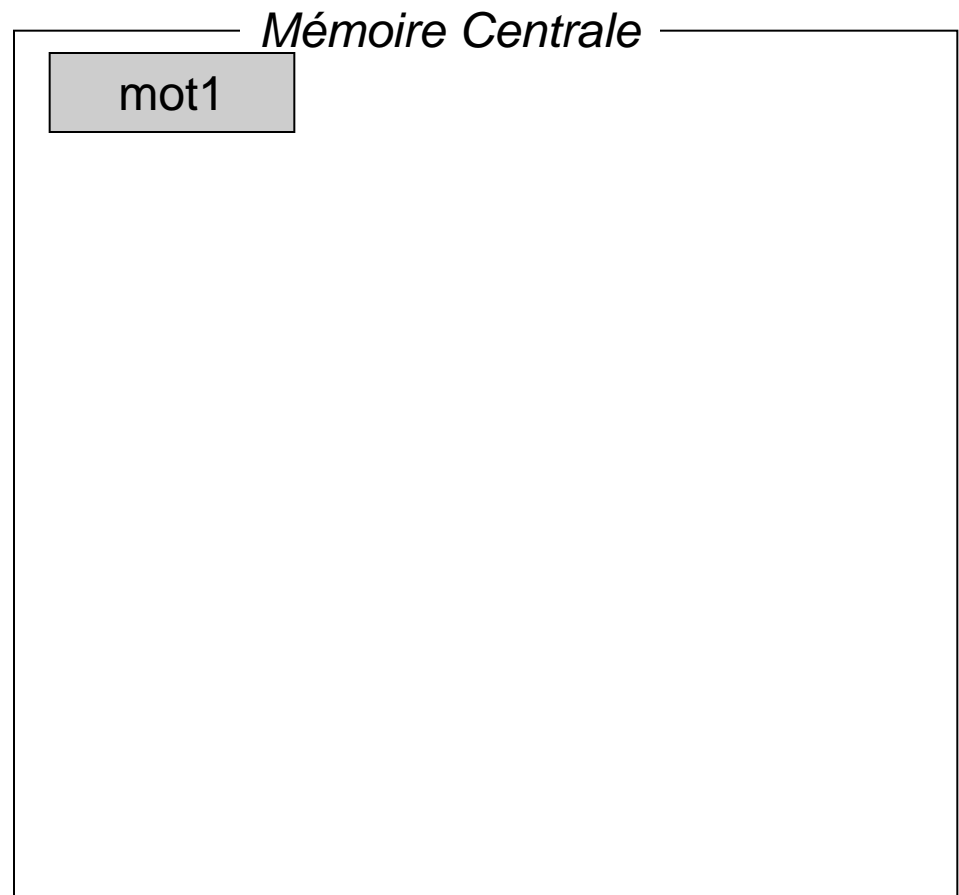
```
let mot1;
```

2 – Construire l'objet avec l'opérateur new et le constructeur de la classe

```
mot1 = new Mot("SALUT", "nom commun");
```

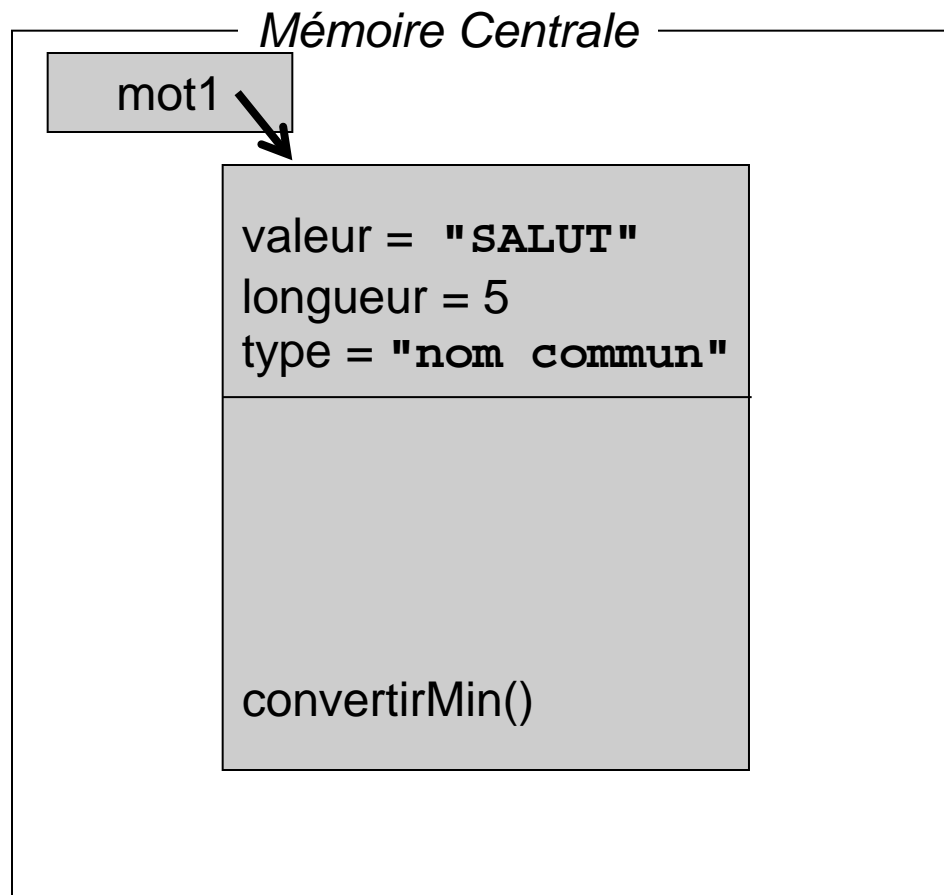

Que se passe-t-il ?

let mot1 ;



Que se passe-t-il ?

```
let mot1 ;  
mot1 = new Mot("SALUT",  
               "nom commun");
```





Objet javascript

- Lorsqu'on définit un objet, il possède automatiquement un attribut nommé prototype (qui est lui-même un objet).
- **prototype** contient les propriétés partagées par tous les objets de la même famille

codage des autres méthodes de la classe Mot

1. Définir la méthode
signature + corps
2. L'ajouter à la classe

classe Mot : convertir en minuscule

// role : mettre un mot en minuscule

// parametres : ?

// retour : ?

☞ On a besoin de connaître la chaîne de caractères
→ c'est l'attribut valeur

classe Mot : convertir en minuscule

```
// role : mettre le mot en minuscule  
// parametres : -  
// retour : -
```

```
function convertirMin()  
{  
  
}
```

classe Mot : convertir en minuscule

```
// role : mettre le mot en minuscule
```

```
// parametres : -
```

```
// retour : -
```

```
function convertirMin()  
{  
this.valeur=this.valeur.toLowerCase();  
  
}
```

Ajouter une méthode à la classe

```
Mot.prototype.convertirMin = convertirMin;
```

→ Ajoute la méthode `convertirMin()` à la classe `Mot`

Comment utiliser une méthode ?

Une fois l'objet créé :

→ appeler une méthode de la classe

→ en l'appliquant à l'objet (notation pointée)

```
mot1 . convertirMin( ) ;
```

- une méthode s'applique toujours à un objet

`mot1.convertirMin ();`

→ OK « mettre mot1 en minuscule »

`mot2.convertirMin ();`

→ !!! mot2 introuvable (n'a pas été créé)

`convertirMin ();`

→ ??? quel objet

Du code javascript...

```
let date = new Date();
```

```
let heure =
```

```
    date.getHours()+":"+date.getSeconds();
```

```
window.document.getElementById('dateAff').  
innerHTML+"<I>"+heure+"</I>";
```

Du code javascript...

```
function Mot(valeur, type) {  
    this.valeur=valeur          ;  
    this.longueur= valeur.length ;  
    this.type=type              ;  
}  
  
Mot.prototype.convertirMin = function()  
{  
    this.valeur=this.valeur.toLowerCase();  
}
```

```
let  mot1;  
mot1 = new Mot("SALUT", "nom commun");
```

Classe Mot

Du code PHP...

```
<?php
class Mot
{
    private $_valeur;
    private $_longueur;
    private $_type;

    public function __construct($valeur, $type) {
        $this->_valeur=$valeur;
        $this->_type=$type;
        $this->_longueur=strlen($valeur);
    }
}
?>
```