



UNIVERSITÉ
TOULOUSE III
PAUL SABATIER

**KEEP
CALM
AND
<CODE/>**

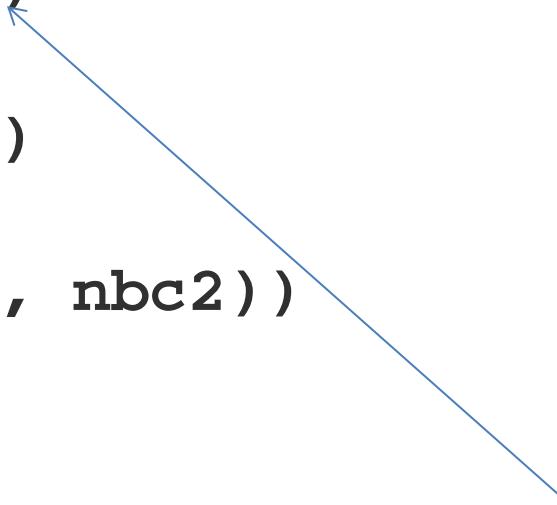
Programmation orientée objet

Classes et objets

Sylvie TROUILHET - www.irit.fr/~Sylvie.Trouilhet

Reprenons l'exercice précédent

```
const nbc1="1+i2"  
const nbc2="2+i1"  
let nbc1_im=im(nbc1)  
let nbc1_reel=re(nbc1)  
let nbc2_im=im(nbc2)  
let nbc2_reel=re(nbc2)  
  
console.log(addC(nbc1, nbc2))
```



Lisibilité ?
Abstraction ?

Notion d'objet

Regrouper les informations d'un objet :

- ses propriétés : valeur de la partie réelle, ...
- son comportement.

→ encapsulation



UNIVERSITÉ
TOULOUSE III
PAUL SABATIER

**KEEP
CALM
AND
<CODE/>**

Programmation orientée objet

Objet

Sylvie TROUILHET - www.irit.fr/~Sylvie.Trouilhet

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 nombre complexe

une expression mathématique

...

décrite par des propriétés → attributs

partie réelle

partie imaginaire

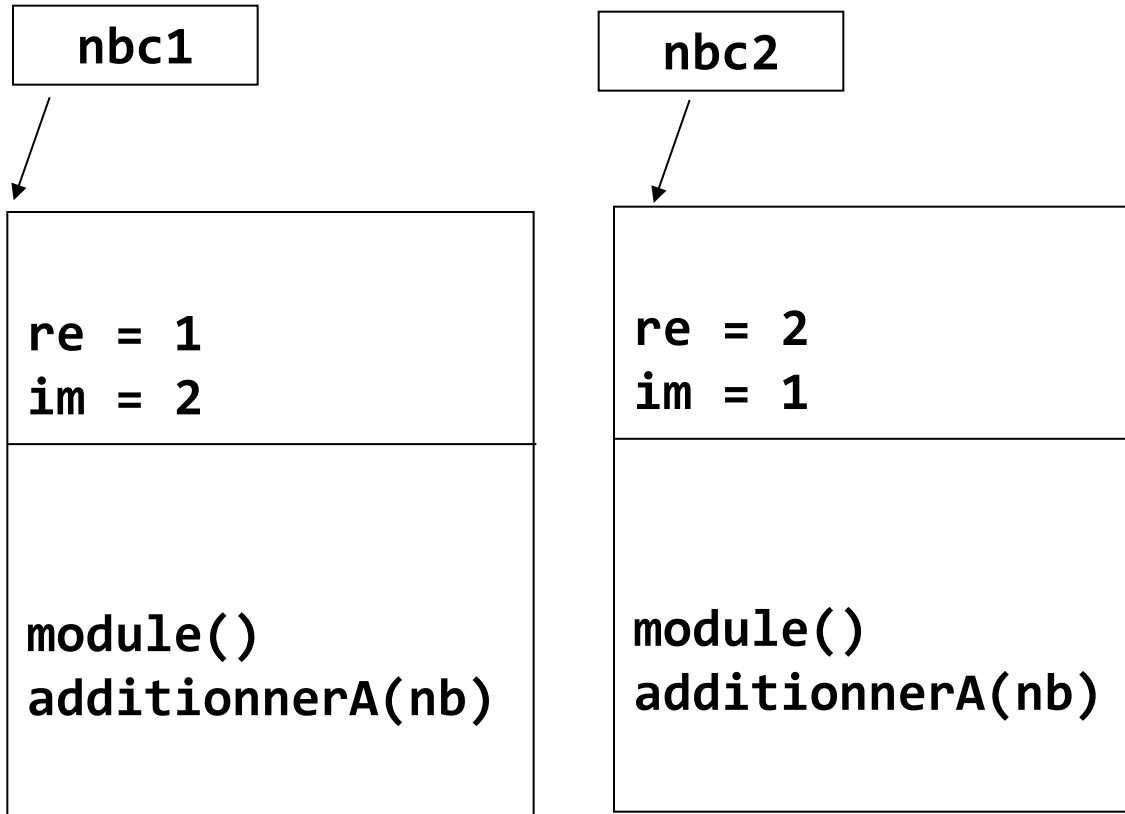
et des comportements → méthodes

connaître son module

additionner à un autre nombre

...

Représentation des objets



Création d'un objet

Initialisateur d'objet

```
nomObjet = { }
```

Exemple :

```
let nbC1 = {  
    re: 1,  
    im: 2  
}
```

```
console.log(nbC1.im)
```

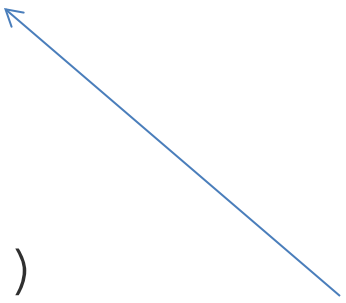

Extension dynamique d'un objet

Ajouter un attribut ou une méthode

```
nomObjet.attribut = ...
```

Exemple :

```
nbC1.module: function () {  
    return Math.sqrt(this.re * this.re  
+ this.im * this.im)  
}
```



```
console.log(nbC1.module())
```

?

→ this représente l'objet courant

`this.attribut`

`// attribut de l'objet
en cours de construction`

Objets prédéfinis

Dans une page web ?

```
let date = new Date()  
document.getElementById( 'dateAff' ).innerHTML  
    ="<I>" + heure + "</I>"
```

Quel est l'objet courant dans un page web ?

```
console.log( this )
```

```
>> console.log(this)
```

```
▶ Window https://intranet.irit.fr  
/SOGGo/so/trouilhe/Mail/view#!/Mail/0  
/INBOX
```

[debugger eval code:1:9](#)

☹ : Comment faire si l'on veut 50 nombres complexes ?



**KEEP
CALM
AND
<CODE/>**

Programmation orientée objet

Classe

Sylvie TROUILHET - www.irit.fr/~Sylvie.Trouilhet

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.

Une classe possède un constructeur pour créer les objets



Constructeur d'objet

Un constructeur

- est une méthode particulière
- qui a 0 ou plusieurs paramètres formels

Son rôle est de créer un objet et d'initialiser son état (c'est-à-dire donner des valeurs aux attributs de l'objet)

En javascript

```
class NombreImaginaire {  
  constructor(im, re) {  
    this.im=im  
    this.re= re  
  }  
}
```


Méthodes autres que le constructeur

On peut ajouter à la classe des méthodes

Pour chaque méthode, définir :
 les paramètres nécessaires
 s'il y a un return ou non

→ signature de la méthode

```
nomMethode( param1, ...)
```

En javascript

```
class NombreImaginaire {  
  constructor (im, re) {  
    this.im = im  
    this.re = re  
  }  
  
  argument () {  
    let res  
    if (this.re == 0.0)  
      if (this.im >= 0) res = Math.PI / 2  
      else res = -Math.PI / 2  
    else res = Math.atan(this.im / this.re)  
    return res  
  }  
}  
  
module.exports.NombreImaginaire = NombreImaginaire
```

Conventions d'écriture

Respecter les conventions d'écriture de la communauté des programmeurs en Javascript :

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

Comment utiliser une classe ?

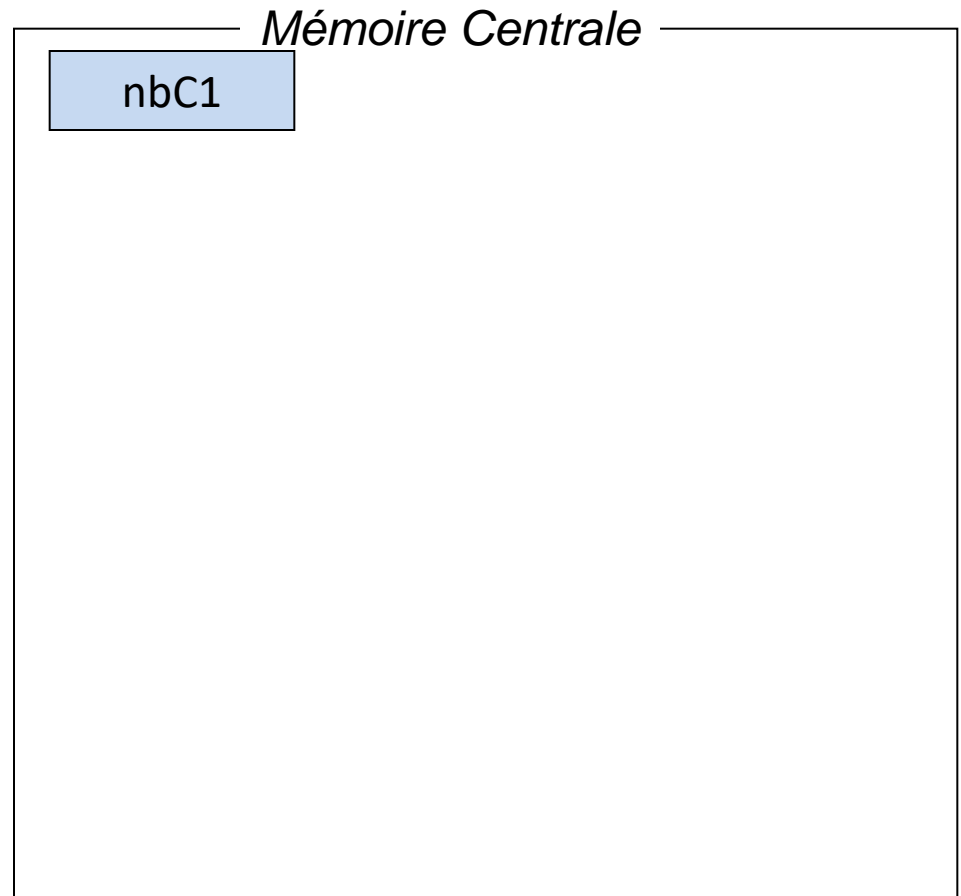
1 - Déclarer l'identificateur d'objet

```
const nbC1
```

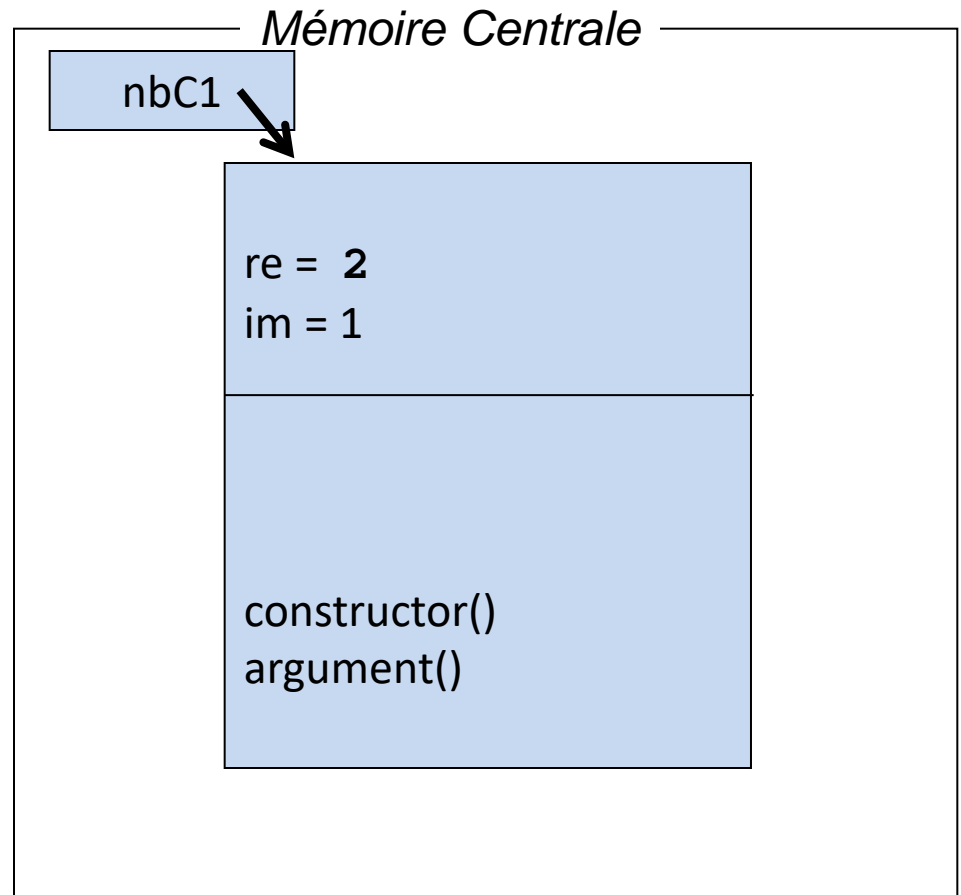
2 – Construire l'objet avec l'opérateur new et le constructeur de la classe

```
nbC1 = new NombreImaginaire(1,2)
```

const nbC1



```
const nbC1 ;  
nbC1 = new Nombrelmaginaire(2,1)
```



Reprenons l'exercice précédent

```
const nbc1="1+i2"  
const nbc2="2+i1"  
let nbc1_im=im(nbc1)  
let nbc1_reel=re(nbc1)  
let nbc2_im=im(nbc2)  
let nbc2_reel=re(nbc2)  
console.log(addC(nbc1, nbc2))
```

```
const nbC1=new NombreImaginaire(1,2)  
const nbC2=new NombreImaginaire(2,1)  
console.log(nbC1.im)  
console.log(nbC1.argument())
```


une méthode s'applique toujours à un objet

nbC1.argument()
→ OK

nbC3.argument()
→ nbC3 introuvable (n'a pas été créé)

argument()
→ c'est équivalent à écrire **this.argument()**

Exemple en PHP

```
<?php
class NombreComplexe
{
    private $_re;
    private $_im;

    public function __construct($re, $im) {
        $this->_re=$re;
        $this->_im=$im;    }
}
?>
```

En réalité Javascript

n'est pas un langage à base de
classe

Mais un langage à base de **prototype**

(voir suite du cours pour ceux que cela intéresse)



UNIVERSITÉ
TOULOUSE III
PAUL SABATIER

**KEEP
CALM
AND
<CODE/>**

Programmation orientée objet

Prototype

Sylvie TROUILHET - www.irit.fr/~Sylvie.Trouilhet

La méthode « constructeur »

La définition « class ... { } » est un sucre syntaxique.

Pour définir une classe d'objet, on définit une fonction ayant le nom de l'objet que l'on veut modéliser :

```
function NombreImaginaire(re, im) {  
  
    this.re=re                ;  
    this.im=im                ;  
  
}
```

Objet javascript

```
const nbC1=new NombreImaginaire(1,2);
```

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.

Ajout de méthode à la classe ?

1. Définir la méthode

```
function argument () {  
    let res  
    if (this.re == 0.0)  
        if (this.im >= 0) res = Math.PI / 2  
        else res = -Math.PI / 2  
    else res = Math.atan(this.im / this.re)  
    return res  
}
```

2. L'ajouter à la classe

```
NombreImaginaire.prototype.argument = argument;
```

→ *Ajoute la méthode argument() à la classe NombreImaginaire*

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

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