



Quelle est la signification de cette phrase : tout est OBJET dans le langage PHARO ?

Dans Pharo, l'expression « tout est objet » signifie que presque tout ce que le langage manipule est représenté comme un objet auquel on peut envoyer des messages. C'est l'un des principes fondamentaux de Pharo, avec l'idée que les objets communiquent par envoi de messages. ^[1]

Que signifie « objet » ?

Un objet regroupe généralement :

- un **état**, stocké dans des variables d'instance ;
- un **comportement**, défini par des méthodes ;
- une **classe**, qui décrit sa structure et son comportement.

Par exemple, le nombre 3 est un objet. On peut lui envoyer des messages :

```
3 + 4  
3 factorial  
3 even
```

Ici :

- `+` est un message envoyé à 3 ;
- `factorial` demande à 3 de calculer sa factorielle ;
- `even` demande si 3 est pair.

Le résultat est lui-même un objet.

Quels éléments sont des objets ?

Dans Pharo, sont notamment des objets :

- les nombres : 3, 42, 2.5 ;
- les chaînes de caractères : 'Bonjour' ;
- les booléens : `true` et `false` ;
- les collections : tableaux, listes, dictionnaires ;
- les fenêtres, boutons et couleurs ;
- les fichiers, sockets et flux ;

- les méthodes et les blocs de code ;
- les classes elles-mêmes.

Même une classe est donc un objet : on peut lui envoyer des messages, l'inspecter ou la modifier pendant l'exécution.^[2]

Conséquence : pas de types primitifs séparés

Dans de nombreux langages, les nombres ou les booléens sont des « types primitifs » distincts des objets. Pharo évite cette distinction : les nombres et les booléens se comportent comme les autres objets. La documentation officielle décrit ainsi Pharo comme un langage où il n'existe pas de types primitifs séparés.^[3]

Par exemple :

```
5 between: 1 and: 10
```

Cette expression demande à l'objet 5 s'il se trouve entre 1 et 10.

« Tout se fait par messages »

Le principe complet est souvent résumé ainsi :

■ Tout est objet, et tout se fait par envoi de messages.

Au lieu d'appeler une fonction indépendante, on demande à un objet d'effectuer une opération :

```
'bonjour' size
```

Le message `size` est envoyé à la chaîne `'bonjour'`.

Autre exemple :

```
Array new: 3
```

Le message `new:` est envoyé à l'objet `Array`, qui représente une classe, afin de créer un tableau de trois éléments.

En résumé, cette phrase signifie que Pharo propose un modèle très uniforme : les données, les classes et une grande partie des éléments du système sont des objets qui collaborent en échangeant des messages.

✱

1. <https://books.pharo.org/updated-pharo-by-example/pdf/2018-09-29-UpdatedPharoByExample.pdf>

2. <https://files.pharo.org/media/flyer-cheat-sheet.pdf>

3. <https://pharo.org/>

4. <http://www.youtube.com/watch?v=xhPIUaXpCU4>
5. <https://files.pharo.org/media/pharoCheatSheet.pdf>
6. <https://pharo.org/documentation>
7. <https://pharo.org/about>
8. <https://dokumen.pub/pharo-9-by-example-9782322394104.html>
9. <https://pharo.org/news/2024-04-26-pharo12-released.html>
10. [https://eng.libretexts.org/Bookshelves/Computer_Science/Programming_Languages/Book%3A_Pharo_by_Example_5.0_\(Ducasse_Zagidulin_Hess_and_Chlopupis\)](https://eng.libretexts.org/Bookshelves/Computer_Science/Programming_Languages/Book%3A_Pharo_by_Example_5.0_(Ducasse_Zagidulin_Hess_and_Chlopupis))
11. [https://eng.libretexts.org/Bookshelves/Computer_Science/Programming_Languages/Book:Pharo_by_Example_5.0\(Ducasse_Zagidulin_Hess_and_Chlopupis\)](https://eng.libretexts.org/Bookshelves/Computer_Science/Programming_Languages/Book:Pharo_by_Example_5.0(Ducasse_Zagidulin_Hess_and_Chlopupis))
12. [https://eng.libretexts.org/Bookshelves/Computer_Science/Programming_Languages/Book:Pharo_by_Example_5.0\(Ducasse_Zagidulin_Hess_and_Chlopupis\)/04:_Syntax_in_a_Nutshell/4.07:_Primitives_and_Pragmas](https://eng.libretexts.org/Bookshelves/Computer_Science/Programming_Languages/Book:Pharo_by_Example_5.0(Ducasse_Zagidulin_Hess_and_Chlopupis)/04:_Syntax_in_a_Nutshell/4.07:_Primitives_and_Pragmas)
13. <https://jiaxianhua.github.io/smalltalk/2015/04/26/pharo-by-example-5>
14. <https://rmod-pharo-mooc.lille.inria.fr/OOPMooc/01-Welcome/W1S04-PharoModelInaNushell.pdf>
15. <https://jiaxianhua.github.io/smalltalk/2015/05/03/pharo-by-example-13>