

Expressions et Messages - Solution

Ces exercices sont principalement pour vous apprendre à lire et à comprendre les expressions Pharo ainsi qu'à identifier les différentes sortes de messages et receveurs.

Notez que dans les expressions pour lesquels on vous demande de les lire et les exécutez, vous pouvez assumer que l'implantation des méthodes correspond généralement à ce que le nom du message implique (par exemple, $2 + 2 = 4$).

De plus, la plupart des expressions que nous utilisons sont des expressions que vous pouvez exécuter dans Pharo donc n'hésitez pas.

Exercice : Objets Littéraux

A quelles sortes d'objets les expressions littérales font-elles référence ?

Exercice :

```
[ 'Hello, Dave'
```

Solution.

```
[ a string
```

Exercice :

```
[ 1.3
```

Solution.

```
[ a float
```

Exercice :

```
[ #node1
```

Solution.

```
[ a symbol (unique string)
```

Exercice :

```
[ #(2 33 4)
```

Solution.

```
[ an array
```

Exercice :

```
[ [ :each | each scale: 1.5 ]
```

Solution.

```
[ a block (lexical closure)
```

Exercice :

```
[ $A
```

Solution.

```
[ a character
```

Exercice :

```
[ true
```

Solution.

```
[ a boolean
```

Exercice :

```
[ 1
```

Solution.

```
[ a smallinteger
```

1.1 Exercice : Messages

Pour chacune des expressions ci-dessous, répondez aux questions suivantes :

- Quel est l'objet receveur ?
- Quel est le selecteur du message ?
- Quel(s) est (sont) les argument(s) ?
- Quel est le résultat retourné par l'exécution de l'expression ?

Exercice :

```
[ 3 + 4
```

Solution.

```
[ receiver: 3  
  selector: +  
  argument: 4
```

Exercice :

```
[ Date today
```

Solution.

```
[ receiver: Date  
  selector: today  
  argument: none
```

Exercice :

```
[ anArray at: 1 put: 'hello'
```

Solution.

```
[ receiver: anArray  
  selector: at:put:  
  argument: 1 and 'hello'
```

Exercice :

```
[ anArray at: i
```

Solution.

```
[ receiver: anArray  
  selector: at:  
  argument: i
```

Exercice :

```
[ #(2 33 -4 67) collect: [ :each | each abs ]
```

Solution.

```
[ receiver: #(2 33 -4 67)
  selector: collect:
  argument: [ :each | each abs ]
```

Exercice :

```
[ 25 @ 50
```

Solution.

```
[ receiver: 25
  selector: @
  argument: 50
```

Exercice :

```
[ SmallInteger maxVal
```

Solution.

```
[ receiver: SmallInteger
  selector: maxVal
  argument: none
```

Exercice :

```
[ #(a b c d e f) includesAll: #(f d b)
```

Solution.

```
[ receiver: #(a b c d e f)
  selector: includesAll:
  argument: #(f d b)
```

Exercice :

```
[ true | false
```

Solution.

```
[ receiver: true
  selector: |
  argument: false
```

1.1 Exercice : Messages

Exercice :

[Point selectors

Solution.

```
[ receiver: Point  
  selector: selectors  
  argument:
```

Exercice : Nommage

- Que peut-on assumer d'une variable nommée Transferator ?

Solution. Transferator est une variable globale : soit une classe, une variable globale ou une variable de classe.

- Que peut-on assumer d'une variable nommée rectangle ?

Solution. rectangle est une variable locale : soit une variable temporaire, une variable d'instance, un argument de méthode.