# Énoncé : **traduire un texte en javanais**

Fait travailler : **expressions rationnelles** (boucles, conditions)

Le javanais est un procédé de codage simple consistant à **ajouter la syllabe 'av' devant chaque voyelle**. Quelques règles supplémentaires s'appliquent, comme indiqué ci-dessous:

- On n’ajoute pas “av” devant un e en fin de mot

- On ajoute bien “av” devant une voyelle en début de mot

- Si plusieurs voyelles se suivent, on n'ajoute “av” que devant la première

*- Par simplification, on s'assurera que la chaîne est traitée en minuscule, ainsi que la traduction attendue.*

*- Par simplification, les caractères accentués ne seront pas traités comme des voyelles*

**Exemples** : wild code school => wavild cavode schavool

bonjour Amélie => bavonjavour avamélavie

Correction :

**class** Javanais

{

**public function** traduire($string)

{

$string = *strtolower*($string);

$words = *explode*(**' '**, $string);

**foreach** ($words **as** $word) {

$wordTraduction = $prevIsVoyelle = **''**;

$letters = *str\_split*($word);

**foreach** ($letters **as** $letter) {

$av = **''**;

**if** (*preg\_match*(**'/[aeiou]/'**, $letter)) {

**if** (($letter == *end*($letters) && $letter == **'e'**)) {

$av = **''**;

} **elseif** ($prevIsVoyelle) {

$av = **''**;

} **else** {

$av = **'av'**;

}

$prevIsVoyelle = 1;

} **else** {

$prevIsVoyelle = 0;

}

$wordTraduction .= $av . $letter;

}

$traduction[]=$wordTraduction;

}

**return** *implode*(**' '**, $traduction);

}

}

// solution alternative en une seule ligne avec uniquement des regex (à montrer en mode refactoring)

return preg\_replace (‘/((?<![aeiou])([aiou]|e\B))/g’, ‘av\1’, $string);

// on ajoute “av” devant tout motif [aiou] ou e n’étant pas un fin de mot

// on exclu également les motifs de la ligne ci-dessus si le caractère précédent est dans [aeiou] (assertion arrière négaitve (?<!motif))

Tests :

**Class** JavanaisTest **extends** PHPUnit\_Framework\_TestCase

{

**public function** testJavanais()

{

$javanais = **new** Javanais;

$this->assertEquals(**'wavild'**, $javanais->traduire(**'wild'**));

$this->assertEquals(**'wavild cavode schavool'**, $javanais->traduire(**'wild code school'**));

$this->assertEquals(**'avarbre'**, $javanais->traduire(**'arbre'**));

$this->assertEquals(**'é'**, $javanais->traduire(**'é'**));

$this->assertEquals(**'bavonjavour avarthavur'**, $javanais->traduire(**'bonjour Arthur'**));

$this->assertEquals(**'bavonjavour avamélavie'**, $javanais->traduire(**'bonjour Amélie'**));

$this->assertEquals(**'lavinavux avest bavien mavieux qavue mavicravosavoft'**

, $javanais->traduire(**'linux est bien mieux que microsoft'**));

$this->assertEquals(**'tavu avas savans davoute mavieux à favaire qavue de lavire laves tavests!'**, $javanais->traduire(**'tu as sans doute mieux à faire que de lire les tests!'**));

}

}