

# W3-S5-E1-decode-zen

December 15, 2014

## 1 Décoder le module `this`

### 1.1 Exercice - niveau avancé

#### 1.1.1 Le module `this` et le *zen de python*

Nous avons déjà eu l'occasion de parler du *zen de python*; on peut lire ce texte en important le module `this` comme ceci

```
In []: import this
```

Il suit du cours qu'une fois cet import effectué nous avons accès à une variable `this`, de type module:

```
In []: this
```

#### 1.1.2 But de l'exercice

Le but de l'exercice est de deviner le contenu du module, et d'écrire une fonction `decode_zen`, qui retourne le texte du manifeste.

```
In []: from corrections.w3_decode_zen import exo_decode_zen
```

#### 1.1.3 Indices

Cet exercice peut paraître un peu déconcertant; voici quelques indices optionnels :

```
In []: dir(this)
```

Vous pouvez ignorer `this.c` et `this.i`, les deux autres variables du module sont importantes pour nous.

```
In []: # le résultat attendu
      resultat = exo_decode_zen.resultat(this)
```

Ceci devrait vous donner une idée de comment utiliser une des deux variables du module

```
In []: len(this.s) == len(resultat)
```

À quoi peut bien servir l'autre ?

```
In []: this.d[this.s[0]] == resultat[0]
```

Le texte comporte certes des caractères alphabétiques

```
In []: len(this.d)
```

mais pas seulement; les autres sont préservés.

#### 1.1.4 À vous de jouer

```
In []: def decode_zen(this):  
        "<votre code>"
```

#### 1.1.5 Correction

```
In []: exo_decode_zen.correction(decode_zen, this)
```