



## Rapports de discussions de recherche et développements

Comparaison tiramisu et Créole

## Rédacteurs

Gwenaël Rémond ([gremond@cadoles.com](mailto:gremond@cadoles.com))

Emmanuel Garette ([egarette@cadoles.com](mailto:egarette@cadoles.com))

## Référence

```
tiramisu/doc/eole-reports
```

```
git clone ssh://gitosis@git.cadol.es:2222/tiramisu.git
```

# COHÉRENCE DES VALEURS DES VARIABLES

## type des variables

---

### Créole

pas d'unicité du type abstrait : `Multivar`, `CreoleVar` et `TypedVar`

- `String`
- `Ip`
- `Netmask`
- `Number`
- `Boolean`
- `OuiNon`

### tiramisu

unicité du type abstrait : `Option()`

pas de nouveau type multivalué, mais un attribut des types existants:

```
>>> from option import BoolOption
>>> boolopt = BoolOption('bool', 'description de bool',
multi=True)
```

tous les types Créole, plus

- `SymlinkOption`
- `CheckOption` qui permet de définir les "oui/non", "On/Off"

## Validations suivant l'organisation en familles

---

### Créole

#### Organisation par accumulation de références sur des dictionnaires (`EoleDict``)

On peut charger un `EoleDict` avec des variables qui pointent vers des familles qui n'existent pas, aucune validation n'est faite (confiance absolument faite au moment du chargement du XML)

exemple, dans l'espace de nommage racine:

```
<variables>
<variable name="adresse_ip_eth0">

from creole.parsedico import parse_dico
flatdict = parse_dico(eoldict)
dico['adresse_ip_eth0']
KeyError: 'adresse_ip_eth0'
```

### Tiramisu

#### Organisation par arborescence.

Un espace de nommage doit systématiquement être défini, la variable n'est accessible **que** par un path.

## Variables présentes deux fois

---

- Créole : pas de validation possible
- tiramisu : comportement réglable (on autorise l'unicité ou pas)
- dans Créole les valeurs sont **fausses** (c'est la dernière variable qui qui gagne)

Il faut faire confiance au XML

```
<family name="general">
<variable name="adresse_ip_eth0">
<valeur>toto

<family name="services">
<variable name="adresse_ip_eth0">
<valeur>tutu
```

dans `gen_config` la valeur retenue est:

```
general/adresse_ip_eth0 -> tutu
services/adresse_ip_eth0 -> tutu
```

dans `parsedico`, la variable est écrasée:

```
>>> from creole.parsedico import parse_dico
>>> d = parse_dico()
>>> d['adresse_ip_eth0']
tutu
```

dans `tiramisu`:

```
>>> config.general.adresse_ip_eth0
toto
>>> config.services.adresse_ip_eth0
tutu
```