

R208 Analyse et traitement de données
BUT R&T : 1A - S2
TD 1 : Git - exception - mutable/immuable

IUT de Lannion

Préparation du TP

Dans un terminal, déplacez-vous à la racine de votre compte et créez les répertoires suivants : *R208/* puis *R208/TD1/*. Placez-vous dans ce dernier répertoire. Vérifiez que vous êtes bien positionné dans le répertoire *~/R208/TD1*.

1 Découverte des notions

1.1 Gestion des exceptions

Vous allez tester la récupération d'exceptions. créez un fichier *td1.py* et éditez-le pour déclarer une variable de type dictionnaire contenant les informations suivantes :

```
d : Dict[str,str] = {'blanc' : 'white', 'noir' : 'black', 'rouge' : 'red'}
```

Demandez à l'utilisateur de saisir une couleur en français puis utilisez la variable *d* pour afficher sa traduction. Regardez ce qu'il se passe si l'utilisateur saisi une couleur non présente dans *d* et identifiez le type (classe) de l'exception levée. Grâce au mécanisme de récupération des exceptions, affichez un message d'erreur lorsque la couleur saisie n'est pas présente dans *d*.

Autres exemples d'exception

Dans un environnement python en mode interactif, déclarez une variable de type tuple (de trois entiers) avec comme valeurs initiales 1, 1 et 0. Essayez de modifier la troisième valeur pour la passer à 1. Que constatez-vous et pourquoi ?

Utilisez le mécanisme des exceptions pour vérifier qu'une information saisie au clavier correspond bien à un entier. Tant que l'information saisie ne peut pas être transformée en un entier, proposez de recommencer la saisie.

1.2 Git en local

Suivez les étapes suivantes pour découvrir le fonctionnement de *git* en local :

- initialisez l'index *git* pour le répertoire $\sim/R208/TD1$ (commande *git init*),
- ajoutez-le fichier *td1.py* à l'index (commande *git add*) pour ensuite versionner cet ajout (commande *git commit*),
- vérifiez l'état de votre répertoire local vis-à-vis du dépôt (commande *git status*),
- modifiez le contenu du fichier *td1.py* en ajoutant par exemple un commentaire avec votre nom, puis vérifiez les différences entre la copie locale et la version dans le dépôt (commande *git diff*),
- versionner la dernière modification,
- supprimer le fichier *td1.py* du dépôt.

2 Python : gestion versionnée des exceptions et types de données

Assurez-vous d'avoir un fichier *td1.py* à jour avec la version *HEAD* de Git.

2.1 Gestion des exceptions

Définissez une procédure qui demande à l'utilisateur de saisir deux entiers. Si l'utilisateur ne saisit pas des chaînes correspondant à des entiers, affichez un message d'erreur. Vous devez ensuite effectuer la division de ces deux nombres et afficher le quotient. Évidemment, vous devez afficher un message lorsque la division n'est pas possible, par exemple si le dénominateur vaut 0.

Finissez par créer une nouvelle version de votre fichier *td1.py* sur *git* avec vos dernières modifications.

2.2 Données mutables et immuables

Clés de dictionnaires

Un dictionnaire a comme particularité d'associer des valeurs à des clés. Ces clés sont utilisées pour accéder aux valeurs de manière très efficace. Pour cela, il faut que les clés soient uniques et peuvent être de n'importe quel type immuable.

Pour une fois, vous n'utiliserez pas le module *typing* pour créer un dictionnaire simple mais uniquement l'instruction suivante :

```
d : dict = dict()
```

Testez l'ajout de la valeur *'v1'* associée à une clé formée par le tuple *(3, 4)*. Puis testez l'ajout de la valeur *'v2'* associée à une clé formée par la liste *{5, 6}*. Et évidemment, après avoir identifié le type d'exception obtenu, récupérez-la pour afficher le message d'erreur *"la clé doit être de type immuable"*.

Après avoir observé que le dépôt git local n'est pas à jour, versionnez votre code.