

Le Langage Go

Les bases du langage

Exercice Find / Replace

Programme qui trouve & remplace un mot dans un fichier

Exemple

Exemple

Source: <code>wikigo.txt</code>	Résultat
Go was conceived in 2007 to improve programming productivity at Google	Python was conceived in 2007 to improve programming productivity at Google

Features

Features

- Affiche le texte de résultat en console

Features

- Affiche le texte de résultat en console
- Affiche des statistiques

Features

- Affiche le texte de résultat en console
- Affiche des statistiques
 - Nombre d'occurences du mot

Features

- Affiche le texte de résultat en console
- Affiche des statistiques
 - Nombre d'occurences du mot
 - Numéro de lignes où le mot a été trouvé

Exemple de résultat

```
$ go run main.go  
== Summary ==  
Number of occurrences of Go: 10  
Number of lines: 7  
Lines: [ 1 - 8 - 15 - 17 - 19 - 23 - 28 ]  
== End of Summary ==
```

Structure

Structure

```
func FindReplaceFile(src, old, new string) (occ int, lines []int, err error)
```

Structure

```
func FindReplaceFile(src, old, new string) (occ int, lines []int, err error)
```

- `src` : nom de fichier source

Structure

```
func FindReplaceFile(src, old, new string) (occ int, lines []int, err error)
```

- `src` : nom de fichier source
- `old` / `new` : ancien mot / nouveau mot

Structure

```
func FindReplaceFile(src, old, new string) (occ int, lines []int, err error)
```

- `src` : nom de fichier source
- `old` / `new` : ancien mot / nouveau mot
- `occ` : nombre d'occurences de `old`

Structure

```
func FindReplaceFile(src, old, new string) (occ int, lines []int, err error)
```

- `src` : nom de fichier source
- `old` / `new` : ancien mot / nouveau mot
- `occ` : nombre d'occurences de `old`
- `lines` : slice des numéros de lignes ou `old` a été trouvé

Structure

```
func FindReplaceFile(src, old, new string) (occ int, lines []int, err error)
```

- `src` : nom de fichier source
- `old` / `new` : ancien mot / nouveau mot
- `occ` : nombre d'occurences de `old`
- `lines` : slice des numéros de lignes ou `old` a été trouvé
- `err` : erreur de la fonction

**Si `FindReplaceFile()` n'arrive pas à
ouvrir le fichier**

Il faut renvoyer une erreur

Structure

```
func ProcessLine(line, old, new string) (found bool, res string, occ int)
```

Structure

```
func ProcessLine(line, old, new string) (found bool, res string, occ int)
```

- `line` : ligne à traiter

Structure

```
func ProcessLine(line, old, new string) (found bool, res string, occ int)
```

- `line` : ligne à traiter
- `old` / `new` : ancien mot / nouveau mot

Structure

```
func ProcessLine(line, old, new string) (found bool, res string, occ int)
```

- `line` : ligne à traiter
- `old` / `new` : ancien mot / nouveau mot
- `found` : vrai si au moins une occurrence trouvée

Structure

```
func ProcessLine(line, old, new string) (found bool, res string, occ int)
```

- `line` : ligne à traiter
- `old` / `new` : ancien mot / nouveau mot
- `found` : vrai si au moins une occurrence trouvée
- `res` : résultat de remplacement (`res == line` si aucun changement !)

Structure

```
func ProcessLine(line, old, new string) (found bool, res string, occ int)
```

- `line` : ligne à traiter
- `old` / `new` : ancien mot / nouveau mot
- `found` : vrai si au moins une occurrence trouvée
- `res` : résultat de remplacement (`res == line` si aucun changement !)
- `occ` : nombre d'occurrences de `old` *dans la ligne*

Outils

Outils - package **bufio**

- Lire le fichier ligne par ligne

```
scanner := bufio.NewScanner(srcFile)
for scanner.Scan() {
    t := scanner.Text() // t contient une ligne
}
```

Outils - package `strings`

- Manipuler des strings

```
c := strings.Contains("go ruby java", "go") // c == true
cnt := strings.Count("go go go", "go") // cnt == 3
res := strings.Replace("old go", "go", "python", 0) // res == "old python"
```

Bonus

Bonus

- Remplacer le mot en miniscule
 - Go et go ==> Python et python
- Stocker le résultat dans un fichier
 - Modifier légèrement `FindReplaceFile()`

Outils - package `bufio`

- Ecrire dans un fichier chaque ligne

```
writer := bufio.NewWriter(dstFile)
defer writer.Flush()
fmt.Fprintln(writer, "Texte d'une ligne")
```