Defer

Repousser l'exécution d'une instruction

Cas d'utilisation

Dans l'exemple si dessous

```
func main() {
	f := os.OpenFile("foo.txt")
	if condition1 {
		return // Oops...! pas de close ici!
	}
	// code
	f.Close()
}
```

Le `f.close()` peut être très éloigné dans le code du `OpenFile`.
Or, quand on ouvre un fichier, on va surement le fermer ensuite, mais pas tout de suite...

On peut même potentiellement arrêter notre fonction avant de fermer le fichier sans faire attention !

Il paraît donc plus "jolie" de mettre le code d'ouverture du fichier et celui de fermeture côte à côte, car il traite le même sujet.

Solution

```
func main() {
	f := os.OpenFile("foo.txt")
	defer f.Close() // exécuté quand on sort de main()

	if condition1 {
```

```
□□return
□}
}
```

> `defer` est rattaché à la fonction qui l'invoque

Ordre d'exécution

Les instructions mises en `defer` fonctionnerons comme une pile **LIFO** (Last In First Out).

Revision #1
Created 2022-05-09 19:14:07 UTC by Noé Larrieu-Lacoste
Updated 2022-05-09 19:15:02 UTC by Noé Larrieu-Lacoste