Les bonnes pratiques

```
#!/usr/bin/python
# -*- coding: utf-8 -*-


## Importation des modules


## Déclaration des constantes


## Déclaration des fonctions


## Programme principal
```

Respecter la structure du programme

Commenter le programme

Commenter les fonctions

Expliciter les paramètres des fonctions

Noms des constantes et des classes commençants par une majuscule

Noms des variables et des fonctions commençants par une minuscule

Pas de caractères accentués, spéciaux, sauf l'underscore _

Noms explicites, différents de ceux du langage

Tester chaque ligne de code

Lire le message d'erreur de bas en haut en cas d'erreur de syntaxe ou d'exécution

Prier Turing en cas d'erreur de sémantique

Séparer les traitements et les calcul de l'affichage

Décomposer les traitements complexes en successions de traitements simples

Regrouper les traitements portant sur les mêmes données

Regrouper les traitements avec leurs données

Tester un programme

Après chaque écriture d'une nouvelle ligne de code

En visualisant et comprenant l'état du système

- Afficheurs, boutons (ON, OFF, ...)
- Donnée d'un capteur
- Commande d'un actionneur

En visualisant et comprenant les résultats intermédiaires

C'est l'utilité de la fonction print()

De plusieurs manières différentes

Le plus de cas possibles

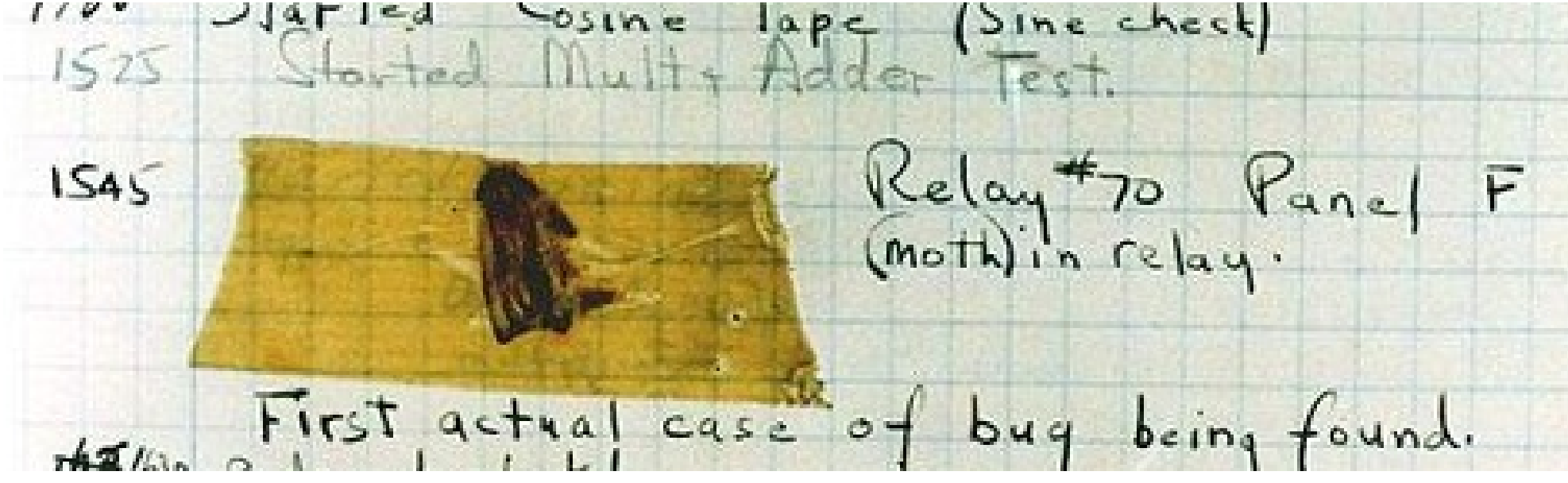
- Quelques cas standards
- Tous les cas extrêmes

Un **bug** informatique est faussement attribué à **Grace Hopper**.

Elle constata dans son journal d'entretien le mauvais fonctionnement du Harvard Mark II, l'un des premiers ordinateurs électromécaniques.

Hopper ne trouva pas le bug, comme elle le reconnaissait volontiers. Les opérateurs qui l'ont trouvé plus tard étaient familiers avec le terme d'ingénierie et amusés ont gardé l'insecte avec l'annotation « premier cas réel de bug trouvé ».

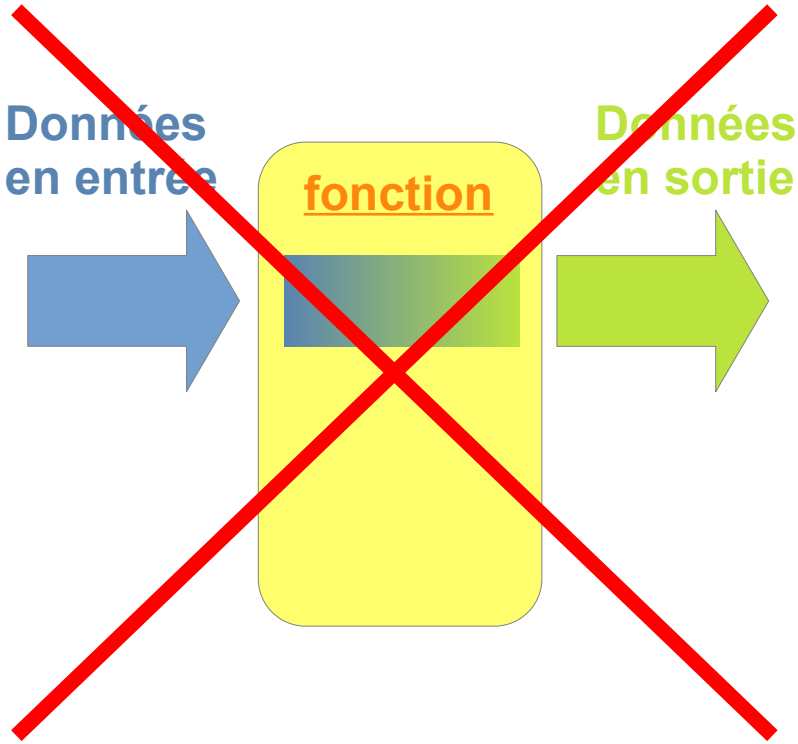
Grace Hopper était informaticienne américaine et Rear admiral de la marine. Elle est à l'origine du concept de langage de programmation indépendant de la machine et a écrit le premier compilateur (A-0 System). *Wikipedia*

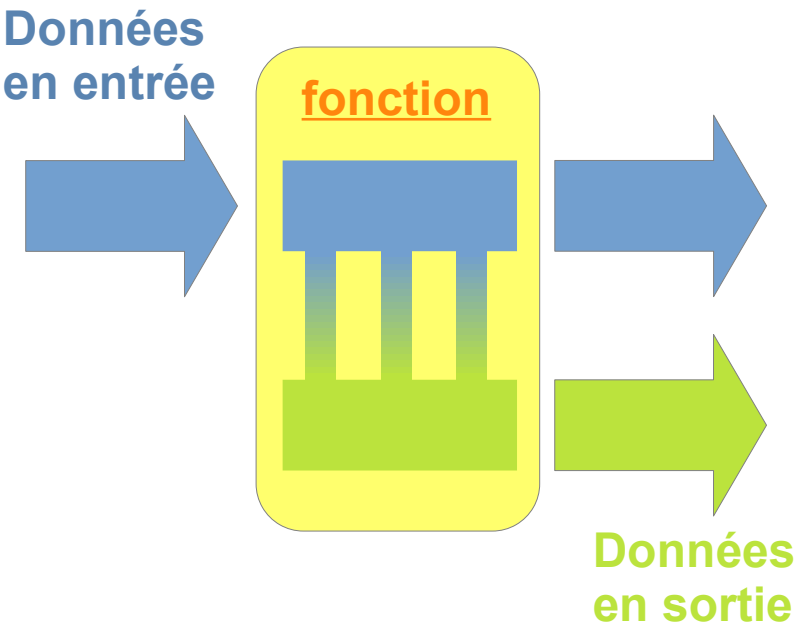


Fonctions et paramètres

Une fonction est un traitement utilisant des **données existantes en entrée** pour produire de **nouvelles données en sortie**

Elle ne doit **pas modifier les données existantes**





Prototypage des fonctions en Python (spécification et tests)

1. décrire la fonction (docstring)

2. décrire les paramètres (noms, types, ...)

3. tester les pré-conditions

```
def charge_texte(nom_fichier):
    '''
    Charge un fichier texte, retourne son contenu sous forme de chaîne de caractères
    et le nombre de caractères lus

    @param entrée : nom_fichier, str, nom du fichier à ouvrir
    @param sortie : texte, str, contenu du fichier
    @param sortie : nbr_caracteres, int, nombres de caractères lus
    '''
    assert type(nom_fichier) == str, 'Le nom de fichier doit être une chaîne de caractères.'
    assert len(nom_fichier) >= 5, 'Le nom de fichier doit contenir au moins 5 caractères.'
    assert nom_fichier[-4:] == '.txt', '''L'extension du nom de fichier doit être .txt'''

    fichier = open(nom_fichier, 'r', encoding='utf-8')
    texte = fichier.read()
    fichier.close()

    nbr_caracteres = len(texte)

    assert type(texte) == str and len(texte) > 0, 'Le fichier est vide.'
    assert type(nbr_caracteres) == int, 'Le nombre de caractères doit être un entier.'

    return texte, nbr_caracteres
```

mot clé de l'assertion

test qui doit être vrai

si le test est faux, le programme s'arrête en affichant l'information correspondante

4. tester les post-conditions