

# Algorithmique Appliquée

BTS SIO SISR

Procédures et fonctions

# Plan

- Procédures : définition et appel
- Arguments
- Valeurs par défaut
- Variables locales et globales
- Fonctions
- Spécifications et contrat
- Modularisation de code
- Nombre variable d'argument
- Retour de plusieurs résultats
- Un mot sur la récursivité
- Fonctions d'ordre supérieur
- Fonctions lambda
- Programmation impérative et fonctionnelle
- Un mot sur les méthodes

# Procédures : définition et appel

Intérêt : réutilisabilité && couches d'abstraction

Syntaxe

Appel

Exemples

# Arguments

# Ajout d'arguments à la procédure

## Syntaxe d'appel

### Exemples

# Valeurs par défaut



Intérêt  
Syntaxe  
Exemples



# Variables locales et globales

Scope

Exemple de 2 procédures avec les mêmes variables  
Exemple avec une var globale et une fonction avec la  
même variable mais valeur différente  
Notion de portée (scope) des variables  
Mot clé global => intérêt, syntaxe, dangers

# Fonctions

Intérêt => output(s)

Mot clé return

Usage

Exemples

# Spécifications et contrat

Retour sur les docstrings

Spécification des entrées

Spécification des sorties

Spécification du contrat

Vérification du contrat : quelques moyens (code d'erreur, exceptions, etc.)

Jusqu'où aller ? Vérification des types

Coût de la vérification dynamique du contrat : runtime performance

Approche pragmatique et bonnes pratiques concrètes



# Modularisation de code et conventions avec la fonction main



# Modularisation de code

Quelle fonction appeler en premier ? => **main**

Syntaxe du main

Responsabilité unique d'une fonction

Taille idéale d'une fonction

Ne pas se répéter

Rassembler les fonctions autour d'un sujet dans un  
fichier/script

Mot clé import

Exemple d'utilisation du mot clé import

# TD : Fonctions géométriques simples

# **TD : Fonctions géométriques simples**

[Lien vers le sujet de TD.](#)

# Nombre variable d'arguments

Exemple avec la fonction print

Exemple avec la fonction sum

Intérêt

Syntaxe

Exemples

# Retour de plusieurs résultats



Intérêt  
Syntaxe  
Exemples

# Un mot sur la récursivité

GNU is Not Unix  
WINE Is Not an Emulator  
Fractal

Définition comme en mathématiques :  $f(0) = 1$ ,  $f(N) = 3 * f(N-1) + 4$ .

Exemple simple

Attention à la stack : exemple de stack overflow

Attention à la stack => on empile tout à chaque call  
récuratif => utilisation massive de mémoire

Conclusion : c'est beau, mais on évite en général

# Fonctions d'ordre supérieur

Fonctions en tant qu'objets

Intérêt = généricité && réutilisation

Tout est objet : `def foo(): pass; type(foo)`

Assignment d'une fonction à une variable

Réassignation d'une autre fonction à une variable

Passage d'une fonction comme argument d'une  
fonction

Générateur de fonction : on retourne une nouvelle  
fonction

# Fonctions lambda

Origine : calcul lambda (et pas un truc quelconque...)  
Intérêt : les dévs sont un peu feignants et n'aiment pas trop taper (sauf aux jeux de baston)

Syntaxe  
Exemples



# Programmation impérative

vs

# Programmation fonctionnelle

Notions de pureté et d'immuabilité

De grands pouvoirs impliquent de grandes  
responsabilités : code mess && spaghetti code  
Programmation impérative => variables (ou état)  
globales, effets de bords non maîtrisé et beau bord\*\*  
A moins d'appliquer des bonnes pratiques stricts =>  
boiled frog && broken windows  
Exemple qui marche : kernel Linux => règles très (très  
très) sévères  
Règle d'immutabilité : les variables constantes  
Règle de pureté : pas d'effet de bord (print, etc.)  
Il existe des langages qui imposent ces règles et qui  
pronent certaines techniques héritées de la théorie des  
catégories, une branche des maths.

# Un mot sur les méthodes

Avec l'exemple du type `str`

Python est *orienté object*

Tout est objet

Définition On appelle méthode une fonction rattachée à un type.

Un type définit ses propriétés (données membres) && capacités (méthodes).

`str` est un type. `upper()` est une méthode de ce type.

Exemples en plus.

# TP : Fonctions d'ordre supérieur

# **TP : Fonctions d'ordre supérieur**

**Lien vers le sujet de TP.**



# Devoir à la Maison 02



# **DM : Retours sur les fonctions et le débogage**

**Lien vers le sujet de DM.**