Programmation Python : Les bases

Cheat Sheet

Général

`# commentaire`	insère un commentaire dans le programme
`print(v)`	affiche la valeur `v` à l'écran
`input("Phrase")`	affiche `Phrase` à l'écran et renvoie la phrase tapée à la console

Opérateurs arithmétiques

`+`	addition	`-`	opposé ou soustraction
`*`	multiplication	`/`	division
`**`	exponentiation		
`//`	division entière	`%`	reste de la division entière

Opérateurs de comparaison

`==`	égal	`!=`	différent
`<`	strictement plus petit	`>`	strictement plus grand
`<=`	plus petit ou égal	`>=`	plus grand ou égal

Opérateurs logiques

`not`	*« non »* logique		
`and`	*« et »* logique	`or`	*« ou »* logique

Conversion de données

`type(v)`	renvoie le type de `v`
`int(v)`	convertit `v` en nombre entier
`float(v)`	convertit `v` en nombre flottant
`complex(v)`	convertit `v` en nombre complexe
`bool(v)`	convertir `v` en booléen
`str(v)`	convertit `v` en chaine de caractères

Instructions conditionnelles

`if` *1re condition*`:` *1er bloc de code*	Exécute le *1er bloc de code* si la *1re condition* vaut `True`,
`elif` *2e condition*`:` *2e bloc de code*	sinon exécute le *2e bloc de code* si la *2e condition* vaut `True`
`else:` *3e bloc de code*	et exécute le *3e bloc de code* sinon

Instruction répétitive

`while` *condition*`:` *1er bloc de code*	répète l'exécution du *bloc de code* tant que la *condition* vaut `True`,
`else:` *2e bloc de code*	et exécute le *2e bloc de code* lorsqu'elle vaut `False`
`break`	interrompt l'exécution de la boucle
`continue`	interrompt l'itération courante de la boucle

Importation

`import` n	importe le module n
`from` n `import` m	importe m depuis le module n

Fonction

`def` $n(p_1, \ldots, p_r)$`:` *bloc de code*	définit une fonction n avec les r paramètres positionnels $p_1, \ldots, p_r$
`def` $n(p_1{=}v_1, \ldots, p_r{=}v_r)$`:` *bloc de code*	définit une fonction n avec les r paramètres optionnels $p_1, \ldots, p_r$ avec comme valeurs par défaut $v_1, \ldots, v_r$
`return v`	quitte la fonction en renvoyant la valeur `v`
`global v`	indique que `v` réfère une variable globale

Séquence

`len(s)`	taille de `s`
`s[i]`	élément d'indice `i` de `s` (premier indice : 0)
`del(s[i])`	supprime l'élément d'indice `i` de `s`
`s[i:j]`	sous-séquence de `i` (inclus) à `j` (exclu) *(on peut omettre les bornes `i` et/ou `j`)*
`s + t`	concatène les deux séquences `s` et `t`
`s * n`	répète `n` fois la séquence `s`
`v in s`	vaut `True` si `v` se trouve dans `s`, `False` sinon
`range(n, m)`	intervalle d'entiers de `n` (inclus) à `m` (exclu)
`for e in s:` *bloc de code*	exécute le *bloc de code* pour chaque élément `e` de `s`