Python (Types de base).

 $int \ , \ float \ , \ bool$ $entier \ , \ flottant \ , \ bool\acute{e}en$

$\{ {f Python} \; ({f Op\'erateurs} \; {f num\'eriques}). \}$

 $\begin{array}{lll} +\;,\; -\;,\; * & \text{addition}\;,\; \text{soustraction}\;,\; \text{multiplication}\\ /\; & \text{division}\; (\text{le résultat est un flottant})\\ ** & \text{exponentiation}\\ //\;,\; \% & \text{quotient}\;,\; \text{reste de la division euclidienne} \end{array}$

Python (Conteneurs).

Non modifiables

str 'hei' ou "hei" chaines de caractères tuple (421,3.14,'hei') t-uplets

Modifiables

 $\begin{array}{ll} list & \left[421, 3.14, \text{'hei'}\right] & listes \\ dict & \left\{\text{cl\'e}: valeur\right\} & dictionnaires \textbf{(MP)} \end{array}$

Python (Opérations sur les conteneurs).

 $\begin{array}{ll} min(cont) \;,\; max(cont) & \text{minimum} \;,\; maximum \; de \; cont \\ sum(cont) & \text{somme } \; des \; \acute{e}l\acute{e}ments \; de \; cont \\ + \;, \; * & \text{concat\'enation} \;, \; duplication \\ cont. \\ cont. \\ count(val) & \text{nombre d'occurrences } \; de \; val \; dans \; cont \\ \end{array}$

Opérations spécifiques aux listes

 $\begin{array}{ll} lst. \\ \hline{insert}(idx, val) & insertion \ de \ val \ dans \ lst \ \grave{a} \ la \ position \ idx \\ \hline{del}(lst[idx]) & suppression \ dans \ lst \ de \ l'élément \ d'index \ idx \\ \end{array}$

Python (Chaines de caractères).

var = eval(ch)

conversion d'une chaine de caractères
en entier, flottant, liste, ...
ch = str(var)

conversion d'un entier, flottant, liste, ...
en chaine de caractères

"-". join(['Jean','Pierre'])

chaine de jointure

séquence de chaines

"2;5.1;12". split(";")

donne ['2','5.1','12']

Python (Indexation des séquences).

Pour les listes, tuples, chaines de caractères, ...

$\{ Python (Affichage/Saisie). \}$

print(ch, "contient",len(ch), "caractères") affichage e=input("message facultatif") affectation à e de la chaine de caractères saisie

Python (Déclaration de fonction).

Python (Fichiers).

 $f{=}open('chemin \setminus nomfichier.txt' \;,\; 'w' \;)\; ouverture\; en \qquad \begin{array}{c} lecture \\ \acute{e}criture \\ \dot{a}_i \end{array}$

f.readline() lecture d'une ligne (caractère de fin de ligne compris) à partir de l'endroit où l'on se trouve dans le fichier

f.readlines() séquence des lignes du fichier

f.write('texte à écrire') écriture à partir de l'endroit où l'on se trouve dans le fichier

f.close() fermeture

$\{ \mathbf{Python} \ (\mathbf{Instructions} \ \mathbf{conditionnelles}). \}$

if *b1* :

 $|bloc\ d$ 'instructions sinon

Python (Boucle itérative).

 $\begin{array}{c} \textbf{for } i \text{ in range}(\overbrace{[a,]} \underbrace{b}_{\text{exclu}} \overbrace{[,p]}^{\text{exclu}}): & \text{i varie de a} \\ |bloc\ d'instructions & \text{avec un pas de p} \end{array}$

Extension du for aux conteneurs

 $egin{array}{lll} ext{for i in cont:} & ext{i prend ses valeurs} \ & bloc \ d'instructions & ext{dans cont} \end{array}$

Python (Boucle conditionnelle).

 while b:
 instructions éxécutées

 bloc d'instructions
 tant que b est vrai