



Python

CPGE TSI

Pr[Pleaseinsertintopreamble]sentation

Syntaxe

Types

Nombres

Chaînes

Listes

Tuples

Tests

Les blocs

Conditionnelle

Boucles for

boucles while

Fonctions

Bibliothèques

Erreurs

Aide

Python

CPGE TSI

18 novembre 2019



Python

CPGE TSI

Pr[Please insert intopreamble]sentation

Syntaxe

Types

Nombres

Chaînes

Listes

Tuples

Tests

Les blocs

Conditionnelle

Boucles for

boucles while

Fonctions

Bibliothèques

Erreurs

Aide

L'interpréteur

```
>>> print("Coucou")  
Coucou
```

L'éditeur

```
# -*- coding: utf-8 -*-  
  
print("Coucou")
```

Généralement, on utilisera la ligne de commande à des fins de tests et l'éditeur pour automatiser un ensemble de tâches.



Python

CPGE TSI

Pr[Pleaseinsertintor

Syntaxe

Types

Nombres

Chaînes

Listes

Tuples

Tests

Les blocs

Conditionnelle

Boucles for

boucles while

Fonctions

Bibliothèques

Erreurs

Aide

Python, c'est joli ...

Python dispose d'une syntaxe élégante :

- docstring
- indentation après " : "
- commentaires commençant par un "#"



Python

CPGE TSI

Pr[Please insert into]

Syntaxe

Types

Nombres

Chaînes

Listes

Tuples

Tests

Les blocs

Conditionnelle

Boucles for
boucles while

Fonctions

Bibliothèques

Erreurs

Aide

... très joli !

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

""" Docstring : Ce programme ne
fait rien d'important
"""

a = "Hello"
b = "Coucou"
choix = 1

if choix == 1: # si choix = 1 alors ...
    print(a)
else: # sinon ...
    print(b)
```



Python

CPGE TSI

Pr[Please insert into]

Syntaxe

Types

Nombres

Chaînes

Listes

Tuples

Tests

Les blocs

Conditionnelle

Boucles for
boucles while

Fonctions

Bibliothèques

Erreurs

Aide

Types numériques

Nombres : int, float, complex, bool

exemples

```
>>> a = 1
>>> type(a)
<class 'int'>
>>> b = 1/3
>>> type(b)
<class 'float'>
>>> c = 2j
>>> type(c)
<class 'complex'>
>>> choix = True
>>> type(choix)
<class 'bool'>
```



Python

CPGE TSI

Pr[Pleaseinsertinto]

Syntaxe

Types

Nombres

Chaînes

Listes

Tuples

Tests

Les blocs

Conditionnelle

Boucles for

boucles while

Fonctions

Bibliothèques

Erreurs

Aide

Autres types

Chaînes, listes, tuples, ...

exemples

```
>>> ch0 = "Je suis une chaîne"
>>> type(ch0)
<class 'str'>
>>> liste0 = ["en vrac", [2, 3], (5,7), True]
>>> type(liste0)
<class 'list'>
>>> tuple0 = (1,5,"ch")
>>> type(tuple0)
<class 'tuple'>
```



Python

CPGE TSI

Pr[Please insert into]

Syntaxe

Types

Nombres

Chaînes

Listes

Tuples

Tests

Les blocs

Conditionnelle

Boucles for

boucles while

Fonctions

Bibliothèques

Erreurs

Aide

Il existe des opérateurs arithmétiques, de comparaison, ...

Opérateurs pour les nombres

`+`, `*`, `/`, `**`, `==`, `round ()`, `abs ()`, `min()`, `max()`, ...

opérations

```
>>> a == b
False
>>> a + 2* b
1.6666666666666665
>>> a < b
False
>>> c + 2*a
(2+2j)
>>> a+=1
>>> a
2
>>> round(b, 2)
0.33
```



Python

CPGE TSI

Pr[Please insert into]

Syntaxe

Types

Nombres

Chaînes

Listes

Tuples

Tests

Les blocs

Conditionnelle

Boucles for
boucles while

Fonctions

Bibliothèques

Erreurs

Aide

Les chaînes de caractères ne sont pas modifiables.

Opérateurs pour les chaînes

+ (concaténation), * (répétition), len (), min (), max (), ...

opérations

```
>>> ch1 = "bla"
>>> ch2 = "bli"
>>> (2*ch1 + 3*ch2)*2
'blablabliblibliblablabliblibli'
>>> len(ch1)
3
>>> ch1[2]
'a'
```



Python

CPGE TSI

Pr[Please insert into

Syntaxe

Types

Nombres

Chaînes

Listes

Tuples

Tests

Les blocs

Conditionnelle

Boucles for
boucles while

Fonctions

Bibliothèques

Erreurs

Aide

Les listes sont des tableaux indexés modifiables.

Méthodes pour les listes

append (), join (), split (), remove (), sort (), ...

listes 1

```
>>> ma_liste = [1,5,8,12,-9.8]
>>> len(ma_liste)
5
>>> ma_liste[3]
12
>>> del ma_liste[4]
>>> ma_liste
[1, 5, 8, 12]
>>> liste_ch = [str(el) for el in ma_liste]
>>> liste_ch
['1', '5', '8', '12']
>>> '--'.join(liste_ch)
'1--5--8--12'
```



Python

CPGE TSI

Pr[Pleaseinsertinto]

Syntaxe

Types

Nombres

Chaînes

Listes

Tuples

Tests

Les blocs

Conditionnelle

Boucles for

boucles while

Fonctions

Bibliothèques

Erreurs

Aide

listes 2

```
>>> ma_liste = []
>>> for cpt in range(12):
...     ma_liste.append(cpt**3)
...
>>> ma_liste
[0, 1, 8, 27, 64, 125, 216, 343, 512, 729, 1000, 1331]
>>> ma_liste_prime = ma_liste[1:5]
>>> ma_liste_prime
[1, 8, 27, 64]
```



Python

CPGE TSI

Pr[Please insert into]

Syntaxe

Types

Nombres

Chaînes

Listes

Tuples

Tests

Les blocs

Conditionnelle

Boucles for

boucles while

Fonctions

Bibliothèques

Erreurs

Aide

Ce sont des listes indexées non modifiables (ou non mutables)

tuples

```
>>> mon_tuple = (3, 0, 9)
>>> mon_tuple[0]
3
>>> ma_liste = list(mon_tuple)
>>> ma_liste
[3, 0, 9]
```



Python

CPGE TSI

Pr[Please insert into]

Syntaxe

Types

Nombres

Chaînes

Listes

Tuples

Tests

Les blocs

Conditionnelle

Boucles for
boucles while

Fonctions

Bibliothèques

Erreurs

Aide

Les blocs



Notez le caractère " : " qui permet de commencer un bloc. Il est suivi d'une indentation (touche Tab).



Python

CPGE TSI

Pr[Please insert intro or table of contents]

Syntaxe

Types

Nombres

Chaînes

Listes

Tuples

Tests

Les blocs

Conditionnelle

Boucles for
boucles while

Fonctions

Bibliothèques

Erreurs

Aide

L'instruction `if` sert à poser une condition :

SI ... ALORS ... SINON ...

L'instruction `elif` peut être absente ou présente plusieurs fois.

if, elif, else

```
choix = 5

if choix == 1:
    print("Hello")
elif choix == 2:
    print("Coucou")
elif choix == 3:
    print("Hallo")
else:
    print("Sympa !")
```



Python

CPGE TSI

Pr[Please insert into preamble]ésentation

Syntaxe

Types

Nombres

Chaînes

Listes

Tuples

Tests

Les blocs

Conditionnelle

Boucles for

boucles while

Fonctions

Bibliothèques

Erreurs

Aide

L'instruction `for` sert à effectuer une boucle dont on connaît le début et la fin, ainsi que le pas. Elle signifie :

POUR ... VARIANT DE ... à ... PAR PAS DE ... FAIRE.

for 1

```
>>> for cpt in range(3,13,2):  
...     print(cpt)  
...  
3  
5  
7  
9  
11
```



Python

CPGE TSI

Pr[Pleaseinsertinto]

Syntaxe

Types

Nombres

Chaînes

Listes

Tuples

Tests

Les blocs

Conditionnelle

Boucles for

boucles while

Fonctions

Bibliothèques

Erreurs

Aide

for 2

```
>>> ma_liste = [5,9,"elem",6.0]
>>> for el in ma_liste:
...     if type(el) != int:
...         print("L'élément ", el, " n'est pas un
...             entier")
...
L'élément  elem  n'est pas un entier
L'élément  6.0  n'est pas un entier
```



Python

CPGE TSI

Pr[Pleaseinsertinto]

Syntaxe

Types

Nombres

Chaînes

Listes

Tuples

Tests

Les blocs

Conditionnelle

Boucles for

boucles while

Fonctions

Bibliothèques

Erreurs

Aide

while 1

L'instruction `while` sert à effectuer une boucle dont on ne connaît pas forcément la fin. Elle signifie :

TANT QUE ... FAIRE.

```
>>> cpt = 25
>>> while cpt > 4:
...     cpt = cpt - 5
...     print(cpt)
...
20
15
10
5
0
```



Python

CPGE TSI

Pr[Pleaseinsertinto]

Syntaxe

Types

Nombres

Chaînes

Listes

Tuples

Tests

Les blocs

Conditionnelle

Boucles for

boucles while

Fonctions

Bibliothèques

Erreurs

Aide

while 2

```
>>> c = 100
>>> while c <= 1e8:
...     c = c * 10
...     print(c)
...     if c == 1e5:
...         print("100 000 !")
...
1000
10000
100000
100 000 !
1000000
10000000
100000000
1000000000
```



Python

CPGE TSI

Please insert the topic to be displayed here

Syntaxe

Types

Nombres

Chaînes

Listes

Tuples

Tests

Les blocs

Conditionnelle

Boucles for
boucles while

Fonctions

Bibliothèques

Erreurs

Aide

La notion de fonction est indispensable en programmation : elle permet notamment d'utiliser plusieurs fois la même suite d'instructions.

fonctions

```
def puiss_n (x,n):  
    resultat = x ** n  
    return resultat  
  
print(puiss_n(2,4))  
print(puiss_n(5,3))
```



Python

CPGE TSI

Pr[Pleaseinsert]to

Syntaxe

Types

Nombres

Chaînes

Listes

Tuples

Tests

Les blocs

Conditionnelle

Boucles for
boucles while

Fonctions

Bibliothèques

Erreurs

Aide

Les bibliothèques : kesako ?

Il existe des fonctions ou méthodes prédéfinies dans des bibliothèques, par exemple pour les graphes, pour la résolution d'équations différentielles, ...

import, from

```
>>> from math import cos
>>> cos(3.14)
-0.9999987317275395
>>> import math as m
>>> m.cos(1.57)
0.0007963267107332633
>>> from math import *
>>> sqrt(2)
1.4142135623730951
```



Python

CPGE TSI

Pr[Pleaseinsertinto]

Syntaxe

Types

Nombres

Chaînes

Listes

Tuples

Tests

Les blocs

Conditionnelle

Boucles for
boucles while

Fonctions

Bibliothèques

Erreurs

Aide

Les erreurs les plus souvent rencontrées :

- l'oubli des " : " en fin de test (if, for, while, else),
- l'oubli de l'indentation,
- la confusion entre "==" (le test d'égalité) et "=" (l'affectation),
- la ligne précédente non terminée (par exemple, oubli d'une parenthèse fermante),
- les indices : de 0 à $n - 1$ sous *Python*,
- la copie de liste.



Python

CPGE TSI

Pr[Pleaseinsertinto]

Syntaxe

Types

Nombres

Chaînes

Listes

Tuples

Tests

Les blocs

Conditionnelle

Boucles for

boucles while

Fonctions

Bibliothèques

Erreurs

Aide

Demander de l'aide

À qui ?

- À moi : `obelix56@free.fr`
- Sur les forums (fora ?)

Par exemple sur le site :

`https://www.developpez.net/forums/f96/autres-langages/python-zope/`