

Structures conditionnelles

Un programme informatique ne travaille pas toujours de façon séquentielle. Il doit pouvoir réagir différemment en fonctions des conditions (données saisies par l'utilisateur, résultats d'un calcul, etc.). Une structure conditionnelle permet d'implémenter ce type de fonctionnement. Dans le cadre du programme du B.T.S, nous n'étudierons que la structure Si-Alors-Sinon. Il faut, cependant, savoir qu'il existe d'autres structures comme la structure de cas par exemple.

I. Présentation du Si-Alors-Sinon

Cette structure exécute un bloc d'instructions si une condition est remplie et un autre bloc dans le cas contraire.

Dans l'environnement Python, elle se présente sous la forme suivante:

if condition :

 Bloc d'instructions exécuté si la condition est vraie

else:

 Bloc d'instructions exécuté si la condition n'est pas remplie

suite du programme

Voici un exemple de programme valide:

```
x=int(input(« Saisir un nombre: »))
```

```
if x>=0:
```

```
    print(« Ce nombre est positif ou nul.»)
```

```
else:
```

```
    print(« Ce nombre est strictement négatif.»)
```

```
print(x*4+2)
```

Sera toujours exécuté

Il n'est pas toujours utile d'utiliser un bloc d'instructions dans le cas où la condition n'est pas remplie. On parle, dans ce cas, d'une structure Si-Alors.

Voici un exemple de code valide:

```
x=int(input(« Saisir un nombre: »))
```

```
if x>=1000:
```

```
    print(« Ce nombre est grand.»)
```

```
print(« Fini »)
```

Sera toujours exécuté

II. Structures imbriquées

Il est tout-à-fait possible d'imbriquer des structures conditionnelles afin de distinguer des sous-cas.

Exemple de code fonctionnel:

```
x=int(input(« Saisir un nombre: »))
```

```
if x>=0:
```

```
    if x>0:
```

```
        print(« Ce nombre est strictement positif. »)
```

```
    else:
```

```
        print(« Ce nombre est nul. »)
```

```
else:
```

```
    print(« Ce nombre est strictement négatif.»)
```

```
print(x*4+2)
```

Sera toujours exécuté