

Introduction au Langage C,C++

Instructions de base

■ opérateurs de base

- +, -, *, / → opérateurs arithmétique de base
- % → reste d'une division entière
- == → test d'égalité
- != test de différence
- <, >, <=, >= → test de comparaison
- ! → négation
- || → **ou** logique pour évaluer une expression
- && → **et** logique pour évaluer une expression

Introduction au Langage C,C++

Instructions de base

▣ opérateurs de base

▣ `a = 2+3`

valeur de a: 5

▣ `r = 3%2`

valeur de a: 1

▣ `a = (3==3)`

valeur de a: 1

▣ `a = (6==5)`

valeur de a: 0

▣ `a = (2!=3)`

valeur de a: 1

▣ `a = (6<=3)`

valeur de a: 0

▣ `a = !1`

valeur de a: 0

▣ `a = ((3==3) || (6<=3))`

valeur de a: 1

▣ `a = ((3==3) && (6<=3))`

valeur de a: 0

Introduction au Langage C,C++

Instructions de base

■ boucle pour

```
for(expr1;expr2;expr3)
{
    instructions
}
```

expr1: évaluée 1 seule fois en début de boucle

expr2: évaluée avant chaque itération. Si vrai alors les instructions de la boucle sont exécutées sinon la boucle est terminée

expr3: évaluée à la fin de chaque itération

Introduction au Langage C,C++

Instructions de base

■ boucle pour

■ exemple

// somme des 100 premiers entiers

```
int i,s;
```

```
s= 0
```

```
for(i=0;i<=100;i = i+1)
```

```
{
```

```
    s = s+i;  // ou s+=i;
```

```
}
```

```
cout <<"s = "<<s<<' \n';
```

Introduction au Langage C,C++

* Instructions de base

- boucle tant que
`while (expression)`
`{`
 instructions;
`}`

expression est évaluée avant chaque itération. Si le résultat est vrai alors les instructions sont exécutées sinon on sort de la boucle

- exemple

```
i = 0;  
n = 20;  
while (i < n)  
{  
    i++;  
}
```

Introduction au Langage C,C++

✦ Instructions de base

▣ boucle répéter

```
do
{
    instructions;
} while (expression)
```

expression est évaluée après chaque itération. Si le résultat est vrai alors les instructions sont exécutées sinon on sort de la boucle

▣ exemple

```
i = 0;
n = 20;
do
{
    i++;
} while(i < n)
```


Introduction au Langage C,C++

* Instructions de base

- instruction conditionnelle simple *si alors*

```
if (expression)
{
    instructions;
}
```

expression est évaluée après chaque itération. Si le résultat est vrai alors les instructions sont exécutées sinon on sort de la boucle.

- exemple

```
int i =5;
int n;
if (i<=20)
{
    n =0;
}
```

Introduction au Langage C,C++

Instructions de base

- instruction conditionnelle simple *si alors sinon*

```
if (expression)
{
    instructions1;
} sinon
{
    instructions2;
}
```

expression est évaluée après chaque itération. Si le résultat est vrai alors les instructions1 sont exécutées sinon on exécute l'ensemble instructions2

- exemple

```
int i =5;int n;
if (i<=20)
    n =0;
else
    n=5;
```


Introduction au Langage C,C++

Instructions de base

- instruction conditionnelle multiple
switch (expression)

```
{  
    case valeur1:  
        instructions1;break;  
    case valeur2:  
        instructions2;break;  
    .  
    .  
    case valeur3:  
        instruction3;break;  
    default:  
        instruction4;break;  
}
```

expression est évaluée. Si le résultat vaut *valeur1*, alors *instruction1* est exécutée et on quitte le switch, sinon si *expression* vaut *valeur2*,, sinon on va dans *default* pour exécuter *instruction4*.

Introduction au Langage C,C++

* Instructions de base

■ instruction conditionnelle multiple

■ exemple

```
int c='a';
switch (c)
{
    case 'a':
        file();break;
    case 'b':
        save();break;
    case 'q':
        quitter();break;
    default:
        beep();break;
}
```

Introduction au Langage C,C++

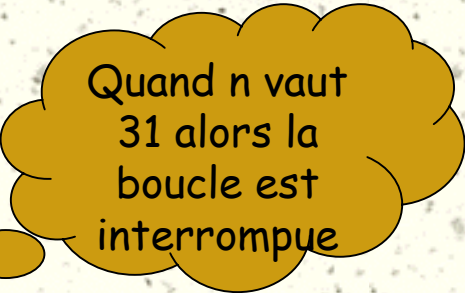
Instructions de base

■ l'instruction break

- permet d'interrompre prématurément une boucle et de se brancher vers la première instruction n'appartenant pas à la boucle

■ exemple:

```
int i;int n=20;  
for (i=0;i<10;i++)  
{  
    if (n==31) break;  
    n=n+2;  
}  
cout <<n<<' \n';
```



Quand n vaut
31 alors la
boucle est
interrompue