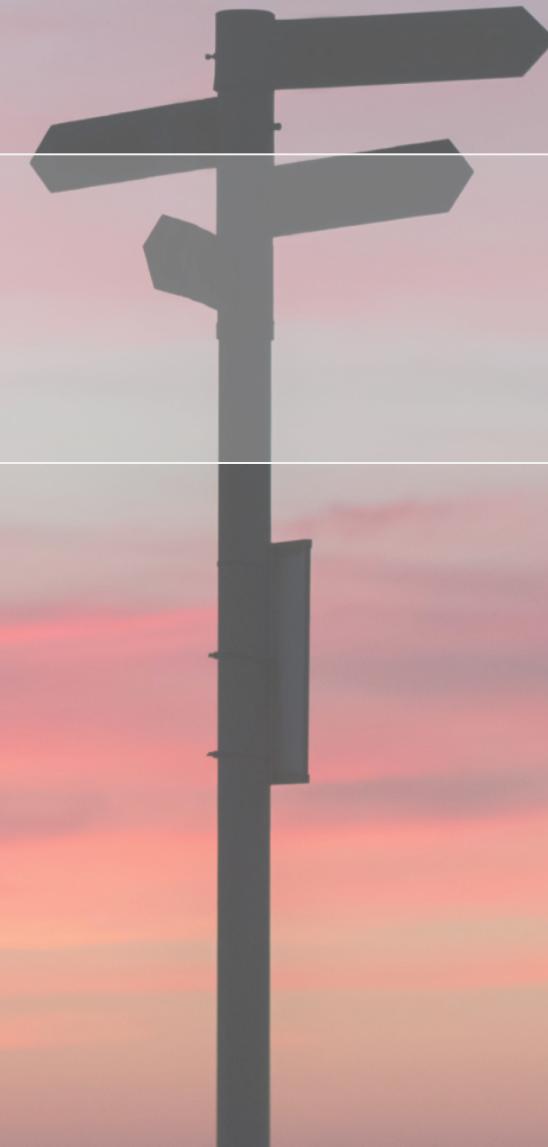


# Alternatives



# Objectifs

- Comprendre les structures de choix
- Connaitre leurs équivalences en C#

# Alternatives – si (<cond>) alors { ... }

- Vous avez besoin d'avoir la possibilité de choisir d'appliquer ou non un traitement
- Ce choix s'appuie sur une condition qui peut-être évaluée vraie ou fausse
- Exemple : demander un nombre entier à l'utilisateur, si le nombre est pair, écrire « Le nombre est pair »

```
entier nombreAValider = 0;  
nombreAValider = lire();  
si (nombreAValider % 2 == 0) alors  
{  
    écrireNL("Le nombre est pair");  
}
```

```
si (<conditionEstVrai>) alors  
{  
    <algorithmeSiConditionEstVrai>  
}
```

# Alternatives – si (<cond>) alors { ... } sinon { ... }

- Dans l'exemple précédent, nous voudrions peut-être écrire que le nombre est impair

```
entier nombreAValider = 0;  
nombreAValider = lire();  
si (nombreAValider % 2 == 0) alors  
{  
    écrireNL("Le nombre est pair");  
}  
sinon // donc ici cas  
// !(nombreAValider % 2 == 0)  
// <=> nombreAValider % 2 != 0  
{  
    écrireNL("Le nombre est impair");  
}
```

```
si (<conditionEstVraie>) alors  
{  
    <algorithmeSiConditionEstVraie>  
}  
sinon  
{  
    <algorithmeSiConditionEstFausse>  
}
```

# Alternatives - Imbrications

- Autre exemple : afficher la mention d'une évaluation notée avec une lettre en suivant le tableau suivant :

Évaluation	Mention
A	Excellent
B	Très bon
C	Bon
D	Passable
E	Échec (faible)
F	Échec (nul)

# Alternatives - Imbrications

Évaluation	Mention
A	Excellent
B	Très bon
C	Bon
D	Passable
E	Échec (faible)
F	Échec (nul)

```
caractère evaluation = ' ';  
evaluation = lire();  
si (evaluation == 'A') alors  
{  
    écrireNL("Excéllent");  
}  
sinon  
{  
    ???;  
}
```

# Alternatives - Imbrications

Évaluation	Mention
A	Excellent
B	Très bon
C	Bon
D	Passable
E	Échec (faible)
F	Échec (nul)

```
caractère evaluation = ' ';  
evaluation = lire();  
si (evaluation == 'A') alors  
{  
    écrireNL("Excéllent");  
}  
sinon  
{  
    si (evaluation == 'B') alors  
    {  
        écrireNL("Très bon");  
    }  
sinon  
    {  
        ???;  
    }  
}
```

# Alternatives - Imbrications

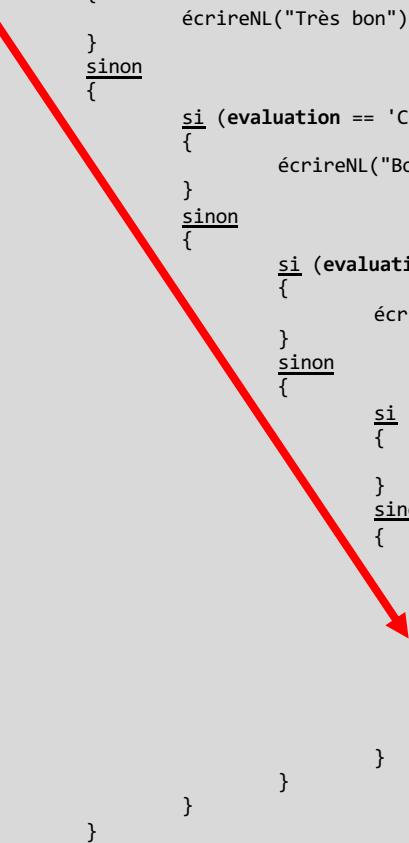
Évaluation	Mention
A	Excellent
B	Très bon
C	Bon
D	Passable
E	Échec (faible)
F	Échec (nul)

```
caractère evaluation = ' ';
evaluation = lire();
si (evaluation == 'A') alors
{
    écrireNL("Excéllent");
}
sinon
{
    si (evaluation == 'B') alors
    {
        écrireNL("Très bon");
    }
    sinon
    {
        si (evaluation == 'C') alors
        {
            écrireNL("Bon");
        }
        sinon
        {
            <...>;
        }
    }
}
```

# Alternatives - Imbrications

Évaluation	Mention
A	Excellent
B	Très bon
C	Bon
D	Passable
E	Échec (faible)
F	Échec (nul)

```
caractère evaluation = ' ';
evaluation = lire();
si (evaluation == 'A') alors
{
    écrireNL("Excéllent");
}
sinon
{
    si (evaluation == 'B') alors
    {
        écrireNL("Très bon");
    }
    sinon
    {
        si (evaluation == 'C') alors
        {
            écrireNL("Bon");
        }
        sinon
        {
            si (evaluation == 'D') alors
            {
                écrireNL("Passable");
            }
            sinon
            {
                si (evaluation == 'E') alors
                {
                    écrireNL("Échec (faible)");
                }
                sinon
                {
                    si (evaluation == 'F') alors
                    {
                        écrireNL("Échec (nul)");
                    }
                    sinon
                    {
                        écrireNL("Lettre d'évaluation inconnue");
                    }
                }
            }
        }
    }
}
```



Deviens difficile à lire !

# Alternatives – Imbrications – Autre syntaxe

Évaluation	Mention
A	Excellent
B	Très bon
C	Bon
D	Passable
E	Échec (faible)
F	Échec (nul)

```
caractère evaluation = ' ';
evaluation = lire();
si (evaluation == 'A') alors
{
    écrireNL("Excéllent");
}
sinon si (evaluation == 'B') alors
{
    écrireNL("Très bon");
}
sinon si (evaluation == 'C') alors
{
    écrireNL("Bon");
}
sinon si (evaluation == 'D') alors
{
    écrireNL("Passable");
}
sinon si (evaluation == 'E') alors
{
    écrireNL("Échec (faible)");
}
sinon si (evaluation == 'F') alors
{
    écrireNL("Échec (nul)");
}
sinon
{
    écrireNL("Lettre d'évaluation inconnue");
}
```

# Alternatives – selon (<valeur>) { cas <constante> : ... sinon : ... }

Évaluation	Mention
A	Excellent
B	Très bon
C	Bon
D	Passable
E	Échec (faible)
F	Échec (nul)

```
caractère evaluation = ' ';  
evaluation = lire();  
selon (evaluation)  
{  
    cas 'A':  
        écrireNL("Excéllent");  
    cas 'B' :  
        écrireNL("Très bon");  
    cas 'C' :  
        écrireNL("Bon");  
    cas 'D' :  
        écrireNL("Passable");  
    cas 'E' :  
        écrireNL("Échec (faible)");  
    cas 'F' :  
        écrireNL("Échec (nul)");  
    sinon :  
        écrireNL("Lettre d'évaluation inconnue");  
}
```

```
selon (<valeur>)  
{  
    cas <constante1>:  
        <algorithmePourConstante1>  
    cas <constante2>:  
        <algorithmePourConstante2>  
    [...]  
    cas <constanteN>:  
        <algorithmePourConstanteN>  
    sinon :  
        <algorithmePourAutreCas>  
}
```

# Équivalence C# – si (<cond>) alors { ... }

```
entier nombreAValider = 0;  
nombreAValider = lire();  
si (nombreAValider % 2 == 0) alors  
{  
    écrireNL("Le nombre est pair");  
}
```

```
int nombreAValider = 0;  
nombreAValider = Console.In.ReadInt();  
if (nombreAValider % 2 == 0)  
{  
    Console.Out.WriteLine("Le nombre est pair");  
}
```

# Équivalence C# – si (<cond>) alors { ... }

```
entier nombreAValider = 0;  
nombreAValider = lire();  
si (nombreAValider % 2 == 0) alors  
{  
    écrireNL("Le nombre est pair");  
}  
sinon  
{  
    écrireNL("Le nombre est impair");  
}
```

```
int nombreAValider = 0;  
nombreAValider = Console.In.ReadInt();  
if (nombreAValider % 2 == 0)  
{  
    Console.Out.WriteLine("Le nombre est pair");  
}  
else  
{  
    Console.Out.WriteLine("Le nombre est impair");  
}
```

# Équivalence C# – Imbrications – Autre syntaxe

```
caractère evaluation = ' ';
evaluation = lire();
si (evaluation == 'A') alors
{
    écrireNL("Excéllent");
}
sinon si (evaluation == 'B') alors
{
    écrireNL("Très bon");
}
sinon si (evaluation == 'C') alors
{
    écrireNL("Bon");
}
sinon si (evaluation == 'D') alors
{
    écrireNL("Passable");
}
sinon si (evaluation == 'E') alors
{
    écrireNL("Échec (faible)");
}
sinon si (evaluation == 'F') alors
{
    écrireNL("Échec (nul)");
}
sinon
{
    écrireNL("Lettre d'évaluation inconnue");
}
```

Évaluation	Mention
A	Excellent
B	Très bon
C	Bon
D	Passable
E	Échec (faible)
F	Échec (nul)

```
char evaluation = ' ';
evaluation = Console.In.ReadChar();
if (evaluation == 'A')
{
    Console.Out.WriteLine("Excéllent");
}
else if (evaluation == 'B')
{
    Console.Out.WriteLine("Très bon");
}
else if (evaluation == 'C')
{
    Console.Out.WriteLine("Bon");
}
else if (evaluation == 'D')
{
    Console.Out.WriteLine("Passable");
}
else if (evaluation == 'E')
{
    Console.Out.WriteLine("Échec (faible)");
}
else if (evaluation == 'F')
{
    Console.Out.WriteLine("Échec (nul)");
}
else
{
    Console.Out.WriteLine("Lettre d'évaluation inconnue");
}
```

# Équivalence C# –

## selon (<valeur>) { cas <constante> : ... sinon : ... }

```
caractère evaluation = ' ';
evaluation = lire();
selon (evaluation)
{
    cas 'A':
        écrireNL("Excéllent");

    cas 'B' :
        écrireNL("Très bon");

    cas 'C' :
        écrireNL("Bon");

    cas 'D' :
        écrireNL("Passable");

    cas 'E' :
        écrireNL("Échec (faible)");

    cas 'F' :
        écrireNL("Échec (nul)");

    sinon :
        écrireNL("Lettre d'évaluation inconnue");
}
```

Évalutation	Mention
A	Excellent
B	Très bon
C	Bon
D	Passable
E	Échec (faible)
F	Échec (nul)

```
char evaluation = ' ';
evaluation = Console.In.ReadChar();
switch (evaluation)
{
    case 'A':
        Console.Out.WriteLine("Excéllent");
        break;
    case 'B':
        Console.Out.WriteLine("Très bon");
        break;
    case 'C':
        Console.Out.WriteLine("Bon");
        break;
    case 'D':
        Console.Out.WriteLine("Passable");
        break;
    case 'E':
        Console.Out.WriteLine("Échec (faible)");
        break;
    case 'F':
        Console.Out.WriteLine("Échec (nul)");
        break;
    default:
        Console.Out.WriteLine("Lettre d'évaluation inconnue");
        break;
}
```