



# Les boucles

Programmation de base sur PC



# Plan du cours

- Mise-en-situation
- La boucle « do... while »
- La boucle « while »
- La boucle « for »

# Les répétitions

- Prenons un exemple de la vie de tout les jours; qui veut du café ?

```
String input = "";  
  
Console.WriteLine("Voulez-vous un café [O/N] ?");  
input = Console.ReadLine();  
  
if(input != "O" && input != "N") {  
    Console.WriteLine("Mauvais choix, veuillez recommencer");  
}
```

- Nous affichons un message d'erreur l'utilisateur entre un choix invalide.
- Mais comment poursuivre ce code en cas d'erreur ?

# Les répétitions

- On continue le dans le SI en demandant de nouveau?

```
String input = "";

Console.WriteLine("Voulez-vous un café [O/N] ?");
input = Console.ReadLine();

if(input != "O" && input != "N") {
    Console.WriteLine("Mauvais choix, veuillez recommencer");
    input = Console.ReadLine();
    if(input != "O" && input != "N") {
        Console.WriteLine("Mauvais choix, veuillez recommencer");
    }
}
```

On demande à nouveau ?



- Que faire si l'utilisateur se trompe encore ?



# Les répétitions

```
//[..]  
if(input != "O" && input != "N") {  
    Console.WriteLine("Mauvais choix, veuillez recommencer");  
    input = Console.ReadLine();  
    if(input != "O" && input != "N") {  
        Console.WriteLine("Mauvais choix, veuillez recommencer");  
        input = Console.ReadLine();  
        if(input != "O" && input != "N") {  
            Console.WriteLine("Mauvais choix, veuillez recommencer");  
            // [... à l'infini...]  
        }  
    }  
}
```

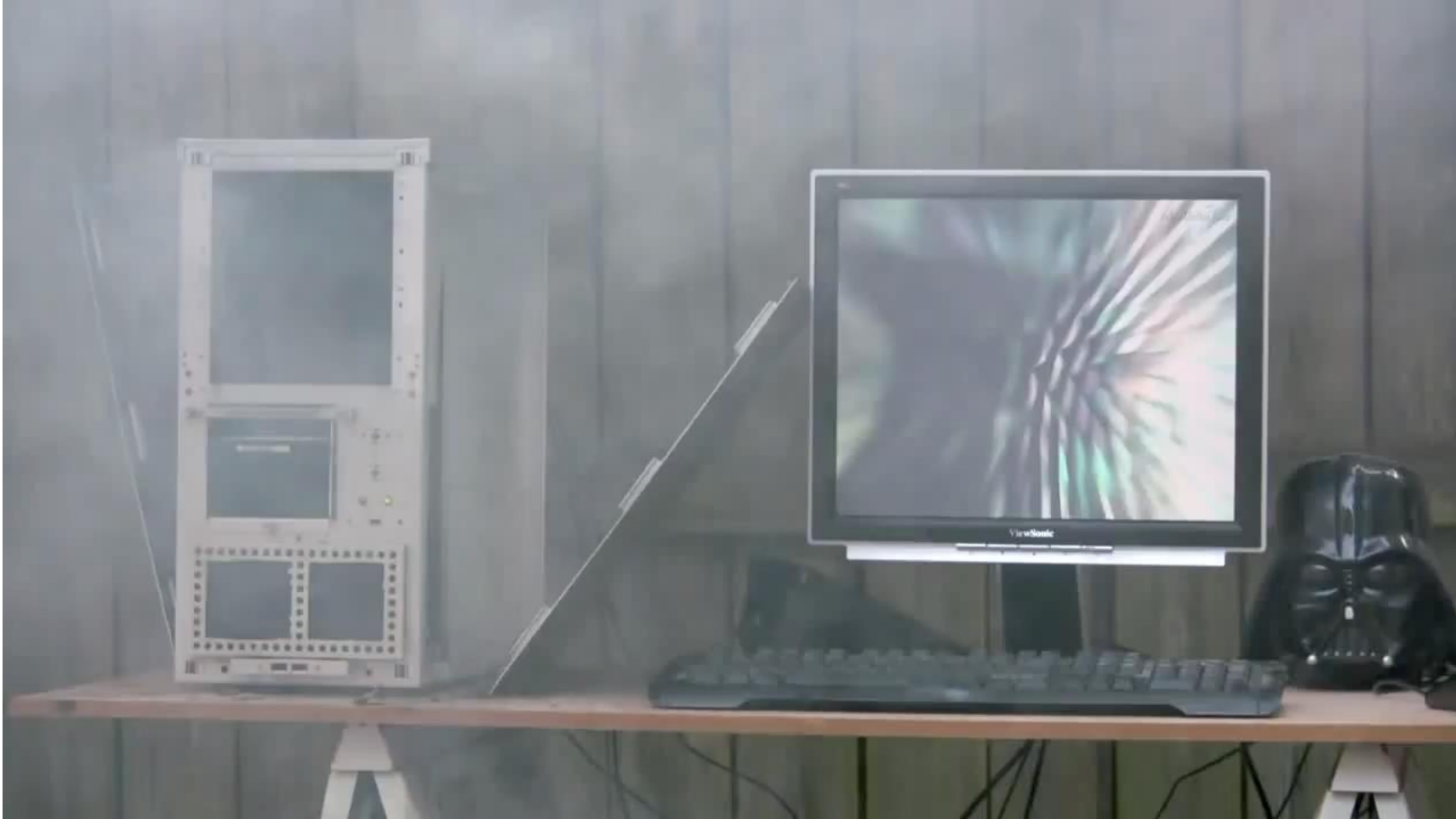
On demande à nouveau ?

On demande à nouveau ?

On demande à nouveau ?

ON EXPLOSE L'ORDI ?

# On explose l'ordi ?





# La boucle « do...while »

Syntaxe

Fonctionnement

Exemples

# La boucle do...while - Syntaxe

- La boucle do...while est une structure répétitive, dont les instructions sont exécutées avant même de tester la condition d'exécution de la boucle.
- S'écrit de la façon suivante :

```
do
{
/*
Plusieurs instructions ici
*/
} while( condition );
```



# Principes de fonctionnement

- Ainsi décrite, la boucle `do...while` s'exécute selon les principes suivants :
  - Les instructions situées à l'intérieur de la boucle sont exécutées tant que l'expression conditionnelle placée entre parenthèses() est vraie
  - Les instructions sont exécutées au moins une fois
  - Si la condition mentionnée entre parenthèses reste toujours vraie, les instructions de la boucle sont répétées à l'infini.
    - On dit que le programme « boucle ».
  - Une instruction modifiant le résultat du test de sortie de boucle est placée à l'intérieur de la boucle, de façon à stopper les répétitions au moment souhaité.
  - Observons qu'un point-virgule est placé à la fin de l'instruction
    - `while (expression);`

# La boucle do...while - Exemple

- L'exemple suivant va demander à l'utilisateur s'il désire quitter le programme et ce, jusqu'à ce qu'un O (majuscule) soit entré.
- S'écrit de la façon suivante :

```
String input = "";  
  
do  
{  
    Console.WriteLine("Voulez-vous quitter le programme ? [O/N]");  
    input = Console.ReadLine();  
} while( input != "O");
```



# La boucle do...while – Exemple de menu

```
int input = 0;

do
{
    Console.WriteLine("-----");
    Console.WriteLine(" | [0] - Ajouter une voiture. |");
    Console.WriteLine(" | [1] - Modifier une voiture. |");
    Console.WriteLine(" | [3] - Supprimer une voiture. |");
    Console.WriteLine(" | [4] - Quitter le programme. |");
    Console.WriteLine("-----");

    do {
        Console.Write("Entrez votre choix [1..4] : ");
        input = int.Parse(Console.ReadLine());
    } while( input < 0 || input > 4);

    //Code ici selon le menu choisi

} while( input != 4);
```



La boucle « while »



# La boucle while - Syntaxe

- Structure répétitive, ressemblant à la boucle « do...while », dont la décision de poursuivre la répétition s'effectue en début de boucle
- La syntaxe est la suivante :

```
while(condition) {  
    /*  
    Plusieurs instructions ici  
    */  
}
```

# Principes de fonctionnement

- Le terme **while** se traduit par **tant que**.
- La structure répétitive s'exécute selon les principes suivants :
  - Tant que l'expression à l'intérieur des parenthèses reste vraie, la ou les instructions composant la boucle sont exécutées
  - Le programme sort de la boucle dès que l'expression à l'intérieur des parenthèses devient fausse
  - Une instruction est placée à l'intérieur de la boucle pour modifier le résultat du test à l'entrée de la boucle, de façon à stopper les répétitions
  - Si l'expression à l'intérieur des parenthèses est fausse dès le départ, les instructions ne sont jamais exécutées.
  - Observons qu'à l'inverse de la boucle `do...while`, il n'y a pas de point-virgule à la fin de l'instruction `while (expression)`



# La boucle while - Exemple

- L'exemple ici permet de compter jusqu'à 10
- La syntaxe est la suivante :

```
int count = 1;

Console.WriteLine("Je compte jusqu'à 10 :");
while (count <= 10)
{
    Console.Write($"{count} ");
    count = count + 1;
}

Console.WriteLine("\nJe suis bon hein ?");
```

The background of the slide is a dark blue gradient filled with a pattern of binary code (0s and 1s) in a lighter blue color. The digits are of varying sizes and are slightly tilted, creating a sense of depth and movement. A large, white rectangular frame is centered on the slide, enclosing the text.

La boucle « for »



# La boucle for

- L'instruction `for` permet d'écrire des boucles dont on connaît à l'avance le nombre d'itérations (de tours) à exécuter :

```
for( initialisation; condition; incrément )  
{  
    /*  
    Plusieurs instructions ici  
    */  
}
```

```
for( int i = 0; i < 10; i++ ) {  
    /*  
    Plusieurs instructions ici  
    */  
}
```

# Principes de fonctionnement

- Les boucles « for » réalisent un nombre précis de boucles dépendant de la valeur initiale, de la valeur finale et du pas d'incrément.

<code>int i;</code> <code>char c;</code>	Valeur initiale	Valeur finale	Pas d'incrément	Nombre de boucles	Valeurs prises par <code>i</code> ou <code>c</code>
<code>for (i = 0; i &lt; 5; i = i + 1)</code>	0	4	1	5	0, 1, 2, 3, 4
<code>for (i = 4; i &lt;= 12; i = i + 2)</code>	4	12	2	5	4, 6, 8, 10, 12
<code>for (c = 'a'; c &lt; 'f'; c = c + 1)</code>	'a'	'e'	1	5	a, b, c, d, e
<code>for (i = 5; i &gt; 0 ; i = i - 1)</code>	5	1	- 1	5	5, 4, 3, 2, 1





# Exercices

# La boucle for

- L'objectif de ce programme est d'afficher les 256 premiers caractères ASCII

```
char character;  
  
for (int i = 0; i < 256; i++)  
{  
    character = (char)i;  
    Console.WriteLine($"{i} { character }");  
}
```

- N.B. Les caractères spéciaux n'ont pas de représentation textuelle