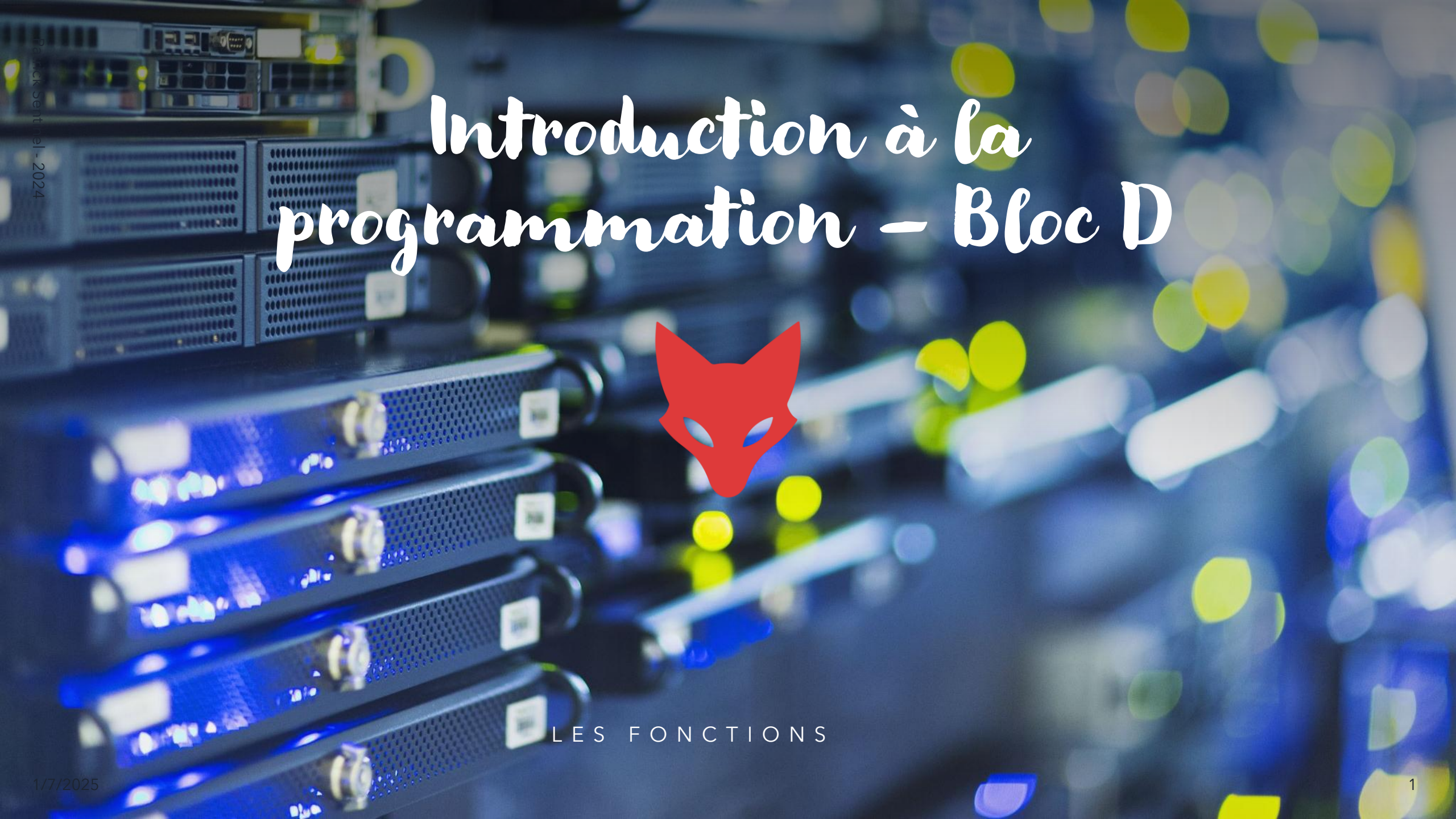

Introduction à la programmation – Bloc D

LES FONCTIONS

Appel de fonctions

```
int longueur = ObtenirLongueurTotal("Osbourne", "Cookie");

int ObtenirLongueurPrenom(string prenom){
        return prenom.Length;
}

int ObtenirLongueurnom(string nom) {
        return nom.Length;
}

int ObtenirLongueurTotal(string nom, string prenom){
        int totalNom = ObtenirLongueurnom(nom);
        int totalPrenom = ObtenirLongueurPrenom(prenom);
        return totalPrenom + totalNom;
}
```

Les valeurs sont copiées dans les paramètres de la fonction ObtenirLongueurTotal

Le résultat sera retourner à la variable longueur

Même résultat – formule différente

```
int longueurNom = ObtenirLongueurTotal("Osbourne", "Cookie");


int ObtenirLongueurPrenom(string prenom){
            return prenom.Length;
}


int ObtenirLongueurnom(string nom){
            return nom.Length;
}


int ObtenirLongueurTotal(string nom, string prenom){
            return ObtenirLongueurPrenom(prenom) + ObtenirLongueurnom(nom);
}
```

1- Appel de la fonction ObtenirLongueurTotal

2- Appel de la fonction ObtenirLongueurPrenom

3- Appel de la fonction ObtenirLongueurNom

Les noms de variables

```
string premierNom = "Blow";

string premierPrenom = "Joe";

string concat = ObtenirConcatenation(premierNom, premierPrenom);



string ObtenirConcatenation(string nom, string prenom){

        return nom + prenom;

}
```

Les valeurs sont passées par copies et les noms sont **indépendants les uns des autres entre les fonctions**

Les noms des variables en paramètre ne sont pas reliés aux noms des variables que la fonction appelle. En d'autres mots, les noms peuvent ÊTRE DIFFÉRENT.

Appels en cascades

```
int GetFirstNumber(){
            return 5;
}

int GetSecondNumber(){
            return 10;
}

int GetThirdNumber(){
            return 15;
}

void ShowTotal(){
            int total = GetFirstNumber() + GetSecondNumber() + GetThirdNumber();
            Console.WriteLine(total);
}
```

Considérons que le système appelle la fonction *ShowTotal*.

On peut appeler plusieurs fonctions en rafales

Même chose mais avec des Nesteds

```
int GetFirstNumber(){
            return 5;
}

int GetSecondNumber(){
            return GetFirstNumber() + 10;
}

int GetTotal(){
            return GetSecondNumber() + 15;
}

void ShowTotal(){
            int total = GetTotal();
            Console.WriteLine(total);
}
```

Considérons que le système appelle la fonction *ShowTotal*.

Même traitement mais avec des fonctions « Nesteds ».

1- ShowTotal appelle GetTotal
2- GetTotal appelle GetSecondNumber
3- GetSecondNumber appelle GetFirstNumber

Quelle approche préférez-vous ?

Fonctions dans les boucles

```
int total = 0;

for (int i=0; i<10; i++){

        total = total + AddOneToNumber(i);

}

int AddOneToNumber(int i){

        return i + 1;

}
```

Appel de fonction dans une boucle. À chaque itération, la fonction AddOneToNumber sera appelée

```
Console.WriteLine("Total : " + total);
```

```
Total : 55
```

Trace de l'algorithme précédent

Valeur de retour de AddOneToNumber

Premier appel 0 +1 = 1

Deuxième appel 1 + 1 = 2

Troisième appel 1 + 2 = 3

Quatrième appel 1 + 3 = 4

Cinquième appel 1 + 4 = 5

Sixième appel 1 + 5 = 6

Septième appel 1 + 6 = 7

Huitième appel 1 + 7 = 8

Neuvième appel 1 + 8 = 9

Dixième appel 1 + 9 = 10

L'addition

Premier appel total = 1

Deuxième appel 1 + 2 = 3

Quatrième appel 3 + 3 = 6

Cinquième appel 6 + 4 = 10

Sixième appel 10 + 5 = 15

Septième appel 15 + 6 = 21

Huitième appel 21 + 7 = 28

Neuvième appel 28 + 8 = 36

Dixième appel 36 + 9 = 45

Dixième appel 45 + 10 = 55

Petite question...

```
GetTotal();


void GetTotal(){
        int i = 10;
        i = i++;
        i = i + 10;
        i++;

        Console.WriteLine(i);
}
```

***Question piège ***

Quelle sera la valeur de 'i' à la fin du traitement ?

Petite question...

```
GetTotal2();


void GetTotal2(){
        int i = 10;
        int j = i++ + (20) + i--;
        Console.WriteLine(j);
}
```

***Question piège ***

Quelle sera la valeur de 'j' à la fin du traitement ?

Petite question...

```
variable();

void variable(){

        byte montant_1 = 234;

        short montant_2 = 34000;

        int total = montant_1 + montant_2;

        Console.WriteLine(total);

}
```

***Question piège ***

Quelle sera la valeur de 'total' à la fin du traitement ?

Petite question...

```
shift();


void shift(){
        byte montant_1 = 2;
        int result = 10 << montant_1;
        Console.WriteLine(result);
}
```

Quelle sera la valeur de ‘result’ à la fin du traitement ?

Petite question...

```
differenceBetween();

void differenceBetween(){

        bool isClown = true;

        bool isACat = true;

        bool isADog = true;


        if (isClown & isACat & isADog){

                Console.WriteLine("Bonjour");

        }

        if (isClown && isACat && isADog){

                Console.WriteLine("Allo");

        }

}
```

Quelle est la différence entre & et && ?

Qu'est-ce qui sera affiché à la console ?

Petite question...

```
unary();

void unary(){
        int i = 10;
        int j = --i;
        j = j + i + 20;
        Console.WriteLine(j);
}
```

Que sera la valeur de 'j' à la fin du traitement ?

Des questions ?

info@29a.ca