

OBJECTIFS

- Découvrir la documentation MSDN
- Distinguer méthode statique/ méthode d'instance
- Manipuler des objets

QUESTIONS

```
public static double Pow (double x, double y);
```

Paramètres

x : Nombre à virgule flottante double précision à élever à une puissance.

y : Nombre à virgule flottante double précision. qui spécifie une puissance.

1. On appelle la ligne surlignée : Une méthode statique
2. Pour élever 3 au carré Avec : `double a = 3` , `b = 2`; choisissez les instructions correctes et utiles:
 - V/F `double res = Math.Pow(double a, double b) ;`
 - V/F `double res = Math.Pow(b, a) ;`
 - V/F `double res = Math.Pow(a, b) ;`
 - V/F `Math.Pow(a, b) ;`
 - V/F `Console.WriteLine(Math.Pow(a, b));`
 - V/F `String txt = "3 à la puissance 2 = " + Math.Pow(a, b);`
 - V/F `Console.WriteLine(double res= Math.Pow(a, b));`
3. Avec : `int a = 3` , `b = 2`; On peut écrire :
 - V/F `double res = Math.Pow(a, b) ;`
 - V/F `int res = Math.Pow(a, b);`
4. On peut écrire :
 - V/F `double res = Math.Pow(3, 2) ;`
 - V/F `int res = Math.Pow(3, 2);`
 - V/F `int res = (int) Math.Pow(3, 2);`
5. Un peu de vocabulaire :
 - Les paramètres effectifs sont : _____
 - Les paramètres formels sont : _____
6. Déduisez la signature de la méthode à partir de l'utilisation suivante et indiquez la classe dans laquelle elle est définie : _____
`String res = Why.Methode("Nom", 2);`

```
public static _____ ( _____ );
```

```
public string Substring (int startIndex, int length);
```

7. Pour extraire la 1ere lettre d'un prénom : `string` prenom = "Leanne"; choisissez les instructions correctes et utiles:

0	1	2	3	4	5
L	e	a	n	n	e

- V/F `string` premiereLettre = `String.Substring(0, 1)` ;
- V/F `string` premiereLettre = `String.Substring(1, 2)` ;
- V/F `string` premiereLettre = `String.Substring(1, 1)` ;
- V/F `string` premiereLettre = `prenom.Substring(0, 1)` ;
- V/F `string` premiereLettre = `prenom.Substring(1, 2)` ;
- V/F `string` premiereLettre = `prenom.Substring(1, 1)` ;

8. Substring est :

- V/F une méthode statique
- V/F une méthode d'instance

```
public int IndexOf (string value);
```

Returns

`Int32`

The zero-based index position of `value` if that string is found, or -1 if it is not. If

`value` is Empty, the return value is 0.

```
string prenom = "Leanne";
```

- `int res = prenom.IndexOf("a")` ; Que vaut res ? _____
- `int res = prenom.IndexOf("n")` ; Que vaut res ? _____
- `int res = prenom.IndexOf("z")` ; Que vaut res ? _____
- `int res = prenom.IndexOf("")` ; Que vaut res ? _____

EXO 1 : MANIPULER UN RECTANGLE

Vous manipulerez ici des variables structurées (des rectangles), c'est comme des objets à la différence que ce sont des types valeurs **tout comme une variable primitive**. La copie se fait par valeur. Le passage en paramètre se fait aussi par valeur... Ce qui n'est pas le cas pour un objet. Ex :

```
Rectangle r1 = new Rectangle(0, 0, 200, 20);
```

```
Rectangle r2 = r1; // ici, c'est une copie de champs et non d'adresse
```

```
r2.X = 10; // en modifiant r2, je ne modifie pas r1
```

1. Dans un moteur de recherche, tapez MSDN Rectangle.
2. Observons la documentation MSDN de la structure Rectangle.

Using System.Drawing ;

- Elle est rangée dans **l'espace de nom** : [System.Drawing](#).
- Intéressons-nous aux 4 sections suivantes :
 - i. **Constructeurs** : méthodes pour initialiser un objet Rectangle
 - ii. **Champs/ Propriétés** : infos stockées dans un objet Rectangle
 - iii. **Méthodes** : traitement dédiés aux objets Rectangle

▼ Rectangle

Rectangle

Constructeurs

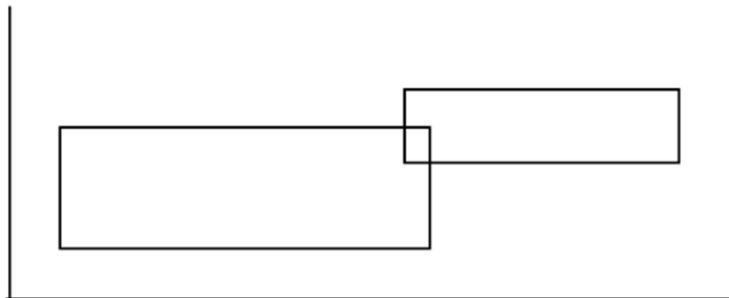
> Champs

> Propriétés

> Méthodes

3. Combien y a-t-il de constructeurs ? Comment appelle t'on cela ?

4. En observant les constructeurs, déduisez combien d'infos pourraient stockées dans un rectangle ?
5. Observez le nombre de propriétés.
6. Au sein du répertoire « **R1_01_InitiationDev_C#** », créez un répertoire :
« **Sequence_3_Methodes** »
7. Démarrez visual studio.
8. Créez un projet « **Exo1_Rectangle** » de type console dans une solution nommée « **Seance1_Utilisation_Methodes** ».
9. Créez 2 objets rectangles : r1 et r2 : dans cet exercice : les rectangles ne sont pas graphiques. Vous trouverez ci-dessous une représentation graphique pour vous aider à vous les représenter.
 - 1^{er} rectangle r1 avec le **constructeur à 4 paramètres** : positionné en (20,20) largeur de 100 et hauteur de 40
 - 2^{ème} rectangle r2 avec le constructeur à 2 paramètres : vous serez obligé de créer des objets Point et Size : positionné en (110,50) largeur de 80 et hauteur de 20



10. Affichez-les avec Console.WriteLine.
11. Observez les méthodes et plus précisément **IntersectWith**. **Est-elle statique ?** Utilisez la méthode **IntersectWith** et affichez le résultat : **Intersection: True**
12. Observez la méthode **Intersect** : elle est **surchargée** : utilisez la version qui renvoie le rectangle d'intersection. **Est-elle statique ?** Affichez le résultat et observez à quoi cela correspond.
13. Observez la méthode **Contains**. Elle est surchargée.
 - Utilisez la méthode Contains pour afficher si r2 est contenu dans r1.
 - Déterminez un point qui vous semble contenu dans r1 puis utilisez la méthode Contains pour tester si c'est bien le cas.

EXO 2 : MANIPULER UN STRING - NUMERO DE SECU

1. Ajoutez un projet « **Exo2_NumeroSecu** ».
2. Votre programme doit demander à l'utilisateur son numéro de sécu : les 7ers numéros. Il doit à partir de celui-ci afficher le mois, l'année et le département de naissance de l'utilisateur. Utilisez la méthode : **Substring** de la classe String. Tapez MSDN String dans un moteur de recherche puis ouvrez toutes les méthodes : observez Substring : est elle statique ?

Donnée en entrée	Résultat attendu		
NumSecu	Mois	Annee	departement
2851074	10	85	74

```
Num secu ?
2851074
Vous etes nee en 10/85 dans le departement 74
```

Pour info : Votre numéro de sécurité sociale est composé de 15 chiffres

- le premier: 1(homme) ou 2(femme) pour le sexe
- les deux suivants: l'année de naissance
- les deux suivants: le mois de naissance
- les deux suivants: le département
- les trois suivants: la ville de naissance
- les trois suivants: numéro d'ordre de naissance
- les 2 derniers : c'est une cle

EXO 3 : MANIPULER UN STRING - MAIL

1. Ajoutez un projet « **Exo3_Mail** ».
2. Votre programme doit demander à l'utilisateur son adresse email puis il doit afficher : Utilisez les **méthodes Substring, IndexOf, LastIndexOf** et la **propriété Length** !

```
public int IndexOf (string value);
public int LastIndexOf (string value);
public string Substring (int startIndex);
public string Substring (int startIndex, int length);
```

- la longueur de la chaîne,
- la position de @ dans cette variable.
- le domaine de cette adresse.
- l'extension de cette adresse.

Donnée en entrée	Résultat attendu			
email	Longueur	positionArrobas	domaine	extension
nat@gmail.com	13	3	gmail	com

EXO 4 : MANIPULER UN DATETIME – DATE DE NAISSANCE

1. Ajoutez un projet « **Exo4_DateTime** ».
2. Observez la classe DateTime et plus particulièrement les constructeurs : vous utiliserez la 3^{ème} surcharge attendant 3 paramètres.
3. Demandez à l'utilisateur (vous) les 3 infos pour créer un objet DateTime stockant votre date de naissance. Puis affichez-le.
4. Observez la **méthode ToLongDateString**. Est-elle statique ? utilisez la et affichez son retour.

Avez-vous utilisé : int.Parse ou Int32.Parse ? à la place utilisez **int.TryParse** ou **Int32.TryParse**. Commettez des erreurs de saisie (des lettres au lieu des chiffres) et observez. Cela plante t'il ?

https://docs.microsoft.com/fr-fr/dotnet/api/system.int32.tryparse?view=net-5.0#System.Int32.TryParse_System.String_System.Int32

5. Observez la méthode AddYears : utilisez là pour indiquer le jour de votre 20ème anniversaire au format long pour ainsi savoir quel jour ça tombe.
6. Observez la **propriété Today** : elle est statique. C'est comme pour les méthodes statiques : pas besoin d'objet, on s'appuie sur la classe. Affichez la date du jour.
7. Affichez le nombre de jour du mois en cours : utilisez la **méthode DaysInMonth** . Observez la documentation : attention, elle est statique, vous lui passerez les informations contenues dans Today.
8. Observez la méthode TryParse pour instancier la date de naissance de votre voisin. Attention : elle est statique.
9. Au sein de la documentation, observez les opérateurs : est-il possible de comparer 2 dates avec < > ?
10. Il est possible de soustraire 2 dates : observez la documentation : cela renvoie un objet TimeSpan. Faites-le pour savoir combien de temps vous avez passé sur terre. Affichez le nombre d'heures puis le total exprimé en heure (cf doc)