

Introduction au framework .NET

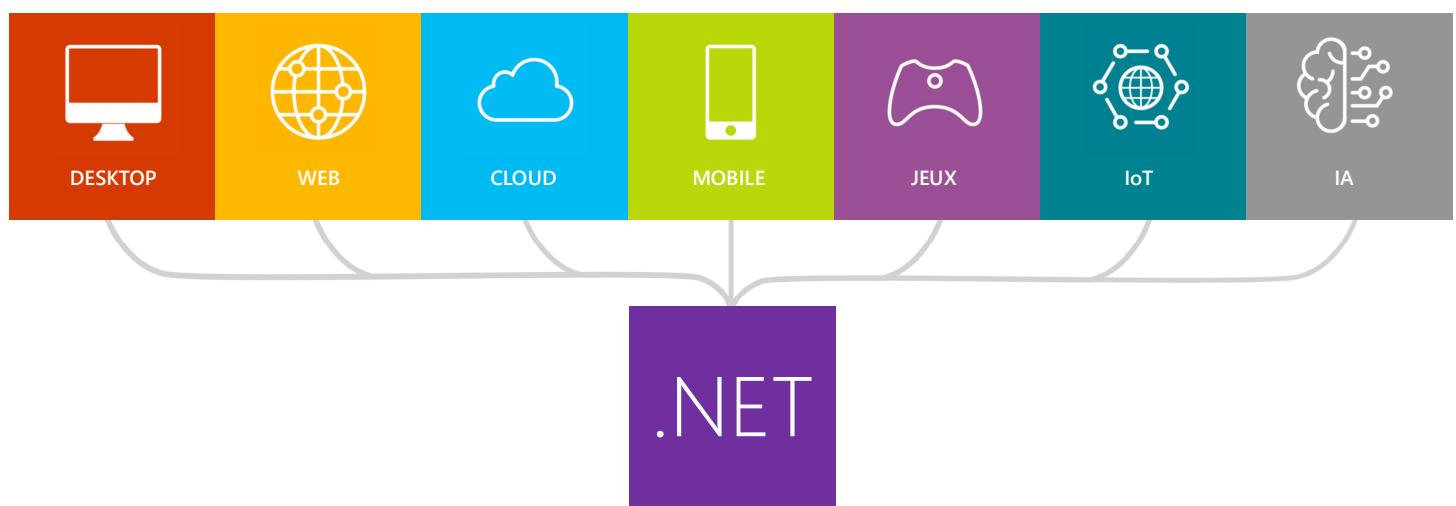
Dr. OUSSAMA BEN KHIROUN

✉ oussamabk.info@gmail.com 🌐 www.oussamabenkhiroun.com

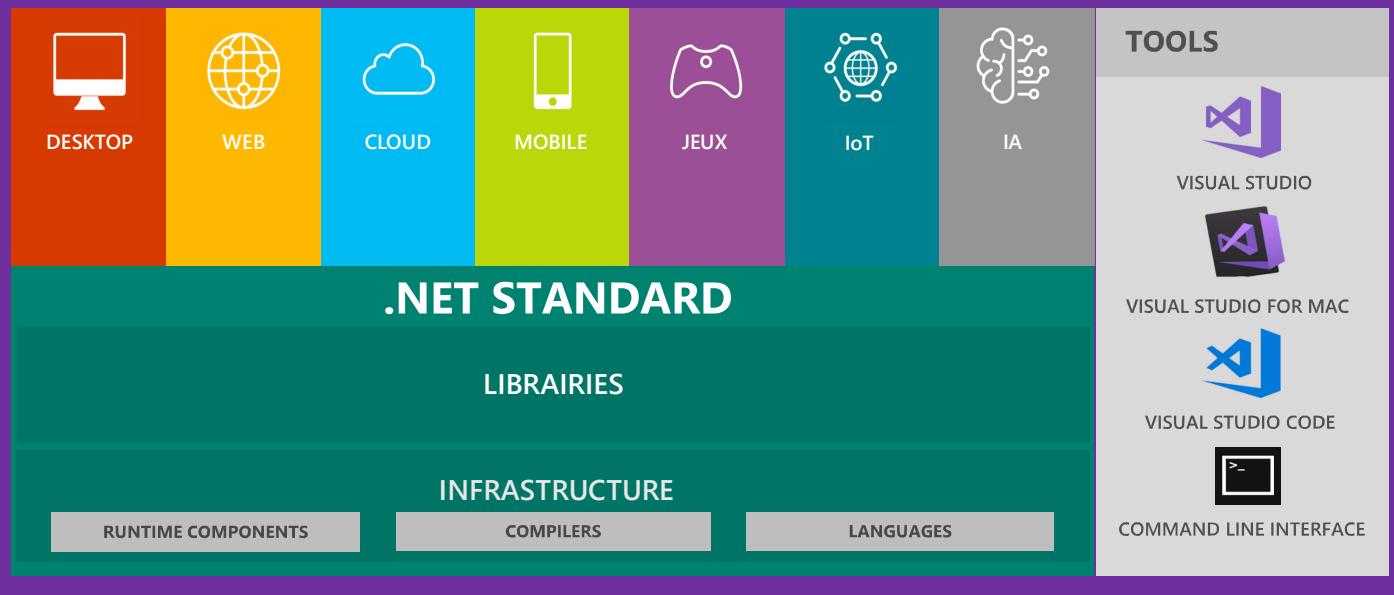
⌚ <https://github.com/oussbenk>



Une seule plateforme pour tout construire



.NET - plateforme de développement logiciel



Plusieurs langages supportés par .NET

C# (c-sharp)

- C# est un langage de programmation simple, moderne, orienté objet et typé
 - Ses origines de la famille du langage C rendent C# directement familier à C, C++, Java et JavaScript

```
var names = new List<String>
{
    "Ana",
    "Felipe",
    "Emillia"
};

foreach (var name in names)
{
    Console.WriteLine($"Hello {name}");
}
```

F# (f-sharp)

- F# est un langage de programmation pour .NET. Il est multiplateformes, fonctionnel et libre
 - Il inclut également l'aspect orienté objet et la programmation impérative

```
let numbers = [ 1; 2; 3; 4; 5; 6; 7; 8; 9; 10 ]

let square x = x * x
let isOdd x = x % 2 > 0

let squaresOfOdds =
    numbers
    |> List.filter isOdd
    |> List.map square

printfn "%A" squaresOfOdds
```

Visual Basic

- Visual Basic est un langage accessible avec une syntaxe simple pour créer des applications orientées objet et sécurisées.

```
Dim names As New List(Of String)({  
    "Ana",  
    "Felipe",  
    "Emilia"  
})  
  
For Each name In names  
    Console.WriteLine($"Hello {name}")  
Next
```

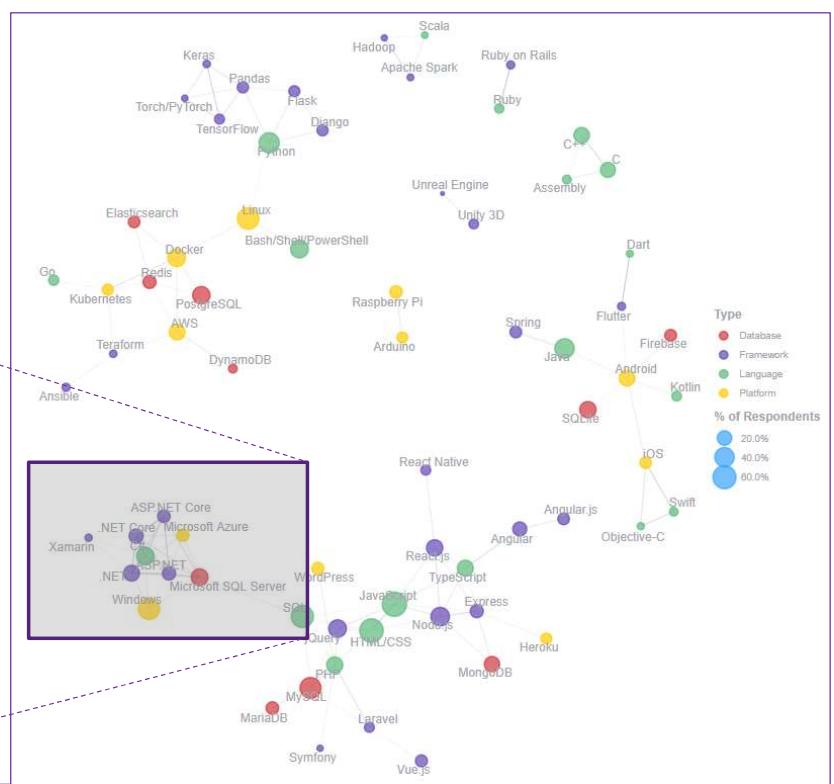
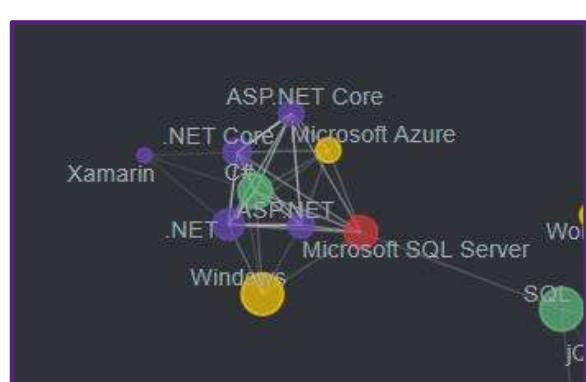
Classement des langages populaires sur TIOBE (mois de Février 2020)

Feb 2021	Feb 2020	Change	Programming Language	Ratings	Change
1	2	▲	C	16.34%	-0.43%
2	1	▼	Java	11.29%	-6.07%
3	3		Python	10.86%	+1.52%
4	4		C++	6.88%	+0.71%
5	5		C#	4.44%	-1.48%
6	6		Visual Basic	4.33%	-1.53%
7	7		JavaScript	2.27%	+0.21%
8	8		PHP	1.75%	-0.27%
9	9		SQL	1.72%	+0.20%

[Source: <https://www.tiobe.com/tiobe-index/>]



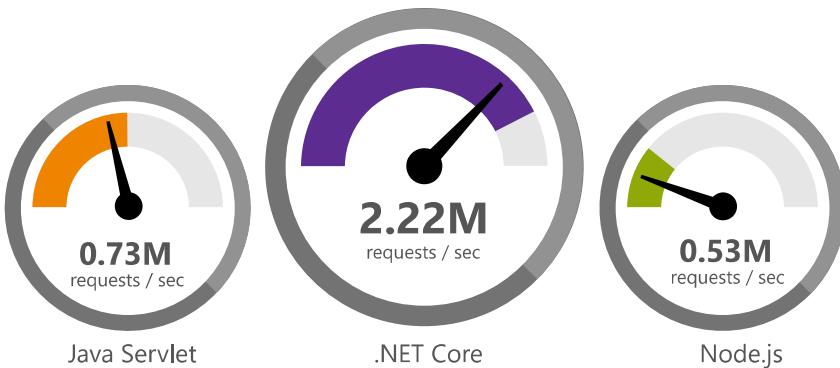
Ecosystème .NET



[Source: Stack Overflow 2020 survey]



.NET est rapide !



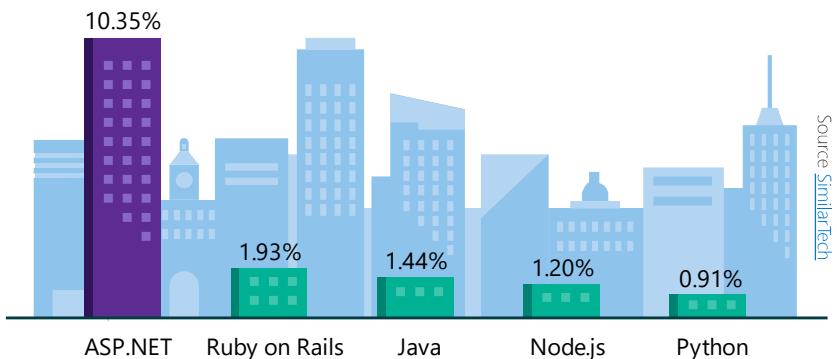
Data sourced from official tests available at [TechEmpower Round 15](#).

" Using the same-size server, we were able to go from 1,000 requests per second per node with Node.js to 20,000 requests per second with .NET Core. "



.NET est répandu !

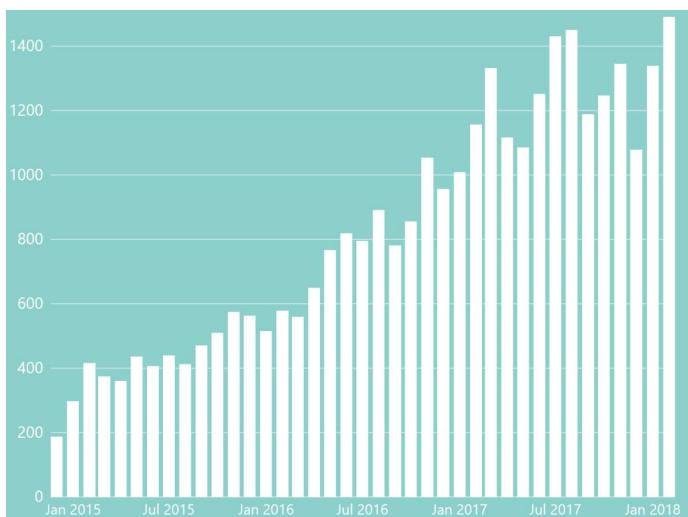
Plus de sites web ont été développés avec l'ASP.NET qu'en Rudy, Java, Python, Node.js et Go combinés.



PHP domine comme langage le plus utilisé dans les sites web existants



.NET est open source



"Samsung is embracing .NET because it is a completely open source project."

SAMSUNG

".NET is open source; that allows us to contribute back to it if we have any performance issues which Microsoft review and together we make a better product."

— Illyriad Games

>16 000 contributions communautaires issues de >3 000 entreprises autres que Microsoft



Dr. Oussama BEN KHIROUN | Framework .NET

9

Le voyage du .NET Libre



.NET 5

.NET Core 3

.NET

Aug 2017- .NET Core 2



2016- Le projet Mono rejoint le .NET Foundation



Nov 2014- .NET Core multiplateforme, Libre

April 2014- .NET Compiler Platform ("Roslyn") Libre
Lancement de la ".NET Foundation"

2008- ASP.NET MVC (plateforme Web) Libre

2002- .NET 1.0 publié pour Windows, le projet Mono commence (concurrent libre de .NET)

2001- ECMA-335



Dr. Oussama BEN KHIROUN | Framework .NET

10

Plus de lecture sur le sujet

Article sur blog personnel:
<https://oussamabenkhiroun.com/blog/>



The screenshot shows a blog post titled "How Microsoft aligned its tactic to empower the open source ecosystem?" by Oussama Ben Khiroun, posted on 11/02/2021. The post features a banner with Microsoft, .NET, Visual Studio, open source, and GitHub logos. The text discusses Microsoft's support for the open source ecosystem, mentioning 7 facts such as the .NET framework being open source and Visual Studio being offered freely.

Being away from the .NET development world (for about 6 years), I have been astonished with the great effort of **Microsoft to support the open source ecosystem** in the last years.

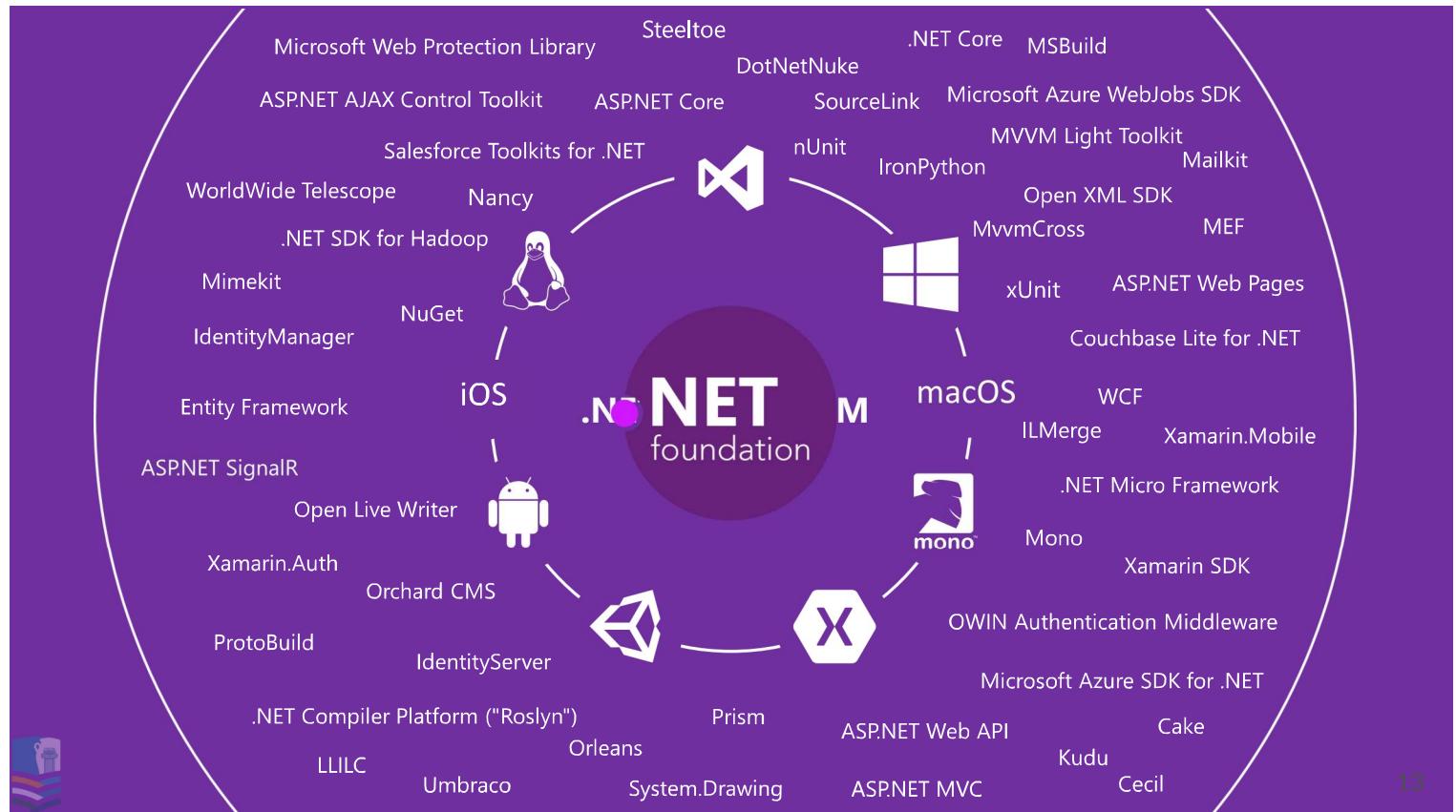
Here are **7 facts**:

- The .NET framework is totally being open source and **Visual Studio** (as an IDE) is **offered freely** for students and open source contributors
- Any .NET application could run without problems **on Windows, Linux and Mac OS** thanks to cross-platform capability
- The .NET foundation aims to assemble passionate developers to contribute to related .NET projects

11

"GitHub is where open source happens..."

The screenshot shows the GitHub interface. On the left, there is a large GitHub logo. The main area displays two repositories: ".NET Foundation" and "dotnet/corefx". The ".NET Foundation" repository page includes a summary card with "1,556" issues, "13,525" stars, and "3,854" forks. Below this, the "dotnet/corefx" repository page is shown, featuring a search bar with "is:pr is:open" and a list of pull requests. One pull request is highlighted with a green border, showing details like "Add GetHashCode test for invariant mode" and "NO MERGE". The right side of the screen shows the "corefx" repository details, including its purpose as the ".NET Core functional library for .NET applications", its codebase, and various files like "coreclr", "corefx", "standard", "orleans", and "docs". It also displays metrics like 16,273 closed issues, 2,197 open issues, and 51 pull requests.



13

Par où commencer ?



14

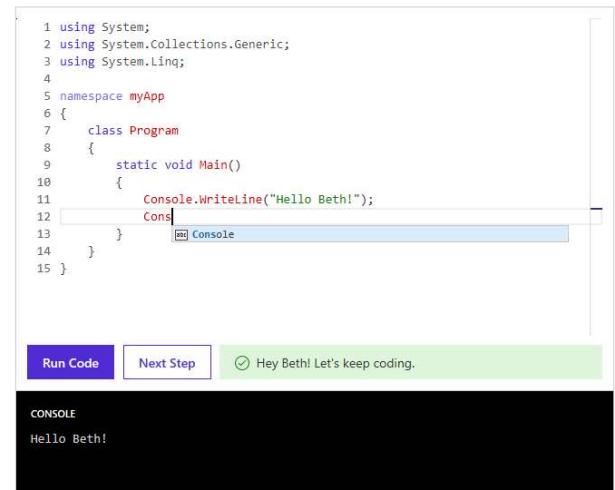
Apprendre à programmer en C#

www.dot.net



A screenshot of the .NET official website. At the top, there's a navigation bar with links like Microsoft, .NET, About, Learn, Architecture, Docs, Downloads, Community, LIVE TV, and All Microsoft. Below the navigation, there's a large purple banner with the .NET logo and the text "Free. Cross-platform. Open source. A developer platform for building IoT apps.". A yellow arrow points to the "Get Started" button. Below the banner, there are four cards: "Web" (Build web apps and services for Windows, Linux, macOS, and Docker), "Mobile" (Use a single codebase to build native mobile apps for iOS, Android, and Windows), "Desktop" (Create beautiful and compelling desktop apps for Windows and macOS), and "Microservices" (Create independently deployable microservices that run on Docker containers). A "Feedback" link is located on the right side of the cards.

In browser tutorial



A screenshot of an in-browser .NET tutorial. It shows a code editor with the following C# code:

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4
5 namespace MyApp
6 {
7     class Program
8     {
9         static void Main()
10        {
11            Console.WriteLine("Hello Beth!");
12        }
13    }
14 }
15 }
```

Below the code editor, there are "Run Code" and "Next Step" buttons. A green bar at the bottom says "Hey Beth! Let's keep coding." To the right, a "CONSOLE" window shows the output: "Hello Beth!".



Dr. Oussama BEN KHIROUN | Framework .NET

Outils et IDEs

www.VisualStudio.com



Visual Studio

S'exécute sur Windows uniquement. A des fonctionnalités intégrées étendues conçues pour fonctionner avec .NET. L'édition Community est gratuite pour les étudiants, les contributeurs Open source et les individus.



Visual Studio Code

S'exécute sur Windows, macOS et Linux. Gratuit et open source. Les extensions sont disponibles pour l'utilisation des langages .NET.



Visual Studio for Mac

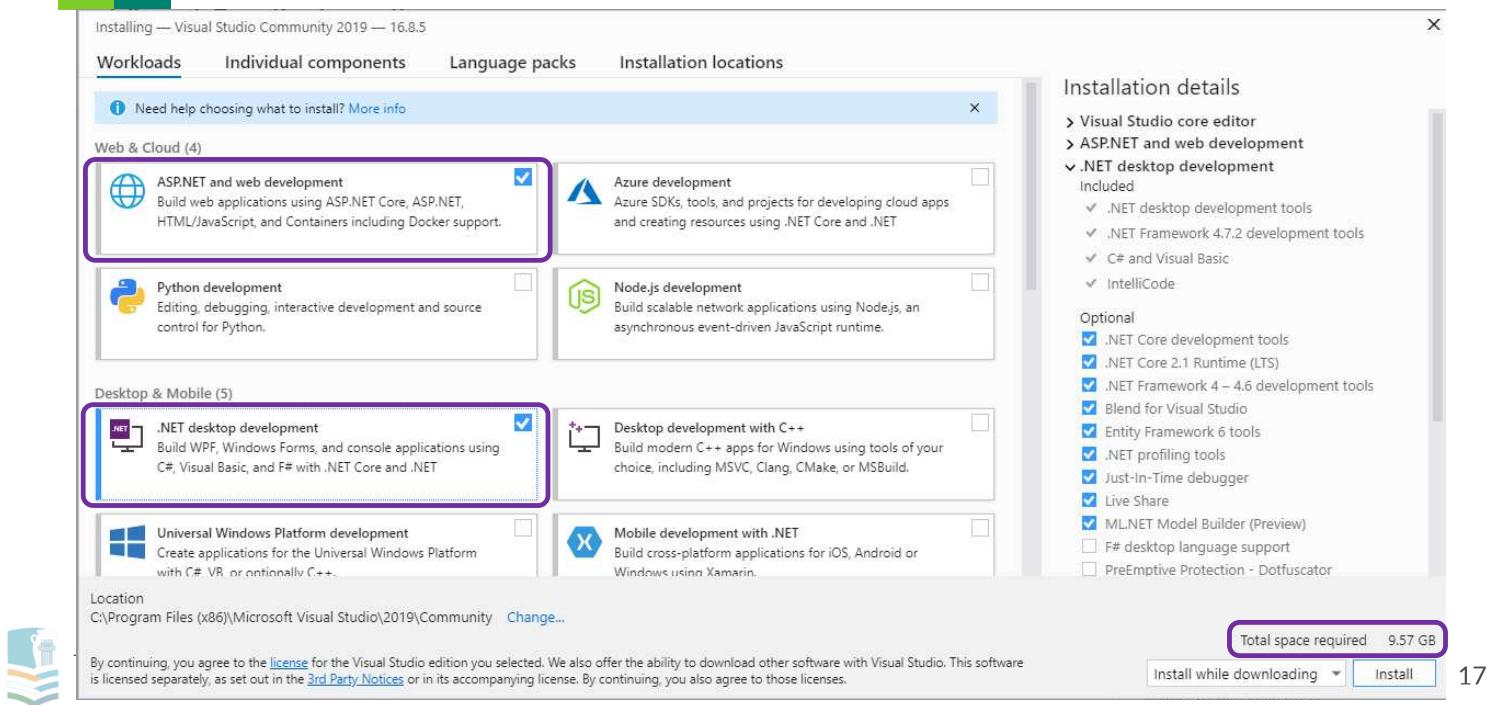
S'exécute sur macOS uniquement. Pour le développement d'applications et de jeux .NET pour iOS, Android et Web.



Dr. Oussama BEN KHIROUN | Framework .NET

16

Installation de Visual Studio 2019 (l'essentiel)



Autres outils .NET

- **MSBuild** : moteur de génération utilisé pour générer des projets
- **NuGet** : gestionnaire de package de Microsoft pour .NET 
- **CLI** : L'interface de ligne de commande .NET est un chaîne d'outils multiplateforme pour le développement, la génération, l'exécution et la publication d'applications .NET.
- Etc.

```
Microsoft (R) Build Engine Version 16.8.0
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\Derrek> dotnet --help
Usage: dotnet [host-options] [command] [arguments] [common-options]

Options:
  [command]
  [host-options]
  [arguments]
  [common-options]

Common options:
  --verbose          Enable verbose output
  -h --help           Show help
  host-options (passed before the command):
    --project         The project to build
    --configuration   Configuration to build (Default: Release)
    --runtime          The runtime to target
    --version         Display .NET CLI version number
    --info            Display .NET CLI information
  Common Commands:
    new              Initializes a basic .NET project
    restore          Restores dependencies specified in the .NET project
    build            Builds the project and its dependencies (using the runtime)
    publish          Publishes the project and its dependencies (using the runtime)
    run              Compiles and immediately executes a .NET application
    pack             Creates a NuGet package
    PS C:\Users\Derrek>
```

Essentiel du langage

Dr. OUSSAMA BEN KHIROUN

✉ oussamabk.info@gmail.com 🌐 www.oussamabenkhiroun.com

⌚ <https://github.com/oussbenk>



.NET



19

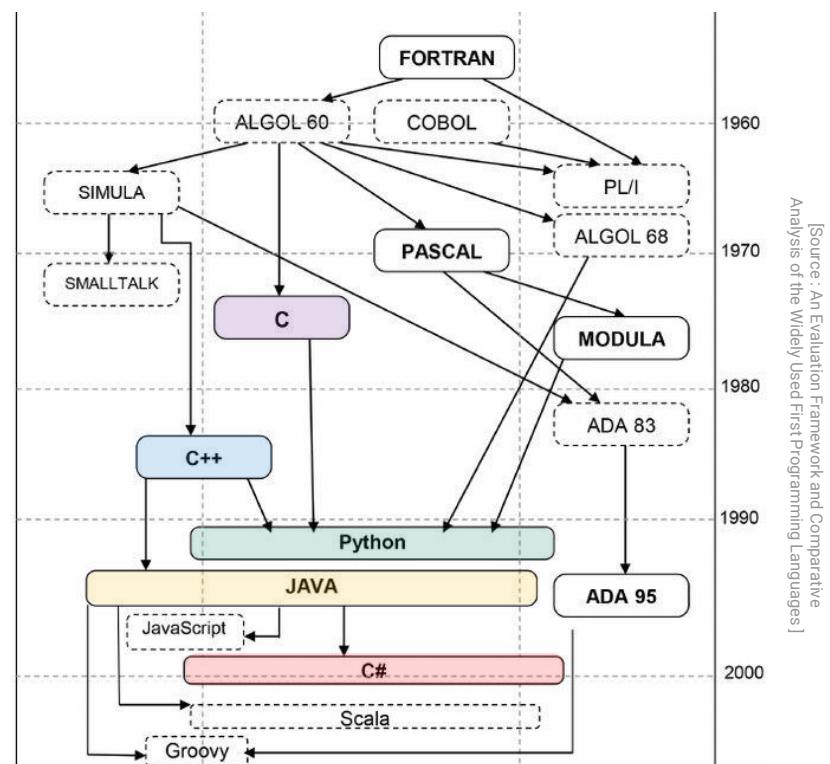
Origines de C# (1/2)

- C# se prononce "C-Sharp"
- Il s'agit d'un langage de programmation **orienté objet** créé par Microsoft qui s'exécute sur le Framework .NET
- La première version est sortie en 2001. La dernière version (sortie en sept. 2021) est la version C# 10.0
- L'extension des fichiers écrits en C# est **".cs"**



Origines de C# (2/2)

C# a des racines de la famille C, et il est proche d'autres langages populaires comme C++ et Java



Dr. Oussama BEN KHIROUN | Framework .NET

Hello World en C#

Fait référence à l'espace de noms System

Nom de la classe "Program"

```
using System;
namespace HelloWorld
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}
```



Définition de l'espace de noms HelloWorld pour la classe en cours

La sortie du programme est générée par la méthode WriteLine de la classe Console dans l'espace de noms System

Dr. Oussama BEN KHIROUN | Framework .NET

22

Conventions de dénomination (*naming convention*)

Quoi nommer ?	Style de dénomination
Class	MyClass
Method	MyMethod
Local variable	myLocalVariable
Private variable	_myPrivateVariable
Constant	MyConstant

Convention différente à celle dans Java



Types de valeur

Types simples

- Intégrale signé: **sbyte** , **short** , **int** , **long** (resp. 1 octet, 2 octets, 4 octets, 8 octets)
- Entier non signé: **byte** , **ushort** , **uint** , **ulong** (resp. 1 octet, 2 octets, 4 octets, 8 octets)
- Caractères Unicode: **char** (2 octets)
- Virgule flottante binaire IEEE: **float** , **double** (4 octets, 8 octets)
- Virgule flottante décimale haute précision : **decimal** (16 octets)
- Boolean : **bool** , qui représente des valeurs booléennes (**true** ou **false**) (1 octet)



Possibilité de vérifier la taille des types (ou des variables) avec l'opérateur **sizeof()**
Ex. `sizeof(int) // retourne 4`



Types référence

- Types de classe :
 - ✓ Classe de base fondamentale de tous les autres types : `object`
 - ✓ Chaînes Unicode : `string`, qui représente une séquence de chars
 - ✓ Types définis par l'utilisateur de la forme `class C {...}`
- Types d'interface : Types définis par l'utilisateur de la forme `interface I {...}`
- Types tableau : `Unidimensionnel`, `multidimensionnel` et `en escalier (jagged array)`.
Par exemple : `int[]`, `int[,]` et `int[][]`



Exemple Tableau (1/2)

- Le nombre de dimensions et la longueur de chaque dimension sont établis lors de la création de l'instance de tableau.
- Les valeurs par défaut des éléments de tableau numériques sont définies sur zéro et les éléments de références sont définis sur null.

```
// Declare a single-dimensional array of 5 integers.  
int[ ] array1 = new int[5];  
int lengthOfArray = array1.Length;  
  
// Declare and set array element values.  
int[ ] array2 = new int[ ] { 1, 3, 5, 7, 9 };  
  
// Alternative syntax.  
int[ ] array3 = { 1, 2, 3, 4, 5, 6 };  
  
// Declare a two dimensional array.  
int[ , ] multiDimensionalArray1 = new int[2, 3];  
  
// Declare and set array element values.  
int[ , ] multiDimensionalArray2 = { { 1, 2, 3 }, { 4, 5, 6 } };
```



Exemple Tableau (2/2) – Tableau en escalier

- Un **tableau en escalier** (*jagged array*) est un tableau de tableaux, et par conséquent ses éléments sont des types référence et sont initialisés sur null.
- Les indices d'un tableau (simple ou en escalier) commencent par 0 (*zero indexed*) : un tableau avec n éléments est indexée de 0 à n-1.

```
// Declare a jagged array (en escalier).
int[ ][ ] jaggedArray = new int[6][ ];

// Set the values of the 1st & 2nd array in the jagged array.
jaggedArray[0] = new int[4] { 1, 2, 3, 4 };
jaggedArray[1] = new int[2] { 5, 6 };
...
```



Concaténation de chaînes

```
string userName = "Toto";
string dateString = DateTime.Today.ToString("dd/MM/yyyy");

// 1- Use the + and += operators for concatenations.
string str = "Hello " + userName + ". Today is " + dateString + ".";
str += " How are you today?";
```



```
// 2- Use string interpolation to concatenate strings.
string str2 = $"Hello {userName}. Today is {date}.";
```



Mots clés C# vs. Java world

C# keyword	Java keyword						
abstract	abstract	Explicit	N/A	object	N/A	this	This
as	N/A	Extern	native	operator	N/A	throw	Throw
base	Super	Finally	finally	out	N/A	true	True
bool	boolean	Fixed	N/A	override	N/A	try	try
break	break	Float	float	params	N/A	typeof	N/A
byte	N/A	For	for	private	private	unit	N/A
case	case	Foreach	N/A	protected	N/A	ulong	N/A
catch	catch	Get	N/A	public	public	unchecked	N/A
char	char	Goto	goto	readonly	N/A	unsafe	N/A
checked	N/A	If	if	ref	N/A	ushort	N/A
class	class	Implicit	N/A	return	return	using	import
const	const	In	N/A	sbyte	byte	value	N/A
continue	continue	Int	int	sealed	final	virtual	N/A
decimal	N/A	Interface	interface	set	N/A	void	void
default	default	Internal	protected	short	short	volatile	volatile
delegate	N/A	Is	instanceof	sizeof	N/A	while	while
do	do	Lock	synchronized	stackalloc	N/A	:	extends
double	double	Long	long	static	static	:	implements
else	else	namespace	package	string	N/A	N/A	strictfp
enum	N/A	New	new	struct	N/A	N/A	throws
event	N/A	Null	null	switch	switch	N/A	transient

[Source : Using McCabe Method to Compare the Complexity of Object Oriented Languages]

Dr. Oussama BEN KHIROUN | Framework .NET

29

C# Cheat sheets

C# Cheat Sheet	
C# Data Types	C# Data Types
string	2 bytes/char
bool	2 bytes
byte	1 byte
short	2 bytes
int	4 bytes
long	8 bytes
float	4 bytes
double	8 bytes
decimal	16 bytes
C# Args	C# Args
Array.Sort(array)	
Array.Sort(array, arr)	
Array.Sort(array, compare)	
Array.Clear(array, array.length)	
Array.Copy(source, target, numelements)	
C# String Operations	C# String Operations
Compare	String.Compare(string, string, StringComparison)
IndexOf	int IndexOf(char val, start, num)
LastIndexOf	Search from end of string
Replace	newstr + oldstr.Replace("old", "new")
Split	char[] delimiter = { ' ' };
C# Function Declaration	C# Function Declaration
*	Match one or more occurrence
*	Match zero or more occurrences
?	Matches zero or one occurrences
id	Match decimal digit or non-number (0-9)
w	Match any word character or non-space
w\W	Match whitespace or non whitespace
\w\W	Match word character in set
{}	Match quantifier not in set
a b	Either a or b
\n \t	New line, carriage return, tab
+	Match one or more occurrence
DateTime and DateTimeOffset	
Date	Date (yy, mm, dd)
DateTime	DateTime (yy, mo, day, hh, min, sec)
DateTime.Parse("1/1/2015")	Parse date/time as string
curr.Day	curr.DayOfDay(1-7)
curr.Subtract(Day(1))	TimeSpan(1, min, sec)
TimeSpan.FromSeconds(120)	TimeSpan.FromSeconds(120)
C# Decimal Processing	
Co c	(0.0) 1.38855 Currency
D d	(0.0) 138855 Must be integer value
E e	1.38855E-05 Must be floating point
F f	(0.9,2) 1.38855, Fixed Point
G g	(0.9) 1.38855 General
N n	(0.9,2) .7865 Convert to percent
P p	(0.9,3) .7865 Percent
R r	(0.9,9) 3.14159 Retain all decimal
X x	(0.9,1) .314 Converts to Hex
C# Date-Time Formatting	
d	1/19/2015
D	Wednesday, January 19, 2015
F	Wednesday, January 19, 2015 4:05:20 PM
G	1/19/2015 4:05:20 PM
M	January'19
Y	January, 2015
o	4:05:20 PM
T	4:05:20 PM
u	2015-01-19T16:05:20Z
U	Wednesday, January 19, 2015 4:05:20 PM
Ending Your Growth Through Application Development & Support Services	
http://www.GeoTouch.com	

C# Cheat Sheet	
by laurence via cheatography.com/42043/cs/12684/	
Data Types	
bool	Boolean value
byte	8-bit unsigned integer
char	16-bit Unicode character
decimal	128-bit precise decimal values with 29-29 significant digits
double	64-bit double-precision floating point
float	32-bit single-precision floating point
int	32-bit signed integer
long	64-bit signed integer
object	Base type for all other types
sbyte	8-bit signed integer
short	16-bit signed integer
string	String value
uint	32-bit unsigned integer
ulong	64-bit unsigned integer
ushort	16-bit unsigned integer
Type Conversion Methods	
ToBoolean	
ToByte	
ToChar	
ToDateTime	
ToDecimal	
ToDouble	
ToInt16	
ToInt32	
ToInt64	
ToSByte	
ToSingle	
ToString	
ToType	
ToUInt16	
ToUInt32	
Type Conversion Methods (cont)	
ToUInt64	
Naming Conventions	
Class	MyClass
Method	MyMethod
Local variable	myLocalVariable
Private variable	_myPrivateVariable
Constant	MyConstant
Arrays	
int[] array = new int[] {1, 2, 3}	
int[] array = {1, 2, 3}	
var array = new int[] {1, 2, 3}	
int[] array = new int[3]	
Statements	
if-else	if (true) {...} else if (true) {...} else {...}
switch	switch (var) { case 1: break; default: break;}
for	for (int i = 1; i < 5; i++) {...}
foreach-in	foreach (int item in array) {...}
while	while (true) {...}
do...while	do {...} while (true);
try-catch-finally	try {...} catch (Exception e) {...} catch {...} finally {...}

Practice Time

Let's Code a
WORDLE
like Game

guess the word

t a r æ g s

Références



Site officiel de Microsoft .NET : <https://dotnet.microsoft.com/>



What is .NET | .NET Core 101 tutorials
<https://dotnet.microsoft.com/learn/dotnet/hello-world-tutorial/intro>

