



ASSOSOFTWARE

ASSOCIAZIONE ITALIANA PRODUTTORI SOFTWARE

.NET Core in C#

Marchetti Filippo

14/11/2023

SOFTWARE HUB
system srl

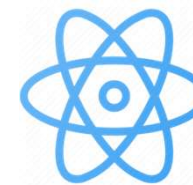


.NET CORE IN C#

Chi sono...



<https://www.linkedin.com/in/filippo-marchetti-06606151/>



.NET CORE IN C#

Agenda

- Lezione 1: Introduzione a .NET e C#
- Lezione 2: Sintassi
- Lezione 3: OOP – Ereditarietà e Polimorfismo
- Lezione 4: Gestione dei Dati e Accesso ai Database
- Lezione 5: Introduzione a EF Core e Multithreading
- Lezione 6: Creazione di Interfacce Utente (UI)
- Lezione 7: Progettazione ed esposizione di API con ASP.NET

.NET CORE IN C#



<https://forms.office.com/e/u2qE4iACfk?origin=lprLink>

.NET CORE IN C#

Lezione 1: Introduzione a .NET e C#

- Cos'è il C#?
- Architettura di .NET
- Installazione e configurazione
- Creazione del primo programma C#: «Hello World»
- Concetti di base di C#

.NET CORE IN C#

Lezione 1: Introduzione a .NET e C#

Cos'è il C#?



C# (pronunciato "See Sharp") è un linguaggio di programmazione moderno, orientato agli oggetti e indipendente dai tipi. C# consente agli sviluppatori di compilare molti tipi di applicazioni sicure e affidabili eseguite in .NET. C# ha le sue radici nella famiglia di linguaggi C e risulterà immediatamente familiare ai programmatori di C, C++, Java e JavaScript.

Multipiattaforma

Open Source <https://github.com/dotnet/csharp-lang>

Ambiti di sviluppo Eterogenei

- ☐ Applicazioni desktop
- ☐ Applicazioni server
- ☐ Applicazioni con o senza interfaccia grafica
- ☐ Applicazioni Web/Cloud (ASP.NET Core)
- ☐ Applicazioni per Smartphone (.NET MAUI)
- ☐ Videogame (Unity)

.NET CORE IN C#

Lezione 1: Introduzione a .NET e C#

Architettura di .NET



.NET è una piattaforma open source e gratuita per lo sviluppo di qualsiasi tipologia di software, che comprende una libreria di oggetti di base riutilizzabili, e un ambiente di esecuzione che al suo interno gestisce l'intero ciclo di vita delle applicazioni e fornisce loro i vari servizi necessari.

Un'applicazione .NET viene sviluppata per una o più implementazioni di .NET. Le implementazioni di .NET includono

- .NET Framework
- .NET 5+ (e .NET Core)
- Mono
- UWP

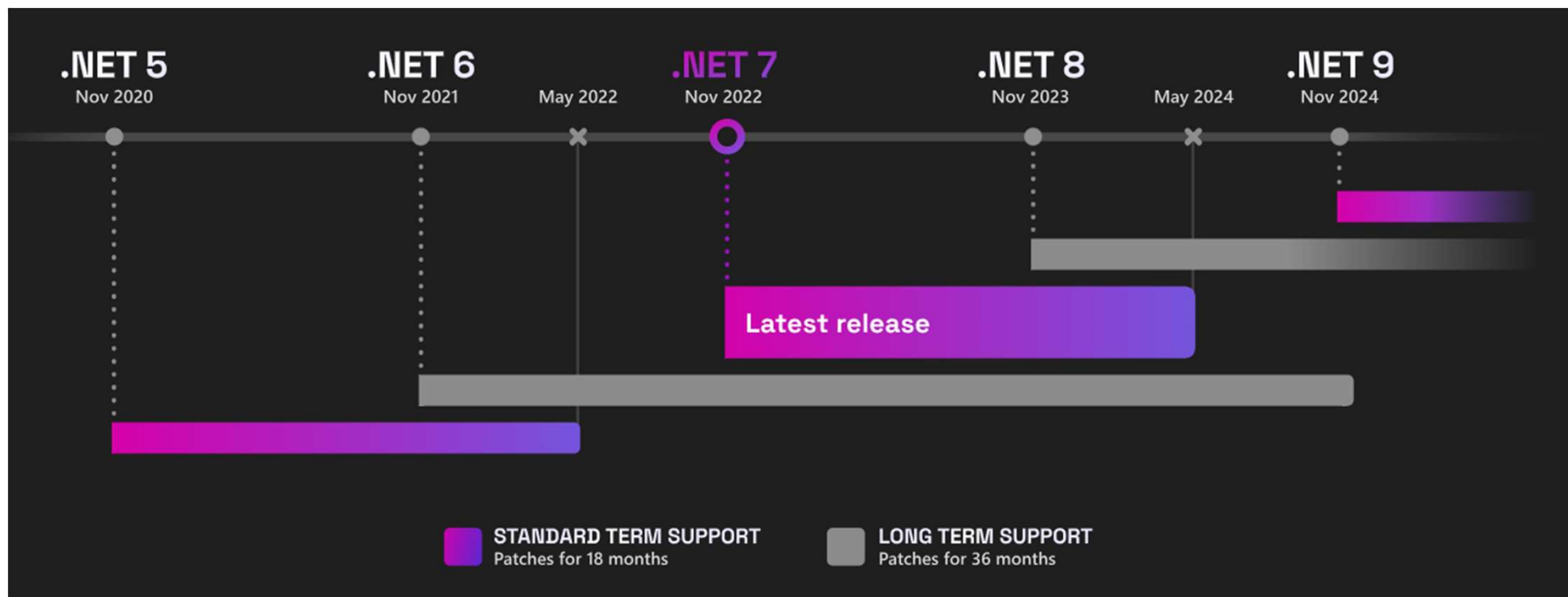
Ogni applicazione .NET include i seguenti componenti:

- Una o più **runtime**, cioè ambienti di esecuzione delle applicazioni, per esempio **.NET 6 CLR** e **.NET Framework CLR**
- Le **librerie di classi di base**, per esempio **.NET Base Class Library**, **.NET Framework Class Library**
- I **framework applicativi** opzionali dedicati a particolari ambiti o scenari: ASP.NET, Windows Forms, WPF, MAUI, Blazor e così via

.NET CORE IN C#

Lezione 1: Introduzione a .NET e C#

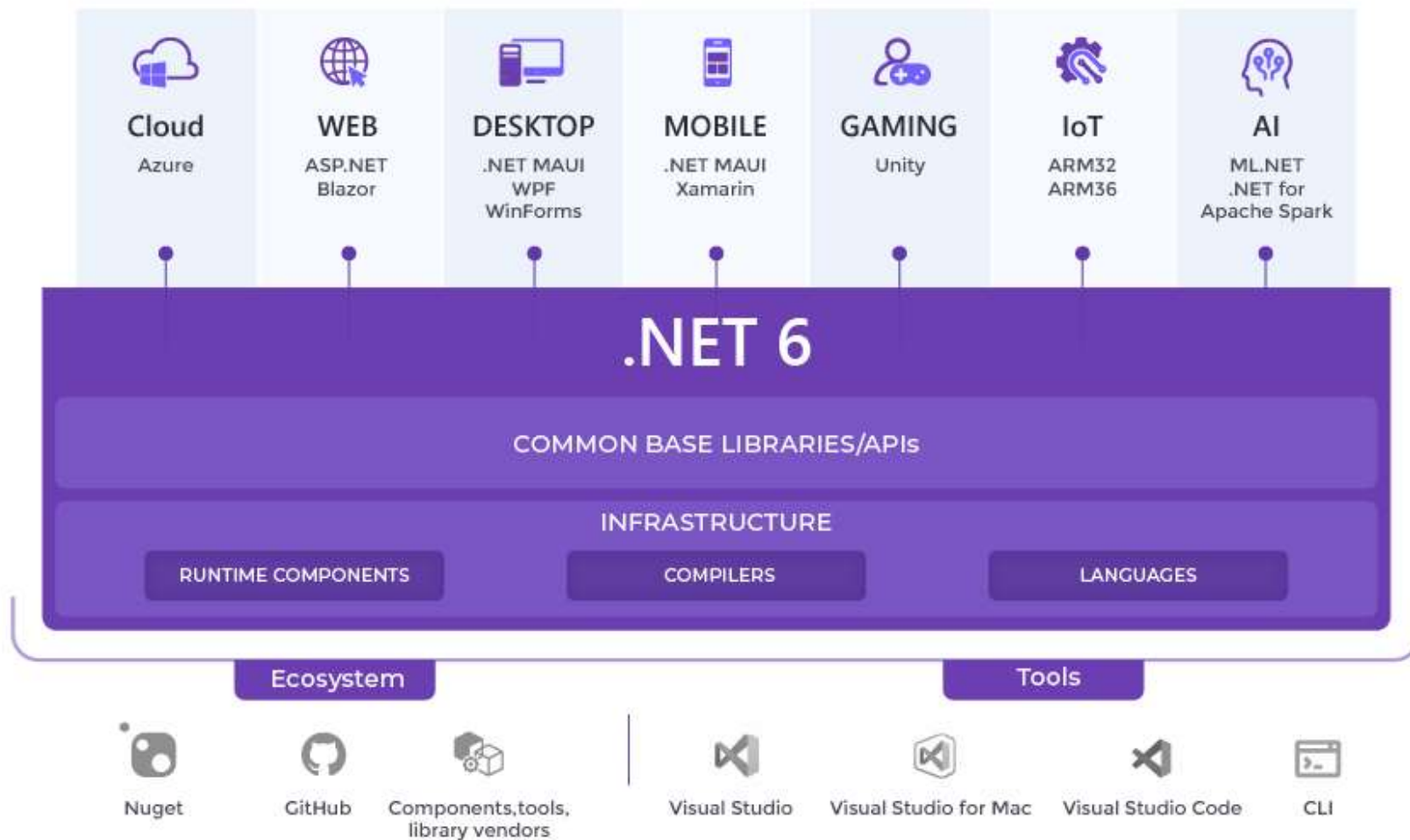
Architettura di .NET



.NET CORE IN C#

Lezione 1: Introduzione a .NET e C#

Architettura di .NET

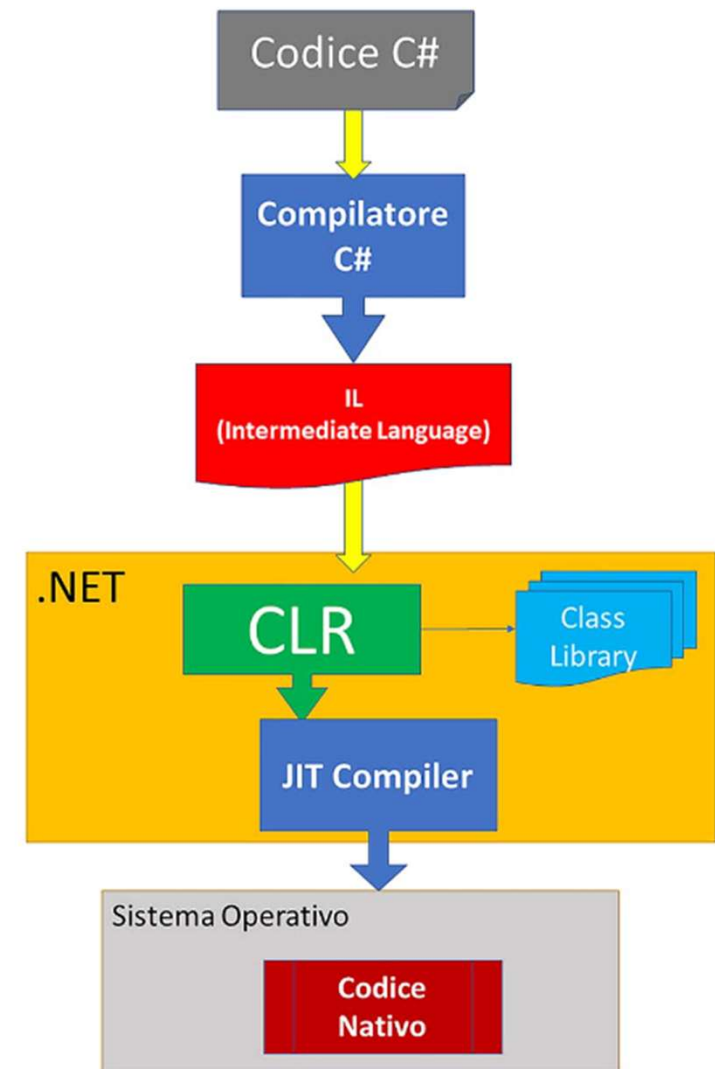


.NET CORE IN C#

Lezione 1: Introduzione a .NET e C#

Architettura di .NET

- Un programma scritto in C#, prima di poter essere eseguito, viene convertito in un linguaggio intermedio chiamato **IL** (Intermediate Language)
- Il codice IL e i metadati sono contenuti in un file detto **assembly** (.exe o .dll)
- In un secondo processo di compilazione, il programma in codice IL viene trasformato nel linguaggio macchina specifico della piattaforma su cui esegue il CLR stesso. Tale processo viene chiamato **JIT Compiler**.



.NET CORE IN C#

Lezione 1: Introduzione a .NET e C#

Versioni di C# e .NET

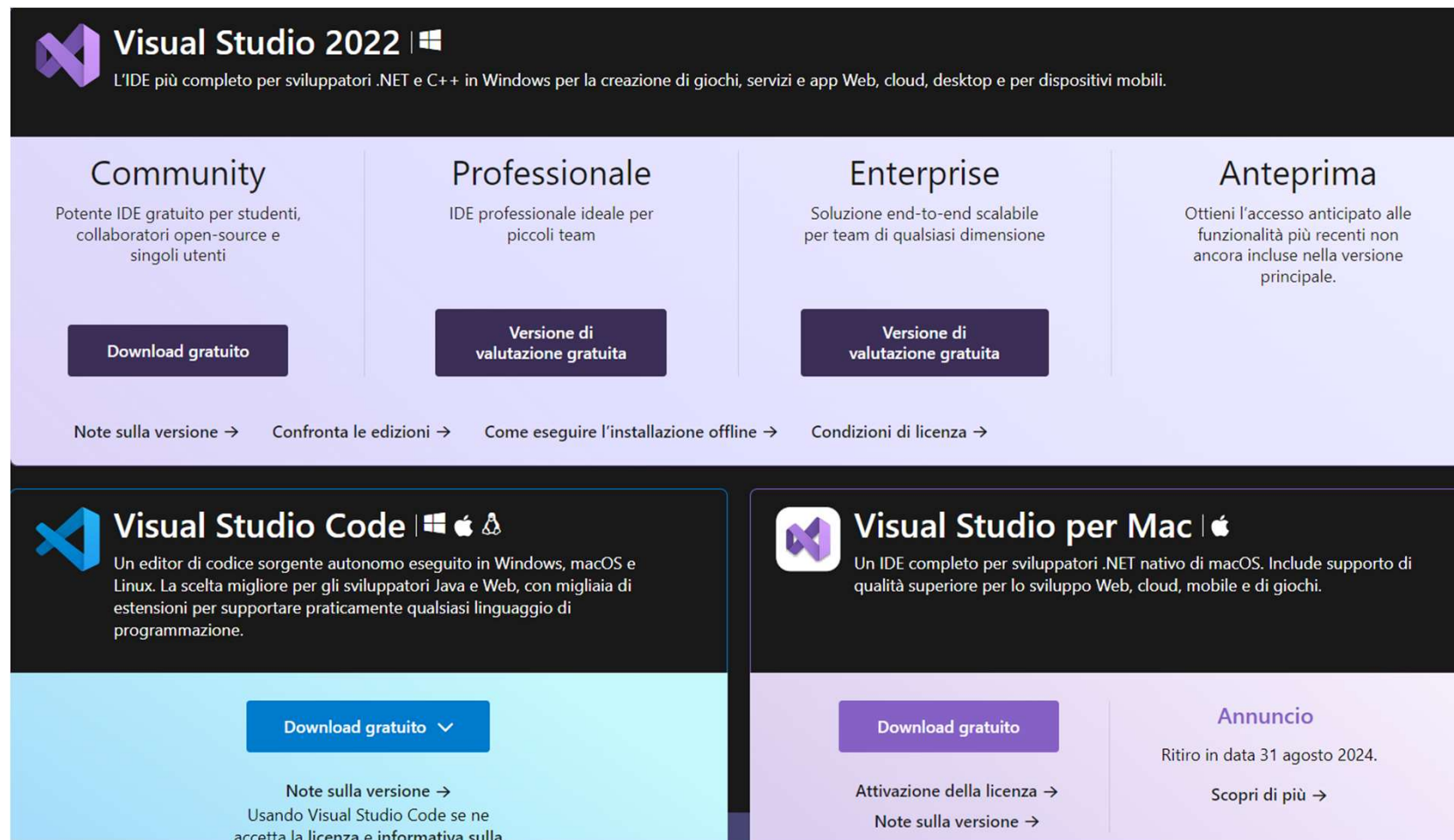
Anno	.NET Framework	.NET Core	.NET	C#	Visual Studio
2002	1.0			1.0	Visual Studio .NET 2002
2003	1.1			1.2	Visual Studio .NET 2003
2005	2.0			2.0	Visual Studio 2005
2006	3.0			2.0	
2007	3.5			3.0	Visual Studio 2008
2010	4.0			4.0	Visual Studio 2010
2012	4.5			5.0	Visual Studio 2012
2013	4.5.1 - 4.5.2			5.0	Visual Studio 2013
2015	4.6 - 4.6.1 - 4.6.2	1.0		6	Visual Studio 2015
2017	4.7 - 4.7.1	2.0 - 2.1		7 - 7.1 - 7.2	Visual Studio 2017
2018	4.7.2	2.2		7.3	
2019	4.8	3.0 - 3.1		8.0	Visual Studio 2019
2020			5.0	9.0	
2021			6.0	10.0	Visual Studio 2022

.NET CORE IN C#

Lezione 1: Introduzione a .NET e C#

Installazione e configurazione

<https://visualstudio.microsoft.com/it/downloads/>



The screenshot displays the official Visual Studio 2022 download page. At the top, the Visual Studio logo is followed by the text 'Visual Studio 2022' and a Windows icon, with a subtitle: 'L'IDE più completo per sviluppatori .NET e C++ in Windows per la creazione di giochi, servizi e app Web, cloud, desktop e per dispositivi mobili.' Below this, four main product categories are listed: 'Community' (described as a powerful free IDE for students, open-source collaborators, and individual users), 'Professionale' (a professional IDE for small teams), 'Enterprise' (a scalable end-to-end solution for teams), and 'Anteprima' (early access to new features). Each category has a corresponding 'Download gratuito' or 'Versione di valutazione gratuita' button. A navigation bar below these categories contains links: 'Note sulla versione →', 'Confronta le edizioni →', 'Come eseguire l'installazione offline →', and 'Condizioni di licenza →'. The bottom section features two additional products: 'Visual Studio Code' (an independent code editor for Windows, macOS, and Linux) and 'Visual Studio per Mac' (a complete IDE for macOS). Both have 'Download gratuito' buttons. The Visual Studio per Mac section also includes an 'Annuncio' (announcement) about a withdrawal on August 31, 2024, and links for 'Attivazione della licenza →' and 'Note sulla versione →'.

Visual Studio 2022 | Windows
L'IDE più completo per sviluppatori .NET e C++ in Windows per la creazione di giochi, servizi e app Web, cloud, desktop e per dispositivi mobili.

Community
Potente IDE gratuito per studenti, collaboratori open-source e singoli utenti
[Download gratuito](#)

Professionale
IDE professionale ideale per piccoli team
[Versione di valutazione gratuita](#)

Enterprise
Soluzione end-to-end scalabile per team di qualsiasi dimensione
[Versione di valutazione gratuita](#)

Anteprima
Ottieni l'accesso anticipato alle funzionalità più recenti non ancora incluse nella versione principale.

[Note sulla versione →](#) [Confronta le edizioni →](#) [Come eseguire l'installazione offline →](#) [Condizioni di licenza →](#)

Visual Studio Code | Windows, macOS, Linux
Un editor di codice sorgente autonomo eseguito in Windows, macOS e Linux. La scelta migliore per gli sviluppatori Java e Web, con migliaia di estensioni per supportare praticamente qualsiasi linguaggio di programmazione.
[Download gratuito](#)

[Note sulla versione →](#)
Usando Visual Studio Code se ne accetta la [licenza](#) e [informativa sulla](#)

Visual Studio per Mac | macOS
Un IDE completo per sviluppatori .NET nativo di macOS. Include supporto di qualità superiore per lo sviluppo Web, cloud, mobile e di giochi.
[Download gratuito](#)

[Attivazione della licenza →](#) [Note sulla versione →](#)

Annuncio
Ritiro in data 31 agosto 2024.
[Scopri di più →](#)

.NET CORE IN C#

Lezione 1: Introduzione a .NET e C#

Installazione e configurazione

Il comando **dotnet** (<https://learn.microsoft.com/en-us/dotnet/core/tools/>)

- `dotnet --version`
- `dotnet --info`
- `dotnet --help`
- `dotnet [command] --help`

new	Inizializza un nuovo progetto, la cui tipologia è definita dal template indicato come argomento
restore	Ripristina dipendenze e strumenti di un progetto
build	Compila un progetto e relative dipendenze
publish	Pubblica l'applicazione e le relative dipendenze inserendole in una cartella per la distribuzione
run	Compila un progetto ed esegue l'applicazione
clean	Pulisce l'output di un progetto compilato in precedenza
help	Mostra la documentazione online (su docs.microsoft.com) del comando specificato

```
C:\Users\f.marchetti>dotnet --version
7.0.402

C:\Users\f.marchetti>dotnet --info
.NET SDK:
Version: 7.0.402
Commit: 791db8e2d8

Ambiente di runtime:
OS Name: Windows
OS Version: 10.0.22621
OS Platform: Windows
RID: win10-x64
Base Path: C:\Program Files\dotnet\sdk\7.0.402\

Host:
Version: 7.0.12
Architecture: x64
Commit: 4a824ef37c

.NET SDKs installed:
7.0.402 [C:\Program Files\dotnet\sdk]

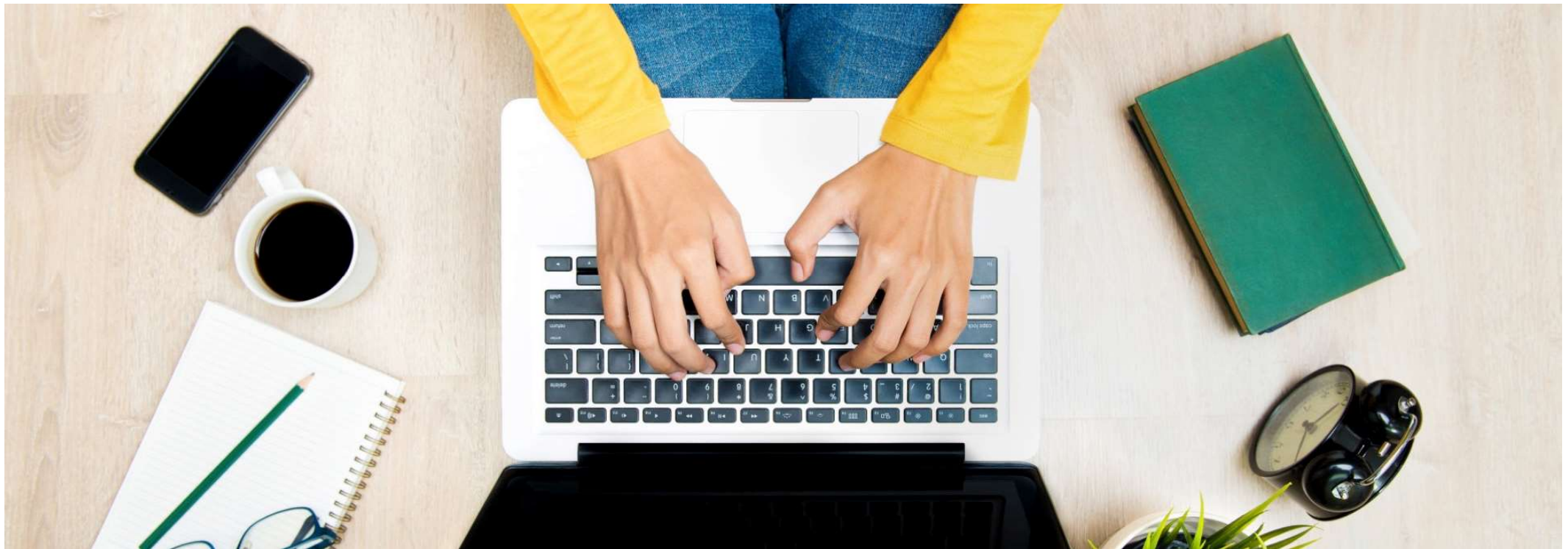
.NET runtimes installed:
Microsoft.AspNetCore.App 6.0.23 [C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App]
Microsoft.AspNetCore.App 7.0.12 [C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App]
Microsoft.NETCore.App 6.0.23 [C:\Program Files\dotnet\shared\Microsoft.NETCore.App]
Microsoft.NETCore.App 7.0.12 [C:\Program Files\dotnet\shared\Microsoft.NETCore.App]
Microsoft.WindowsDesktop.App 6.0.23 [C:\Program Files\dotnet\shared\Microsoft.WindowsDesktop.App]
Microsoft.WindowsDesktop.App 7.0.12 [C:\Program Files\dotnet\shared\Microsoft.WindowsDesktop.App]

Other architectures found:
x86 [C:\Program Files (x86)\dotnet]
registered at [HKLM\SOFTWARE\dotnet\Setup\InstalledVersions\x86\InstallLocation]
```


.NET CORE IN C#

Lezione 1: Introduzione a .NET e C#

Creazione del primo programma C#: «Hello World»

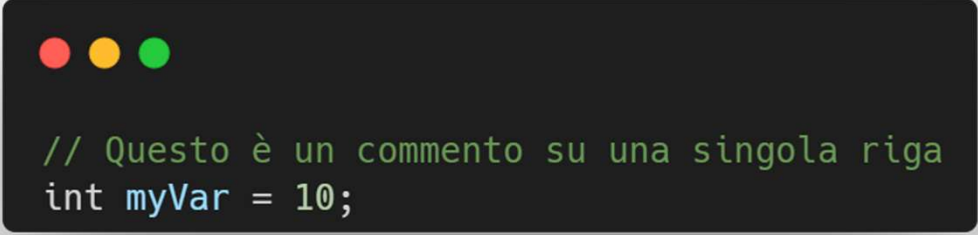


.NET CORE IN C#

Lezione 1: Introduzione a .NET e C#

Concetti di base di C#

- Commenti
 - Su una riga



```
// Questo è un commento su una singola riga  
int myVar = 10;
```

- Multipli



```
/*  
 * Questo è un commento su più righe  
 */  
int myVar = 10;
```

.NET CORE IN C#

Lezione 1: Introduzione a .NET e C#

Concetti di base di C#

- Commenti
 - Di documentazione

```
///<summary>
/// Il metodo esegue la somma di due numeri interi
///</summary>
///<param name="x">Primo numero</param>
///<param name="y">Secondo numero</param>
///<returns>Risultato della somma</returns>
public int Add(int x, int y)
{
    return x + y;
}
```

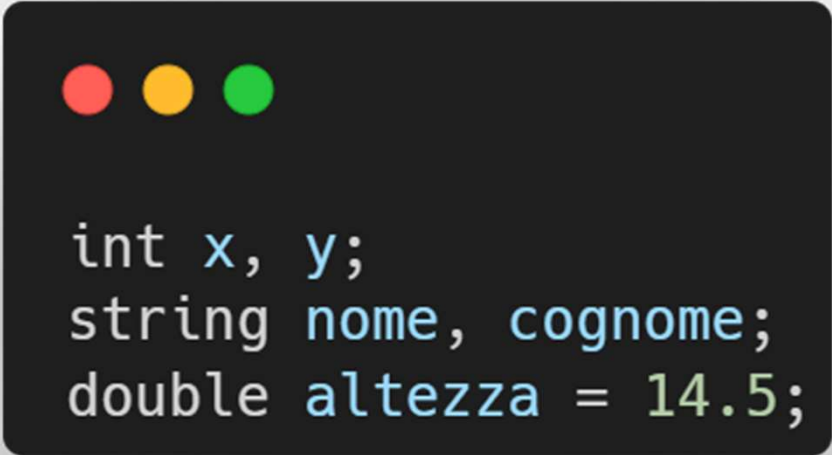
.NET CORE IN C#

Lezione 1: Introduzione a .NET e C#

Concetti di base di C#

➤ Variabili

- Contenitore di dati **dinamico**
- `<tipoDato> <identificatore>;`
- Naming convention (*camelCase*)



```
int x, y;  
string nome, cognome;  
double altezza = 14.5;
```

.NET CORE IN C#

Lezione 1: Introduzione a .NET e C#

Concetti di base di C#

➤ Costanti

- Contenitore di dati **fisso/immutabile/costante**
- `const <tipoDato> <identificatore> = <value>;`
- Naming convention (*pascalCase*)



```
const string MyFirstConst = "Prima Costante!";
```



.NET CORE IN C#

Lezione 1: Introduzione a .NET e C#

Concetti di base di C#

➤ Primitive Types

Short Name	.NET Class	Type	Width	Range (bits)
byte	Byte	Unsigned integer	8	0 to 255
sbyte	SByte	Signed integer	8	-128 to 127
int	Int32	Signed integer	32	-2,147,483,648 to 2,147,483,647
uint	UInt32	Unsigned integer	32	0 to 4294967295
short	Int16	Signed integer	16	-32,768 to 32,767
ushort	UInt16	Unsigned integer	16	0 to 65535
long	Int64	Signed integer	64	-9223372036854775808 to 9223372036854775807
ulong	UInt64	Unsigned integer	64	0 to 18446744073709551615
float	Single	Single-precision floating point type	32	-3.402823e38 to 3.402823e38
double	Double	Double-precision floating point type	64	-1.79769313486232e308 to 1.79769313486232e308
char	Char	A single Unicode character	16	Unicode symbols used in text
bool	Boolean	Logical Boolean type	8	True or false
object	Object	Base type of all other types		
string	String	A sequence of characters		
decimal	Decimal	Precise fractional or integral type that can represent decimal numbers with 29 significant digits	128	$\pm 1.0 \times 10e-28$ to $\pm 7.9 \times 10e28$

.NET CORE IN C#

Lezione 1: Introduzione a .NET e C#

Concetti di base di C#

➤ Keywords

C# Keywords				
abstract	as	base	bool	break
byte	case	catch	char	checked
class	const	continue	decimal	default
delegate	do	double	else	enum
event	explicit	extern	false	finally
fixed	float	for	foreach	goto
if	implicit	in	int	interface
internal	is	lock	long	namespace
new	null	object	operator	out
override	params	private	protected	public
readonly	ref	return	sbyte	sealed
short	sizeof	stackalloc	static	string
struct	switch	this	throw	true
try	typeof	uint	ulong	unchecked
unsafe	ushort	using	var	virtual
void	volatile	while		
Special C# Identifiers (Contextual Keywords)				
add	alias	async	await	dynamic
get	global	nameof	partial	remove
set	value	when	where	yield
Contextual Keywords (Used in Queries)				
ascending	by	descending	equals	from
group	in	into	join	let
on	orderby	select	where	



.NET CORE IN C#

Lezione 1: Introduzione a .NET e C#

Concetti di base di C#

➤ Casting

- **Implicit** (automatically) - converting a smaller type to a larger type size
int -> long -> float -> double

```
int myInt = 9;
double myDouble = myInt;           // Automatic casting: int to double

Console.WriteLine(myInt);          // Outputs 9
Console.WriteLine(myDouble);       // Outputs 9
```

- **Explicit** (manually) - converting a larger type to a smaller size type
double -> float -> long -> int

```
double myDouble = 9.78;
int myInt = (int) myDouble;        // Manual casting: double to int

Console.WriteLine(myDouble);       // Outputs 9.78
Console.WriteLine(myInt);          // Outputs 9
```

.NET CORE IN C#

Lezione 1: Introduzione a .NET e C#

Concetti di base di C#

- Lavorare con le stringhe
 - *Concatenation*
 - *Length*
 - *ToUpper()*
 - *ToLower()*
 - *Substring()*
 - *IndexOf()*

```
string myFirstString = "Ciao mondo";  
int myLengthString = myFirstString.Length; // 10  
string myUpperString = myFirstString.ToUpper(); // CIAO MONDO  
string myLowerString = myFirstString.ToLower(); // ciao mondo  
string mySubString = myFirstString.Substring(5); // mondo  
int myIndexOfString = myFirstString.IndexOf('o'); // 3
```

.NET CORE IN C#

Lezione 1: Introduzione a .NET e C#

Concetti di base di C#

- Lavorare con i numeri
 - Operatori: + / * - %
 - *Increment/Decrement*
 - La classe *Math*

```
int a = 10;
int b = 2;
int sum = a + b; // 12
int sub = a - b; // 8
int mul = a * b; // 20
int div = a / b; // 5
int module = a % b; // 0
int expression = (a * b) + b;
double abs = Math.Abs(f);
double pow = Math.Pow(b, 2);
double min = Math.Min(a, b);
double max = Math.Max(a, b);
double sqrt = Math.Sqrt(b);
```

.NET CORE IN C#

Lezione 1: Introduzione a .NET e C#

Concetti di base di C#

- Operatori Logici e di Comparazione
 - *Comparazione*: <, >, <=, >=, ==, !=
 - *Logici*: &&, ||, !

```
bool minor = 5 < 6; // True
bool major = 5 > 6; // False
int myLogicalIntVar = 10;
bool firstLogicalExpression = (5 < 6) && (myLogicalIntVar == 10); // True
bool secondLogicalExpression = (5 < 6) || (myLogicalIntVar == 5); // True
bool thirdLogicalExpression = !(myLogicalIntVar == 10); // False
```

.NET CORE IN C#

Lezione 1: Introduzione a .NET e C#

Concetti di base di C#

Operators	Category / name
<code>x.y, x?.y, x?[y], f(x), a[i], x++, x--, new, typeof, checked, unchecked, default, nameof, delegate, sizeof, stackalloc, x->y</code>	Primary
<code>+x, -x, !x, ~x, ++x, --x, (T)x, await, &x, *x, true and false</code>	Unary
<code>*, /, %</code>	Multiplicative
<code>+, -</code>	Additive
<code><<, >></code>	Shift
<code><, >, <=, >=, is, as</code>	Relational and type-testing
<code>==, !=</code>	Equality
<code>&</code>	Boolean logical AND or bitwise logical AND
<code>^</code>	Boolean logical XOR or bitwise logical XOR
<code> </code>	Boolean logical OR or bitwise logical OR
<code>&&</code>	Conditional AND
<code> </code>	Conditional OR
<code>??</code>	Null-coalescing operator
<code>? :</code>	Conditional operator
<code>=, +=, -=, *=, /=, %=, &=, =, ^=, <<=, >>=, =></code>	Assignment and lambda declaration

.NET CORE IN C#

Lezione 1: Introduzione a .NET e C#

Concetti di base di C#

- Input/Output Console
 - `Console.WriteLine();`
 - `Console.Write();`
 - `Console.ReadLine();`
 - `Console.ReadKey();`

```
Console.WriteLine("Inserisci il primo numero: ");  
var firstNumberString = Console.ReadLine();  
Console.WriteLine("Inserisci il secondo numero: ");  
var secondNumberString = Console.ReadLine();  
Console.ReadKey();
```


.NET CORE IN C#

Lezione 1: Introduzione a .NET e C#



<https://www.menti.com>

2552 2351