Cheatography

C# Cheat Sheet

by nbmx99 via cheatography.com/208740/cs/44863/

Comments

```
// This is a comment.
/* This is a multi-line comment. */
```

Variables

```
T variableName = value;
```

String Interpolation

```
int a = 2;
int b = 2;
int sum = a + b;
Console.WriteLine($"Sum of {a} and {b} is {sum}."); // outputs : Sum of 2 and 2 is 4.
```

String Concatenation

```
string str1 = "Hello";
string str2 = "World";
Console.WriteLine(str1 + " " + str2); // outputs : Hello World
```

If-Else If-Else Statement

```
if (condition) { ... }
else if (condition) { ... }
else { ... }
```

Switch-Case Statement

```
switch (value)
{
  case value1: ...
  break;
  case value2: ...
  break;
  ...
  default: ...
  break;
}
```

For Loop Statement

```
for (initializer; condition; iterator) { ... }
```

Foreach Loop Statement

```
foreach (iterator in collection) { ... }
```

While Loop Statement

```
while (condition) { ... }
```

Do-While Loop Statement

```
do { ... }
while (condition);
```

Functions

```
[accessModifier] [returnType] FunctionName(parameters) { return ...; }
[accessModifier] static [returnType] FunctionName(parameters) { return ...; }
```

Arrays

```
T[] arrayName = new T[length];
T[] arrayName = { value, value, value, ... };
T[] arrayName = new T[] { value, value, value, ... };
```

Lists

```
List<T> listName = new List<T>();
List<T> listName = new List<T> { value, value, value, ... };
```

Classes

```
class ClassName { ... }
```

Objects

```
Object objectName = new Object(arguments);
```

2D Arrays

```
T[,] arrayName = new T[row, column];
T[,] arrayName = { { value, value, ... }, { value, value, ... }, ... };
T[,] arrayName = new T[,] { { value, value, ... }, { value, value, ... }, ... };
```

Enums

```
enum EnumName
{
    value1,
    value2,
    value3,
    ...
}
```

Properties

```
[accessModifier] T PropertyName { get; set; }
[accessModifier] static T PropertyName { get; set; }
```

Try-Catch-Finally Statement

```
try { ... }
catch (Exception e) { ... }
finally { ... }
```

Operators

Operator	Symbols
Arithmetic operators	+, -, *, /, %
Assignment operators	=, +=, -=, *=, /=, %=, &=, |=, ^=, <<=, >>=, >>>=
Bitwise operators	&, |, ^, ~, <<, >>, >>>
Comparison operators	==, !=, <, >, <=, >=
Logical operators	&&, ||, !
Lambda operator	=>
Member access operators	., ->, []
Null-conditional operators	?., ?[]
Ternary operator	?:
Unary operators	++, --

C# Keywords

abstract	event	namespace	static
as	explicit	new	string
base	extern	null	struct
bool	false	object	switch
break	finally	operator	this
case	fixed	out	throw
catch	float	override	true
char	for	params	try
checked	foreach	private	typeof
class	goto	protected	uint
const	if	public	ulong
continue	implicit	readonly	unchecked
decimal	in	ref	unsafe
default	int	return	ushort
delegate	internal	sbyte	using
do	interface	sealed	virtual
double	is	short	void
else	lock	sizeof	volatile
enum	long	stackalloc	while

Escape Characters

Character	Meaning
\n	New line
\t	Horizontal tab
\v	Vertical tab
\a	Alert (bell sound)
\\	Backslash
\"	Double quote
\'	Single

Naming Conventions

Type	Style	Example
Namespaces	**PascalCase**	MyNamespace
Classes	**PascalCase**	MyClass
Structs	**PascalCase**	MyStruct
Interfaces	**IPascalCase**	IMyInterface
Enums	**PascalCase**	MyEnum
Events	**PascalCase**	MyEvent
Delegates	**PascalCase**	MyDelegate
Methods	**PascalCase**	MyMethod
Parameters	**camelCase**	myParameter
Properties	**PascalCase**	MyProperty
Public fields	**PascalCase**	MyField
Private fields	**_camelCase**	_myField
Protected fields	**_camelCase**	_myField
Constants	**SNAKE_CASE**	MY_CONSTANT
Local variables	**camelCase**	myLocalVariable