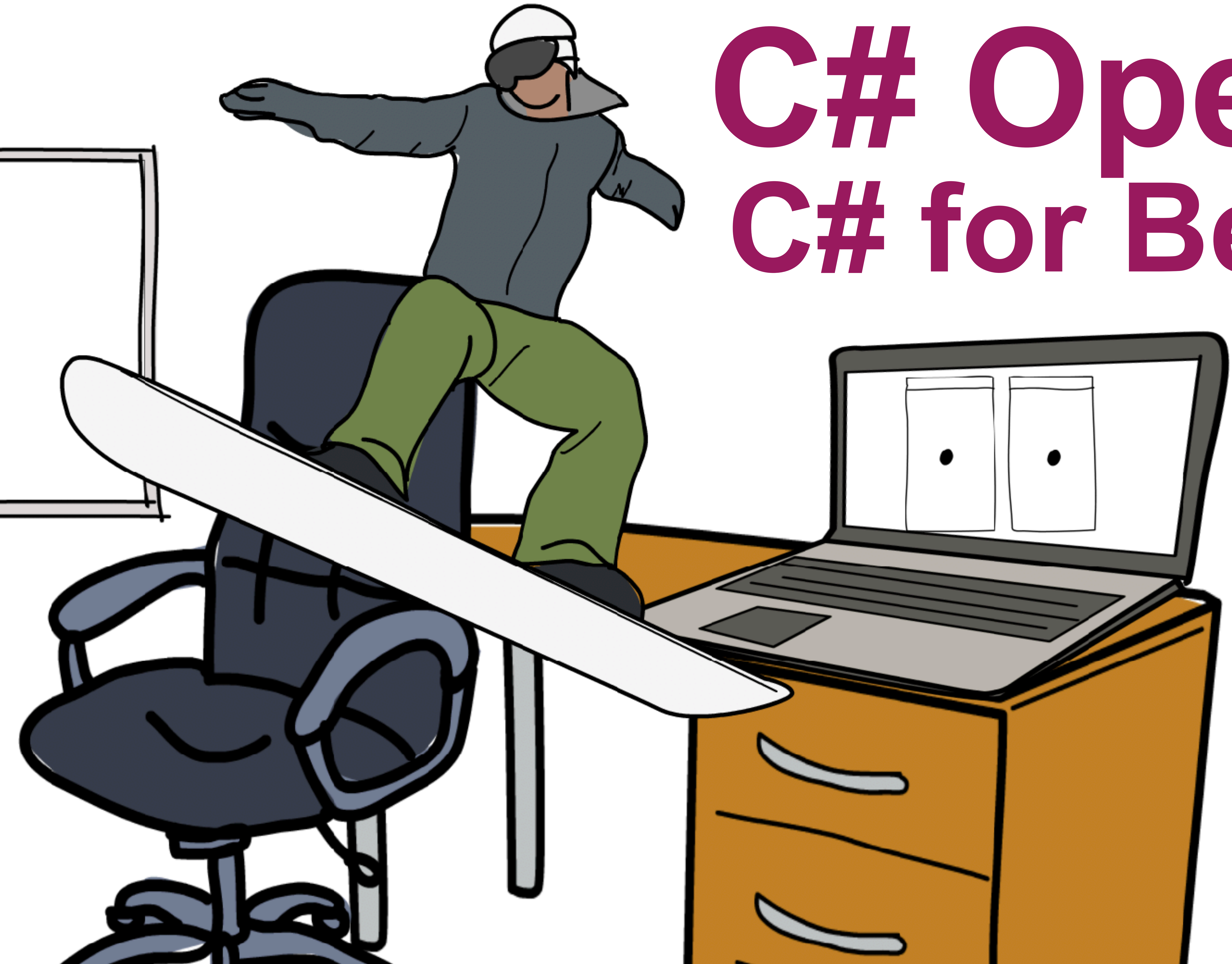
C# Operators Documentation

C# Operators

C# for Beginners



C# Operators

Primary Operators

Expression	Description
x.y	Member access
f(x)	Method and delegate invocation
a[x]	Array and indexer access
x++	Post-increment
x--	Post-decrement
new T(...)	Object and delegate creation
new T(...){...}	Object creation with initializer.

C# Operators

Primary Operators

new T(...){...}	Object creation with initializer.
new {...}	Anonymous object initializer.
new T[...]	Array creation.
typeof(T)	Obtain System.Type object for T
checked(x)	Evaluate expression in checked context
unchecked(x)	Evaluate expression in unchecked context
default (T)	Obtain default value of type T
delegate {}	Anonymous function (anonymous method)

C# Operators

Unary Operators

Expression	Description
+x	Identity
-x	Negation
!x	Logical negation
~x	Bitwise negation
++x	Pre-increment
--x	Pre-decrement
(T)x	Explicitly convert x to type T

C# Operators

Multiplicative Operators

Expression	Description
*	Multiplication
/	Division
%	Remainder

C# Operators

Additive Operators

Expression	Description
x + y	Addition, string concatenation, delegate combination
x - y	Subtraction, delegate removal

C# Operators

Equality Operators

Expression	Description
x == y	Equal
x != y	Not equal

C# Operators

Relational and Type Operators

Expression	Description
x < y	Less than
x > y	Greater than
x <= y	Less than or equal
x >= y	Greater than or equal
x is T	Return true if x is a T, false otherwise
x as T	Return x typed as T, or null if x is not a T

C# Operators

Logical, Conditional, and Null Operators

Category	Expression	Description
Logical AND	x & y	Integer bitwise AND, Boolean logical AND
Logical XOR	x ^ y	Integer bitwise XOR, Boolean logical XOR
Logical OR	x | y	Integer bitwise OR, Boolean logical OR
Conditional AND	x && y	Evaluates y only if x is true
Conditional OR	x || y	Evaluates y only if x is false
Null coalescing	x ?? y	Evaluates to y if x is null, to x otherwise
Conditional	x ? y : z	Evaluates to y if x is true, z if x is false

C# Operators

Assignment and Anonymous Operators

Expression	Description
=	Assignment
x op= y	Compound assignment. Supports these operators: +=, -=, *=, /=, %=, &=, |=, ^=, <<=, >>=
(T x) => y	Anonymous function (lambda expression)

C# Operators

Shift Operators

Expression	Description
x << y	Shift left
x >> y	Shift right

Summary

- We have discussed that an operator in C# is an element applied to one or more operands in a statement to perform a specific operation.
- We have discussed what is meant by binary, unary and ternary with regards to C# operators.
- We went through multiple examples of using C# operators in code.
- We have discussed the significants in C# of precedence, associativity and parentheses regarding multiple operators in a statement.

gl

Digital