Python Basics for C# Developers

- navan

Feature	C#	Python
Code File Extension	.cs	.py
End of line	;	Enter key
Code block grouping	{ }	: Enter key indent
Variable Declaration	var x=“navan”	x = “navan” or x=‘navan’
Code Indentation	Visual appeal	Part of syntax
Output string to console	Console.WriteLine()	print()
Read user input	Console.ReadLine()	Input()
Base class keyword	base	super()
Base-child class inheritance	ChildClass:BaseClass	ChildClass(BaseClass)
Comment	//	#
Multi-line comment	/* */	“”” ”””

Feature	C#	Python
While block	while{}	while: else:
For loop	for (int i=0; i<6;i++) {}	for i in range(6): else:
List for loop	foreach (var name in names) {}	For name in names:

If-else	If(){}
	else if(){}
	else{}
		If():
		elif():
		else:

Single-line if else	if (a>b) Console.WriteLine(“A"); else Console.WriteLine(“B");	print("A") if a > b else print("B")
String interpolation	$”{x}”	F”{x}”
Trim spaces in string	x.Trim()	x.strip()
Casting	Convert.ToString(variable)	str(variable)
Multiple variable value assignment	var (k,j,m)=(20,40,"dfsdf")	k,j,m=20,40,"dfsdf"
Item removal	RemoveAt(index)	pop(index)
Unordered, unindexed collection	HashSet	set

Feature	C#	Python
Nested Function	✗	✓
global / nonlocal keywords	✗	✓

try block	try{}
	catch{}
	finally{}
		try:
		except:
		else:
		finally:

Function Definition	int returnInt(int k)
	{
	  return k;
	}
		def returnInt(k):
		  return k

Class definition	class Classname
	{
	  internal Classname(string name,int age)
	  {
	  this.Name = name;
	  this.Age = age;
	  }
	  public string Name { get; set; }
	  public int Age { get; set; }
	}
		class Classname:
		  def __init__(self, name, age):
		  self.name = name
		  self.age = age

Feature	C#	Python
Package manager	nuget	PIP
Constructor	Classname() {}	def __init__():
Function to return object as string	✕	def __str__():
Allow blank body	{}	pass keyword
External code integration	using <namespace>;	import <module>
Invoke external function	<namespace>.<funcname>();	<module>.<funcname>()
Array	In-built	list (or) import numpy
Logical operators	|| &&	or and
Substring	x.SubString(2,3)	x[2:6]
Deconstruct/unpacking	Does not support asterisk	Supports asterisk
Lambda function	Func<int,int,string> funcname = (a,b) => { return (a+b+10).ToString(); };	funcname = lambda a,b: str(a+b + 10)

Feature	C#	Python
Create list	var newlist = new List<string> {“navan", "kkkk"};	newlist=[“navan", "kkkk"]
Create tuple	var tup = new Tuple<int,int,string>(1, 2, "navan");	tup = (1, 2, "navan")
Create set	HashSet<string> names = new HashSet<string> { "nanana", "boooobbooo"};	names ={"nanana", "boooobbooo”}
Create dictionary	Dictionary<string,string> dict= new Dictionary<string, string> {{ "1", "navan" },{ "2", "dff" }};	dict ={“1": “navan", “2": “dff”}
Access collections from last with negative index	✗	✓
Get type of object	var x=dict.GetType();	x=type(dict)

List Query	int[] marklist = new int[4];
	marklist[0]=97;
	marklist[1] = 92;
	marklist[2]=81;
	marklist[3] = 8;
	IEnumerable<int> marklist80 = from mark in marklist where mark > 80 select mark;
		marklist=[97,92,81,8]
		marklist80=[mark for mark in marklist if mark>80]