**Python vs C#**

|  |  |
| --- | --- |
| **Python** | **C#** |
| **Comments**  ## This is a single-line comment  ''' This is a  multi-line  comment ''' | **Comments**  **//** This is a single-line comment  **/\*** This is a  multi-line  comment **\*/** |
| **Variables**  name = "John"  height = 180  weight = 70.5  height = 180.5 ## no error | **Variables**  **string** name = "John" **;**  **int** height = 180 **;**  **double** weight = 70.5 **;**  **Note: Data Type declared only once for each variable**  **Can only assign values of the same type**  height = 180.5 ERROR! |
| **Basic Input/Output**  data = input(<prompt>) ## need to cast  e.g. int data = (int) (input(“Enter age : ”)  print data | **Basic Input/Output**  data = Console.ReadLine() // need to convert  e.g. int age = Convert.ToInt32(Console.ReadLine());  Console.Write(data)  Console.WriteLine(data) |

**Python vs C#**

|  |  |
| --- | --- |
| **Python** | **C#** |
| **Arithmetic Operators**  y = 5  z = 3  x = y + z ## x = 8  x = y - z ## x = 2  x = y \* z ## x = 15  x = y / z ## x = 1.6666666667  x = y % z ## x = 2  ## Operators not available in C#  x = y // z ## x = 1  x = y \*\* z ## x = 125 | **Arithmetic Operators**  y = 5  z = 3  x = y + z // x = 8  x = y - z // x = 2  x = y \* z // x = 15  x = y / z // x = 1  x = y % z // x = 2  // Operators not available in Python  x = y++ // x = 6  x = z-- // x = 2 |
| **Relational Operators (same)**  **==**  **!=**  **>**  **>=**  **<**  **<=**  **Chained Operators**  1 <= a <= 10  a >= 1 and a <= 10  a <= 1 or a >= 10 | **Relational Operators (same)**  **==**  **!=**  **>**  **>=**  **<**  **<=**  **Chained Operators**  Not available  a >= 1 **&&** a <= 10  a <= 1 **||** a >= 10 |

**Python vs C#**

|  |  |
| --- | --- |
| **Python** | **C#** |
| **Selections**  if mark >= 50:  result = "Pass"  else:  result = "Fail | **Selections**  **int mark;** // need to be declared  **string result;** // need to be declared  . . .  . . .  if **(**mark >= 50**)**  **{**  result = "Pass"**;**  **}**  else  **{**  result = "Fail"**;**  **}** |
| if mark >= 80:  grade = "A"  elif mark >= 70:  grade = "B"  . . .  . . .  else:  grade = "F" | **int mark;** // need to be declared  **char grade;** // need to be declared  . . .  . . .  if **(**mark >= 80**)**  **{**  grade = 'A'**;**  **}**  else if **(**mark >= 70**)**  **{**  grade = 'B'**;**  **}**  **. . .**  else  grade = 'F'**;** |

**Python vs C#**

|  |  |
| --- | --- |
| **Python** | **C#** |
| **Repetitions (while-loop)**  count = 1  sum = 0  while count < 10:  sum += count  count += 1 | **Repetitions (while-loop)**  **int** count = 1**;**  **int** sum = 0**;**  while **(**count < 10**)**  **{**  sum += count**;**  count += 1**;**  **}** |
| **Repetitions (for-loop)**  for count in range(1, 10 ):  sum += count | **Repetitions (for-loop)**  **int** sum = 0**;**  for (int count=1; count<10; count++)  **{**  sum += count**;**  **}** |

**Python vs C#**

|  |  |
| --- | --- |
| **Python** | **C#** |
| **List**  EmptyList = []  Friends = ['Peter', 'John', 'Mary']  PriceList = [ 899.00, 999.00, 799.00 ]  Friends.append('Agnes')  # display all the data in a List  print(Friends)  # reduce all the prices by 10 %  for **i** **in** range(len(PriceList)):  PriceList[i] = PriceList[i] \* 0.9 | **List**  List<int> EmptyList = new List<int>();  List<string> Friends = new List<string> **{ "**Peter**"**, **"**John**"**, **"**Mary**" }**;  List<double> PriceList = new List<double> **{** 899.00, 999.00, 799.00 **}**;  Friends.add(**"**Agnes**"**);  // display all the data in a List  foreach (string s in Friends)  {  Console.WriteLine(s);  }  // reduce all the prices by 10 %  for (int i=0; i<PriceList.Count; i++)  {  PriceList[i] = PriceList[i] \* 0.9;  } |

**Python vs C#**

|  |  |
| --- | --- |
| **Python** | **C#** |
| **Function / Method with return value**  def Add(a,b):  **return** (a + b)  c = Add(3, 5) ## c = 8 | **Function / Method with return value**  public **int** Add(**int** a, **int** b)  {  **return** (a + b);  }  **int** c = Add(3, 5) // c = 8 |
| **Function / Method with NO return value**  def Add(a,b):  total = a + b ## total is NOT returned  Add(3, 5) ## NO error but not meaningful | **Function / Method with NO return value**  public **void** Add(**int** a, **int** b)  {  **int** total = a + b; // total is NOT returned  }  Add(3, 5) // NO error but not meaningful |