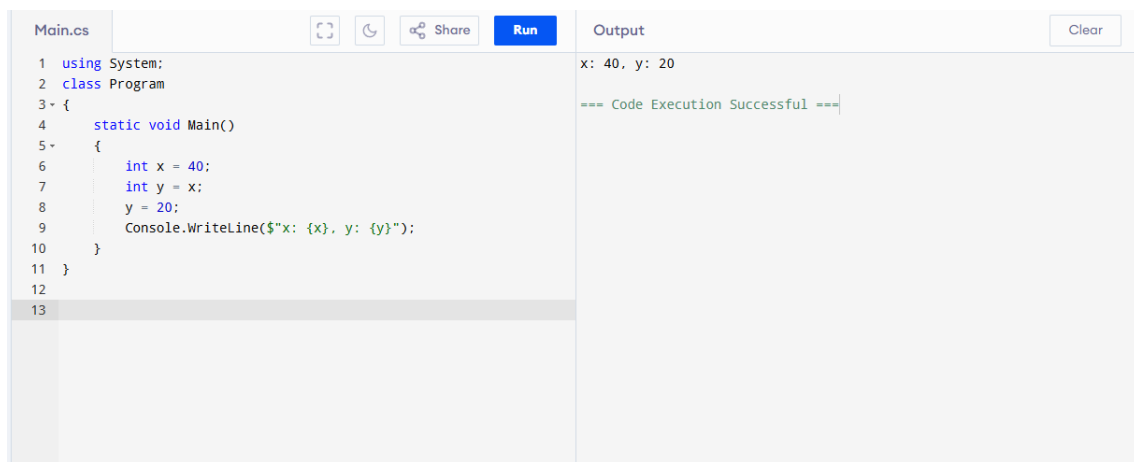


ATV1

Exemplos de tipos primitivos:

- int
- float
- double
- bool
- char

Neste caso, as variáveis do tipo int (x,y) onde o valor de x é copiado diretamente para y sem alteração do valor de x



```
1 using System;
2 class Program
3 {
4     static void Main()
5     {
6         int x = 40;
7         int y = x;
8         y = 20;
9         Console.WriteLine($"x: {x}, y: {y}");
10    }
11 }
12
13
```

Output

x: 40, y: 20

=== Code Execution Successful ===

Exemplos de tipos de referência:

- string
- array
- class
- delegate

Já neste caso a passagem é sempre por referência, ou seja, neste caso p1.nome recebe “igor” e p2.nome recebe “Ronaldo”.



```
1 using System;
2 class Pessoa
3 {
4     public string Nome;
5 }
6 class Program
7 {
8     static void Main()
9     {
10         Pessoa p1 = new Pessoa();
11         p1.Nome = "Igor";
12         Pessoa p2 = p1;
13         p2.Nome = "Ronaldo";
14
15         Console.WriteLine($"p1.Nome: {p1.Nome}, p2.Nome: {p2
16                               .Nome}");
17     }
18 }
```

Output

p1.Nome: Ronaldo, p2.Nome: Ronaldo

=== Code Execution Successful ===

ATV 2

Com erro:

Main.cs	Output
<pre>1 using System; 2 class Program 3 { 4 static void Main() 5 { 6 object obj = "123"; 7 int x = (int)obj; 8 Console.WriteLine(x); 9 } 10 }</pre>	<pre>ERROR! /tmp/8DbBiuq48U/Main.cs(7,25): error CS1002: ; expected === Code Exited With Errors ===</pre>

O objeto `obj` contém uma **string** que precisa ser convertida para **int** no meu caso usei `Convert.ToInt32()`

Ajustado:

Main.cs	Output
<pre>1 using System; 2 class Program 3 { 4 static void Main() 5 { 6 object obj = "123"; 7 int x = Convert.ToInt32(obj); 8 Console.WriteLine(x); 9 } 10 }</pre>	<pre>123 === Code Execution Successful ===</pre>

ATV3

Main.cs	Output
<pre>1 using System; 2 class Program 3 { 4 static void Main() 5 { 6 string numStr = "99"; 7 int num = int.Parse(numStr); 8 Console.WriteLine(num); 9 } 10 }</pre>	<pre>99 === Code Execution Successful ===</pre>