

PI Triangular e Colorido

Mário Leite

...

Se você pegar uma figura plana, perfeitamente circular, abri-la e dividir seu comprimento pelo diâmetro SEMPRE vai encontrar um valor muito próximo de **3,14**, conhecido como PI e representado pela letra grega π . Este número é uma constante universal conhecida desde a Grécia antiga, e que para muitos da época era quase como uma divindade; um número mágico! Assim, estas igualdades se confirmam:

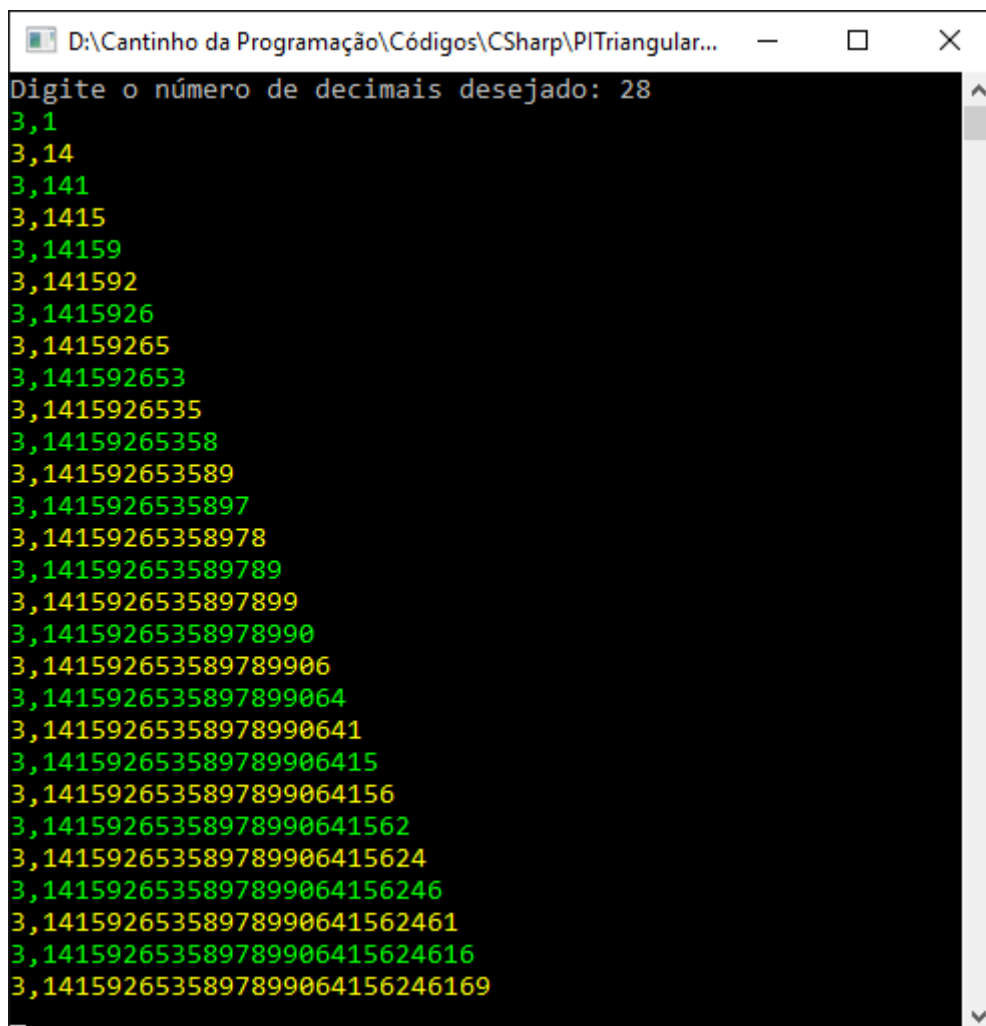
$\frac{\text{Perímetro(roda)}}{\text{Diâmetro(roda)}} = \frac{\text{Perímetro(pneu)}}{\text{Diâmetro(pneu)}} = \frac{\text{Perímetro(moeda)}}{\text{Diâmetro(moeda)}} = \frac{\text{Perímetro(anel)}}{\text{Diâmetro(anel)}} = \pi$

O valor de **PI** mencionado: é **3.14**, é usado na maioria dos cálculos; entretanto, este número é irracional, não podendo ser escrito como uma razão entre dois números inteiros e que, portanto, não é possível determinar um padrão que se repita infinitamente nos dígitos que o constitui. Com relação ao número de casas decimais, este é outro assunto de vital importância para alguns ramos da ciência, como na cosmologia para cálculos de paralaxes; pois um número muito pequeno de casas decimais pode fazer com que o ângulo de observação fique completamente errado! Daí a importância de se calcular um valor de **PI** com uma precisão melhor do que apenas **3.14**. Utilizando computadores bem modernos já se conseguiu encontrar um valor com até 50 trilhões de casas decimais (até 2020). E com relação à lógica de programação computacional, existem vários recursos para obtê-lo com boa precisão; desde expressões simples como a desenvolvida por Aryabhata[1] (com 62832/20000) até sofisticados métodos numéricos usando *cluster* de CPI's.

O programa “**PiTriangularColorido**” é uma solução bem simples para calcular PI com um número de decimais desejado pelo usuário. Aqui neste exemplo (vide **figura 1**) foi mostrado **PI** com até 28 decimais num formato triângulo, em linhas alternadas de cores. Este programa é bem interessante para os programadores iniciantes que precisam se exercitarem na lógica de programação e desenvolver algoritmos para estimular suas capacidades de criar novas soluções computacionais.

Nota: É importante enfatizar que, dependendo da linguagem de programação e da capacidade dessa linguagem trabalhar com bibliotecas matemáticas, pode ser que partir de determinado número de decimais o valor obtido comece a se repetir; este é, exatamente, o caso de programas codificados em C# (como o apresentado aqui): a partir da **28ª** decimal (C# começa a contagem por 0) os valores começam a se repetir. Este problema está ligado ao tratamento de cálculos em ponto flutuante, que é sempre muito problemático na computação. Deste modo, caso seja importante obter PI com centenas ou milhares de decimais o algoritmo deve ser implementado em linguagens que oferecem bibliotecas matemáticas mais potentes: como o Python, por exemplo. Mas, normalmente, o PI com 15 decimais atende perfeitamente às necessidades; é o caso do valor usado pela NASA em cálculos cosmológicos: apenas **15** decimais: **3,141592653589793**, é suficiente!

¹ **Aryabhata** (476 - 550) foi o primeiro entre os grandes matemáticos-astrônomos da Idade clássica dos matemáticos e astrônomos indianos. Foi dele a ideia de calcular **PI** através da razão: 62832/20000.



The screenshot shows a Windows console window titled "D:\Cantinho da Programação\Códigos\CSharp\PITriangular...". The prompt "Digite o número de decimais desejado: 28" is displayed. Below it, a list of 28 decimal approximations of Pi is shown, each on a new line. The numbers are color-coded: the integer part "3" is green, the decimal separator "," is yellow, and the digits are green. The sequence starts with "3,1" and ends with "3,1415926535897899064156246169".

```
Digite o número de decimais desejado: 28
3,1
3,14
3,141
3,1415
3,14159
3,141592
3,1415926
3,14159265
3,141592653
3,1415926535
3,14159265358
3,141592653589
3,1415926535897
3,14159265358978
3,141592653589789
3,1415926535897899
3,14159265358978990
3,141592653589789906
3,1415926535897899064
3,14159265358978990641
3,141592653589789906415
3,1415926535897899064156
3,14159265358978990641562
3,141592653589789906415624
3,1415926535897899064156246
3,14159265358978990641562461
3,141592653589789906415624616
3,1415926535897899064156246169
```

Figura 1 - PI com 28 decimais

```

using System;

namespace PiTriangularColorido
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.Write("Digite o número de decimais de Pi desejado: ");
            int ndecimais = Math.Abs(int.Parse(Console.ReadLine()));
            CalcularPI(ndecimais);
            Console.ReadKey();
        }

        static void CalcularPI(int decimais)
        {
            //Define a precisão para 50 decimais
            int precisao = decimais + 10; //adiciona 10 para garantir precisão suficiente
            //Calcula Pi
            string pi_str = CalculatePi(precisao);
            for (int i = 2; i <= decimais + 2; i++)
            {
                string linha = i < pi_str.Length ? pi_str.Substring(0, i + 1) : pi_str;
                if (i % 2 == 0)
                {
                    Console.ForegroundColor = ConsoleColor.Green; //linhas ímpares
                    Console.WriteLine(linha);
                }
                else
                {
                    Console.ForegroundColor = ConsoleColor.Yellow; //linhas pares
                    Console.WriteLine(linha);
                }
                Console.ResetColor();
            }
        }

        static string CalculatePi(int precisao)
        {
            decimal pi = 0;
            for (int k = 0; k < precisao; k++)
            {
                pi += (decimal)((Math.Pow(16, -k)) * ((4.0 / (8.0 * k + 1)) - (2.0 / (8.0 * k + 4)) - (1.0 / (8.0 * k + 5)) - (1.0 / (8.0 * k + 6))));
            }
            return pi.ToString("#." + new string('#', precisao));
        }
    }
} //Fim do programa "PiTriangularColorido"

```