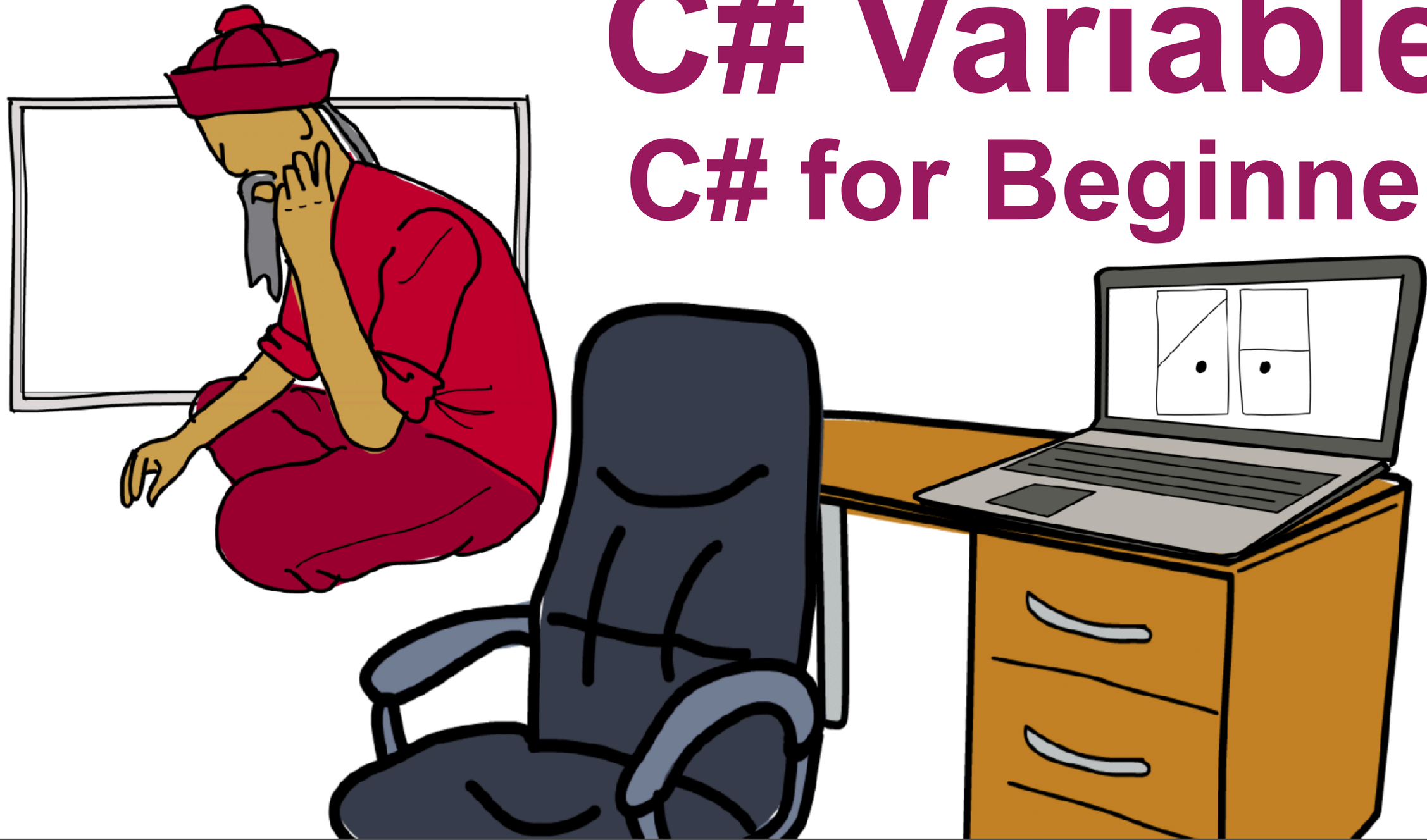
C# Variables Documentation

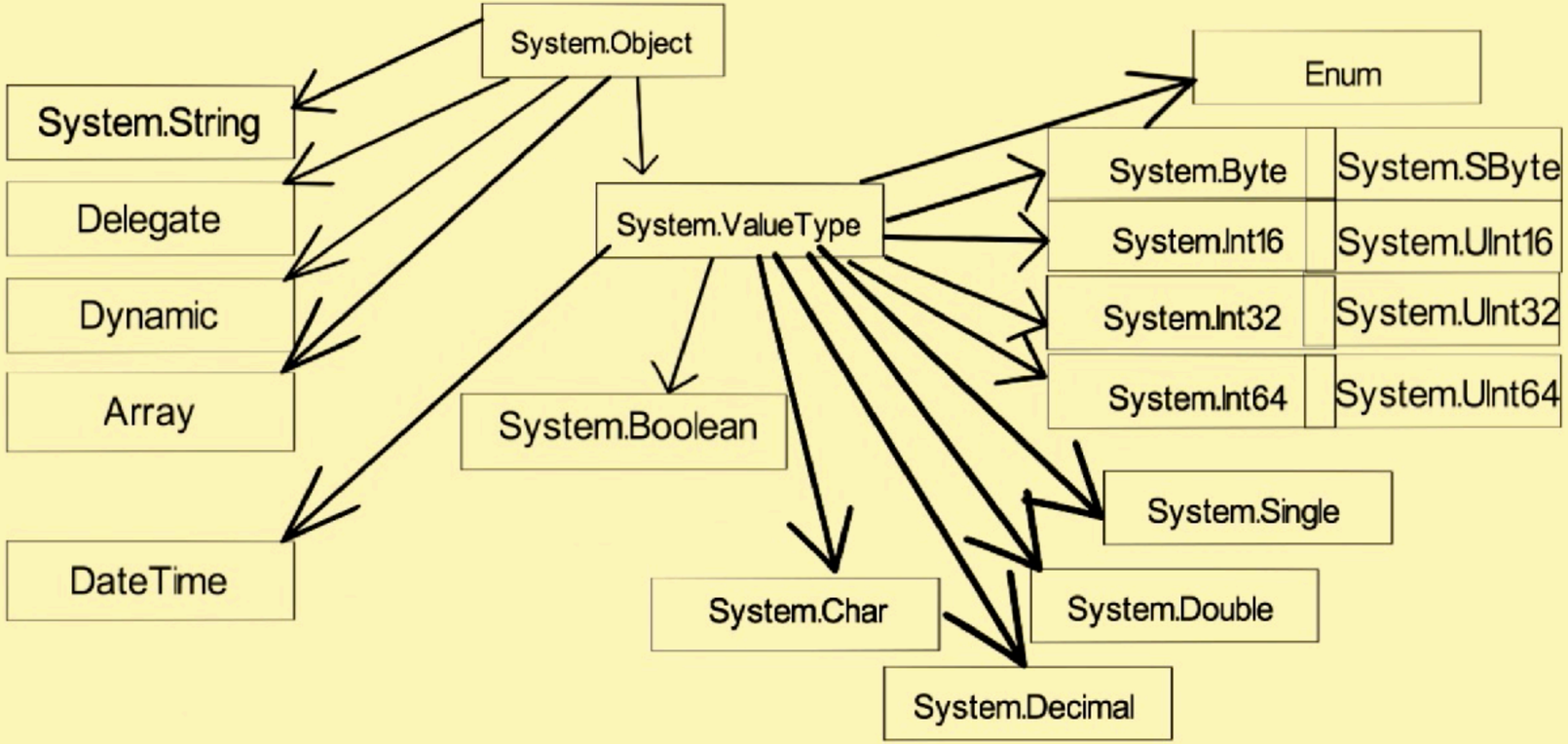
C# Variables

C# for Beginners



Variables are simply names in code
that represent a storage location in memory
where data is stored.

The data stored for a particlar variable can be changed.


System.Object
System.String
Delegate
Dynamic
Array
DateTime
System.ValueType
System.Boolean
System.Char
Enum
System.Byte
System.SByte
System.Int16
System.UInt16
System.Int32
System.UInt32
System.Int64
System.UInt64
System.Single
System.Double
System.Decimal

Type	C# Keyword	.Net Type	Range	Size	Precision
Value Types					
Byte	byte	System.Byte	0 to 255	Unsigned 8 bit integer	
sByte	sbyte	System.SByte	-128 to127	Signed 8 bit Integer	
Short Integer	short	System.Int16	-32678 to 32677	Singed 16 bit Integer	
uShort	ushort	Systen.UInt16	0 to 65355	Unsigned 16 bit Integer	
Integer	int	System.Int32	-2,147,483,648 to 2,147,483,647	Signed 32 bit Integer	
uInt		System.UInt32	0 to 4,294,967,295	Unsigned 32 bit Integer	
Long Integer	long	System.Int64	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	Signed 64 bit Integer	
uLong		System.UInt64	0 to 18,446,744,073,709,551,615	Unsigned 64 bit Integer	
Float	float	System.Single	$\pm 1.5 \times 10^{-45}$ to $\pm 3.4 \times 10^{38}$	32 bit floating point value	~6-9 digits
Double	double	System.Double	$\pm 5.0 \times 10^{-324}$ to $\pm 1.7 \times 10^{308}$	64 bit floating point value	~15-17 digits
Decimal	decimal	System.Decimal	$\pm 1.0 \times 10^{-28}$ to $\pm 7.9228 \times 10^{28}$	128 floating point value	28-29 significant digits
Boolean	bool	System.Boolean	true/false	8 bits	
Character	char	System.Char		16 bit unicode value	
Enum	enum				
Struct	struct				
Datetime	datetime	System.DateTime		64 bit Integer	
Reference Types					
String	string	System.String			
Object	object	System.Object			
Dynamic	dynamic				
Delegate	delegate				
Class	class				
Array	Array				

Int32

- This integer data type is a 32 bit data type.
- It is a signed data type which means it supports both positive and negative values.
- This data type supports a range from, -2147483648 to 2147483647.
- The integer data type has an unsigned counter part named uInt. Unsigned means it only supports positive values.
- uInt supports a range of positive whole number values between 0 to 4294967295.
- An integer is a value type.

String

- Internally a string object stores a read-only sequence of char objects.
- The data stored for a string object is immutable.
- A string is a reference type.

Decimal

- A decimal is a precise fractional or integral type that can represent decimal numbers with 29 significant digits.
- It differs from the float and double data types because it supports less range but a much higher precision which makes the data type preferable for financial and monetary calculations.
- The float and double are faster than the decimal.
- The float (Single data type) supports 32 bits of data.
- The Double data type supports double the amount of bits (i.e. 64 bits of data).
- The decimal supports 128 bit of data.
- The decimal data type is a value type.

Char

- The Char data type is a 16 bit unicode data type.
- It supports character encodings i.e. values that represent characters.
- The Char data type is a value type.

Boolean

- The Boolean data type supports one of two values, true or false.
- The data type supports 8 bits of data.
- The Boolean data type is a value type.

Byte

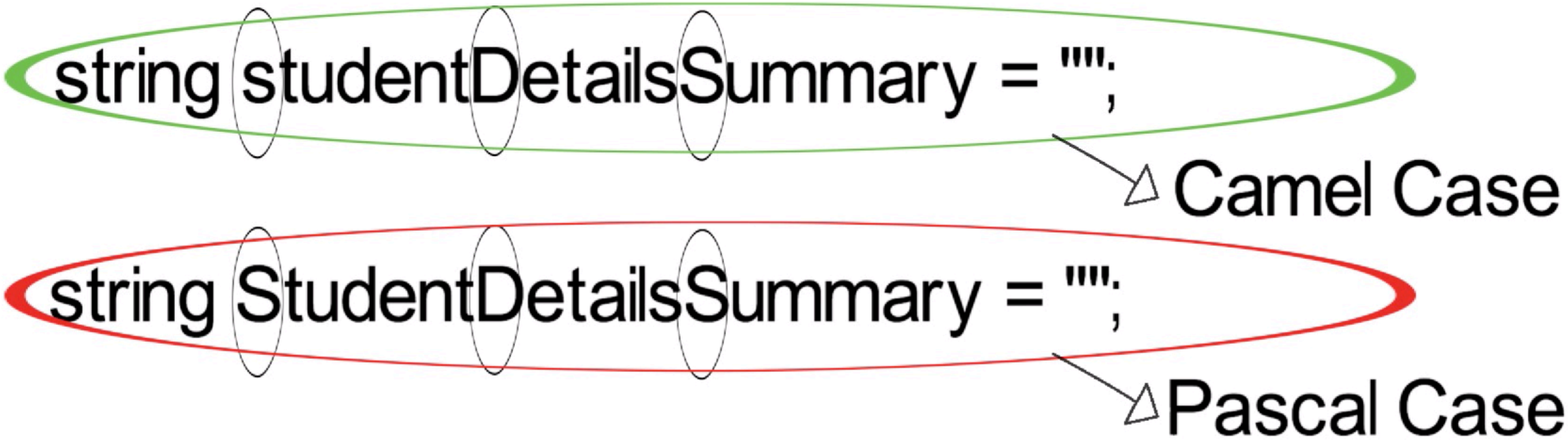
1 1 1 1 1 1 1 1

$$2^0 + 2^1 + 2^2 + 2^3 + 2^4 + 2^5 + 2^6 + 2^7 = 255$$

```
int stuId = 1000;
string studentId = "1000";
int increment = 1;
string concat = "1";


stuId = stuId + increment;

studentId = studentId + concat;

Console.WriteLine($"stuId = {stuId}, studentId = {studentId}");
```

Result

stuId = 1001, studentId = 10001



Pascal Case is not recommended for naming local and member variables

Camel case is recommended for naming local and member variables.

Pascal Case for Classes and Methods

```
public class StudentLoanManager
{

        public StudentLoanManager()
        {
        }

        public void ProcessData()
        {
                //do stuff
        }
}
```

Concise and Descriptive Variable Names

```
decimal annualSalary = 50000;
```

Summary

- We have demonstrated how the context of operators are effected when applied to variables defined as different data types.
- We demonstrated how the number of bits supported for value types effects the magnitude and range for variables defined as these data types.
- We briefly discussed the difference between signed and unsigned integer defined variables.
- We briefly discussed rules and conventions associated with C# variable names.
- We created a user defined type for storing student data. We defined member variables for the ‘Student’ class which were defined as int, string, decimal, char and bool respectively.
- We briefly discussed how the scope of variables limits or enhances access to the values stored within those variables.
- We created a basic application named, ‘StudentApplication’ to demonstrate the use of variables defined as both inbuilt data types and also a user defined type.
- We demonstrated a fundamental difference between value types and reference types.


gl
Digital