C#

github.com/JKramerDD
Jeremy Kramer

Data Types:

bool	boolean
byte	8-bit unsigned
char	16-bit unicode
decimal	128-bit precise 28-29 sf
double	64-bit double-precision
float	32-bit single-precision
int	32-bit signed
long	64-bit signed
object	
sbyte	8-bit signed
short	16-bit signed
string	
uint	32-bit unsigned
ulong	64-bit unsigned
ushort	16-bit unsigned

Type Conversions:

ToBoolean	ToSbyte
ToByte	ToSingle
ToChar	ToString
ToDateTime	ToType
ToDecimal	ToUInt16
ToInt16	ToUInt32
ToInt32	ToUInt64
ToInt64	

Arrays:

```
int[] array = new int[] {x, y, z}
int[] array = {x, y, z}
var array = new int[] {x, y, z}
int[] array = new int[3]
```

Control Flow:

```
if ( conditional ) { ... }
else if ( conditional ) { ... }
else { ... }
switch (var) {
        case1: break;
        default: break;
        }
for (int i = _ ; i _ _; i __ ) { ... }
foreach( type item in arrayName) { ... }
while ( conditional ) { ... }
do { ... } while ( conditional );
try { ... }
catch (Exception e) { ... }
catch { ... }
finally { ... }
```

Classes:

public class Dog { ... }	
public class Dog: Pet { ... }	inheritance from pet
public static class Dog { ... }	static class

Other Modifiers:

abstract	class is a base of other classes only
async	method is asynchronous
const	value is immutable
event	declares an event
extern	method is implemented externally
new	hides member inherited from base class
override	new implementation of member from base class
partial	defines partial classes, structs, & methods
readonly	values assigned in same class only
sealed	member cannot be inherited
static	member belongs to the type instead of specific obj
unsafe	unsafe context
virtual	implementation can be overwritten
volatile	field can be modified by OS, hardware, or thread

String Methods:

```
int Compare(str1,str2)
string Concat(str1,str2)
bool Contains(str)
string Copy(str)
bool EndsWith(str)
bool Equals(str)
int IndexOf(char)
int IndexOf(str)
int IndexOf(char, startindex)
string Insert(startIndex, val)
bool IsNullOrEmpty(str)
string Join(separator, string[] value)
string Remove(startIndex)
string Replace(oldValue, newValue)
string[] Split (separator [, count])
char[] ToCharArray([start, length])
string ToLower()
string ToUpper()
string Trim()
```