# Mini-Tutorial: ConversionExtensions – Useful Conversion & Manipulation Methods

This mini-tutorial summarizes all public methods available in the static class ConversionExtensions. It provides concise descriptions, parameter details, and return values to help you integrate these helpers into your own codebase quickly.

## ToInt(this string text)

Converts a string to an integer by parsing it as a double and then casting to int. If conversion fails, returns 0.

* Parameters:

**- text:** The input string to convert.

* Returns:

**-** Integer value, or 0 if conversion fails.

Example:  
var i = "3.5".ToInt(); // i == 3

## ToDouble(this string text)

Converts a string to a double value. Supports numbers in decimal, hexadecimal (e.g. "FFH", "0xFF"), binary (e.g. "1010B"), and certain keywords ("LONG", "BYTE"). Invalid or unrecognized inputs return double.NaN.

* Parameters:

**- text:** The input string to convert.

* Returns:

**-** Parsed double value, or double.NaN if conversion is not possible.

## Tolerant(this string text)

Returns a tolerant (normalized) version of the string: all uppercase, spaces/tabs removed, and German umlauts replaced (e.g., "Ä" -> "AE"). Returns null if input is null.

* Parameters:

**- text:** The input string (nullable).

* Returns:

**-** Normalized string, or null if input is null.

## RemoveInvalidCharsFromFilename(this string filename)

Replaces all invalid filename characters in the input string with a dot ('.').

* Parameters:

**- filename:** The file name to sanitize (not null).

* Returns:

**-** A safe filename with all invalid characters replaced.

## RemoveRedundantBrackets(string orig)

Removes redundant outer square brackets from a string if present. Returns the original string if no redundant brackets are found.

* Parameters:

**- orig:** Input string (can be null or empty).

* Returns:

**-** String without redundant outer brackets.

## ToImage(this string base64String)

Decodes a Base64-encoded string and returns an Image object. Returns null if decoding fails or input is null/empty.

* Parameters:

**- base64String:** The Base64 string representing image data.

* Returns:

**-** Image object or null if conversion fails.

## ChangeBrightness(this Color color, double percent)

Adjusts the brightness of a color. Positive percent lightens, negative percent darkens. Returns black if operation fails.

* Parameters:

**- color:** The original color.

**- percent:** Brightness adjustment (positive = lighter, negative = darker).

* Returns:

**-** Color with adjusted brightness, or black if failed.

## ChangeTransparency(this Color color, double percent)

Changes the alpha channel of a color based on the percent parameter (0 = fully transparent, 100 = opaque). Returns black if input is out of range or operation fails.

* Parameters:

**- color:** The original color.

**- percent:** Transparency as a percentage (0-100).

* Returns:

**-** Color with new alpha value, or black if failed.

## ToColor(this object color)

Converts an object to a System.Drawing.Color. Accepts color names (e.g. "Red"), hex codes (e.g. "#FF0000"), or Color objects. Returns black for unsupported types or if conversion fails.

* Parameters:

**- color:** The input object (Color or string).

* Returns:

**-** Corresponding Color object, or black if conversion fails.

## ToBase64String(this Image image, ImageFormat format)

Converts an Image to a Base64 string using the specified image format (e.g. PNG, JPEG). Returns null if conversion fails.

* Parameters:

**- image:** The image to convert.

**- format:** The image format.

* Returns:

**-** Base64 string, or null if conversion fails.

## IsBase64ImageStringIsValid(string base64)

Checks if a string is a syntactically valid Base64 string. Returns true only if the string matches Base64 format.

* Parameters:

**- base64:** String to validate.

* Returns:

**-** true if valid Base64, otherwise false.

## ToFormatedBase64String(this Image image, int width = 0, int height = 0, RotateFlipType flipType = RotateFlipType.RotateNoneFlipNone, ImageFormat format = null)

Converts an Image to a Base64 string after optional scaling and rotation. Width and height specify the target size; both zero uses the original size. Format defaults to JPEG. Returns null if conversion fails.

* Parameters:

**- image:** The image to convert.

**- width:** Target width in pixels (optional).

**- height:** Target height in pixels (optional).

**- flipType:** Image rotation/flip (optional, default: none).

**- format:** Image format (optional, default: JPEG).

* Returns:

**-** Base64 string of the processed image, or null if failed.

## Summary

ConversionExtensions provides a set of handy extension methods for robust string, color, and image conversion. These utilities simplify parsing, normalization, and format conversion in various .NET projects.