.NET

Style guidelines

A reference for assets, design, and writing.

These style guidelines can help have a unified visual identity and facilitate effective communication within the .NET developer community. This reference empowers designers, writers, and developers in crafting a unified, on-brand content.

Contents

01 Story and principles

02 Essential elements

03 Imagery

04 Expression and application

01
Story and principles

What is .NET?

A free, cross-platform, open-source developer framework for building modern apps and powerful cloud services.

Modern
Build feature-rich experiences with cutting-edge, performant technologies, leveraging .NET's cross-platform support and integration with cloud and AI.

Productive
Use top editors, AI coding assistants, and cloud environments to save time and deploy efficiently across platforms.

Adaptive
Get started fast and reuse your skills to build web, cloud, mobile, and desktop apps with .NET.

Creative principles

Our creative principles are important and useful attributes to help craft copy, create designs, engage with customers, and help us define how we want to show up in the world.

Universal

Open

Powerful

Simple

Dependable

Approachable

Adaptable

Innovative

Tone of voice

When writing, speaking or designing, take into account these qualities.

Approachable	Arrogant
Helpful	Dismissive
Authentic	Fake
Simple	Complicated
Clear	Ambiguous
Good-natured	Sarcastic
Inclusive	Exclusive

02
Essential elements

Logo

Download logo

Treatment and usage

The square logo is the primary logo for .NET

Primary .NET logo

.NET logotype

Logo minimum size and clear space

Primary Logo

On screen, it should never appear smaller than 50 pixels.

Logotype

On screen, it should never appear smaller than 27 pixels.
In print, it must appear at least .28” (5.5 mm).

Clear space

Make the logo stand out by giving it space on all sides equivalent to the height of the ‘N’.

Minimum height

On screen: 50 px
Print: .52” (12.7 mm)

On screen: 27 px
Print: .28” (5.5 mm) tall

Minimum width

On screen: 50 px wide
Print: .52” (13.2 mm) wide

On screen: 70 px wide
Print: .72” (17.7 mm) wide

Minimum clear space

Color

Our color palette is a source of clarity and relevance and should be applied thoughtfully. It creates familiarity and in a community centric atmosphere evokes emotion.

Before this color palette iteration, the .NET brand color was primarily purple. The expanded palette still features purple as the primary color, with additional colors to make it more flexible depending on tonality. This addition helps express growing skills, technological advances, and user experience.

Primary palette

BLUE
Shade 30
#042B66

BLUE
Tint 30
#9DC4FF

BLUE
Tint 45
#E7F0FF

BLUE
Primary
#512BD4
Pantone 2194 U
CMYK: 96, 58, 0, 0

BRAND PURPLE
Shade 25
#311A7F

BRAND PURPLE
Tint 30
#B9AAEE

BRAND PURPLE
Tint 45
#EEEAFB

BRAND PURPLE
Primary
#512BD4
Pantone 2736 U
CMYK: 51, 66, 0, 17

MAGENTA
Shade 20
#800066

MAGENTA
Tint 30
#EF99DD

MAGENTA
Tint 45
#FBE5F6

MAGENTA
Primary
#D600AA
Pantone Rhodamine Red U
CMYK: 0, 84, 17, 16

Secondary palette

MIDNIGHT BLUE
Shade 20
#0F042C

MIDNIGHT BLUE
Shade 10
#14053A

MIDNIGHT BLUE
Primary
#190649
Pantone 2372 U
CMYK: 19, 26, 0, 71

BLUE
Shade 30
#18747D

BLUE
Tint 45
#E9F9FA

CYAN
Primary
#28C2D1
Pantone 319 U
CMYK: 66, 6, 0, 18

YELLOW
Shade 30
#70511D

YELLOW
Tint 40
#FDF0DA

YELLOW
Primary
#F7B548
Pantone 129 U
CMYK: 0, 26, 69, 3

FLAMINGO
Shade 30
#74222B

FLAMINGO
Tint 40
#FDDCE0

FLAMINGO
Primary
#F65163
Pantone 1788 U
CMYK: 0, 65, 58, 4

Color

Choose between two primary gradient wheels, one for light theme and one for dark theme.

GRADIENTS
Purple >
Magenta

GRADIENTS
Magenta >
Flamingo

GRADIENTS
Flamingo >
Purple

GRADIENTS
Purple >
Blue

GRADIENTS
Blue >
Purple

GRADIENTS
Purple >
Magenta

GRADIENTS
Magenta >
Flamingo

GRADIENTS
Flamingo >
Purple

GRADIENTS
Purple >
Cyan

GRADIENTS
Cyan >
Purple

Type

Download Space Grotesk

Download Open Sans

Use Space Grotesk for headlines and Open Sans for secondary headlines and body copy.

Best for bigger headlines

Space Grotesk

Light Medium **Bold** Regular

AaBbCcDdEeFfGgHhIiJjKkLlMmN
nOoPpQqRrSsTtUuVvWwXxYyZz

Best for smaller titles and paragraphs

Open Sans

Light Medium Bold Semibold Extra bold **Regular**

AaBbCcDdEeFfGgHhIiJjKkLlMmNn
OoPpQqRrSsTtUuVvWwXxYyZz

Type scale

H1 Space Grotesk Bold

H2 Space Grotesk Medium

B1 Open Sans Regular

H3 Open Sans Bold

B2 Open Sans Regular

H4 Space Grotesk Medium

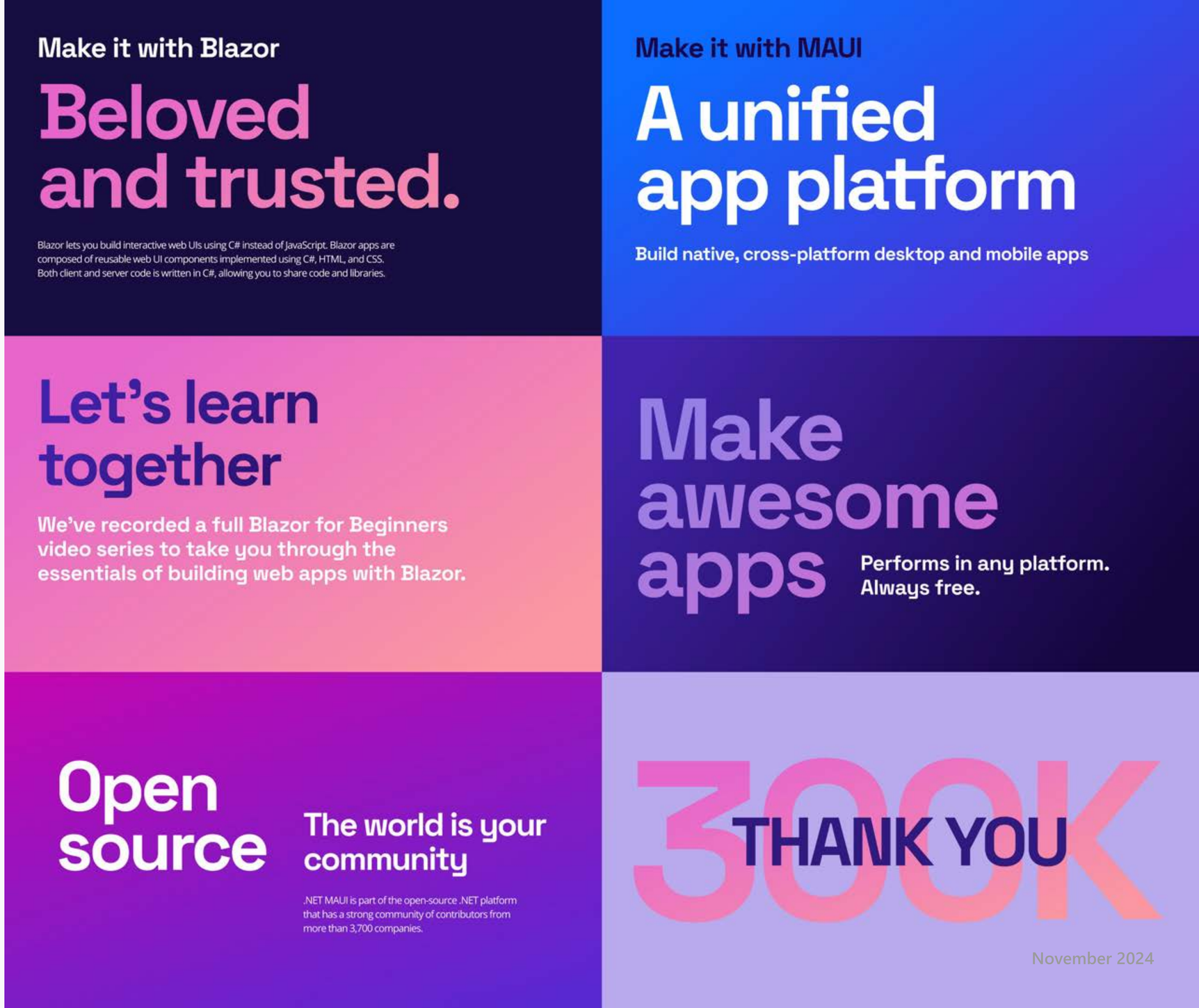
B4 Open Sans Regular

8 12 20 32 52 84

Example scale sequence generated from 20pt body copy

Type examples

Type can be expressive and can be used creatively, applying color, scale and thoughtful copy writing.



03 Imagery

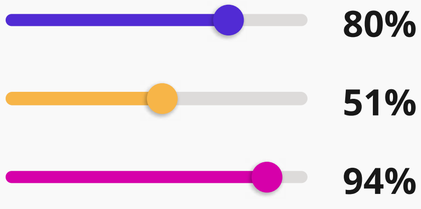
Imagery

Iconography

Best for many blocks of text. Based on the Fluent design language.

Spot illustration

Best used small, as supplementary graphics in communications.

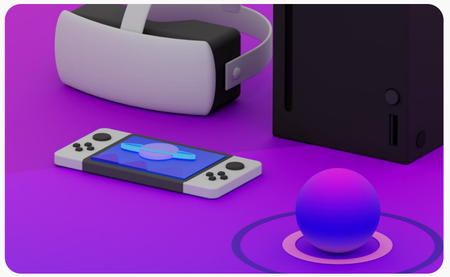


Data visualization

How we show data, numbers and other visualizations.

Code visualization

This is how we focus on code.



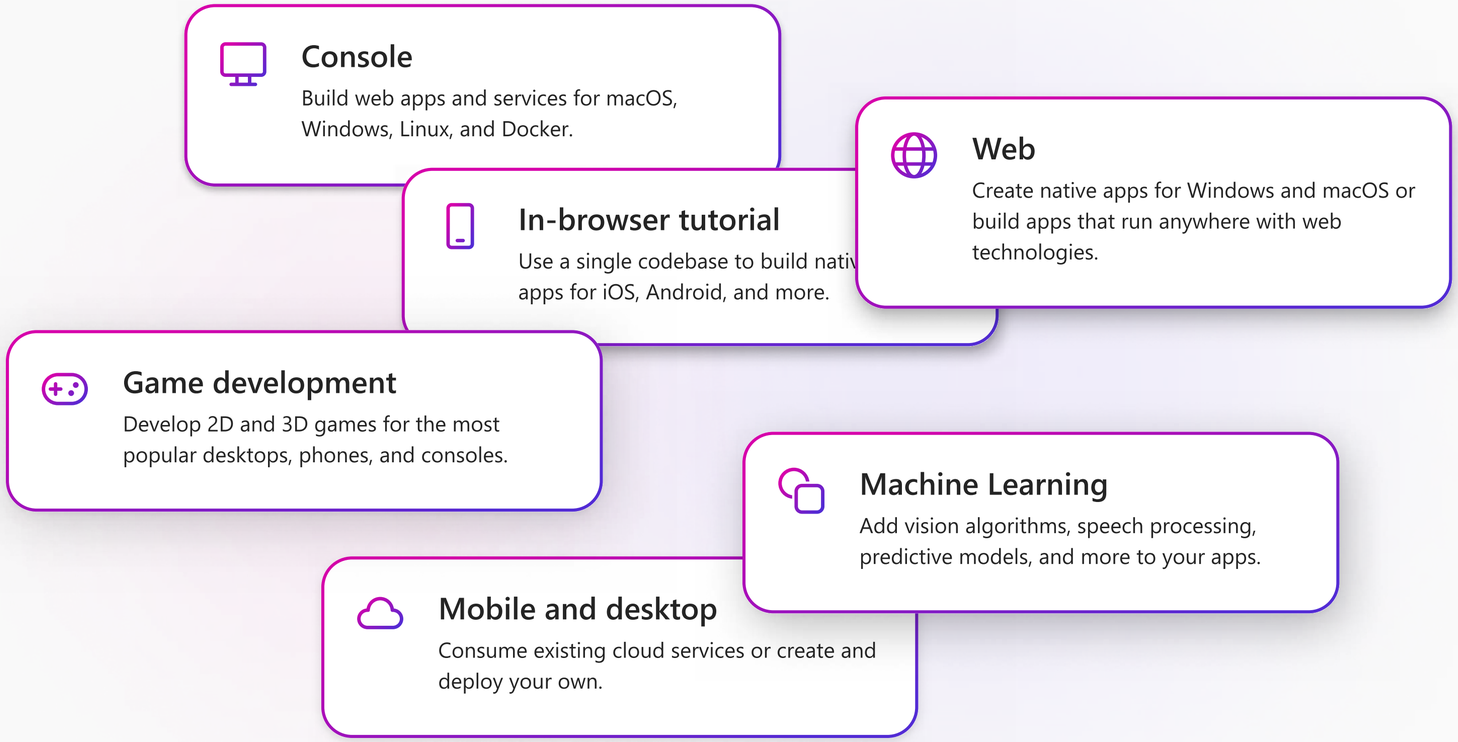
Hero illustration

Artworks that have more depth and storytelling and can be used as hero images or backgrounds.

Iconography

Icons are great at supporting concepts and ideas at a glance and are most effective when used to group similar types of content or break up large blocks of text into smaller components, making them easier to scan.

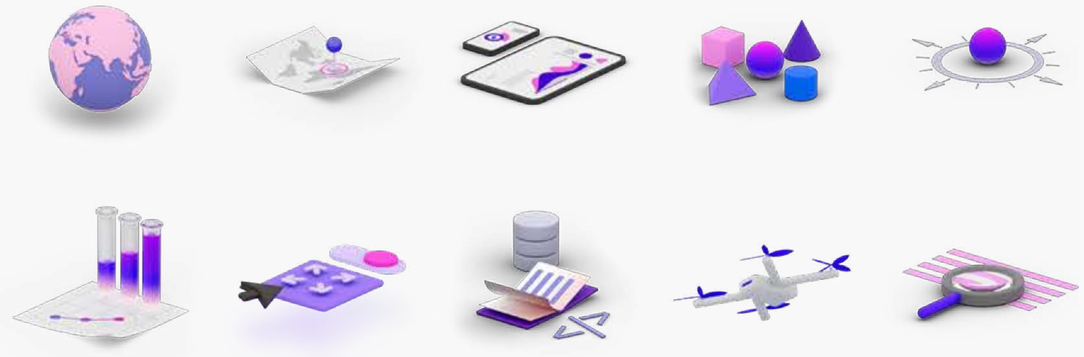
.NET icons are based on the Fluent design language.



Spot illustrations

Meant to accompany blocks of text, these illustrations are useful for wayfinding or to represent specific features. They are isometric, best used small (up to 200 pixels) and depict simple metaphors in 3D.

Consider making the spot illustrations universal across .NET for a clear user experience.



Data visualization

Data visualizations help with clarity for metrics, figures, and business-level information in an impactful way while leveraging brand elements like color, typography, and simple shapes.

General guidelines

Color

Be sure to use .NET brand colors to ensure consistency. Utilize gradients to create emphasis or dimension wherever necessary. Use high-contrast color combinations to clearly distinguish separate data points.

Typography

Make sure type feels balanced and appropriately scaled with the overall graphics.

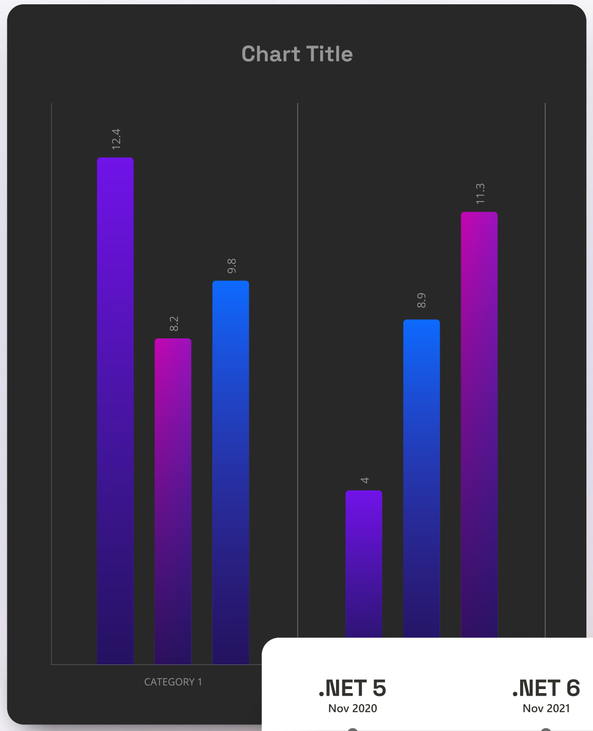
Accessibility (WCAG AA)

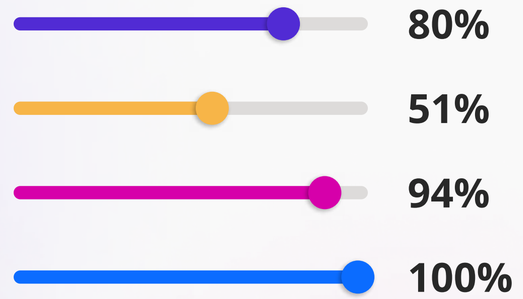
Graphics need to have at least a 3:1 contrast ratio.

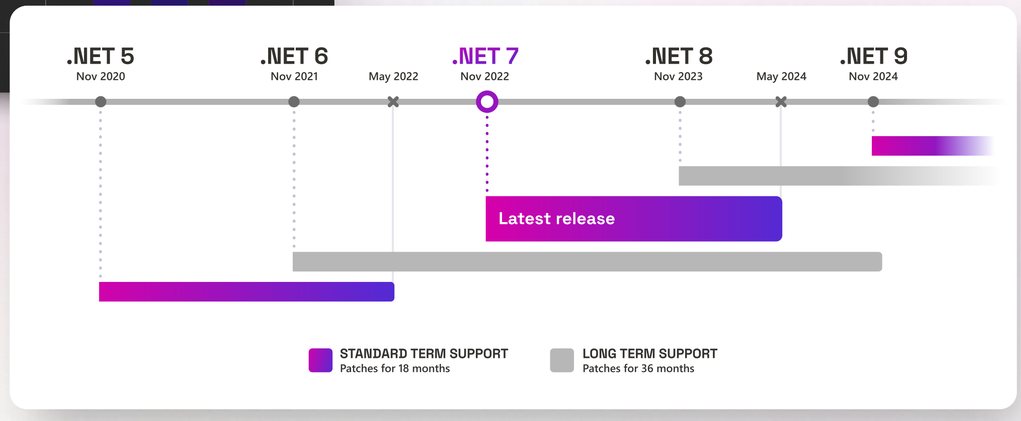
Normal text needs to have at least 4.5:1 contrast ratio.

Large text needs to have at least 3:1 contrast ratio.

Use this contrast checker tool to ensure your designs are accessible.







Code visualization

Code visualizations are important storytelling elements within the .NET platform. They can be enhanced with motion to create engaging, interactive experiences that capture the attention of developers and inspire them to learn more about what's possible with .NET.

```
1  service Greeter {
2   rpc SayHello (HelloRequest) returns (HelloReply) { }
3  }
```

```
1   await db.Blogs.Where(b => b.Slug == slug)
        .FirstAsync();
```

Corner radius: 16px

```
1  builder.Services
           .AddHttpClient<OrderServiceClient>(c =>
2              c.BaseAddress = new("http://orders");
3          )
4          .AddStandardResilienceHandler();
```

Space Mono – Regular, 18pt
- #00467E – Code
- #7A7A7A – Line #s

```
1  service Greeter {
2   rpc SayHello (HelloRequest) returns (HelloReply) { }
3  }
```

```
1   await db.Blogs.Where(b => b.Slug == slug)
        .FirstAsync();
```

```
1  builder.Services
           .AddHttpClient<OrderServiceClient>(c =>
2              c.BaseAddress = new("http://orders");
3          )
4          .AddStandardResilienceHandler();
```

Space Mono – Regular, 18pt
- #B6D3FF– Code
- #7A7A7A – Line #s

Hero illustration

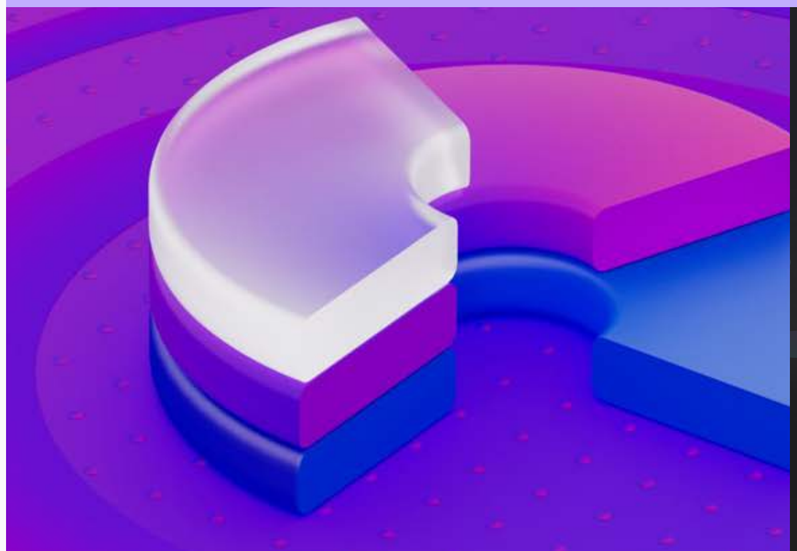
Style and usage

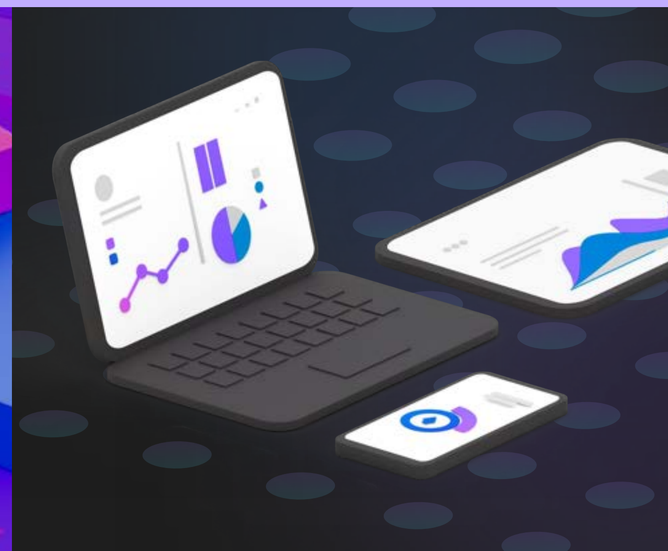
Isometric perspective

Hero illustrations are built using an isometric grid to create dynamic compositions by layering 2D/3D objects, UI components, and devices. All elements within individual compositions must be visually skewed at the same angle to ensure consistency.

Visual elements and metaphor

3D designs are used to represent apps, concepts, and UI components. These elements can also be combined with 2D wireframes, code samples, and devices.





dotnet-bot

dotnet-bot's story and value

The dotnet-bot was originally created by a member of the .NET community and has since become our mascot. The bot has gone through a series of makeovers over the years and we are currently using a new 3D style which enhances its personality and opens the opportunity for creative stories.

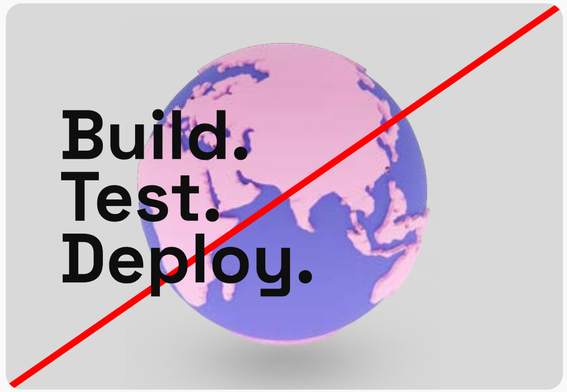
Don't overdo it

The value that the bot brings to the our storytelling is preferred for community events, swag, and collectables. In this way we avoid its exhaustion. Refrain from over-using the bot in communications, web experiences, or presentations that aren't related to the .NET community or events.

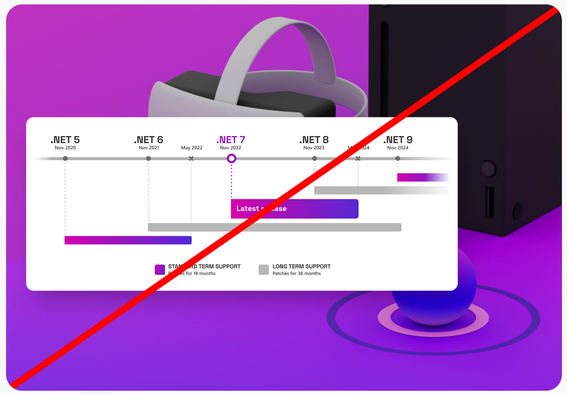
Don't reuse old artwork

Refrain from reusing old 2D bot artwork for any new communications.

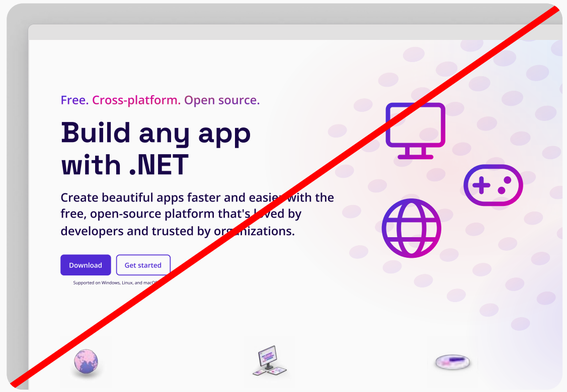
Things to avoid



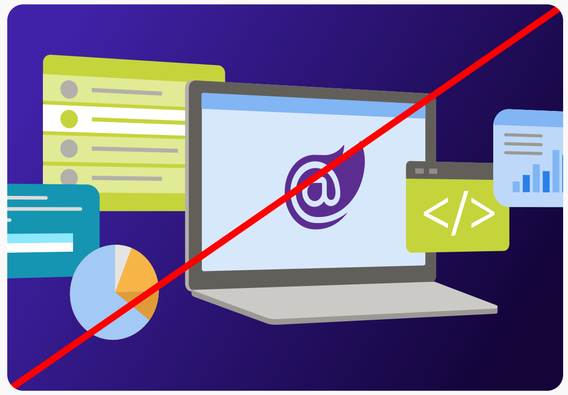
Don’t place type over illustrations.



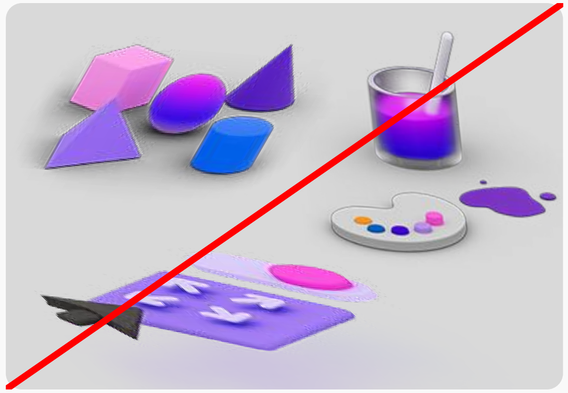
Don’t combine different illustration types within a single composition.



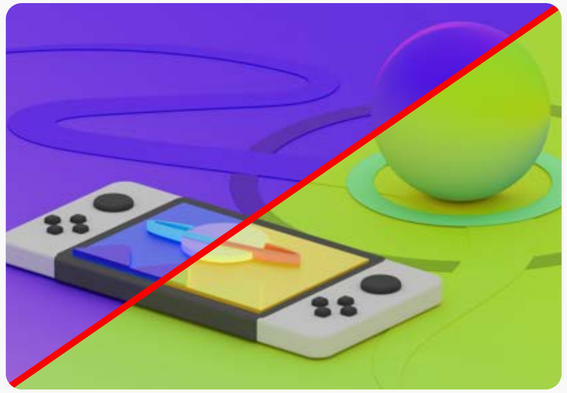
Don’t use 2D illustrations as hero images.



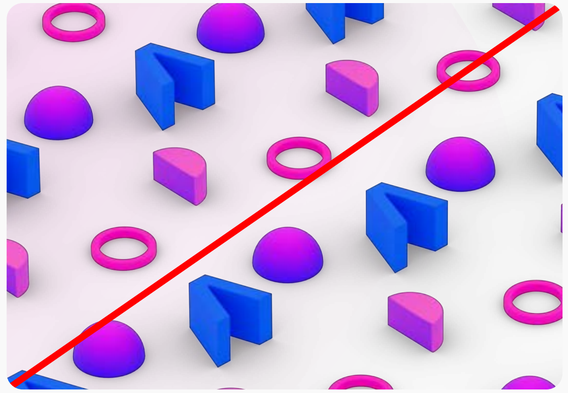
Don’t use outdated illustrations.



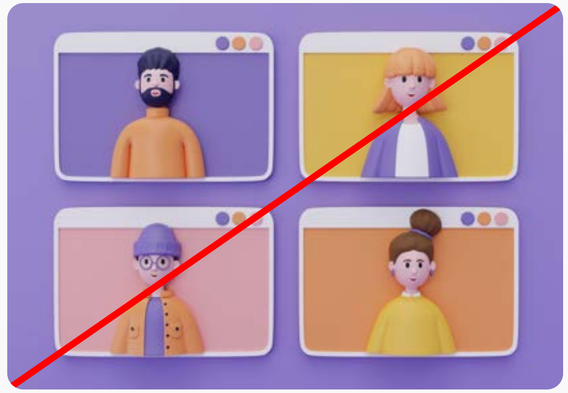
Don’t skew or stretch illustrations.



Don’t recolor, edit, or change existing assets.



Don’t use 3D illustrations as patterns or background textures.



Don’t depict humans or animals in illustrations.

04
Expression and application

Web

Build. Test. Deploy.

.NET is the free, open-source, cross-platform framework for building all your apps and powerful cloud services.

Download | Get started

Supported on Windows, Linux, and macOS

Build it with .NET

Web
Build web apps and services for macOS, Windows, Linux, and Docker.

Mobile and desktop
Use a single codebase to build native mobile apps for iOS, Android, and more.

Cloud
Consume existing cloud services or create and deploy your own.

Microservices
Create independently deployable microservices that run on docker containers.

Machine learning → | Game Development → | Internet of Things → | Desktop → | Front-end web → | Back-end APIs →

Faster response times, less compute power, better applications

The Fortunes test simulates a simple web application where HTML is rendered server-side after querying a database. See TechEmpower's Round 22 Results.

Fortunes responses per second

33.9K
Node.js
(Express)

342.5K
ASP.NET Core
(Minimal APIs)

62.6K
Go
(Gin)

Build beautiful, web apps with Blazor

Use the power of .NET and C# to build full-stack web apps without writing a line of JavaScript.

Get started | Read docs

Run anywhere

Host Blazor components in any web browser on WebAssembly, server-side in ASP.NET Core, or in native client apps.

Start today

Create beautiful user experiences fast. The flexible, reusable Blazor component models are simple, composable, declarative, and efficient.

Web and native

Use Blazor components on the web and in hybrid native apps for mobile & desktop.

Blazor for Beginners

The Blazor for Beginners video series shows you how to build a simple user interface using Blazor. Blazor expert Jeff Fritz unpacks exactly what Blazor is and how you can use it to make your own web applications—including layout, components, data access, and form field validation.

Watch on YouTube →

Ebook templates

Ebook covers should follow the following layout setup to create consistency in the Ebook series.

Page dimensions 1274 x 1650px

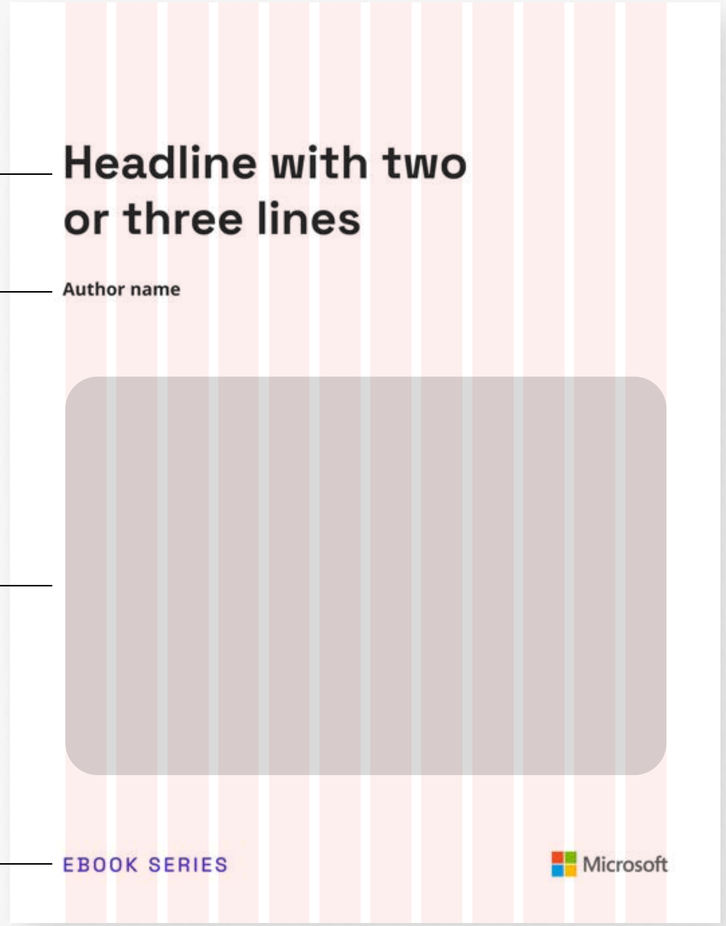
12 columns with 95px margin

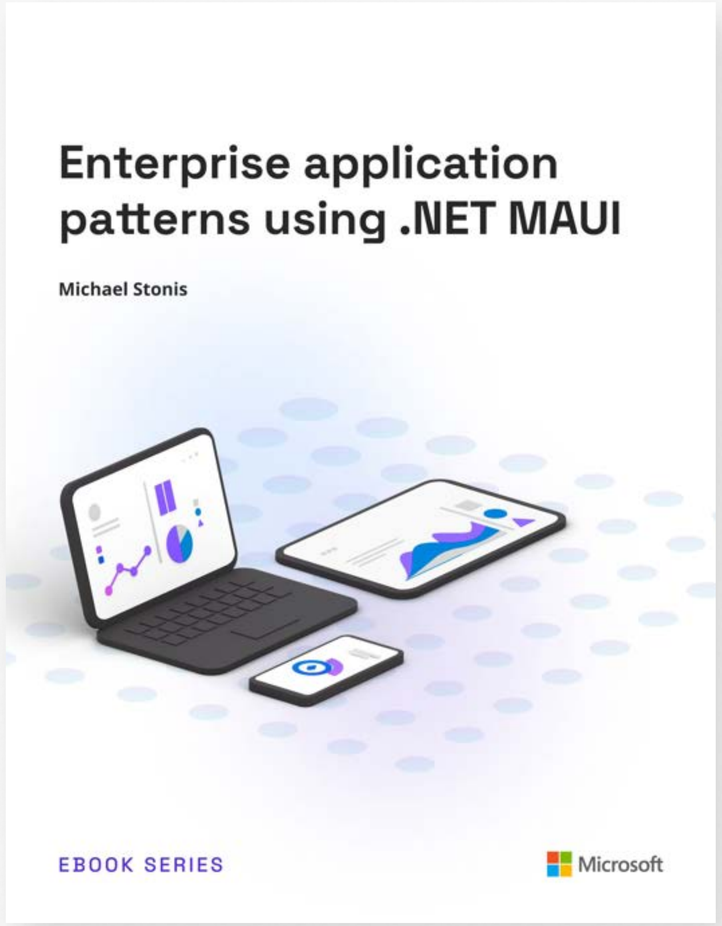
Headline: Space Grotesk bold 84pt

Author: Open Sans bold 32 pt

Illustration: Usable area 1084 x 665px

Ebook series: Space Grotesk Medium 36pt with 5px tracking





Swag

Streaming and video

Example of streaming assets for live events and lower thirds for video material.