

# Next.js Image Component Cheat Sheet

Hey there!

Matti Saari

Thank you for downloading this cheat sheet!

Image component is one of those features of Next.js that you should use but it's not always clear how to use it correctly. Or at least that's how I feel.

I've collected here different situations on how to show images, how to calculate correct aspect ratio and ChatGPT prompts to use for determining the aspect ratio. At the end of the document you'll also find example scenarios that you might find yourself in.

If you have any feedback or questions, shoot me an email at [tuomo@tuomokankaanpaa.com](mailto:tuomo@tuomokankaanpaa.com).

## Table of contents

---

## Local Image - Path String

For local images, you need to specify the `src` as a path string and provide the `width` and `height` props.

```
import Image from 'next/image'

export default function LocalImagePath() {
  return (
    <Image
      src="/images/profile.jpg"
      alt="Profile picture"
      width={500}
      height={500}
    />
  )
}
```

### When should I use this method?

Use this approach when you have images stored in your project's public directory and you know their dimensions. This is ideal for static assets that are part of your project, such as logos or icons.

---

## Local Image - Static Import

Using a static import allows Next.js to optimize the image at build time.

```
import Image from 'next/image'
import profilePic from '../public/images/profile.jpg'

export default function LocalImageImport() {
  return (
    <Image
      src={profilePic}
      alt="Profile picture"
      // width and height are automatically provided
      placeholder="blur"
    />
  )
}
```

```
)  
}
```

#### When should I use this method?

Use static imports when you want to leverage build-time optimization for your local images. This method is best for images that are part of your source code and won't change between builds. It's particularly useful for critical images that benefit from automatic optimization and lazy loading.

---

## Remote Image - Width & Height Known

For remote images, you need to add the domain to the `next.config.js` file and provide `width` and `height`.

```
// next.config.js  
module.exports = {  
  images: {  
    remotePatterns: [  
      {  
        protocol: 'https',  
        hostname: 's3.amazonaws.com',  
        port: '',  
        pathname: '/my-bucket/**',  
      },  
    ],  
  },  
}
```

`width` and `height` are provided for the image component as props.

```
import Image from 'next/image'  
  
export default function RemoteImageKnownDimensions() {  
  return (  
    <Image  
      src="https://example.com/profile.jpg"  
      alt="Profile picture"  
      width={500}  
      height={500}  
    />  
  )  
}
```

#### When should I use this method?

Use this method for external images hosted on other domains when you know their exact dimensions. This is useful for integrating images from content management systems, APIs, or other external sources where the image sizes are consistent or known in advance.

---

## Remote image - Width & Height Unknown

When dimensions are unknown, you can use the `fill` prop and wrap the Image in a container with `position: relative`. Use `objectFit` value `cover` or `contain` depending on your preference.

```
import Image from 'next/image'  
  
export default function RemoteImageUnknownDimensions() {
```

```
return (  
  <div style={{ position: 'relative', width: '100%', height: '300px' }}>  
    <Image  
      src="https://example.com/profile.jpg"  
      alt="Profile picture"  
      fill  
      style={{ objectFit: 'cover' }}  
    />  
  </div>  
)  
}
```

Remember to add the domain to your `next.config.js` file as shown in the previous example.

### When should I use this method?

Use the `fill` property when working with remote images of unknown or variable sizes, or when you need the image to adapt to its container's dimensions. This is particularly useful for responsive designs, hero images, or when integrating user-generated content where image dimensions may vary.

---

## Aspect Ratio

### What is aspect ratio?

Aspect ratio is the proportional relationship between the width and height of an image, video, or screen. It's typically expressed as two numbers separated by a colon, such as 16:9, 4:3 or 1:1.

In web development, understanding aspect ratio is particularly important when working with responsive designs and image components like the Next.js Image component. For example, when using the `fill` property with the Image component, you likely want to maintain a specific aspect ratio for the container so the image won't be distorted.

### How to calculate it?

Aspect ratio is calculated by dividing both the width and height by their greatest common divisor.

For example for an image with a width of 800 pixels and a height of 600 pixels, the aspect ratio is calculated as follows:

```
Greatest common divisor = 200  
  
Width: 800 / 200 = 4  
Height: 600 / 200 = 3  
  
Aspect ratio = 4:3
```

### ChatGPT Prompt

You can also use this prompt to ask ChatGPT for help:

```
I need your help determining the aspect ratio for an image. I will give you the width and height of th
```

I need your help determining the aspect ratio for an image. I will give you the width and height of the image and you will tell me what the aspect ratio is. Ask me for the width and height of the image if you understand.



I understand! Please provide me with the width and height of the image, and I'll help you determine the aspect ratio.

Width: 800  
Height: 600



The aspect ratio for an image with a width of 800 pixels and a height of 600 pixels is **4:3**.

Here's how it's calculated:

**1. Identify the Width and Height:**

- Width = 800 pixels
- Height = 600 pixels

**2. Find the Greatest Common Divisor (GCD):**

- The GCD of 800 and 600 is 200.

**3. Divide Both Dimensions by the GCD:**

- $\frac{800}{200} = 4$
- $\frac{600}{200} = 3$

**4. Express the Aspect Ratio:**

- The aspect ratio is 4:3.

This means the width is 4 units for every 3 units of height.

If you have your image at hand, you can also upload it to ChatGPT and ask what the aspect ratio is for that given image.



What is the aspect ratio of this image?



To determine the aspect ratio of the image you provided, I need to first extract its width and height. I'll do that now.

The image you provided has a width of 1000 pixels and a height of 750 pixels.

### Calculating the Aspect Ratio:

#### 1. Identify the Width and Height:

- Width = 1000 pixels
- Height = 750 pixels

#### 2. Find the Greatest Common Divisor (GCD):

- The GCD of 1000 and 750 is 250.

#### 3. Divide Both Dimensions by the GCD:

- $\frac{1000}{250} = 4$
- $\frac{750}{250} = 3$

#### 4. Express the Aspect Ratio:

- The aspect ratio is 4:3.

So, the aspect ratio of your image is 4:3. [\[-\]](#)



## Common Scenarios

► "I have a logo image in my project, what Image component method should I use?"

Use the local image with static import method. This allows for build-time optimization and is ideal for static assets like logos.

```
import Image from 'next/image'
import logo from '../public/images/logo.png'
```

```
export default function Header() {
  return (
    <Image
      src={logo}
      alt="Company Logo"
      placeholder="blur"
    />
  )
}
```

- "I'm building a product catalog with images from an external API, and I know the dimensions of the product images. How should I implement the Image component?"

Use the remote image with known height and width method. This is perfect for external images with consistent sizes.

```
import Image from 'next/image'

export default function ProductCard({ product }) {
  return (
    <Image
      src={product.imageUrl}
      alt={product.name}
      width={300}
      height={300}
    />
  )
}
```

- "I want to display a full-width hero image on my homepage, but the image dimensions may vary. What's the best way to use the Image component?"

Use the remote image when height and width are not known method. This allows the image to adapt to its container.

```
import Image from 'next/image'

export default function HeroSection() {
  return (
    <div style={{ position: 'relative', width: '100%', height: '50vh' }}>
      <Image
        src="https://example.com/hero-image.jpg"
        alt="Hero Image"
        fill
        style={{ objectFit: 'cover' }}
      />
    </div>
  )
}
```

- "I have several team member profile pictures stored in my project, and I want to optimize them during the build process. Which Image component method is most suitable?"

Use the local image with static import method again. This is ideal for optimizing multiple images that are part of your source code.

```
import Image from 'next/image'
import profilePic from '../public/images/team/john-doe.jpg'

export default function TeamMember() {
  return (
    <Image
      src={profilePic}
      alt="John Doe"
    />
  )
}
```

```
        placeholder="blur"  
      />  
    )  
  }  
}
```

- "I'm integrating user-generated content with images hosted on various platforms, and I don't have control over the image sizes. How should I implement the Image component?"

Use the remote image with width and height unknown -method. This is suitable when you're storing user-uploaded images in various platforms e.g. headless cms or a AWS bucket.

```
import Image from 'next/image'  
  
export default function UserContent({ image }) {  
  return (  
    <div style={{ position: 'relative', width: '100%', height: '300px' }}>  
      <Image  
        src={image.src}  
        alt={image.alt}  
        fill  
        style={{ objectFit: 'contain' }}  
      />  
    </div>  
  )  
}
```

---

## Learn more

---

*Follow me for more Next.js content*

