



Ministry of SMEs
and Startups

KOSME
Korea SMEs and Startups Agency

L LIKELION

JOBKOREA

LESSON 13

Images and Fonts Optimization

WEEK 03

Introduction to Optimization in Next.js

❖ What is Optimization?

- Process of improving performance by reducing file sizes and load times for assets like images and fonts.

❖ Importance in Next.js

- Enhances user experience, SEO, and Core Web Vitals in server-rendered React apps.

❖ Official Reference

- Next.js Documentation

Why Optimize Images and Fonts?

❖ Performance Issues

- Large images and fonts cause slow loading, high bandwidth usage.

❖ User Impact

- Faster sites reduce bounce rates and improve engagement.

❖ SEO Benefits

- Google prioritizes optimized sites in search rankings.

Image Optimization Overview

❖ Challenges with Images

- High-resolution images increase page weight and load time.

❖ Next.js Solution

- Built-in optimization via next/image for automatic resizing and format conversion.

❖ Key Features

- Lazy loading, responsive sizes, WebP support.

Introducing Next/Image Component

❖ What is Next/Image?

- A React component that extends with optimization features.

❖ Benefits

- Automatic optimization, placeholder support, and layout stability.

❖ Official Reference

- Next/Image Docs

Basic Usage of Next/Image

❖ Import and Use

- Import from next/image and use like
`<Image src="/example.jpg" alt="Example" width={500} height={300} />`.

❖ Required Props

- src, alt, width, height for proper rendering.

Responsive Images

❖ Sizes Prop

- Defines image width based on viewport
(e.g., sizes="(max-width: 768px) 100vw, 50vw").

❖ SrcSet Generation

- Next.js auto-generates multiple sizes for different devices.

Lazy Loading and Placeholders

❖ Lazy Loading

- Images load only when entering viewport, reducing initial load.

❖ Placeholder

- Use placeholder="blur" for blurred low-res preview.

Image Formats and Quality

❖ Automatic Format

- Converts to WebP or AVIF for modern browsers.

❖ Quality Prop

- Adjust compression level (e.g., quality={75} for balance).

Custom Image Loaders

❖ What is a Loader?

- Function to customize image URL generation for CDNs.

❖ Usage

- Define in next.config.js or per-image.

Font Optimization Overview

❖ Font Challenges

- Custom fonts increase load time if not optimized.

❖ Next.js Solution

- next/font for automatic self-hosting and subsetting.

❖ Benefits

- Reduces requests, improves privacy and performance.

Introducing Next/Font

❖ What is Next/Font?

- Module to import and optimize fonts in Next.js apps.

❖ Features

- Automatic subsetting, preloading, and CSS injection.

❖ Official Reference

- Next/Font Docs

Using Google Fonts

❖ Import Google Fonts

- Use @next/font/google for optimized import.

```
import { Inter } from 'next/font/google';
const inter = Inter({ subsets: ['latin'] });
export default function RootLayout({ children }) {
  return <html lang="en" className={inter.className}>{children}</html>;
}
```

Using Local Fonts

❖ Local Font Import

- Use @next/font/local for self-hosted fonts.

```
import localFont from 'next/font/local';
const myFont = localFont({ src: './my-font.woff2' });
```

Font Display and Preloading

❖ Display Prop

- Controls font rendering (e.g., display: 'swap' to avoid invisible text).

❖ Preloading

- Next.js auto-preloads fonts for faster availability.

```
const inter = Inter({ subsets: ['latin'], display: 'swap' });
```

Subsetting Fonts

❖ What is Subsetting?

- Load only necessary characters (e.g., Latin subset).

❖ Benefits

- Reduces font file size significantly.

❖ Example

- Use subsets: ['latin', 'latin-ext'] in font import.

Performance Impact

❖ Metrics Improved

- Lower TTFB, better LCP and FID with optimized assets.

❖ Tools to Measure

- Use Lighthouse or WebPageTest for audits.

❖ Reference

- Google PageSpeed Insights

Best Practices for Image Optimization

❖ Choose Right Format

- Use SVG for icons, JPEG for photos, WebP for general.

❖ Compress Images

- Tools like TinyPNG before uploading.

❖ Avoid Over-Sizing

- Set appropriate width and height.

Best Practices for Font Optimization

❖ Limit Font Variants

- Use only necessary weights and styles.

❖ Self-Host Fonts

- Avoid external requests for critical fonts.

❖ Combine with CSS

- Apply fonts selectively to elements.

Common Pitfalls

- ❖ **Missing Alt Text**
 - Impacts accessibility and SEO.
- ❖ **Overusing Custom Fonts**
 - Increases load time; stick to 2-3 fonts.
- ❖ **Ignoring Mobile**
 - Test responsive images on devices.

Conclusion

❖ Key Takeaways

- Use next/image and next/font for built-in optimization in Next.js.

❖ Next Steps

- Implement in a project, measure performance improvements.

❖ Resources

- Next.js Optimization Guide