

IBM Carbon Design System

Enterprise Design System Guide

Open Source Design System

Complete Reference & Implementation Guide

Version 2024 Edition

"With the IBM Design Language as its foundation, the system consists of working code, design tools and resources, human interface guidelines, and a vibrant community"

Table of Contents

- 1. Introduction to Carbon Design System3
- 2. Design Philosophy & Principles4
- 3. Getting Started6
- 4. Design Tokens & Foundations8
- 5. Typography System12
- 6. Color System14

7. Spacing & Layout	17
8. Iconography	19
9. Component Library	21
10. Patterns & Templates	28
11. Accessibility Guidelines	31
12. Implementation Guide	34
13. Best Practices	37
14. Migration & Adoption	39
15. Community & Resources	41

1. Introduction to Carbon Design System

IBM Carbon Design System is a comprehensive, open-source design system created by IBM to enable teams to build consistent, accessible digital products at scale. Built on the foundation of IBM Design Language, Carbon provides a unified approach to design and development across IBM's extensive portfolio of enterprise software.

What Makes Carbon Unique

Carbon stands out in the design system landscape through its enterprise-first approach, robust accessibility features, and comprehensive component library specifically designed for complex business applications. It bridges the gap between design and development with living code examples and detailed implementation guidance.

Core Components of Carbon

The Carbon Design System encompasses four main pillars that work together to create cohesive digital experiences:

Working Code

Carbon provides production-ready components built with modern web technologies including React, Angular, Vue, and vanilla JavaScript. Each component includes comprehensive APIs, accessibility features, and responsive behavior out of the box.

Design Tools and Resources

Comprehensive design assets including Sketch libraries, Figma kits, Adobe XD resources, and design tokens ensure seamless handoff between design and development teams. These tools maintain consistency across all touchpoints of the design process.

Human Interface Guidelines

Detailed documentation covers interaction patterns, visual design principles, accessibility requirements, and content guidelines. These guidelines ensure that products built with Carbon meet enterprise-grade standards for usability and accessibility.

Vibrant Community

An active open-source community contributes to Carbon's continuous evolution, providing feedback, contributions, and real-world usage examples that help refine and expand the system.

Enterprise Focus

Carbon is specifically designed for enterprise applications, addressing complex use cases such as data visualization, form-heavy interfaces, dashboard design, and workflow management that are common in business software.

2. Design Philosophy & Principles

Carbon's design philosophy is rooted in IBM's rich design heritage while embracing modern digital product needs. The system is built on fundamental principles that guide every design decision and component creation.

Clarity

Every element serves a purpose. Carbon prioritizes clear communication and removes unnecessary complexity to help users accomplish their tasks efficiently. This principle manifests in clean interfaces, purposeful interactions, and straightforward information architecture.

Efficiency

Carbon optimizes for productivity and workflow efficiency. Components are designed to minimize cognitive load and support rapid task completion, particularly important in enterprise environments where users interact with software throughout their workday.

Consistency

Predictable patterns and behaviors across all touchpoints create familiarity and reduce learning curves. Carbon's systematic approach ensures that once users learn one Carbon-based application, they can navigate others with confidence.

Beauty

Aesthetic appeal enhances usability and creates emotional connection with software. Carbon balances functional requirements with visual refinement, creating interfaces that are both beautiful and highly functional.

The IBM Design Language Foundation

Carbon builds upon IBM's broader Design Language, which emphasizes human-centered design, systematic thinking, and restless reinvention. This foundation ensures that Carbon-based products align with IBM's overall design strategy while remaining flexible for diverse use cases.

Accessibility as a Core Principle

Accessibility isn't an afterthought in Carbon—it's fundamental to every component and pattern. The system adheres to WCAG 2.1 AA standards and incorporates inclusive design practices throughout.

Universal Design Approach

Carbon's components are designed to work for users with diverse abilities, using techniques such as sufficient color contrast, keyboard navigation support, screen reader compatibility, and flexible interaction methods.

Progressive Enhancement

All Carbon components work without JavaScript and are enhanced progressively, ensuring baseline functionality across all environments and assistive technologies.

3. Getting Started

Implementing Carbon Design System in your project involves several key steps, from initial setup to component integration. This section provides a comprehensive roadmap for teams beginning their Carbon journey.

Installation Options

Carbon offers multiple installation methods to accommodate different project types and team preferences:

Package Managers

```
# npm npm install @carbon/react # yarn yarn add @carbon/react # pnpm
pnpm add @carbon/react
```

CDN Integration

```
<link rel="stylesheet"
href="https://unpkg.com/@carbon/styles/css/styles.css"> <script
src="https://unpkg.com/@carbon/web-components/dist/carbon-web-components.js"></script>
```

Framework Support

Carbon provides native support for major JavaScript frameworks and libraries:

React

Full component library with hooks and TypeScript support

Angular

Native Angular components with form integration

Vue

Vue.js compatible components and directives

Web Components

Framework-agnostic web components for any environment

Vanilla JavaScript

Pure JavaScript implementation for maximum flexibility

Svelte

Community-maintained Svelte bindings

Basic Setup Example

```
import React from 'react'; import { Button, Theme } from
 '@carbon/react'; import '@carbon/styles/css/styles.css'; function
 App() { return ( <Theme theme="white"> <div className="App">
 <Button>Get Started with Carbon</Button> </div> </Theme> ); } export
 default App;
```

Theme Configuration

Carbon supports multiple themes including white, gray10, gray90, and gray100. Themes can be applied globally or to specific sections of your application, enabling flexible branding and user preference support.

Design Asset Setup

For design teams, Carbon provides comprehensive asset libraries:

Figma Kit

The Carbon Design Kit for Figma includes all components, icons, and design tokens. Teams can duplicate the kit to their workspace and stay updated with the latest design assets.

Sketch Libraries

Sketch users can access Carbon's component libraries through Sketch Cloud, ensuring automatic updates and symbol synchronization across team members.

Adobe XD Resources

Adobe XD asset packages provide component libraries and design specifications for teams using Adobe's design tools.

4. Design Tokens & Foundations

Design tokens form the foundation of Carbon's visual language, providing a systematic approach to maintaining consistency across all products and platforms. These atomic design decisions ensure coherence while enabling flexibility for different contexts and brand requirements.

Token Categories

Carbon organizes design tokens into several key categories that address different aspects of visual design:

Color Tokens

Carbon's color system uses semantic naming conventions that describe the token's purpose rather than its appearance, enabling theme flexibility and maintaining meaning across different visual contexts.

Token Name	Purpose	Example Usage
------------	---------	---------------

\$background	Primary background color	Page backgrounds, card surfaces
\$text-primary	Primary text color	Headings, body text
\$interactive	Interactive elements	Links, buttons, form controls
\$danger	Error and warning states	Error messages, destructive actions

Spacing Tokens

Consistent spacing creates visual rhythm and hierarchy. Carbon uses a base-8 spacing system that provides flexibility while maintaining proportional relationships.

\$spacing-02 (2px) - Minimal spacing for tight layouts

\$spacing-04 (4px) - Small component spacing

\$spacing-05 (8px) - Standard component spacing

\$spacing-06 (16px) - Layout spacing

\$spacing-07 (24px) - Section spacing

Typography Tokens

Typography tokens define font families, sizes, weights, and line heights that create consistent text hierarchies across all Carbon implementations.

Token Implementation

Design tokens are available in multiple formats to support different tools and workflows:

CSS Custom Properties


```
:root { --cds-background: #ffffff; --cds-text-primary: #161616;
--cds-interactive: #0f62fe; --cds-spacing-05: 1rem; } .my-component {
background: var(--cds-background); color: var(--cds-text-primary);
padding: var(--cds-spacing-05); }
```

Sass Variables

```
@use '@carbon/styles/scss/config' with ( $prefix: 'cds' ); @use
 '@carbon/styles/scss/theme'; .my-component { background:
theme.$background; color: theme.$text-primary; padding:
theme.$spacing-05; }
```

JavaScript/JSON

```
import { theme } from '@carbon/themes'; import { spacing } from
 '@carbon/layout'; const styles = { background: theme.background,
color: theme.textPrimary, padding: spacing.spacing05 };
```

Token Naming Convention

Carbon uses a systematic naming convention: category-property-variant-state. For example, "button-primary-background-hover" clearly indicates the token's purpose and usage context, making it easy for teams to find and implement the correct values.

5. Typography System

Carbon's typography system is built on IBM Plex, a custom typeface family designed specifically for IBM's digital and print communications. The system provides a comprehensive type scale and clear hierarchy guidelines that enhance readability and create visual consistency across all touchpoints.

IBM Plex Font Family

IBM Plex consists of four main subfamilies, each optimized for specific use cases:

IBM Plex Sans

The primary typeface for user interfaces and digital communications. Plex Sans offers excellent readability at all sizes and includes a wide range of weights and language support.

IBM Plex Serif

Used for editorial content and brand communications where a more traditional, literary feel is appropriate. The serif variant maintains Plex's distinctive character while providing classical typography aesthetics.

IBM Plex Mono

Essential for code display, technical documentation, and any context requiring monospaced characters. Designed to be highly legible and comfortable for extended reading of code.

IBM Plex Condensed

Space-efficient variant for headlines, labels, and situations where horizontal space is limited while maintaining readability.

Font Loading Strategy

Carbon provides optimized font loading strategies including font-display: swap for better performance, and fallback font stacks that maintain similar metrics to IBM Plex during load times.

Type Scale

Carbon's type scale follows a systematic approach with preset sizes that create clear hierarchical relationships:

Token	Size	Line Height	Usage
\$heading-01	14px	18px	Small headings, labels

\$heading-02	16px	22px	Sub-headings, section titles
\$heading-03	20px	26px	Component headings
\$heading-04	28px	36px	Page headings
\$heading-05	32px	40px	Major section headings
\$heading-06	42px	50px	Display headings
\$heading-07	54px	64px	Hero headings

| Typography Guidelines

Hierarchy Best Practices

Establish clear information hierarchy by using consistent heading levels, appropriate sizing relationships, and sufficient contrast between different text elements. Avoid skipping heading levels and maintain logical content structure.

Line Length and Readability

Optimal line length for body text is 45-75 characters per line. Carbon's grid system naturally supports these proportions, but consider using measure utilities for long-form content.

Color and Contrast

All typography meets WCAG 2.1 AA contrast requirements. Use semantic color tokens to ensure text remains readable across different themes and maintains meaning for users with visual impairments.

```
/* Typography implementation example */ .page-title { @include
type.type-style('heading-05'); color: var(--cds-text-primary);
margin-bottom: var(--cds-spacing-06); } .section-heading { @include
type.type-style('heading-03'); color: var(--cds-text-primary);
margin: var(--cds-spacing-07) 0 var(--cds-spacing-05) 0; } .body-text
{ @include type.type-style('body-01'); color:
var(--cds-text-secondary); max-width: 66ch; /* Optimal reading width
*/ }
```

6. Color System

Carbon's color system provides a comprehensive palette designed for enterprise applications, emphasizing accessibility, usability, and visual harmony. The system includes functional colors for interface elements and expressive colors for data visualization and brand expression.

Color Philosophy

Carbon's approach to color balances several key considerations:

Accessibility First

All color combinations meet or exceed WCAG 2.1 AA contrast requirements. The system provides automatic color pairing recommendations and warns against inaccessible combinations.

Semantic Meaning

Colors carry consistent meaning across all Carbon implementations. Red indicates errors or danger, blue represents primary actions, green signals success, and yellow warns of caution.

Cultural Sensitivity

Color choices consider global cultural contexts and avoid assumptions about color meaning that may not translate across different user populations.

Core Color Palette

Blue 60

#0f62fe

Primary interactive

WCAG AA Compliant

Gray 100

#161616

Primary text

High contrast

White

#ffffff

Background

Maximum contrast

Red 60

#da1e28

Error/danger

Warning states

Green 60

#24a148

Success states

Positive feedback

Yellow 30

#f1c21b

Warning/caution

Attention states

Theme Support

Carbon supports multiple themes that can be applied globally or to specific components:

White Theme