Angular is a powerful, open-source framework developed by Google for building dynamic, single-page web applications (SPAs). It uses TypeScript, a superset of JavaScript, to provide developers with robust tools for creating scalable, maintainable, and efficient web applications.

Key Features of Angular

1. **Component-Based Architecture**:
   Applications are structured into reusable, self-contained components, making the code modular and easier to manage.

2. **Two-Way Data Binding**:
   Angular automatically synchronizes data between the model (business logic) and the view (UI), enabling a seamless user experience.

3. **Dependency Injection**:
   Angular's built-in dependency injection system helps manage services and dependencies efficiently, improving code modularity and testability.

4. **Directives**:
   Angular provides powerful directives for extending HTML's behavior. For example, `ngIf` and `ngFor` allow conditional rendering and iteration over data.

5. **TypeScript Integration**:
   Angular fully supports TypeScript, bringing strong typing and modern JavaScript features that improve code quality and readability.

6. **Routing**:
   The Angular Router enables developers to build SPAs by managing navigation and URL routing efficiently.

7. **Comprehensive Tooling**:
   Angular includes a CLI (Command Line Interface) for scaffolding projects, generating components, and optimizing builds, streamlining development.

8. **Reactive Programming**:
   Angular supports reactive programming through RxJS, a library for composing asynchronous and event-based programs using observable sequences.

9. **Performance Optimization**:
   Features like Ahead-of-Time (AOT) compilation, lazy loading, and efficient change detection enhance application performance.

Basic Building Blocks of Angular

1. **Modules** (`NgModule`):
   Organize an application into cohesive blocks of functionality. The `AppModule` is the root module in every Angular application.

2. **Components**:
   Define the UI and behavior. Each component includes an HTML template, TypeScript class, and CSS/SCSS for styling.

3. **Templates**:
   Define the view's structure using Angular's extended HTML syntax with bindings and directives.

4. **Services**:
   Encapsulate business logic or shared functionality and can be injected into components using dependency injection.

5. **Directives**:

   - **Structural Directives**: Modify the DOM structure, e.g., `*ngIf`, `*ngFor`.
   - **Attribute Directives**: Modify the appearance or behavior of elements, e.g., `[ngStyle]`, `[ngClass]`.

6. **Pipes**:
   Transform data in templates, e.g., `| date`, `| uppercase`.

Advantages of Angular

- Strong community support and regular updates from Google.
- Built-in tools for testing, such as Karma and Jasmine.
- Comprehensive ecosystem for enterprise-level applications.
- Scalability for complex, large-scale projects.

Example: A Simple Angular Component

```
import { Component } from '@angular/core';

@Component({
  selector: 'app-hello-world',
  template: `<h1>{{ title }}</h1>`,
  styles: [`h1 { font-family: Lato; }`]
})
export class HelloWorldComponent {
  title = 'Hello, Angular!';
}
```

This is a basic introduction to Angular. Let me know if you'd like more details on a specific topic, such as setting up an Angular project, routing, or advanced features!