# Chapter 02

1. The theming concept used in PrimeFaces is similar to what?

Answer: The jQuery ThemeRoller CSS Framework.

1. All PrimeFaces components are designed to allow a developer to what?

Answer: Integrate them seamlessly into the look and feel of an entire web application.

1. Powered by ThemeRoller, PrimeFaces does what?

Answer: Separates structural CSS from skinning CSS.

1. Each component is styled with CSS and contains what?

Answer: Two layers of style information—structural or component-specific and skinning or component-independent styles.

1. The first two style classes’ ui-picklist-list and ui-picklist-source are generated by what and what are they provide?

Answer: They are generated by PrimeFaces and provide a semantic presentation to indicate the role of an element within a component.

1. PrimeFaces leverages what?

Answer: The jQuery ThemeRoller CSS Framework.

1. Describe the ui-widget-content and ui-corner-all classes.

Answer: The ui-widget-content and ui-corner-all classes are defined by ThemeRoller and they affect the look and feel. These are skinning style classes, which define CSS properties such as text colors, border colors, and background images.

1. What are the main advantage of the presented PrimeFaces selectors?

Answer: The main advantage of the presented PrimeFaces selectors is a great flexibility in theming, because you don't need to know each and every skinning selector to change the styles of all available components in your web application consistently.

1. What is Maven?

Answer: Maven is a build and project management tool, which manages installation of all dependencies in an easy way. PrimeFaces is a Maven-based project and offers all artifacts, including themes, as Maven dependencies.

1. Describe the JSF standard Head component.

Answer: The PrimeFaces component library has a special implementation for the JSF standard Head component. PrimeFaces provides the HeadRenderer class, which is responsible for the rendering of the <h:head> tag. HeadRenderer automatically detects the current configured theme in web.xml—regardless if static or dynamic via managed bean—and renders theme-related resources on the page.

1. Describe Aristo.

Answer: Aristo is the built-in default theme of PrimeFaces. There is no separate JAR file for it; the theme is delivered with the core PrimeFaces JAR file itself. Therefore, you don't need to install it via Maven or have it extra in the classpath.

1. How Custom styles will work?

Answer: Custom styles will be rendered after the PrimeFaces theme (skinning) and component (structural) styles. The correct output is ensured by the PrimeFaces resource ordering. Therefore, custom styles, being rendered after PrimeFaces ones, overwrite the default settings.

1. Can we disable theming completely with the same look and feel in all browsers?

Answer: No.

1. What is the disadvantage in a multi-theme application?

Answer: Each PrimeFaces theme has a specific font family and font size, which can be different from theme to theme. This may have a disadvantage in a multi-theme application because switching from one theme to another would cause a broken layout.

1. How we can change fonts globally?

Answer: A simple way to change fonts globally can be accomplished by using the .ui-widget style class.

1. Describe the ui-widget style class.

Answer: All PrimeFaces components are styled by the ui-widget style class. It is a skinning style specified by jQuery ThemeRoller and applied to HTML elements rendered by PrimeFaces.

1. Describe The default PrimeFaces' theme switcher.

Answer: The default PrimeFaces' theme switcher is a component, which enables switching themes on the fly without a round-trip to the server. We speak about a stateless theme switcher because the current selected theme is only known on the client side.

1. By default, PrimeFaces theme switcher do what?

Answer: By default, PrimeFaces theme switcher only changes the theme on the fly without sending an AJAX or full-page request.