

Arkusze CSS

Zadania

- Sformatować Dokument HTML za pomocą arkusza CSS

Wymagania

- Formatowanie elementów strukturalnych HTML (ciało dokumentu, nagłówki, akapity, tabele, ...)
- Wykorzystanie klas CSS
np. p – akapit, p.rys – podpis rysunku, ...
- Wykorzystanie relacji selektorów CSS
np. #menu a, #main a
- Wykorzystanie pseudoklas łączy

Arkusze CSS

Domyślne formatowanie przeglądarki

Version Control with Subversion

Preface

It is important not to let the perfect become the enemy of the good, even when you can agree on what perfect is. Doubly so when you can't. As unpleasant as it is to be trapped by past

- Greg Hudson, Subversion developer

In the world of open source software, the Concurrent Versions System (CVS) was the tool of choice for version control for many years. And rightly so. CVS was open source software itse of geographically dispersed programmers to share their work. It fit the collaborative nature of the open source world very well. CVS and its semi-chaotic development model have since b

But CVS was not without its flaws, and simply fixing those flaws promised to be an enormous effort. Enter Subversion. Subversion was designed to be a successor to CVS, and its originator design (and "look and feel") similar to CVS, and by attempting to avoid most of CVS's noticeable flaws. While the result isn't necessarily the next great evolution in version control design newly started open source projects now choose Subversion instead of CVS.

Chapter 1. Fundamental Concepts

It is upon the Trunk that a gentleman works.

- Confucius

This chapter is a short, casual introduction to Subversion. If you're new to version control, this chapter is definitely for you. We begin with a discussion of general version control concep Subversion in use.

Even though the examples in this chapter show people sharing collections of program source code, keep in mind that Subversion can manage any sort of file collection-it's not limited to

The Repository

Subversion is a centralized system for sharing information. At its core is a repository, which is a central store of data. The repository stores information in the form of a filesystem tree- and then read or write to these files. By writing data, a client makes the information available to others; by reading data, the client receives information from others. Figure 1.1, "A typic

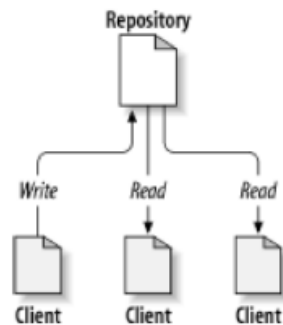


Figure 1.1. A typical client/server system

Arkusze CSS

Formatowanie CSS

Version Control with Subversion

Preface

It is important not to let the perfect become the enemy of the good, even when you can agree on what perfect is. Doubly so when you can't. As unpleasant as it is to be trapped by past mistakes, you can't make any progress by being afraid of your own shadow during design.

- Greg Hudson, Subversion developer

In the world of open source software, the Concurrent Versions System (CVS) was the tool of choice for version control for many years. And rightly so. CVS was open source software itself, and its nonrestrictive modus operandi and support for networked operation allowed dozens of geographically dispersed programmers to share their work. It fit the collaborative nature of the open source world very well. CVS and its semi-chaotic development model have since become cornerstones of open source culture.

But CVS was not without its flaws, and simply fixing those flaws promised to be an enormous effort. Enter Subversion. Subversion was designed to be a successor to CVS, and its originators set out to win the hearts of CVS users in two ways-by creating an open source system with a design (and "look and feel") similar to CVS, and by attempting to avoid most of CVS's noticeable flaws. While the result isn't necessarily the next great evolution in version control design, Subversion is very powerful, very usable, and very flexible. And for the most part, almost all newly started open source projects now choose Subversion instead of CVS.

Chapter 1. Fundamental Concepts

It is upon the Trunk that a gentleman works.

- Confucius

This chapter is a short, casual introduction to Subversion. If you're new to version control, this chapter is definitely for you. We begin with a discussion of general version control concepts, work our way into the specific ideas behind Subversion, and show some simple examples of Subversion in use.

Even though the examples in this chapter show people sharing collections of program source code, keep in mind that Subversion can manage any sort of file collection-it's not limited to helping computer programmers.

The Repository

Subversion is a centralized system for sharing information. At its core is a repository, which is a central store of data. The repository stores information in the form of a filesystem tree-a typical hierarchy of files and directories. Any number of clients connect to the repository, and then read or write to these files. By writing data, a client makes the information available to others; by reading data, the client receives information from others. Figure 1.1, "A typical client/server system" illustrates this.

