

Course Title

XPP (9.4/9.5) CSS Style Development eLearning

Variations on course:

For XPP 9.2 or XPP 9.3 users specific course materials will be delivered.

Learning Objectives:

You will learn best practices for developing CSS style sheets for XPP.

Topics covered include:

- Overview of XML
- Overview of XPP concepts
- How to use CSS with XPP
- Hands-on exercises

Key Benefits:

This course allows developers creating XPP styles to leverage the industry standard CSS style development language while harnessing the power and unique capabilities XPP provides to produce automated, high typographic quality content deliverables.

Target Audience:

This course is for:

- Style Developers

Prerequisites:

The prerequisites for this course are:

- XPP Fundamentals or equivalent understanding of the XPP environment
- Familiarity with XML or structured markup
- Basic style, typesetting and page composition concepts
- Knowledge of the requirements for the publications/deliverables you need to produce

Further benefits will accrue to those who have experience with XML markup, CSS, Perl, and/or experience as an operator or style developer for

“classic” (that is, non-XML) or non-CSS (that is, traditional XML) XPP.

CSS styling was introduced with XPP 9.0 and has been enhanced in subsequent version releases as outlined in the XPP Release Notes. This course was developed with XPP 9.4 and is also applicable for XPP 9.5 since no additional CSS changes were made in that version release.

If you are still using XPP 9.2 or XPP 9.3, please contact us to obtain instructions for accessing the XPP CSS course which was developed for your specific version release.

Course Delivery Mode:

This course is available in an eLearning delivery mode and is accessed via the RWS University online learning portal. The self-paced, on-demand course will take an estimated 11 hours to complete. You should allow up to a further 8 hours to complete the practical tasks. The eLearning format enables the learner to bookmark their place in the course and restart when ready. Once the course has been taken the learner can go back to any of the modules.

Further Information:

For further information, please contact us at learn@rws.com.

Topics covered during the course

1 **Module 1: Before you start**

- Downloading and installing the course material

2 **Module 2: Setting up your working environment**

- Setting up an example working environment
- Using other editors
- Setting up a Perl validation tool in Notepad++

3 **Module 3: Introduction to XML**

- XML naming conventions and best practices
- Elements, attributes and XML structure

4 **Module 4: Introduction to CSS for XPP**

- The structure of CSS stylesheets
- Learning about selectors
- Learning about properties

5 **Module 5: Course results preview**

- Reviewing the XML structure
- Reviewing the delivered JOBs and finished XPP publication

6 **Module 6: Introduction to page layout and typographic terms**

- Basic page layout terms and descriptions
- Common terms such as: units, leading, etc.
- Unicode character set
- Fonts

7 **Module 7: Applying basic styling**

- Setting basic CSS properties
- Exercises: beginning the sample job

8 Module 8: Inheritance, multiple rules and default values

- Inheritance, specificity and multiple rules
- Default values
- Exercises: reviewing the CSS for inheritance

9 Module 9: Applying basic page sequences and layouts

- Page sequence and page layouts
- Exercises: adding page sequences and layouts

10 Module 10: Advanced selectors and their uses

- Pseudo element selectors
- Outputting fixed content
- Calling XPP macros
- Calling a XyPerl function
- Outputting text via XyPerl
- Some other useful advanced selectors
- Exercises: using pseudo element selectors

11 Module 11: Working with attributes

- Basic CSS attribute selectors
- Using the *attr()* CSS3 function
- Using attributes inside XyPerl
- Exercises: using *attr()* and attributes inside XyPerl

12 Module 12: Working with registers

- Different types of registers
- Concept of streams in XPP
- Effect of streams on registers
- Stream diversion at XML import
- Exercises: working with registers

13 Module 13: Numbering

- Setting up numbering for chapters and sections
- Naming CSS based counters
- Reading out CSS based counters
- Exercises: setting up and outputting named registers

14 Module 14: Lists

- Unordered lists
- Ordered lists
- Nested lists
- Exercises: working with lists

15 Module 15: Frills processing for running heads and folios

- Frills processing
- Exercises: working with frills

16 Module 16: Further enhancements

- XPP specific CSS properties
- Inheritance for XPP specific CSS properties
- Vertical justification
- Widow and orphan control
- Unsupported CSS properties
- Exercises: adding XPP specific CSS properties

17 Module 17: Adding footnotes

- Footnotes and the PS spec
- Creating a link with the anchor
- Outputting the footnotes
- Exercises: working with footnotes

18 Module 18: Adding inline images

- Difference between inline images and pickups
- How to output an inline image
- Exercises: outputting inline images

19 Module 19: Introduction to pickups

- An introduction to pickups
- Exercises: outputting the pickups

20 Module 20: CSS boxes in XPP

- How to use CSS in XPP to draw boxes
- Exercises: outputting a variety of boxes

21 Module 21: Introduction to CALS tables

- An introduction to tables in XPP
- Set-up for CALS tables in CSS
- Exercises: outputting CALS tables

22 Module 22: Introduction to HTML5 tables

- Set-up for HTML5 tables in CSS
- Exercises: outputting HTML5 tables

23 Module 23: Resources

- Helpful resources

24 Complementary Module 1: Using the Supporting Data Editor (SDedit)

- Using the SDeditor to populate style specifications
- Other Supporting Data configuration files

25 Complementary Module 2: Page Column Override Spec

- Page Column (PC) Overrides – definition and purpose
- How to use a PC spec and where to use it
- What are some important PC fields and the impact on text layout
- Examples of using the Page Column Override.

26 Complementary Module 3: Vertical Justification and the Pagination Tries Spec

- Vertical justification (Vj) and Column and Page breaks.
- PT spec file and the fields most commonly used.
- "Status" window and an example on using the PT spec

27 Complementary Module 4: Hyphenation and Justification Spec

- Tables and Rules in the Hyphenation and Justification spec.
- Chaining and how to control the width of Space Bands.
- Types of letter spacing
- Algorithm and Dictionaries Fields
- Ragged text and Exception rules

About us

RWS is a content solutions company, powered by technology and human expertise. We grow the value of ideas, data and content by making sure organizations are understood. Everywhere.

Our proprietary technology, 45+ AI patents and human experts help organizations bring ideas to market faster, build deeper relationships across borders and cultures, and enter new markets with confidence – growing their business and connecting them to a world of opportunities.

It's why over 80 of the world's top 100 brands trust RWS to drive innovation, inform decisions and shape brand experiences.

With 60+ global locations, across five continents, our teams work with businesses across almost all industries. Innovating since 1958, RWS is headquartered in the UK and publicly listed on AIM, the London Stock Exchange regulated market (RWS.L).

More information: rws.com

© 2025 All rights reserved. Information contained herein is deemed confidential and the proprietary information of RWS Group*.

*RWS Group shall mean RWS Holdings plc for and on behalf of its affiliates and subsidiaries.