Scripting Languages

1. Give a one sentence definition of "scripting language."

2. List the principal ways in which scripting languages differ from conventional "systems" languages.

3. From what two principal sets of ancestors are modern scriping languages descended?

4. What IBM creation is generally considered the first general-purpose scripting language?

5. What is the most popular language for server-side web scripting?

6. How does the notion of context in Perl differ from coercion?

7. What is globbing? What is a wildcard?

8. What is a pipe in Unix? What is redirection?

9. Describe the three standard `I/O` streams provided to every Unix process.

10. Explain the significance of the #! convention in Unix shell scripts.

11. What is meant by the pattern space in `sed`?

12. Briefly describe the fields and associative arrays of `awk`.

13. In what ways even early versions of Perl improve on `sed` and `awk`?

14. Explain the special relationship between `while` loops and file handlers in Perl. What is the meaning of the empty file handle, `<>`?

15. Name three widely used commercial packages for mathematical computing.

16. List several distinctive features of the R statistical scripting language.

17. Explain the meaning of the $ and @ characters at the beginning of variable names in Perl. Explain the different meanings of $, @, and @@ in Ruby.

18. Which of the languages described in Section 12.2.4 uses indentation to control syntactic grouping?

19. List several distinctive features of Python.

20.Describe, briefly, how Ruby uses blocks and iterators.

21. What capabilities must a scripting language provide in order to be used for extension?

22. Name several commercial tools that use extension languages.

23. Explain the distinction between server-side and client-side web scripting.

24. List the tradeoffs between CGI scripts and embedded PHP.

25. Why are CGI sripts usually installed only in a special directory?

26. Explain how a PHP page can service its own requests.

27. Why might we prefer to execute a web script on the server rather than the client? Why might we sometimes prefer the client instead?

28. What is the HTML Document Object Model? What is its significance for client side scripting?

29. What is the relationship between JavaScript and Java?

30. What is an applet? Why applets are usually not considered an example of scripting?

31. What is HTML? XML? XSLT? How are they related to one another?

32. Name a scripting language that uses dynamic scoping.

33. Summarize the strategies used in Perl, PHP, Ruby, and Python to determine the scope of variables that are not declared.

34. Describe the conceptual model for dynamically scoped variables in Perl.

35. List the principal features found in POSIX regular expressions, but not in the regular expressions of formal language theory (Section 2.1.1).

36. List the principal features found in Perl REs, but not in those of POSIX.

37. Explain the purpose of search modifiers (characters following the final delimiter) in Perl-type regular expressions.

38. Describe the three main categories of escape sequences in Perl-type regular expressions.

39. Explain the difference between greedy and minimal matches.

40. Describe the notion of capture in regular expressions.

41. Contrast the philosophies of Perl and Ruby with regard to error checking and reporting.

42. Compare the numeric types of popular scripting languages to those of compiled languages like C or Fortran.

43. What are bignums? Which languages support them?

44. What are associative arrays? By what other names are they sometimes known?

45. Why don't most scripting languages provide direct support for records?

46. What is a typeglob in Perl? What purpose does it serve?

47. Describe the tuple and set types of Python.

48. Explain the unification of arrays and hashs in PHP and Tcl.

49. Explain the unification of arrays and objects in JavaScript.

50. Explain how tuples and hashes can be used to emulate multidimensional arrays in Python.

51. Explain the concept of context in Perl. How is it related to type compatibility and type inference? What are the two principal contex defined by the language's operators?

52. Compare the approaches to object orientation taken by Perl 5, PHP 5, JavaScript, Python, and Ruby.

53. What is meant by the blessing of a reference in Perl?

54. What are prototypes in JavaScript? What purpose do they serve?

55. Explain the relationship among SGML, HTML, and XML. What are their corresponding stylesheet languages?

56. Why does XML work so hard to distinguish between content and presentation?

57. What are the four main components of XSL? What are their respective purposes?

58. What is XHTML? How does it differ from "ordinary" HTML?

59. Explain the correspondence between XML documents and trees.

60. What does it mean for an XML document to be well formed.

61. Explain the distinctions (syntactic and semantic) among elements, declarations, and processing instructions in XML. Also explain the distinctions among elements, tags, and attributes.

62. Summarize the execution model of XSLT. In a nutshell, how does it work?

63. Explain the difference between applying templates and calling them in XSLT.