

Arrays are always made up of what?

Since scalars and arrays can always be distinguished syntactically ...

What happens if an array index isn't exactly an integer?

Give two examples of Perl's "no unnecessary limits" philosophy.

What happens if you exceed the bounds of an array?

Find the last index of an array.

What do negative array indexes do?

What shortcut is provided for creating ranges?



Give a shortcut for creating lists of strings.

What are the limitations of quotes lists?

Describe multiple scalar assignment.

What are its boundary cases?

What happens if you create a list literal some of whose elements are lists?

What does assigning one list variable to another do?

Why should array indexing be avoided?

What is preferred in its place?

What are `shift` and `unshift`?

What happens to arrays in string interpolation?



After a `foreach` loop is done, what's the value of the control variable?

What shortcut is provided for `foreach` loops?

Many operators do what when not provided a needed argument?

Describe `reverse`.

Describe `sort`.

What are the two most important contexts?

What happens if you use an expression in a context for which it's not defined.

What is the `$` before a scalar identifier or the `@` before a list identifier called?



Expressions evaluate differently depending on  
what?

Give an example.

What does *list-producing expression* even mean in a language whose evaluation is determined by context?

What is context is provided by assigning to  
`( $ident ) ?`

What context does `print` provide?

What happens if you use a scalar in list context?

How can you force the scalar context?

What does the expression `<STDIN>` evaluate to?

What does `chomp` do in list context?



Give the idiomatic way of reading all the lines of a file.