

Arrays are always made up of what?

Scalars.

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

... their identifiers are in separate namespaces. There can be an array with the same name as a scalar.

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

It is truncated.

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

Strings and arrays can be any size. Contrast with Java arrays which are of limited size.



What happens if you exceed the bounds of an array?

If you exceed at the end by assignment, the array is automatically expended. Any "empty slots" that need to be created are filled with `undef`.

If you exceed at the beginning, or at the end by lookup, `undef` is returned.

Find the last index of an array.

\$#ident

What do negative array indexes do?

As in Python, they index the set starting from the end.

What shortcut is provided for creating ranges?

The range operator ( . . ) placed between two numbers. The numbers are truncated.



Give a shortcut for creating lists of strings.

qw( no quotes necessary whitespace delimited )

Any punctuation can be used in place of the parens.

What are the limitations of quotes lists?

- You can't put comments inside.
- If you want to use the delimiter as an element of the list you must backslash escape.

Describe multiple scalar assignment.

What are its boundary cases?

$(\$id1, \dots, \$idN) = (val1, \dots, valN)$

The RHS need not be a list *literal*.

If there are too many values they will be ignored. If there are too many variables the extra variables will get their default values.

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

The resulting list will be flattened, containing the elements of any "nested list" along with any scalar values in the literal.



What does assigning one list variable to another do?

Copies the whole array.

Why should array indexing be avoided?

What is preferred in its place?

It's slow.

`pop` which mutates the array and returns the last element (or `undef`)

`push` which takes an array and a scalar or two arrays, mutating the first

What are `shift` and `unshift`?

`shift` is like popping from the beginning.

`unshift` is like pushing to the beginning (*cons*).

What happens to arrays in string interpolation?

They are printed with the elements with spaces interposed.



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

The same as it was before the loop (*i.e.*, quite possibly undefined).

What shortcut is provided for `foreach` loops?

You can leave off the control variable. The default `$_` will be used.

Many operators do what when not provided a needed argument?

They use \$\_ instead.

Describe `reverse`.

It returns a new array with the argument reversed. The original argument is not mutated.



Describe `sort`.

Returns a sorted array, using ascii order. Does not mutate the argument.

What are the two most important contexts?

List and scalar.

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

undef will result.

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

A qualifier.



Expressions evaluate differently depending on  
what?

Give an example.

Context, the location in the program text where the expression occurs.

Bare list variables evaluate to lists only in list context. They evaluate to the size of the list in scalar context.

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

An expression that would be more surprising if it were used in scalar context than if it was used in list context.

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

List context. That's a list of length one, not a scalar with extraneous parens.

What context does `print` provide?

List.



What happens if you use a scalar in list context?

The scalar will be wrapped into a singleton list.

How can you force the scalar context?

Use the `scalar` operator.

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

In scalar context, the next line of input. In list context, all remaining lines.

What does `chomp` do in list context?

It chomps each element, in place.



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

```
chomp(@lines = <FILEHANDLE>)
```