

`print` in perl doesn't include what?

A newline character.

List the variable type syntax.

\$ scalar

@ array

% hash

& subroutine

* typeglob

What's the difference between double, single quotes, and back ticks?

Double quotes perform variable interpolation and backslash interpretation. Single quotes do not.

Back ticks capture the output from executing a command.

What are the default values?

" " or 0, as appropriate.

How are scalar values interpreted?

As the expected type, depending on context.

What context do double quotes provide?

Assignment to arrays/ hashes?

interpolative context

list context

Create a list using literal syntax.

Unpack that list.

```
@threeprimes = (2, 3, 5);  
($a, $b, $c) = @threeprimes;
```

Index an array.

Update an array.

Either involves a scalar so use \$ not @.

```
$lst[n] = new_el  
print $lst[n]
```


Create a hash using literal syntax.

```
@birthmonths = (  
  "John" => "February",  
  "Mary" => "March",  
);
```

Arrows are just a nicer way of writing more arrows.

Look up an element in a hash.

Update a hash.

```
$hash{"key"}  
$hash{"key"} = val
```

As with arrays, notice the use of `$` when dealing with individual elements.

Describe the Perl noun/verb analogy.

Nouns can be singular (scalars) or plural (arrays and hashes).
Verbs can be procedures or functions.

Run some perl straight from the terminal.

Run some perl stored in a file.

Get interpreter warnings.

```
perl -e 'some perl'  
perl file.pl
```

The `-w` option prints warnings.

What is a filehandle?

Which are built in?

A data type that can represent files, devices, sockets, and pipes.

`STDOUT` and `STDERR` are provided by default.

Get a filehandle.

Use `open`, whose simplest form is:

```
open (HANDLENAME, "filename");
```

What read/write options are available when requesting a new filehandle?

Readonly (default): "<filename"

Write (clobber): ">filename"

Write (append): ">>filename"

Read a line from a file.

Read a line from the terminal.

```
$str = <FILEHANDLE>
```

```
$str = <STDIN>
```


Write to a file.

Write to the terminal.

```
print FILEHANDLE 'str'
```

These two are the same:

```
print STDOUT 'str'
```

```
print 'str'
```

What's the difference between `chop` and
`chomp`?

`chop` removes the last character of the string passed to it, and returns it.

`chomp` removes `endl` from the string passed to it, and returns the *number* of characters removed.

What is Perl's string concatenation operator?

Why does Perl have separate operators for string concatenation and arithmetic addition?

Use a period (.) for string concatenation.

Because of weak typing addition of scalars created as strings, but that can be interpreted as numbers, would result in a sum.