Unlike Java, F#'s primitive types ...

How can you use an integer as the base of an exponent?

What are the conversion functions for primitive types?

Why are conversion functions more widely used in F#?

For arbitrary precision integers F# features what?

What are the bitwise operators?

Index a string.

How are multi-line strings written?

What is the verbatim string syntax?

What shortcut is provided for converting a string literals to arrays of bytes?

How is negation written?

How is equality written?

Inequality?

What is the syntax of function definitions?

The result of a function is ...

How are function types written?

Unlike in Haskell, functions using operators like + will default to ...

How are type annotations written?

How are type parameters written?

Unlike in Scala, shadowing is allowed where?

What is the syntax of conditionals?

Unlike in Scala, each branch of a conditional must have what?

How are tuples written?

How are their types written?

Access the elements of a tuple.

What is the list literal syntax?

What is the syntax of list comprehensions?

How are list comprehensions evaluated?

Loop over a range.

Loop over a collection.

What shortcut `for` syntax is provided for list comprehensions?

In the F# API, what's the difference between `fold` and `reduce`?

Run a block of code on each element of a list.

What is used in place of `null`?

Why is `printfn` more useful than using `System.Console's` print methods?

What are the `printfn` format specifiers?

If no module is specified, you are actually using what?

How can it be accessed?

Create a module explicitly.

Create a nested module.

How are namespaces different from modules?

What are the ideal use-cases for modules and namespaces?

How does an F# program execute?

Create an explicit entry point.