

F# Cheat Sheet (Author: Tyler P. Berkshire)

Key Features	
Strongly typed with algebraic-style types	
Easy concurrency	
Extensive multi-paradigm features	
Integrated with .NET	
Metaprogramming	

Core syntax	
let	Bind value to a name
let rec	Recursive definition
//, (**)	Inline and multi-line comments
indentation	Separates code blocks
whitespace	Separates parameters

Control Flow	
for x in { } do	Standard for loop
while x do	Standard while loop
match x with 	Pattern matching against a discriminated union
when	Guards help with complicated pattern matching
>	Pipes emulate UNIX pipes. Output of the left becomes input of the right
<	Reverse pipe
> >	Function composition returns a function instead of immediately evaluating
< <	Reverse function composition
currying	Functions can be curried and partially applied

Algebraic Types	
fun()	Function
unit	A function with one range value
type	Keyword to declare custom types
(x,y)	Tuples can be heterogeneous
{x:int, y:float}	Records are labelled tuples
I of int B of bool	Discriminated unions
<'a>	Single quote denotes generic type
Option	The option type is either <i>Some</i> or <i>None</i>
[]	Lists are immutable and homogeneous
seq{}	Sequences use <i>yield</i> and <i>yield!</i> to create infinite data types

Metaprogramming	
quotations	Compiled into objects representing the program
<@ @>	Quotation with type information
<@@ @@>	Quotation without type information
Expr<'T>	Resulting type of a quotation
%	Splice a typed expression into a quotation
%%	Splice a non-typed expression into a quotation
Patterns	Active pattern module used to analyze expression objects
ExprShape	Module to traverse expression trees with fewer active patterns

Asynchronous Workflows	
async { }	Expressions in the curly braces become <i>async</i>
let!	Wait for the <i>async</i> task to return, similar to <i>await</i>
Async.Start	Start a task asynchronously
Async.RunSynchronously	Block until all <i>async</i> events are completed
Async.Parallel	Run the tasks in parallel
CancellationTokenSource	Workflows can be cancelled by invoking the cancel token

Actor Model	
MailBoxProcessor	Built in agent class
inbox.Receive()	Read a message
agent.Post	Post a message

Useful Libraries	
FSharp.Data	Providers for working with structured data files formats (CSV, HTML, JSON, etc.)
FsUnit	Builds upon the .NET testing framework
FSharp.Charting	Compositional library for creating charts

Copyright © 2019 by Tyler Berkshire.