F# - DATA TYPES

The data types in F# can be classified as follows −

- Integral types
- Floating point types
- Text types
- Other types

Integral Data Type

The following table provides the integral data types of F#. These are basically integer data types.

F# Type	Size	Range	Example	Remarks
sbyte	1 byte	-128 to 127	42y -11y	8-bit signed integer
byte	1 byte	0 to 255	42uy 200uy	8-bit unsigned integer
int16	2 bytes	-32768 to 32767	42s -11s	16-bit signed integer
uint16	2 bytes	0 to 65,535	42us 200us	16-bit unsigned integer
int/int32	4 bytes	-2,147,483,648 to 2,147,483,647	42 -11	32-bit signed integer
uint32	4 bytes	0 to 4,294,967,295	42u 200u	32-bit unsigned integer
int64	8 bytes	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	42L -11L	64-bit signed integer
uint64	8 bytes	0 to 18,446,744,073,709,551,615	42UL 200UL	64-bit unsigned integer

bigint	At least 4 bytes	any integer	42I 1499999 9999999 9999999 9999999 9999I	arbitrary precision integer

Example

```
(* single byte integer *)
let x = 268.97f
let y = 312.58f
let z = x + y

printfn "x: %f" x
printfn "y: %f" y
printfn "z: %f" z

(* unsigned 8-bit natural number *)

let p = 2uy
let q = 4uy
let r = p + q

printfn "p: %i" p
printfn "q: %i" q
printfn "r: %i" r

(* signed 16-bit integer *)

let a = 12s
let b = 24s
let c = a + b

printfn "a: %i" a
printfn "b: %i" b
printfn "c: %i" c

(* signed 32-bit integer *)

let d = 212l
let e = 504l
let f = d + e

printfn "d: %i" d
printfn "e: %i" e
printfn "f: %i" f
```

When you compile and execute the program, it yields the following output −

```
x: 1
y: 2
z: 3
p: 2
q: 4
r: 6
a: 12
b: 24
c: 36
```

```
d: 212
e: 504
f: 716
```

Floating Point Data Types

The following table provides the floating point data types of F#.

F# Type	Size	Range	Example	Remarks
float32	4 bytes	±1.5e-45 to ±3.4e38	42.0F -11.0F	32-bit signed floating point number 7*significantdigits*
float	8 bytes	±5.0e-324 to ±1.7e308	42.0 -11.0	64-bit signed floating point number 15 − 16*significantdigits*
decimal	16 bytes	±1.0e-28 to ±7.9e28	42.0M -11.0M	128-bit signed floating point number 28 − 29*significantdigits*
BigRational	At least 4 bytes	Any rational number.	42N -11N	Arbitrary precision rational number. Using this type requires a reference to FSharp.PowerPack.dll.

Example

```
(* 32-bit signed floating point number *)
(* 7 significant digits *)

let d = 212.098f
let e = 504.768f
let f = d + e

printfn "d: %f" d
printfn "e: %f" e
printfn "f: %f" f

(* 64-bit signed floating point number *)
(* 15-16 significant digits *)
let x = 21290.098
let y = 50446.768
let z = x + y

printfn "x: %g" x
printfn "y: %g" y
printfn "z: %g" z
```

When you compile and execute the program, it yields the following output −

```
d: 212.098000
e: 504.768000
f: 716.866000
x: 21290.1
y: 50446.8
z: 71736.9
```

Text Data Types

The following table provides the text data types of F#.

F# Type	Size	Range	Example	Remarks
char	2 bytes	U+0000 to U+ffff	'x' '\t'	Single unicode characters
string	20 + $2 * string's length$ bytes	0 to about 2 billion characters	"Hello" "World"	Unicode text

Example

```
let choice = 'y'
let name = "Zara Ali"
let org = "Tutorials Point"

printfn "Choice: %c" choice
printfn "Name: %s" name
printfn "Organisation: %s" org
```

When you compile and execute the program, it yields the following output −

```
Choice: y
Name: Zara Ali
Organisation: Tutorials Point
```

Other Data Types

The following table provides some other data types of F#.

F# Type	Size	Range	Example	Remarks
bool	1 byte	Only two possible values, true or false	true false	Stores boolean values

Example

```
let trueVal = true
let falseVal = false

printfn "True Value: %b" (trueVal)
printfn "False Value: %b" (falseVal)
```

When you compile and execute the program, it yields the following output −

```
True Value: true
False Value: false
```