

BIT115: What Happens When You Declare an int Variable ?

When you **declare**, **name**, and **initialize** a **variable** with an **int** (*integer*) data type, you are setting up a named container in memory that has **32** memory cells. Each memory cell is called a **bit** (i.e., **binary element** for short) and **8 bits** make up a **byte**.

An **int** data type consists of **4 bytes**, enough to hold a range of numbers between **- 2,147,483,648** to **2,147,483,647** . These numbers are stored by turning the 32 memory cells **on** and **off** in a specific way and this represents binary (denoted by using a **1** or a **0**).

Example: `int someNumber = 15;`

0	0	0	0	1	1	1	1	1 byte
0	0	0	0	0	0	0	0	2 byte
0	0	0	0	0	0	0	0	3 byte
0	0	0	0	0	0	0	0	4 byte

someNumber (all 32 bit memory cells are turned off except the first four in byte 1)

Other Data Types

Just for reference, there are other data types available in Java. These take up different memory container sizes based on the type and range size of the data being stored there. These are shown in the handy-dandy chart below:

Type	Contains	Default Value	# Bits	# Bytes	Range
boolean	True or False	False	1	N/A	N/A
char	Unicode Char.	\u0000	16	2	\u0000 to \uFFFF
byte	Signed Integer	0	8	1	-128 to 127
short	Signed Integer	0	16	2	-32,768 to 32,767
int	Signed Integer	0	32	4	-2,147,483,648 to 2,147,483,647
long	Signed Integer	0	64	8	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 (± 9 quintillion)
float	Decimal	0.0	32	4	$\pm 1.4E-45$ to $\pm 3.4028235E+38$
double	Decimal	0.0	64	8	$\pm 4.9E-324$ to $\pm 1.7976931348623157E+308$