The int type: whole numbers

1. The `int` type

If you want to store *whole number* in variables, then you need to use the `int` type.

The word `int` is short for `Int`eger , which of course is a good hint that this type lets you store *integer numbers*.

Variables whose type is `int` are capable of storing *integer numbers* ranging from `-2 billion` to `+2 billion`. To be more precise, from `-2,147,483,648` to `+2,147,483,647`.

Interesting fact

These decidedly non-round numbers are related to how the computer's memory is organized.

In Java, **4 bytes** of memory are allocated for the `int` type. Each byte of memory consists of **8 bits**. Each bit can only represent 2 values: 0 or 1. An `int` variable contains **32 bits** and can represent `4,294,967,296` values.

Half of this range was set aside for negative numbers, and the other half for positive numbers. And that's how we get the range from `-2,147,483,648` to `+2,147,483,647`.

---

2. Creating an `int` variable

The `int` type is for storing integers. To create a variable in code that can store *integer numbers*, you need to use a statement like this:

```
int name;
```

Declaring an `int` variable

Where name is the name of the variable. Examples:

Statement	Description
`int x;`	An `x` integer variable is created
`int count;`	A `count` integer variable is created
`int currentYear;`	A `currentYear` integer variable is created

The case of the letters matters. That means the commands `int color` and `int Color` will declare two different variables.

And the commands `Int Color` and `INT COLOR` won't make any sense to the compiler, causing it to report an error. `int` is a special keyword for the integer type and it must be written in lowercase.

---

3. Shorthand for creating variables

If you need to create many variables of the same type in the same place in a program, you can use this shorthand notation:

```
int name1, name2, name3;
```

Shorthand for creating multiple variables of the same type

Examples:

Statements	Shorthand

`int x;`
`int y;`
`int z;`
	`int x, y, z;`

`int count;` `int totalCount;`	`int count, totalCount;`

`int day;`
`int month;`
`int year;`
	`int day, month, year;`

4. Assigning values

To *put* a *value* into an `int` *variable*, you need to this statement:

```
name = value;
```

Assigning a value to a variable

Where the value can be any integer expression. Examples:

Statement	Note
`int a;` `a = 5;`	
`int b;` `b = 2*1000*1000*1000;`	
`int c;` `c = -10000000;`	
`int d;` `d = 3000000000;`	This code won't compile, because `3,000,000,000` is greater than the maximum possible value for an `int`, which is `2,147,483,647`

5. Shorthand for creating and initializing a variable

You can use a single command to create (declare) a *variable* and *assign* a *value* to it. This is what is done most often, since we usually declare a variable when we need to store a value.

Here's what the command looks like:

```
int name = value;
```

Shorthand for creating and initializing a variable

Statement	Note
`int a = 5;`	
`int b = 2*1000*1000*1000;`	The value of the variable will be 2 billion
`int c = -10000000;`	The value of the variable will be negative 10 million
`int d = 3000000000;`	This code won't compile, because 3,000,000,000 is greater than the maximum possible value for an int: 2,147,483,647

You can also declare several variables in a single line. In this case, the command will look like:

```
int name1 = value1, name2 = value2, name3 = value3;
```

Shorthand for creating and initializing multiple variables

Examples:

Statement	Note
`int a = 5, b = 10, c = a + b;`	`a` equals `5`, `b` equals `10`, `c` equals `15`