Data types and variables in Java

Data types
Declaring variables
Assigning variables

Data Types

Java have 8 primitive data types:

Type	Description
byte	A single byte
short	A small integer, taking up 2 bytes
int	A regular integer, taking up 4 bytes
long	A long integer, taking up 8 bytes
float	A 32 bit floating point number
double	A 64 bit floating point number
boolean	Represents a boolean value
char	A single 16-bit Unicode character

These are the fundamental core data types in Java.

Data Types: Integers

Type	Size (bytes)	Range
byte	8	-2^7 to 2^7-1
short	16	-2^{15} to $2^{15}-1$
int	32	-2^{31} to $2^{31}-1$
long	64	-2^{63} to $2^{63}-1$

```
byte age = 46;
short year = 2017;
int number = 123456;
long largeNumber = 1234567890;
```

The preferred types to use are **int** and **long**.

Data Types: Real Types

Type	Size (bytes)	Litteral	Range
float	4	f	~$1.4e^{-45}$ to $3.4e^{38}$
double	8	d	~$4.9e^{-324}$ to $1.8e^{308}$

```
float f1 = 3.141592653589793f;
System.out.println(f1);

double d1 =3.141592653589793d;
double d2 =3.141592653589793;
System.out.println(d1);
System.out.println(d2);
```

Float and double literals

Default is double

```
3.1415927
3.141592653589793
3.141592653589793
```

These data types should never be used for precise values, such as currency.

Data Types: Other

Type	Size (byte)	Range/Values
boolean	1	true/false
char	2	Represents a single Unicode character
String	Varied	Maximum length 2^{31} -1

```
char c = 'A';
boolean enabled = true;
String name = "Edument";
```

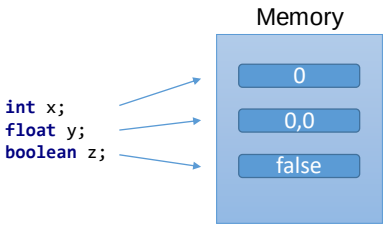
Notice the upper-case S

Declaring variables

Declaring variables

Variables have to be **declared** before usage.

Declaring a variable in Java:



Variables represent storage locations in memory.

Every **variable has a type** that determines what values can be stored in the variable.

Declaring variables

Example of variable declarations:

```
// Create a String variable named "text"
String text;

// Create an integer variable named "number"
int number;

// Create a boolean variable named "hasCandy"
boolean hasCandy;

// Create a floating point variable named "percentage"
float percentage;

// Create a double named "highPrecision"
double highPrecision;
```

To just **declare** a variable is generally not enough
In order to use it, we have to **assign** a value to it as well

Assigning values to variables

Examples of assigning values:

```
// Create a String variable named "text" and assign a value
String text = "Hello";

/* Create an integer variable named "number" and assign a
value */
int number = 2;

/* Create a boolean variable named "hasCandy" and assign a
value */
boolean hasCandy = true;

/* Create a floating point variable named "percentage" and
assign a value */
float percentage = 0.7961F;

// Create a double named "highPrecision" and assign a value
double highPrecision = 1.2631852722174092;
```

Assigning values to variables

Assignment and declaration can be separated

```
// Assignment when declaring
String text = "Hello";


// Splitting declaration and assignment:
String text;
text = "Hello";
```

The two methods are equivalent, but can be used in different situations.

Assigning values: Special Cases

You can also assign hexadecimal, octal and binary values:

```
// Assignment with a hexadecimal value (251 decimal)
int hex = 0xFB;

// Assignment with a octal value (83 decimal)
int oct = 0123;

// With Java 7 or higher, you can assign binary values.

// 32-bit
int bits = 0b10100001010001011010000101000101;

// 16-bit
short bits = 0b1010000101000101;

// 8-bit
byte bits = 0b10011001;
```

When an integer literal starts with 0 in Java, it's assumed to be in octal notation.

Why Assign and Declare?

Why do we assign and declare variables?

```
// Assignment when declaring
String text = "Hello";
```

Declaring and assigning variables allows the compiler to find typos when we program.

The result is more robust code with fewer bugs.

Exercise 4

Let's do exercise 4