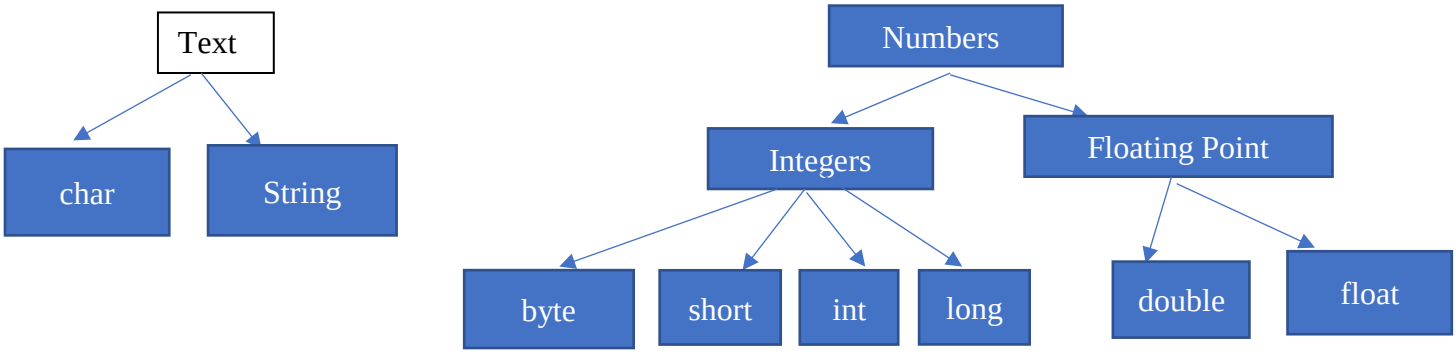
4-Unit 1 Lesson 2 -Storing values

All programming languages have built in ability to use the computer memory in order to store information used by applications created with them. The way that programming languages store information depends on the type of information being stored.
The main factors in deciding which kind of identifier to be used depends first on the type of data – text or numbers as well as on the amount of Information which needs to be stored.
Processing can work with two types of numbers – integers (whole numbers) and real (numbers with decimals)



Integers

There are 4 types of integers depending on how large is the number. Bigger numbers use more computer memory.

Data type	Computer memory use	Range of values
byte	8 bits	-128 to 127
short	16 bits	-32768 to 32767
int	32 bits	-2 147 483 648 to 2 147 483 647
long	64 bits	-9 223 372 036 854 775 808 to 9 223 372 036 854 775 807

Floating Point (Real) Numbers

Floating point (numbers with decimals) are represented with double and float data types. The difference is both the size and the precision of the numbers you can store. The numbers always contain a decimal point.

Data type	Memory Use	Range of Values	Precision
float	32 bits	$-3.40282347 \times 10^{38}$ to $3.40282347 \times 10^{38}$	at least 6 digits
double	64 bits	-1.8×10^{308} to 1.8×10^{308}	at least 15 digits

Storing single characters and text

Char - Storing single characters

Processing uses a system called Unicode which is capable of representing characters of all alphabets in the world. Unicode uses 16 bits to represent a character and this makes it possible to represent over 65 000 characters (2^{16}).

Some examples of characters are : 'x' , '%' '9'

Note: The apostrophe character is represented by '\'' (where the backslash is the escape character). The backslash character is represented by '\\'.

String – storing text

String is the only type introduced here known as Object (Class) type. More lessons about its use and functionality will be discussed later in the course.

Boolean identifiers

Sometimes when a decision or a comparison needs to be made or the outcome of one stored Boolean identifiers are used. There are only 2 possible vales – **true or false.**

Declaration statements

Rules

- ✓ Before declaring an identifier (variable) a few rules need to be remembered:
- ✓ Identifiers cannot start with numbers
- ✓ Identifiers cannot be reserved(command words)
- ✓ Identifiers cannot contain special characters except underscore.
- ✓ Case sensitive: Processing is case sensitive, so lowercase and uppercase of the same letter/word are considered different identifiers. Keep in mind that if you call one variable x and another X you may have a hard time remembering what is the purpose of each one. Using identifiers with the same name but different lower/uppercase characters is a bad practice and should be avoided.
- ✓ Use descriptive names, not too short, not too long.
- ✓ Use camel notation if name is made of two words.

data type identifier name

int number; or int number=10;

String word; or String word=”school”;

char letter; or char letter=’A’;

Boolean state; or Boolean state=true;

Application 1 **Student Name:** Jessica Chen

Practice variables. Complete and return to the teacher for signature. Due by the end of class.

I. Answer the following questions:

1. Why do programming languages use identifiers (variables)?

2. Why are there so many different types of integer and real identifiers?
   Those types each hold a different amount of data and later I'll do this later.

3. In the space provided write the declaration statement which will BEST match the values given:

a. 31111	int number = 31111;
b. -5000.000000	float number = -5000.000000;
c. 'A'	char letter = 'A';
d. 127	int number = 127;
e. true	Boolean state = true;
f. "processing"	String word="processing"
g. $123,456.00	String money = "$123,456.00
h. -12e-3	double number = -12e-3;

Completed: ☐ Teacher signature:_____

Application 2 (Q 1-3 No Computer) /12 (T/K) **Student Name:** Jessica Chen

II. Complete ***the variable declaration statements only*** for the following programming problems. For full marks use descriptive variable names, following the rules of variable declaration.

1. A program which records the results of a hockey game. It has to record the values of each team's score, the percentage of fouls , the name of the player who has scored or who is in the penalty box. Declare the best possible variables for the purpose holding the least, yet adequate space in memory.

```
//Global Variables

//team scores
int team1score = 0;
int team2score = 0;

//foul percentages
float team1foulspercent =  ;
float team2foulspercent =  ;

//foul statements
String team1fouls = "       : " + team1foulspercent + "%";
String team2fouls = "       : " + team2foulspercent + "%";
```

2. Declare variables which will help you calculate a student's semester average mark – one variable for each subject and one for the overall average.

```
//Global Variables

//subject averages              //overall average
float subject1average=  ;       float averageOverall = (subject1average+subject2average)/2;
float subject2average=  ;
```

3. Declare variables which you can use to store a friend's information in an address directory. You must use at least one String variable, one numeric variable and one Boolean variable.

```
//Numeric Values
int streetNumber =  ;
float travelTime =  ;



//Booleans
Boolean addressVerified = true;
```