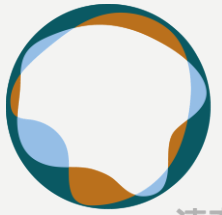
VARIABLE

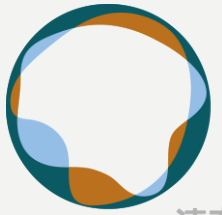
MEMORY – 内存



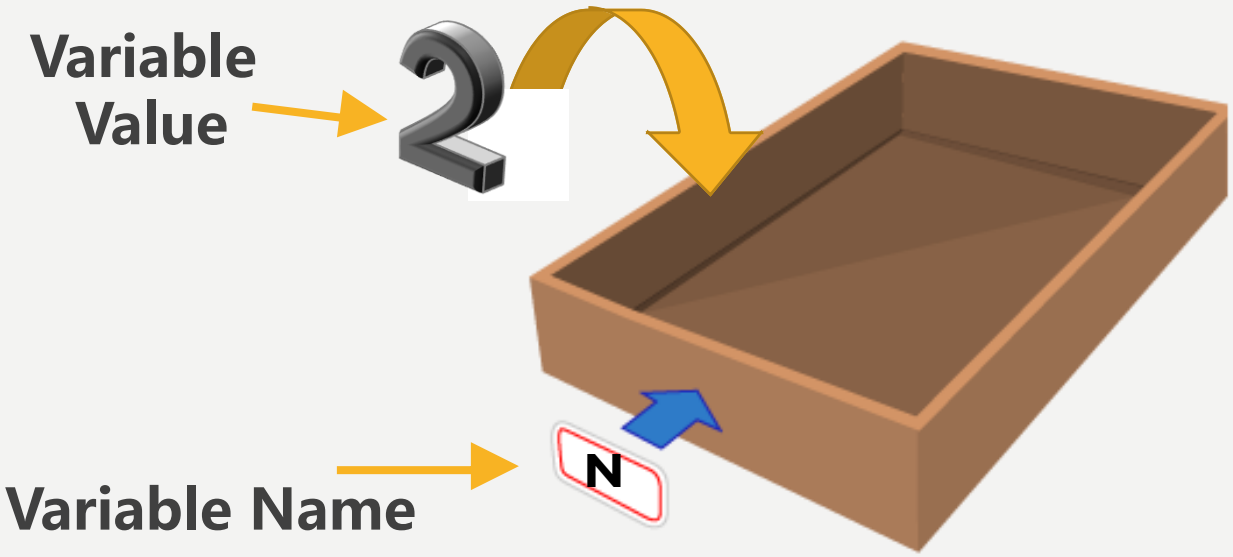
清森学校
BEIJING QINGSEN SCHOOL

VARIABLE 变量

Variable: A piece of the computer's memory that is given a name and type, and can store a value.

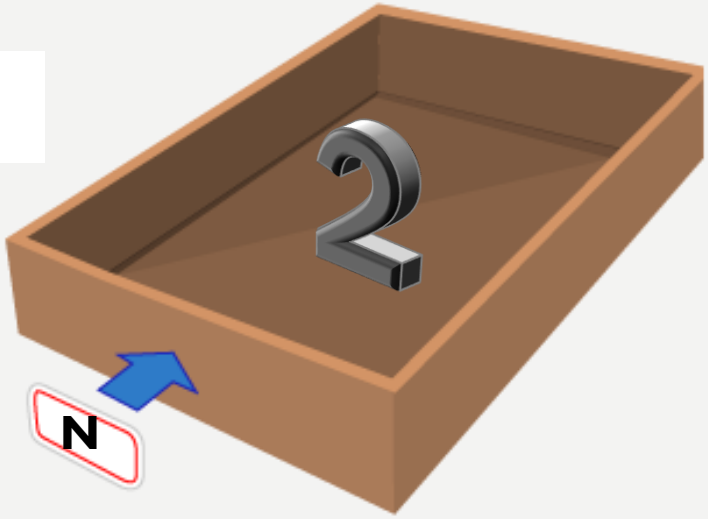


Using Variables



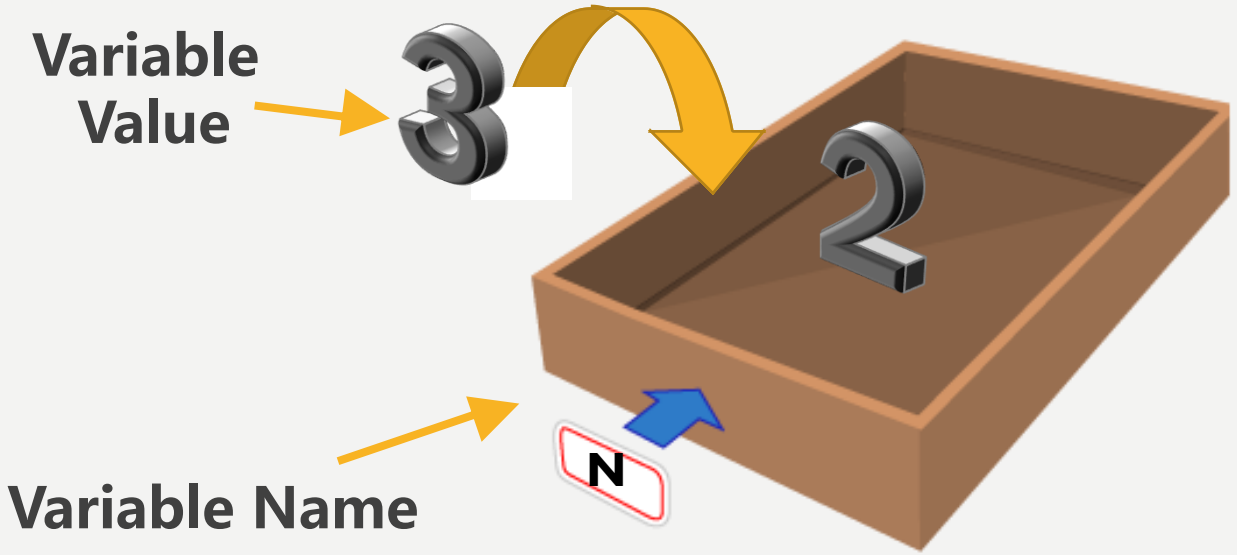
Using Variables

Now, variable N is 2.



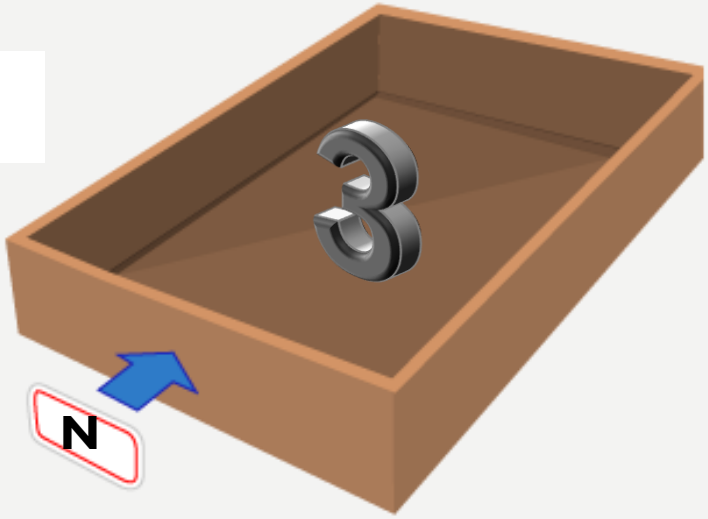
Variable Name

Using Variables



Using Variables

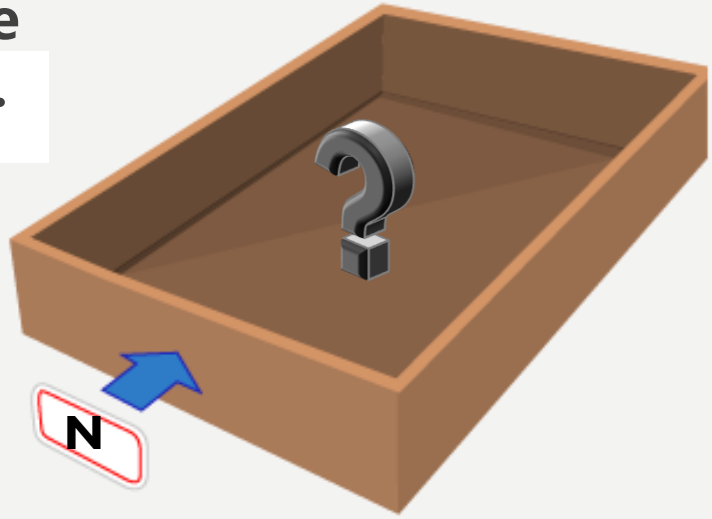
Now, variable N is 3.



Variable Name

Using Variables

Can not use a variable without an init value.



Variable Name

STEPS FOR USING A VARIABLE

1. Declare variable 声明变量

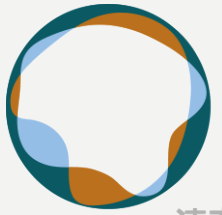
   state its name and type

2. Assignment variable 赋值变量

   store a value into it

3. Use variable 使用变量

   print it or use it as part of an expression



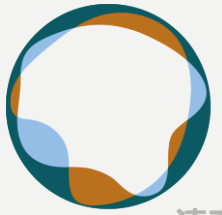
1. DECLARE VARIABLE

- **variable declaration**: Sets aside memory for storing a value.
  - Variables must be declared before they can be used.

- Syntax: **<type of data> <name of variable>;**

- Example:

```
int num;
double score;
String name;
```

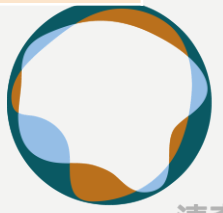


TYPES OF DATA

A **type** is a set of values (e.g. integers, String, etc..)

- ✓ Primitive data 基本数据类型
- ✓ Reference data 引用数据类型

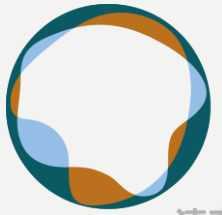
Name	Age	Score	boarding
Tom	18	4.7	Yes
Ben	19	4.2	No
Lisa	18	4.0	Yes



EXERCISE 1 - PRIMITIVE

Task 1: Fill in the forms

Time limitation: 10 mins for self studying



清森学校
BEIJING QINGSEN SCHOOL

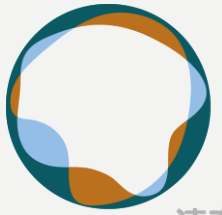
PRIMITIVE TYPE

The primitive types on the Advanced Placement Computer Science A exam are:

- **int** - which store integers
- **double** - which store floating point numbers
- **boolean** - which store Boolean values (either true or false).

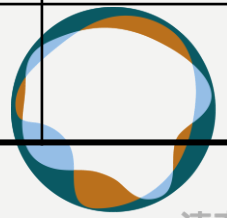
NAME OF VARIABLE

1. Case sensitive
2. Begin with a letter (A-Z, a-z) or an underline(_) or the character "$"
3. Only contain letters、numbers、underline(_)、$
4. Do not use keyword as name, e.g. public, void, for, if
5. No any Space
6. Readability
7. CamelCase Notation



JAVA KEYWORDS

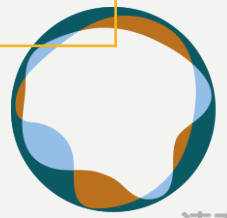
abstract	boolean	break	byte	case	catch	char
class	const	continue	default	do	double	else
extends	final	finally	float	for	goto	if
implements	import	instanceof	int	interface	long	native
new	package	private	protected	public	return	short
static	super	switch	synchronized	this	throw	throws
transient	try	void	volatile	while		



EXERCISE 2

Which name of variable is not correct?

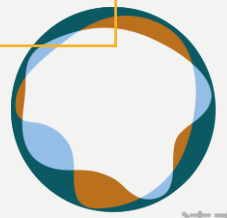
print*	o_0
ak47	public
_interface	0_o
BMW	$$$
300warriors	User Name



EXERCISE 2 - ANSWER

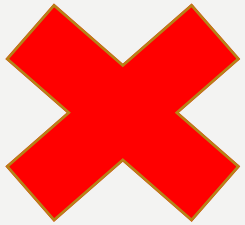
Which name of variable is not correct?

print*	o_0
ak47	public
_interface	0_o
BMW	$$$
300warriors	User Name



READABILITY

```
int a = 2.0 ;
int b = 3.1415926 ;
int c = 2 * a * b ;
```



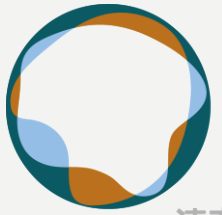
```
int radius = 2.0 ;
int PI = 3.1415926 ;
int circlePerimeter = 2 * PI * radius ;
```

CAMELCASE NOTATION
驼峰命名

CamelCase Notation :

First letter of identifier is in lower case and the first letters of each subsequent letters would be capital else lower case.

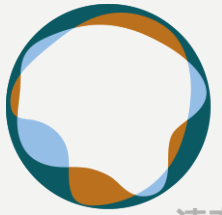
eg. studentName, workAddress, midScore



CAMELCASE NOTATION
驼峰命名

TIPS:

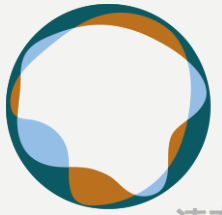
1. Do not be so long.
2. Keep origin meaning
3. Do not use single letter, except for temporary value.
4. Not recommend using more than three words



EXERCISE 3

Naming these variables

1) A student's home address;
2) An order's service identity number;
3) The page number of a book;
4) A person's income in 2017.



2. ASSIGNMENT VARIABLE
赋值变量

Tip:
A variable can only store a value of its own type.

- **Assignment**:

  Stores a value into a variable.

- Syntax: **<name of variable> = value;**

- Example

```
int num;
num = 12;
```

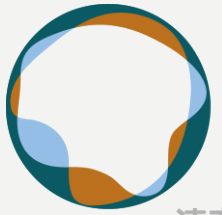
DEFINE VARIABLE/ INITIALIZATION

Syntax:

<type of data> <name of variable> = value;

Example:

```
int num = 12;
double score = 85.5;
String name = “Ying";
```



COMPARE

Declare

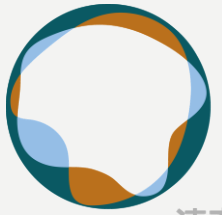
```
int  classNum;
classNum = 1;
```

Define

```
int classNum = 1;
short a = classNum;
```

What happens here?

```
int x = 3;
x = x + 2;
```

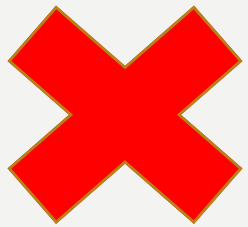


清森学校
BEIJING QINGSEN SCHOOL

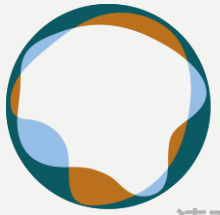
What happens here?

```
int x = 3;
double y =3.7;
y = x + y;
```

```
int x = 3;
double y =3.7;
x = x + y;
```



An int value can be stored in a double variable.



PRINTING A VARIABLE'S VALUE

- Use + to print a string and a variable's value on one line.

```
double grade = (95.1 + 71.9 + 82.6) / 3.0;
System.out.println(" Your grade was  " + grade);
```

Output:

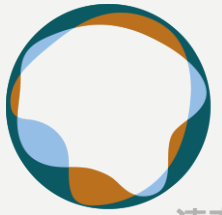
Your grade was 83.2

PRINTING A VARIABLE'S VALUE

- Try to code

```
int students = 11 + 17 + 4 + 19 + 14 ;
System.out.println("There are " + students + "
students in the course." );
```

What is your output?



EXERCISE 4 -Compiler Errors

Find errors in the code, and how can this code be fixed

1.

```
int num = 3000000000;
System.out.println(num);
```

2.

```
int num1 = 25;
int num2 = 2;
num3 = num1+num2;
System.out.println(num3);
```

清森学校
BEIJING QINGSEN SCHOOL

EXERCISE 4 -Compiler Errors

3.

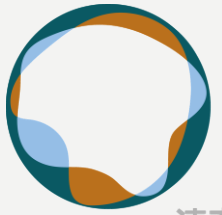
```
int num;
7 = num;
```

4.

```
int num1 ;
System.out.println(num1);
```

5.

```
int num1 = 4;
int num1 = 5;
```



MULTIPLE VARIABLES

- Multiple variables of the same type can be declared and initialized at the same time.

- Syntax:

type name1, name 2, name3;

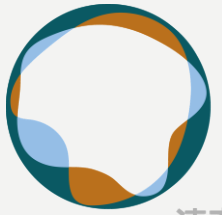
```
int x, y, z;  // declare three integers.
```

type name1 = value1, name2 = value2, name3 = value3;

```
int a = 1, b = 2, c = 3;
// declare and initialize three integers
```

WARM UP

1. How to set a variable?
2. List some types of variable
3. How to name a variable?
4. What is the complier/syntax error? Give some examples



TYPE BOOLEAN

boolean: A logical type whose values are true and false.

Try to code, what is output?

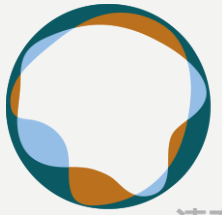
```
boolean notEqual = 4 <= 5;
boolean lovesAPCS = false;
System.out.println(lovesAPCS);
System.out.println(notEqual);
```

TYPE OF STRING

A String variable contains a collection of characters surrounded by double quotes.

Example

```
String string1 = "I am Ying";
System.out.println(string1);
```



FINAL

The keyword **final** can be used in front of a variable declaration to make it a constant that cannot be changed.

```
public class TestFinal
{
   public static void main(String[] args)
   {
     final double PI = 3.14;
     System.out.println(PI);
     PI = 4.2; // This will cause a syntax error
   }
}
```

EXERCISE 5

Improve the receipt program using variables.

```
public class Receipt {
    public static void main(String[] args) {
        // Calculate total owed, assuming 8% tax / 15% tip
        System.out.println("Subtotal:");
        System.out.println(38 + 40 + 30);

        System.out.println("Tax:");
        System.out.println((38 + 40 + 30) * .08);

        System.out.println("Tip:");
        System.out.println((38 + 40 + 30) * .15);

        System.out.println("Total:");
        System.out.println(38 + 40 + 30 + (38 + 40 + 30) * .15 +(38 + 40 + 30) * .08);
    }
}
```

EXERCISE 5

Improve the receipt program using variables.

```
public class Receipt {
    public static void main(String[] args) {
        // Calculate total owed, assuming 8% tax / 15% tip
        int subtotal = 38 + 40 + 30;
        double tax = subtotal * .08;
        double tip = subtotal * .15;
        double total = subtotal + tax + tip;

        System.out.println("Subtotal: " + subtotal);
        System.out.println("Tax: " + tax);
        System.out.println("Tip: " + tip);
        System.out.println("Total: " + total);
    }
}
```

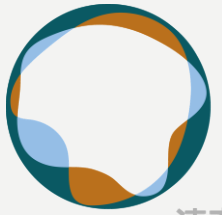
INPUT AND SYSTEM.IN (NOT ON AP)

Interactive program: Reads input from the console.

- While the program runs, it asks the user to type input.
- The input typed by the user is stored in variables in the code.

Scanner: An object that can read input from many sources.

- Communicates with System.in (the opposite of System.out)
- Can also read from files, web sites, databases, ...



SCANNER SYNTAX (NOT ON AP)

- The Scanner class is found in the java.util package.

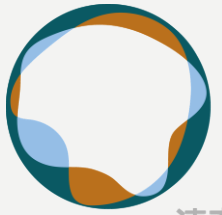
```
import java.util.*;   // so you can use Scanner
```

- Constructing a Scanner object to read console input:

```
Scanner name = new Scanner(System.in);
```

Example:

```
Scanner console = new Scanner(System.in);
```



SCANNER METHODS (NOT ON AP)

- Using a variable to store data from input.

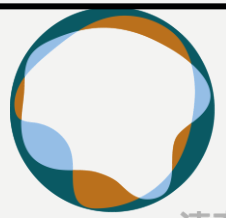
varaibleName = name.next();

Example:

```
Scanner console = new Scanner(System.in);
System.out.print("How old are you? ");  // prompt
int age = console.nextInt();
System.out.println("You typed " + age);
```

SCANNER METHODS (NOT ON AP)

Method	Description
`nextInt()`	reads an `int` from the user and returns it
`nextDouble()`	reads a `double` from the user
`next()`	reads a one-word `String` from the user
`nextLine()`	reads a one-*line* `String` from the user

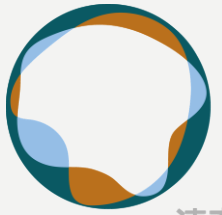


INPUT TOKENS (NOT ON AP)

- **token**: A unit of user input, as read by the Scanner.

  Tokens are separated by whitespace (spaces, tabs, new lines).

- When a token is not the type you ask for, it crashes.



EXERCISE 6 (NOT ON AP)

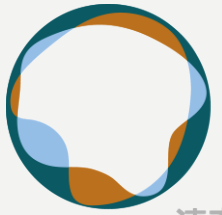
- Create two integer variables : pens and pencils.
- Ask the user to enter the number pens and pencils
- Output the total number of pens and pencils

Output:

Enter the number of pens: 10

Enter the number of pencils: 18

Total is 28 pens and pencils.



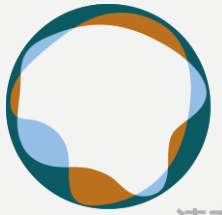
清森学校
BEIJING QINGSEN SCHOOL

OPERATION

OPERATION

Try Coding:

```
System.out.println(2 + 3);
System.out.println(2 - 3);
System.out.println(2 * 3);
System.out.println(2 / 3);
System.out.println(2.0/3)
System.out.println(2%3);
System.out.println(2 == 3);
System.out.println(2 != 3);
```



ARITHMETIC OPERATORS

Operation	Meaning	Example
+	Addition	2+3=5
-	Subtraction/ Negation	2-3=-1
*	Multiplication	2*3=6
/	Division	2/3=0 (integer) 2.0/3= 0.66667 (double)
%	Remainder	2%3=2

EXERCISE 1

Design a program to give the exact change with the least number of coins for a given number of cents (user input).

Using quarters (25 cents), dimes (10 cents), nickels (5 cents), and 1 cent.

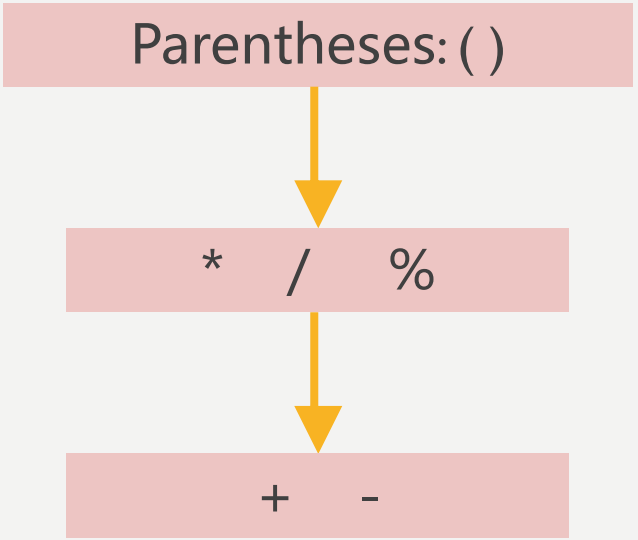
Output:

```
Input the number of cents 137
The quarters is 5
The dimes is 1
The nickels is 0
The cents is 2
```

PRECEDENCE 优先级

precedence: Order in which operators are evaluated.

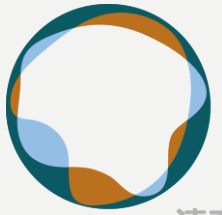
1. Generally operators evaluate left-to-right.
2. But * / % have a higher level of precedence than + -
3. Parentheses can force a certain order of evaluation



EXAMPLE

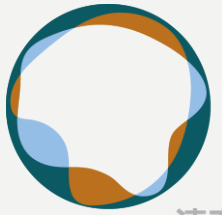
a. 1 * 2 + 3 * 5 % 4

b. 1 + 8 % 3 * 2 - 9



MIX TYPE (int & double)

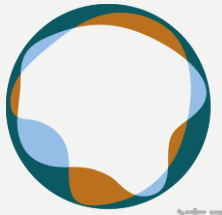
When int and double are mixed, the result is a double.



EXAMPLE

a. 7 / 3 * 1.2 + 3 / 2

b. 2.0 + 10 / 3 * 2.5 - 6 / 4



TYPE CASTING

Type Cast:

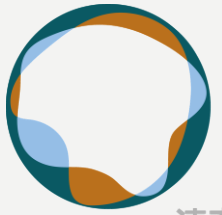
A conversion from one type to another.

Syntax:

TypeA variableA = (typeA)variable B

Example:

```
double dNum = 1.7;
int num = (int)dNum;
```



PRECISION

Low precision

(byte)

(short)

(int)

(float)

(double)

High precision:

Low To High:

```
int a = 124;
double b = a;
System.out.println(b);
```

High To Low:
(Precision loss)

```
double a = 128.5;
int b = (int)a;
System.out.println(b);
```

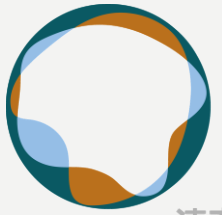
MORE CASTING

```
System.out.println((double)1/2);

System.out.println((double)(1/2));

System.out.println(1.0/2);
```

What happens here?



清森学校
BEIJING QINGSEN SCHOOL

ROUND TO THE NEAREST INTEGER

```
double number = 7.0 / 3; // round a positive number to its nearest integer
int nearestInt = (int)(number + 0.5);
double negNumber = -20.0 / 3; // round a negative number to its nearest integer

int nearestNegInt = (int)(negNumber -0.5);
```

What is the value of nearestInt and nearestNegInt?