**Selection** is another name for "Decision Making"

In regard to a group of commands it answers the question:

> Shall we do these or not?

The "we" is the computer.

The choice is usually not random - but depends on the answer to a specific question.

This question is called "the condition".
Here is some simple but silly Java code for a selection:

```java
If( 21 < 34 ) {
    out.println("YES");
    out.println("we like it");
}
```

Because the value of the expression

21 < 34

is **true**, the group of commands WILL **ALWAYS** be done by the computer.
The previous Java code was silly because, we programmers already know that 21 is less than 34 and so the computer will ALWAYS do the two commands:

```java
out.println("YES");
out.println("we like it");
```

There is NO need for selection in this case.
Here is something a little less transparent:

If( (5313*27+5)%4==2 ) {
    out.println("YES");
    out.println("we like it");
}

So, what do YOU think, will the computer say "YES" and "we like it" or not ???
Unless you are an arithmetic whizz, you are not going to instantly know the answer.

Instead you will have to take a little time to compute:

$$ (5313 \times 27 + 5) \mod 4 $$

You should get 0 from doing the %4 operation.

Thus the condition

$$ (5313 \times 27 + 5) \mod 4 = 2 $$

is false because 0 is NOT EQUAL to 2.

Hence the computer WILL NOT say "YES" nor will it say "we like it".
But an even better example comes when we use a selection inside a simple interaction. Here is the Java code:

```java
out.print("Give a number: ");
int x = cin.getInt();

if( x>5000 ) {
    out.println("That is big enough.");
    out.println("Its square is" + x*x );
}
```

User prompt and input

Condition

To do or not to do

Selection
Suppose the user types 5001 when prompted. What will the console show?
Suppose the user types 5001 when prompted. What will the console show:

Give a number: 5001
That is big enough.
Its square is 25010001

User input is shown in blue.
Suppose the user types 4990 when prompted. What will the console show?
Suppose the user types 4990 when prompted. What will the console show:

```
Give a number: 4990
```

User input is shown in blue.

Note that the code did NOT respond to the user's input.
A more interesting kind of selection has this format:

\[
\text{If}(\text{some\_condition}) \{ \\
\text{one\_group\_of\_steps} \\
\ldots \\
\} \text{ else } \\
\text{another\_group\_of\_steps} \\
\ldots \\
\}
\]

Can be true or false

ONLY done when the condition is true.

ONLY done when the condition is false.

NEVER does BOTH groups of steps.
The interaction is more satisfying when an ELSE phrase is used. Here is the Java code:

```java
out.print("Give a number: ");
int x = cin.getInt();

if (x > 5000) {
    out.println("That is big enough.");
    out.println("Its square is" + x*x);
} else {
    out.println("TOO small");
}
```

User prompt and input

Condition

To do when condition is true

To do when condition is false

Note group might have only one step
Suppose the user types 5001 when prompted. What will the console show?
Suppose the user types 5001 when prompted. What will the console show:

```
Give a number: 5001
That is big enough.
Its square is 25010001
```

User input is shown in blue.

Note this is the same answer as before - since the ELSE phrase is not used.
Suppose the user types 4990 when prompted. What will the console show?

User input is shown in blue.
Suppose the user types 4990 when prompted. What will the console show:

```
Give a number: 4990
TOO small
```

User input is shown in blue.

Note that the code now responds to the user input because of the ELSE phrase.
Another place selection is often used is INSIDE loops.

Consider the series of numbers which starts with 3 and which gets each successive number by tripling the previous number and adding 5.

So the first number is 3

The second number is 3 times 3 plus 5 : namely 14

The third number is 3 times 14 plus 5 : namely 47

Etc. etc. etc.
Suppose we are only interested in those numbers in the series (see previous page) which are multiples of 7 - but that we only want to look at numbers no larger than 5000.

For example, the second number in the series is 14 and that IS a multiple of 7.

```java
int item = 3;
while (item <= 5000) {
    if (item % 7 == 0) {
        out.println(item);
    }
    item = 3 * item + 5;
}
```
Suppose we are only interested in **counting** those numbers in the series (see previous page) which are multiples of 7 - but that we only want to look at numbers no larger than 5000

```java
int count = 0;
int item = 3;
while (item <= 5000) {
    if (item%7 == 0) {
        count = count + 1;
    }
    item = 3*time + 5;
}
out.println("count="+count);
```

No items have been counted yet

TOP of Loop Logic

Selection logic

Instead of printing selected items on console, count how many there are

BOTTOM of Loop Logic

Output of FINAL count.

Note: NOT IN loop
OR - Suppose we are only interested in summing up those numbers in the series (see previous page) which are multiples of 7 - but that we only want to look at numbers no larger than 5000

```java
int sum = 0;
int item = 3;
while (item <= 5000) {
    if (item % 7 == 0) {
        sum = sum + item;
    }
    item = 3*item + 5;
}
out.println("sum=\"+sum);
```

Instead of counting selected items on console, sum them up. Note "item" on right instead of "1".

No items have been summed yet

TOP of Loop Logic

Selection logic

BOTTOM of Loop Logic

Output of FINAL SUM.
Note: NOT IN loop