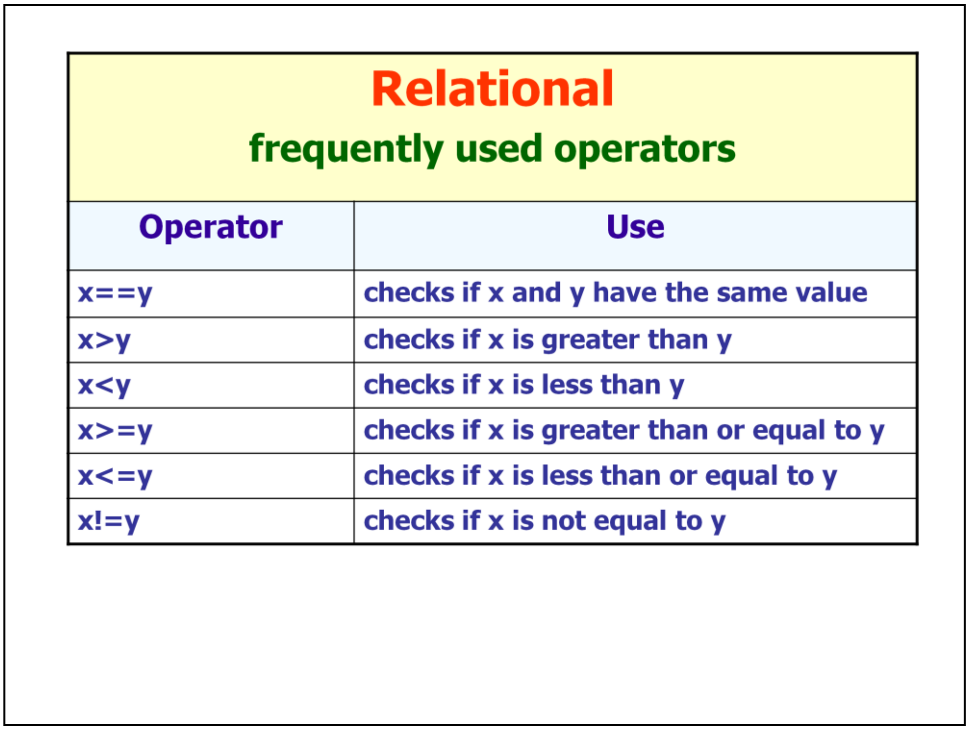
Jeroo Ifs



Relational

frequently used operators

Operator	Use
x==y	checks if x and y have the same value
x>y	checks if x is greater than y
x<y	checks if x is less than y
x>=y	checks if x is greater than or equal to y
x<=y	checks if x is less than or equal to y
x!=y	checks if x is not equal to y

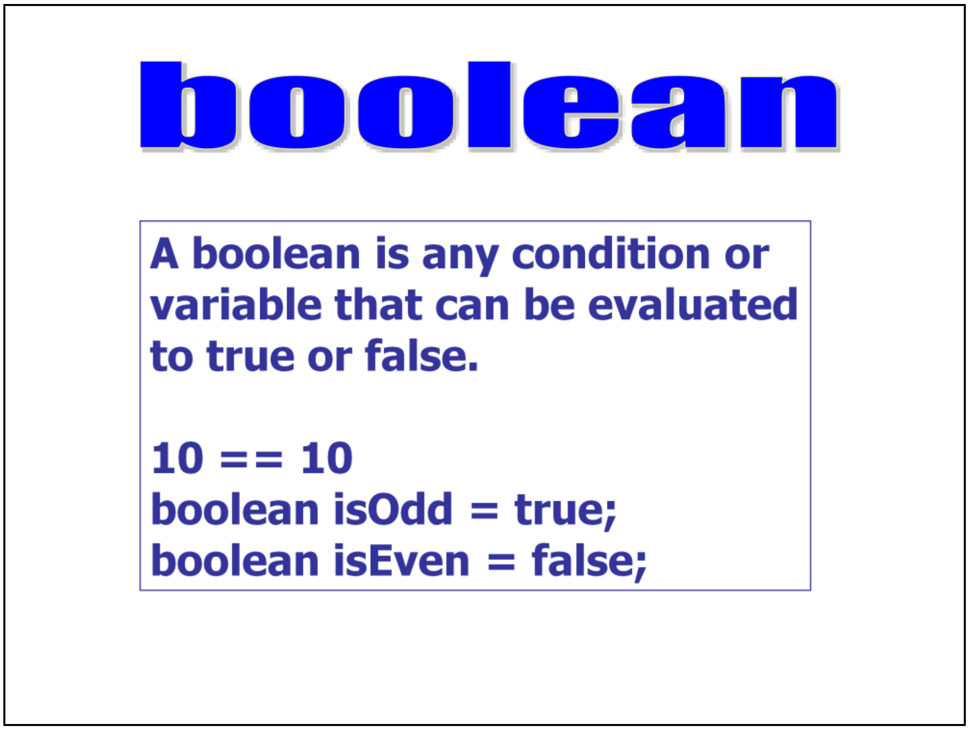
Relational operators are used to compare values for equality, less than, and greater than.

90<2 is false.

90>2 is true.

90==2 is false.

2==2 is true.



boolean

A boolean is any condition or variable that can be evaluated to true or false.

```
10 == 10
boolean isOdd = true;
boolean isEven = false;
```

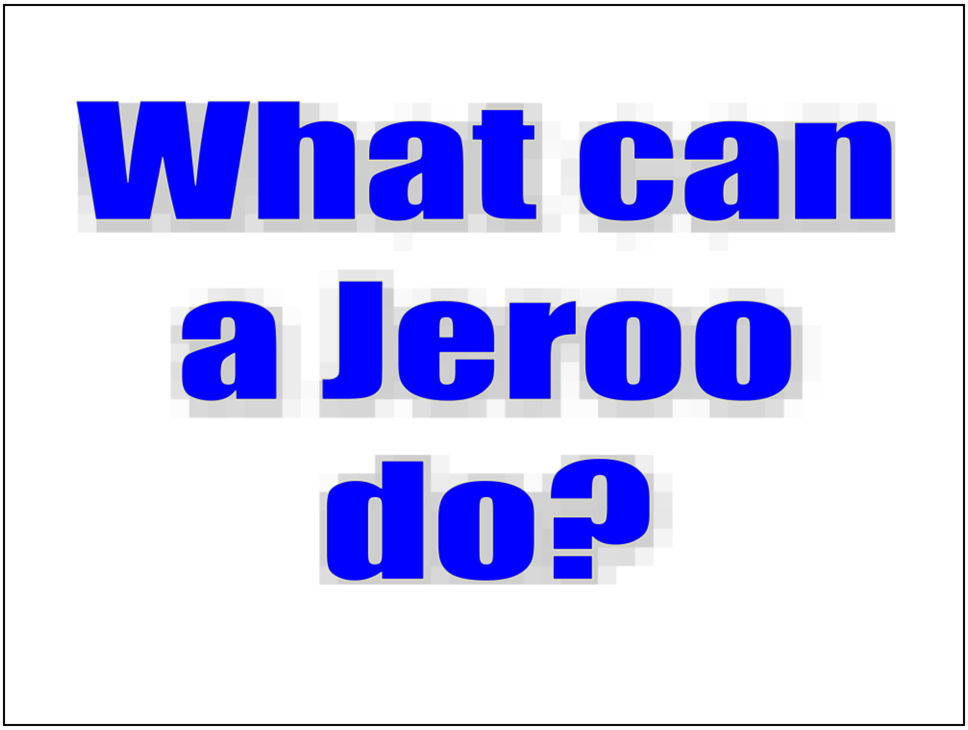
A boolean is a variable or condition that can be evaluated as true or false.

90<2 is false.

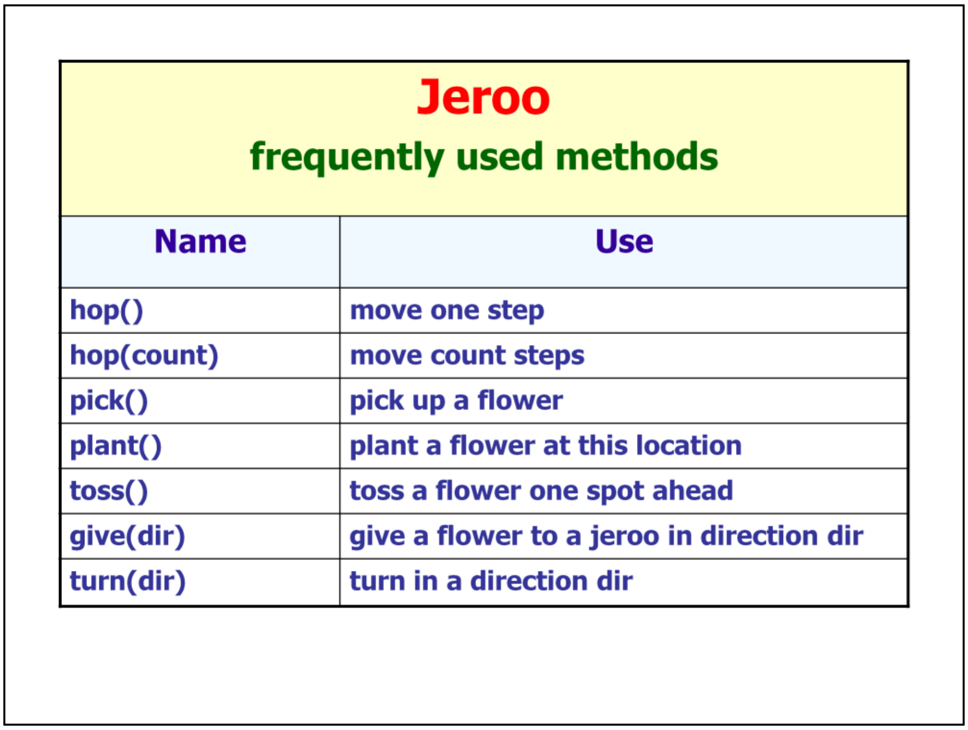
90>2 is true.

90==2 is false.

2==2 is true.



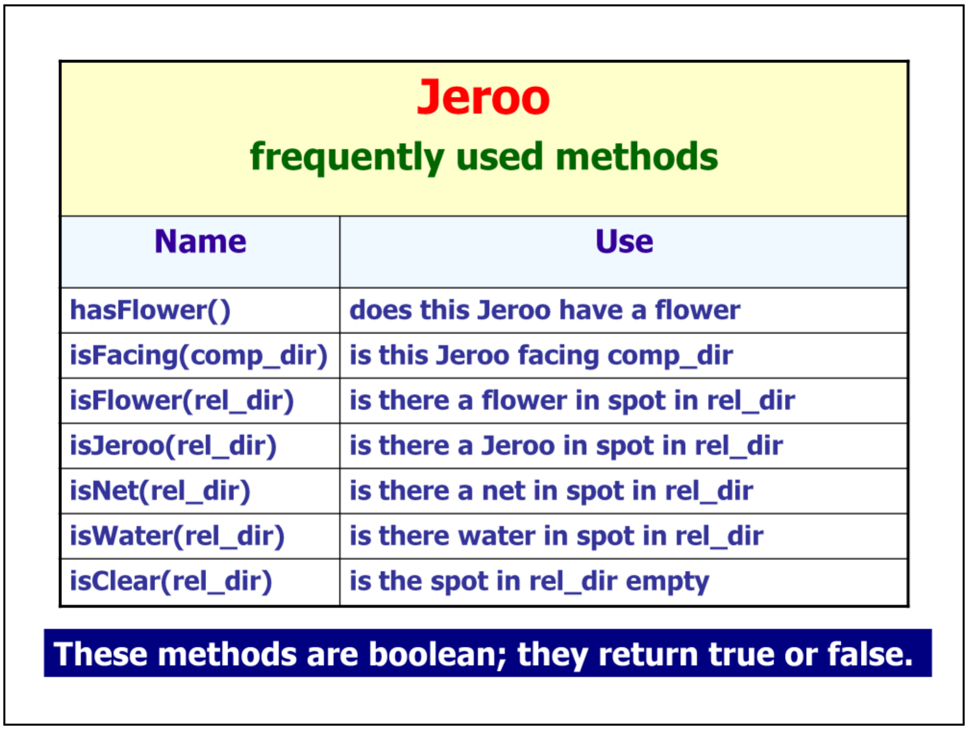
What can a Jeroo do?



Jeroo

frequently used methods

Name	Use
hop()	move one step
hop(count)	move count steps
pick()	pick up a flower
plant()	plant a flower at this location
toss()	toss a flower one spot ahead
give(dir)	give a flower to a jeroo in direction dir
turn(dir)	turn in a direction dir



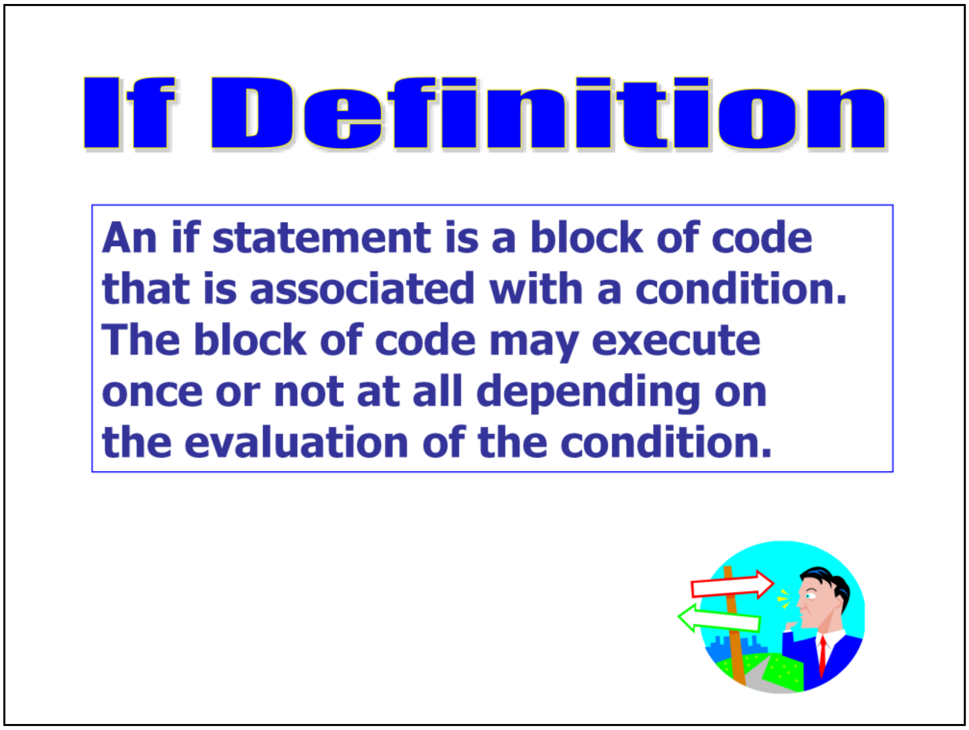
Jeroo

frequently used methods

Name	Use
hasFlower()	does this Jeroo have a flower
isFacing(comp_dir)	is this Jeroo facing comp_dir
isFlower(rel_dir)	is there a flower in spot in rel_dir
isJeroo(rel_dir)	is there a Jeroo in spot in rel_dir
isNet(rel_dir)	is there a net in spot in rel_dir
isWater(rel_dir)	is there water in spot in rel_dir
isClear(rel_dir)	is the spot in rel_dir empty

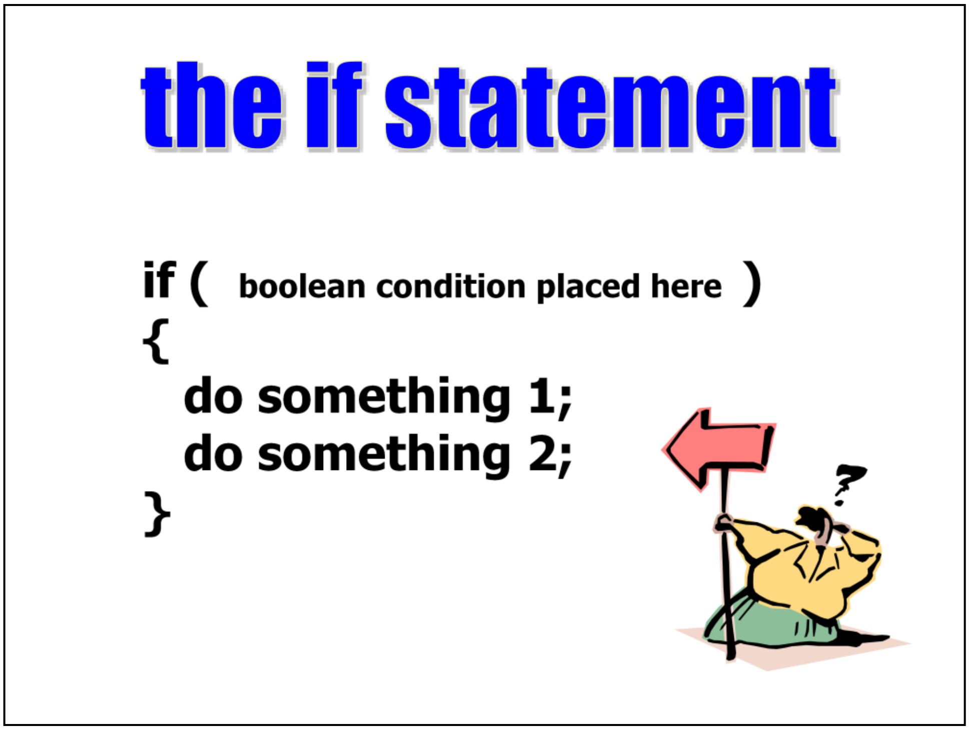
These methods are boolean; they return true or false.

The if statement!



If Definition

An if statement is a block of code that is associated with a condition. The block of code may execute once or not at all depending on the evaluation of the condition.



the if statement

```
if (  boolean condition placed here  )
{
   do something 1;
   do something 2;
}
```

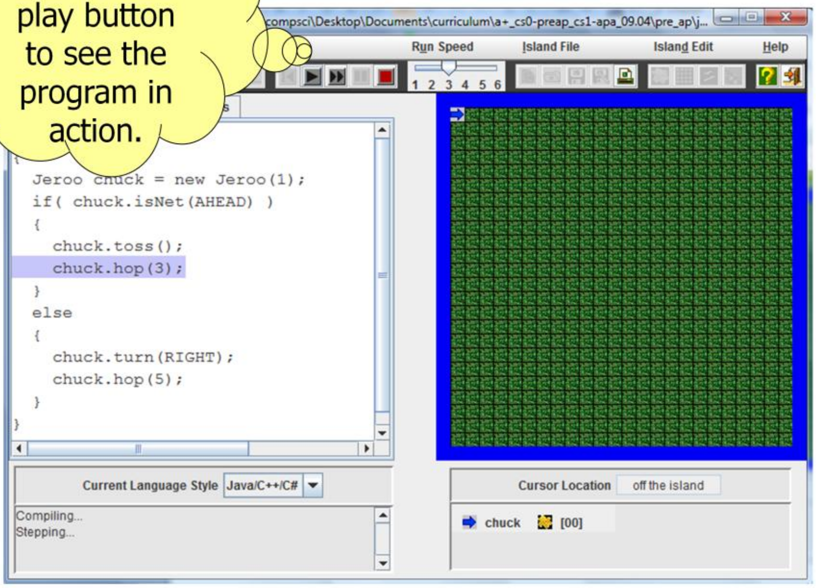
`do something 1` and `do something 2` will occur if the condition is true.

If the condition is false, `do something 1` and `do something 2` will not occur.

the if statement

```
if( jill.isWater(AHEAD) )
{
  jill.turn(RIGHT);
}
jill.hop();
```

the if statement



Click on the play button to see the program in action.

```
{
  Jeroo chuck = new Jeroo(1);
  if( chuck.isNet(AHEAD) )
  {
    chuck.toss();
    chuck.hop(3);
  }
  else
  {
    chuck.turn(RIGHT);
    chuck.hop(5);
  }
}
```

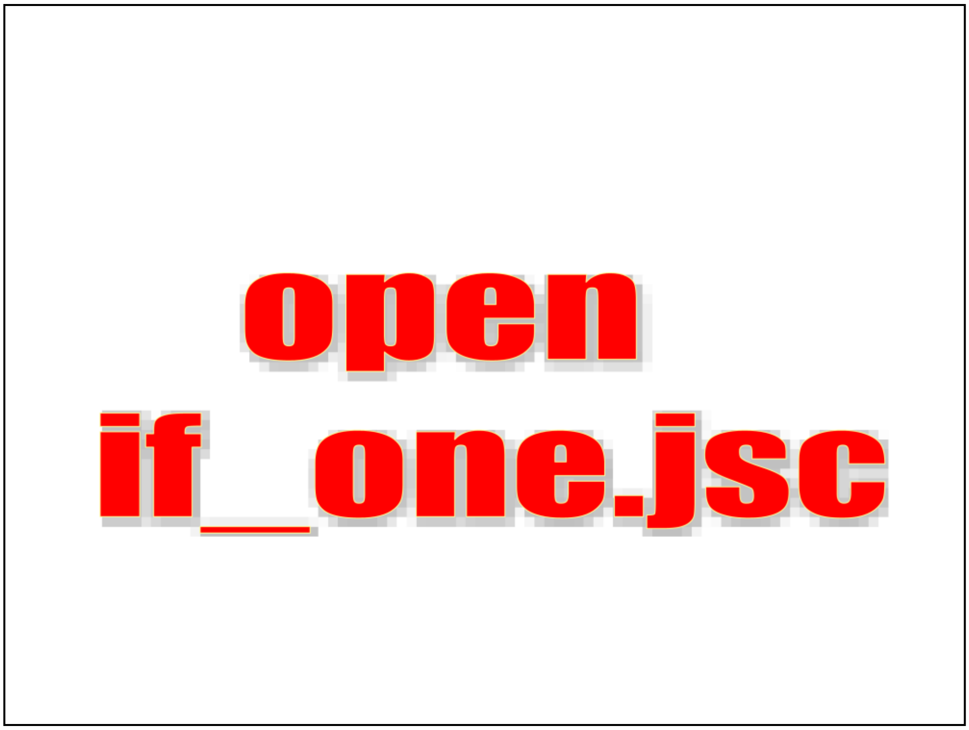
Current Language Style: Java/C++/C#

```
Compiling...
Stepping...
```

Cursor Location: off the island

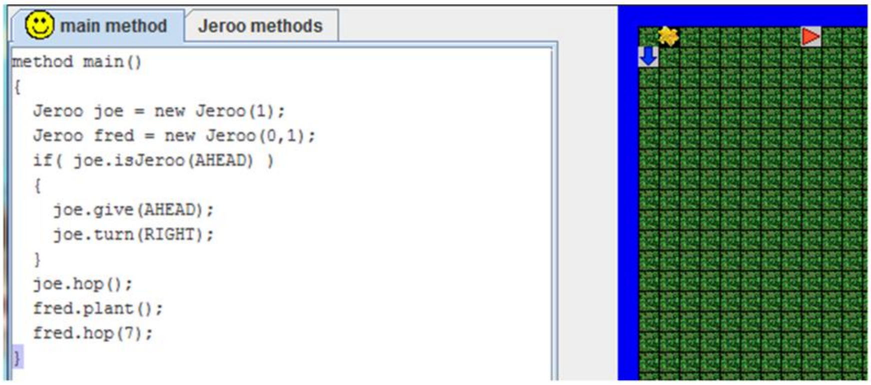
chuck [00]

compsci\Desktop\Documents\curriculum\a+_cs0-preap_cs1-apa_09.04\pre_ap\j...
Run Speed
1 2 3 4 5 6
Island File
Island Edit
Help



open
if_one.jsc

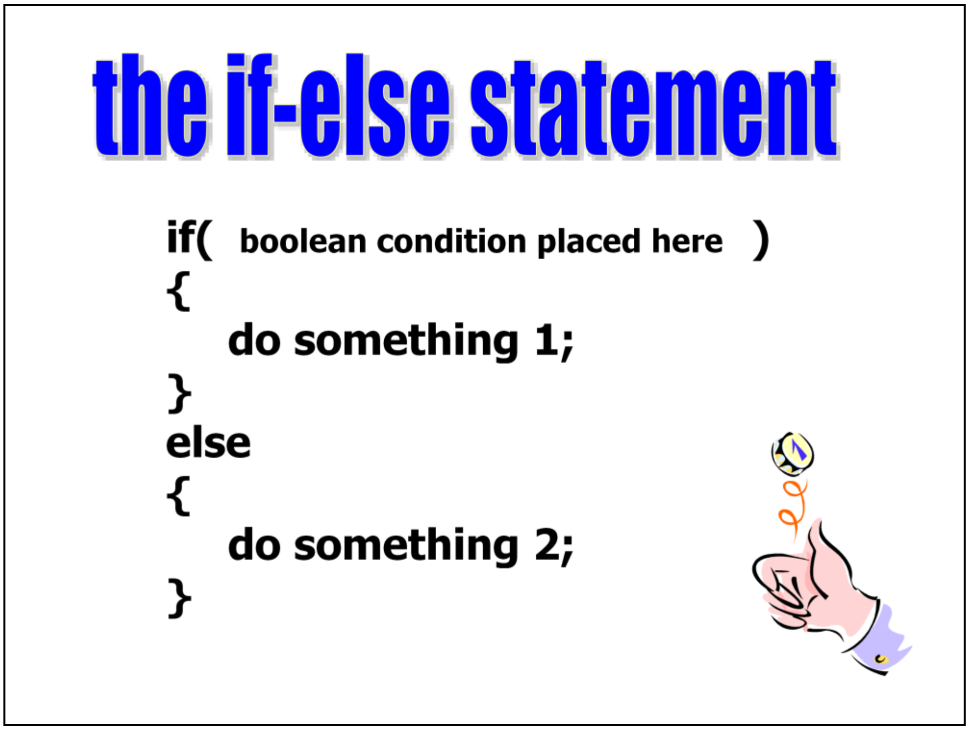
the if statement



```
method main()
{
  Jeroo joe = new Jeroo(1);
  Jeroo fred = new Jeroo(0,1);
  if( joe.isJeroo(AHEAD) )
  {
    joe.give(AHEAD);
    joe.turn(RIGHT);
  }
  joe.hop();
  fred.plant();
  fred.hop(7);
}
```

open

if_two.jsc

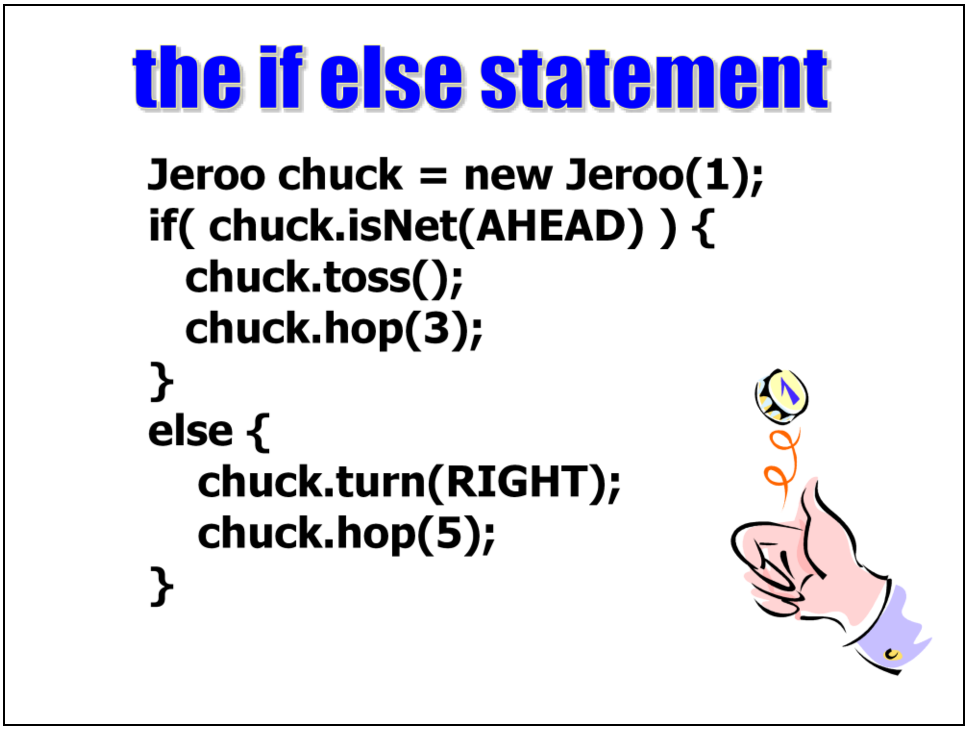


the if-else statement

```
if(  boolean condition placed here  )
{
    do something 1;
}
else
{
    do something 2;
}
```

If the condition is true, `do something 1` will occur.

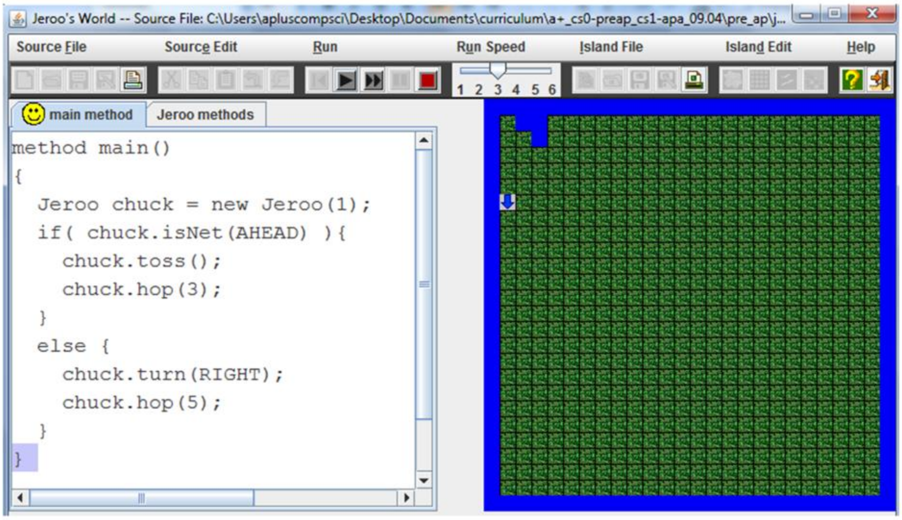
If the condition is false, `do something 2` will occur.



the if else statement

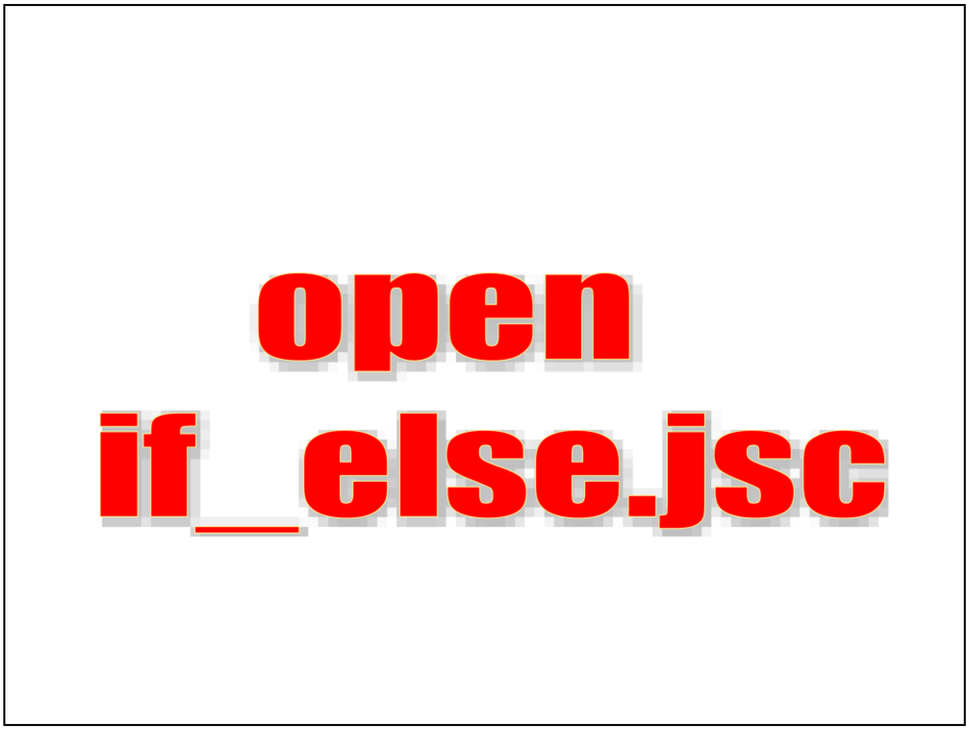
```
Jeroo chuck = new Jeroo(1);
if( chuck.isNet(AHEAD) ) {
  chuck.toss();
  chuck.hop(3);
}
else {
   chuck.turn(RIGHT);
   chuck.hop(5);
}
```

the if else statement

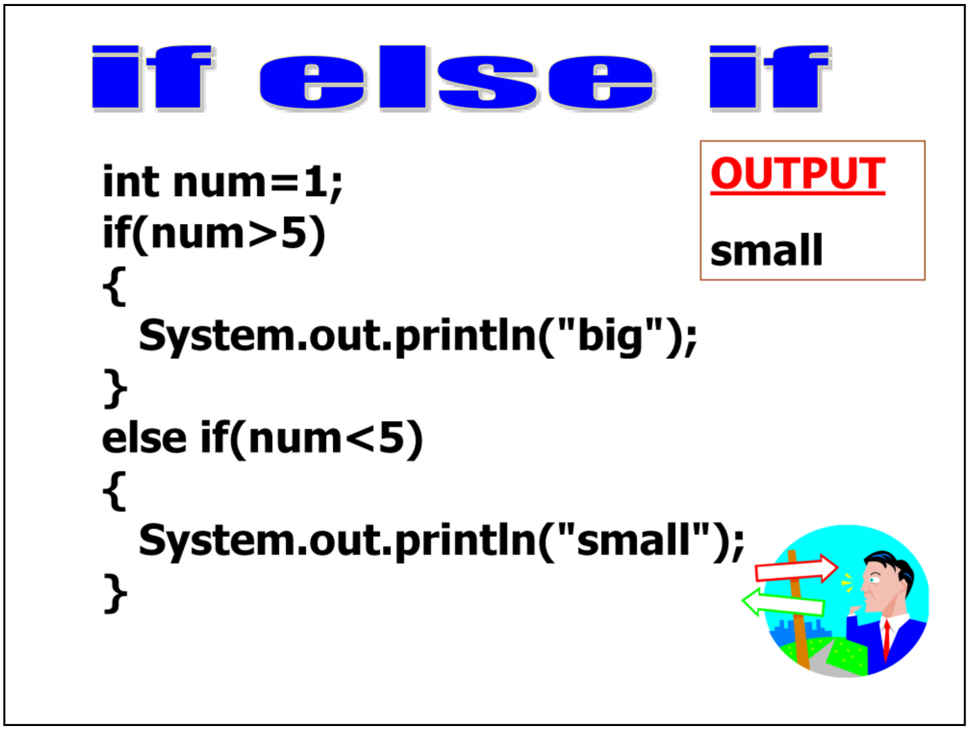


Jeroo's World -- Source File: C:\Users\apluscompsci\Desktop\Documents\curriculum\a+_cs0-preap_cs1-apa_09.04\pre_ap\j...
Source File
Source Edit
Run
Run Speed
Island File
Island Edit
Help
1 2 3 4 5 6
main method
Jeroo methods

```
method main()
{
  Jeroo chuck = new Jeroo(1);
  if( chuck.isNet(AHEAD) ){
    chuck.toss();
    chuck.hop(3);
  }
  else {
    chuck.turn(RIGHT);
    chuck.hop(5);
  }
}
```



open
if_else.jsc



if else if

```
int num=1;
if(num>5)
{
  System.out.println("big");
}
else if(num<5)
{
  System.out.println("small");
}
```

OUTPUT

```
small
```

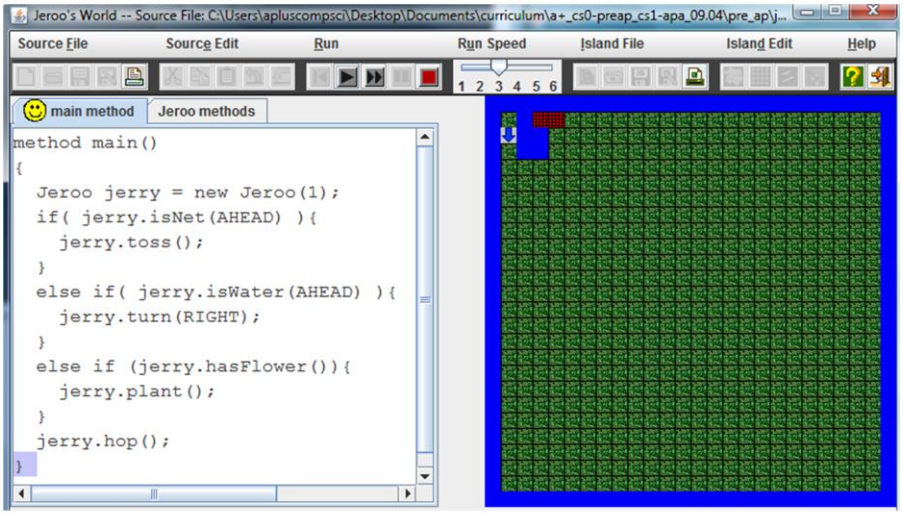
Nesting occurs when one thing is placed inside of another thing.

`if(num<10)` has been nested inside of `if(num>2)`

`if(num<10)` will only be tested if `if(num>2)` is true.

The else is associated with `if(num>2)`. Without the braces, the else would be associated with `if(num<10)` as if and else are paired based on proximity.

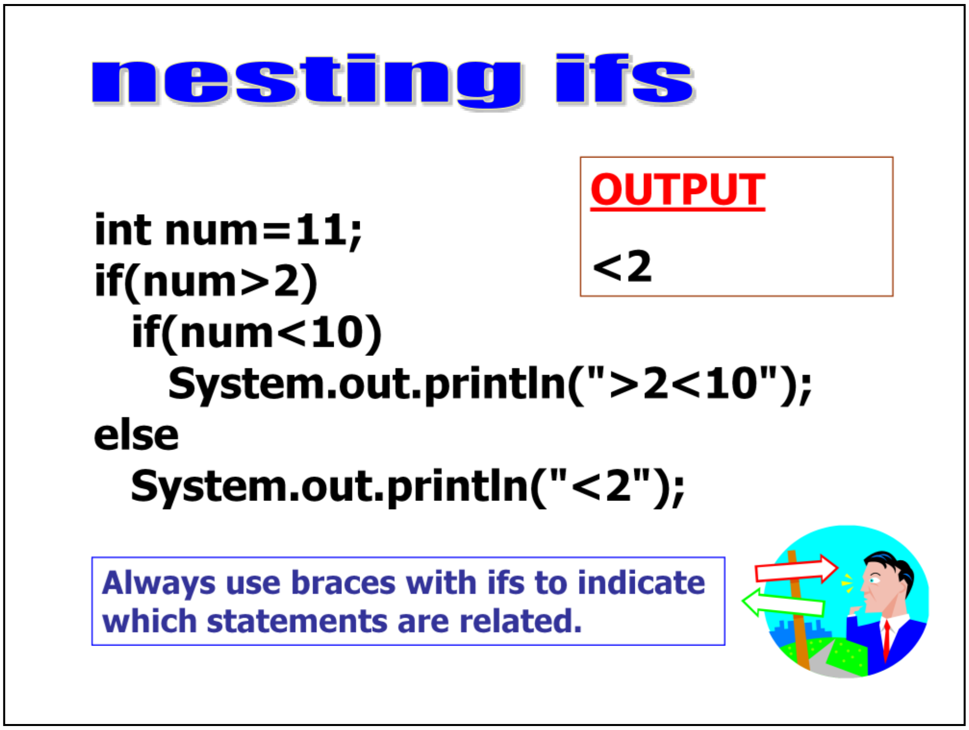
if else if



Jeroo's World -- Source File: C:\Users\apluscompsci\Desktop\Documents\curriculum\a+_cs0-preap_cs1-apa_09.04\pre_ap\j...
Source File
Source Edit
Run
Run Speed
1 2 3 4 5 6
Island File
Island Edit
Help
main method
Jeroo methods

```
method main()
{
  Jeroo jerry = new Jeroo(1);
  if( jerry.isNet(AHEAD) ){
    jerry.toss();
  }
  else if( jerry.isWater(AHEAD) ){
    jerry.turn(RIGHT);
  }
  else if (jerry.hasFlower()){
    jerry.plant();
  }
  jerry.hop();
}
```

open if_else_if.jsc



nesting ifs

```
int num=11;
if(num>2)
  if(num<10)
    System.out.println(">2<10");
else
  System.out.println("<2");
```

OUTPUT

```
<2
```

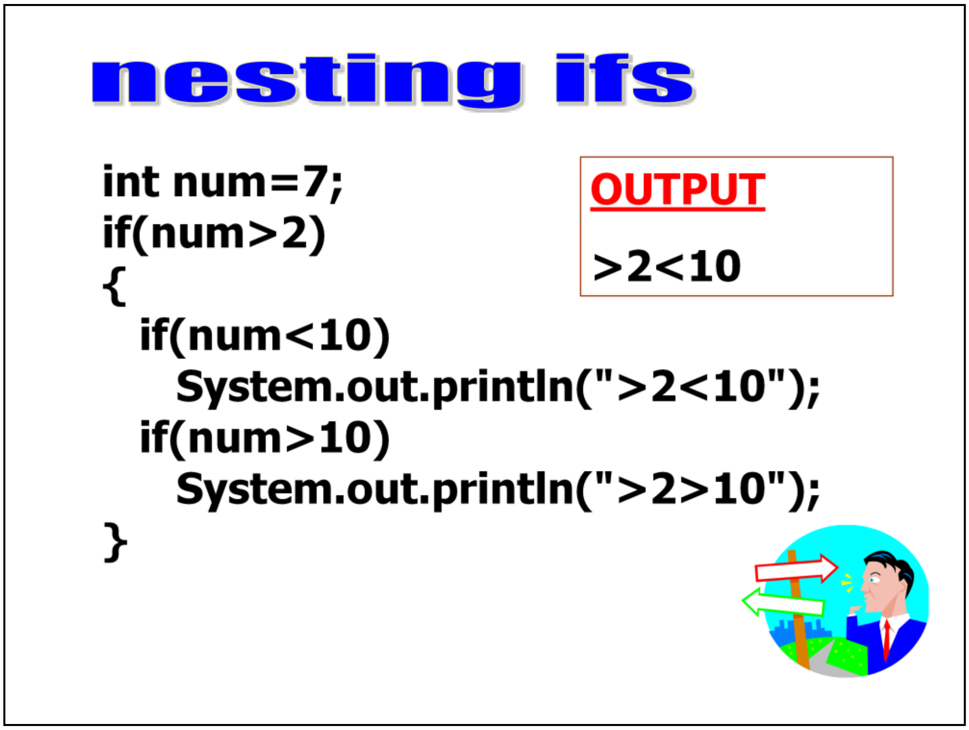
Always use braces with ifs to indicate which statements are related.

Nesting occurs when one thing is placed inside of another thing.

`if(num<10)` has been nested inside of `if(num>2)`

`if(num<10)` will only be tested if `if(num>2)` is true.

The else is associated with `if(num<10)`. If braces were present around `if(num<10)`, the else would be associated with `if(num>2)` as if and else are paired based on proximity.



nesting ifs

```
int num=7;
if(num>2)
{
  if(num<10)
    System.out.println(">2<10");
  if(num>10)
    System.out.println(">2>10");
}
```

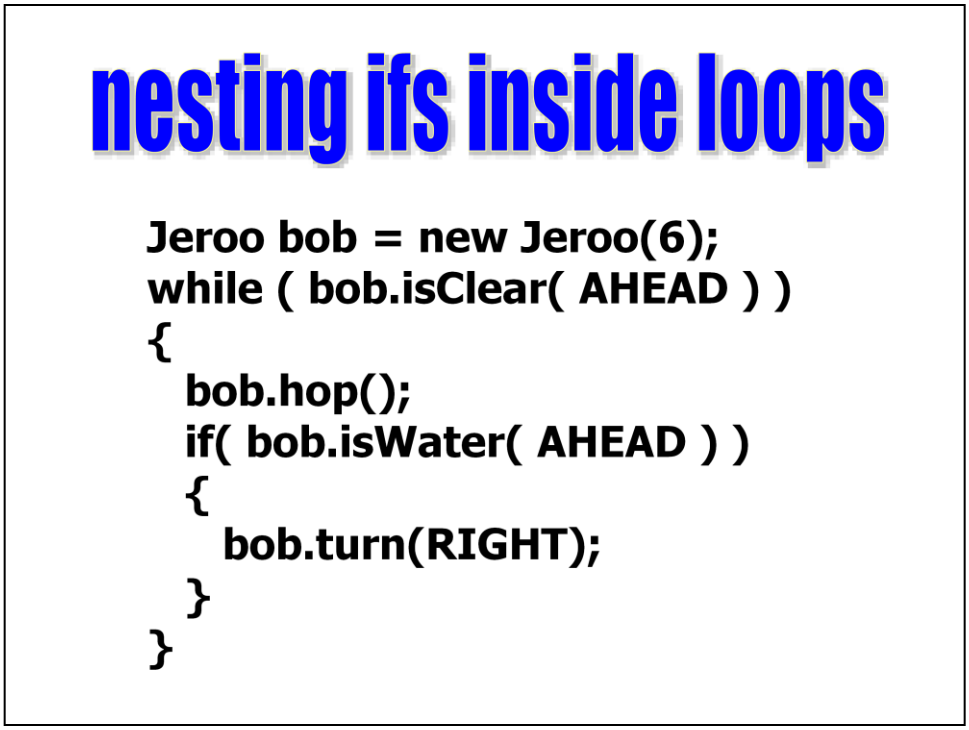
OUTPUT

```
>2<10
```

Nesting occurs when one thing is placed inside of another thing.

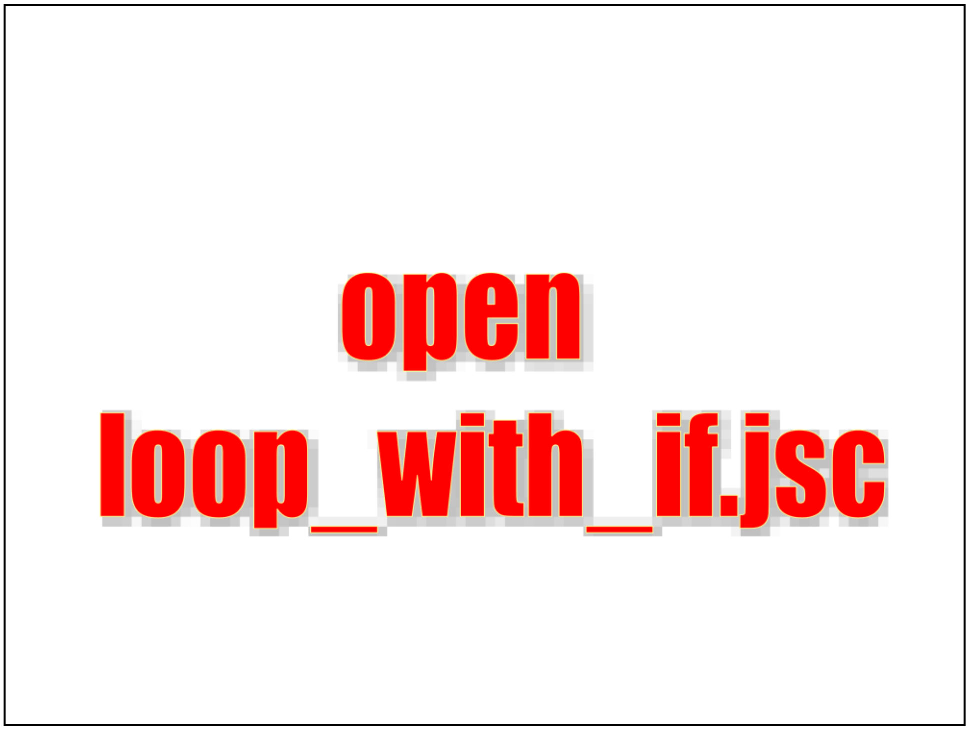
In the example, `if(num>2)` contains 2 ifs.   The 2 ifs have been nested in side of `if(num>2)`.

`if(num>2)` is true, the 2 nested ifs will be evaluated.

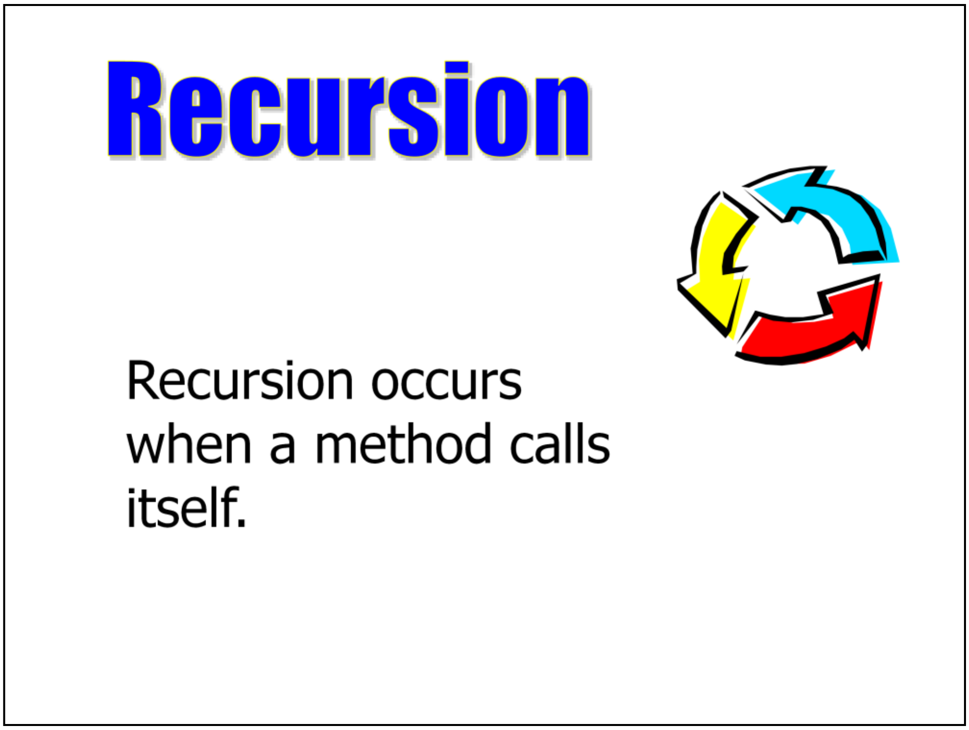


nesting ifs inside loops

```
Jeroo bob = new Jeroo(6);
while ( bob.isClear( AHEAD ) )
{
  bob.hop();
  if( bob.isWater( AHEAD ) )
  {
    bob.turn(RIGHT);
  }
}
```



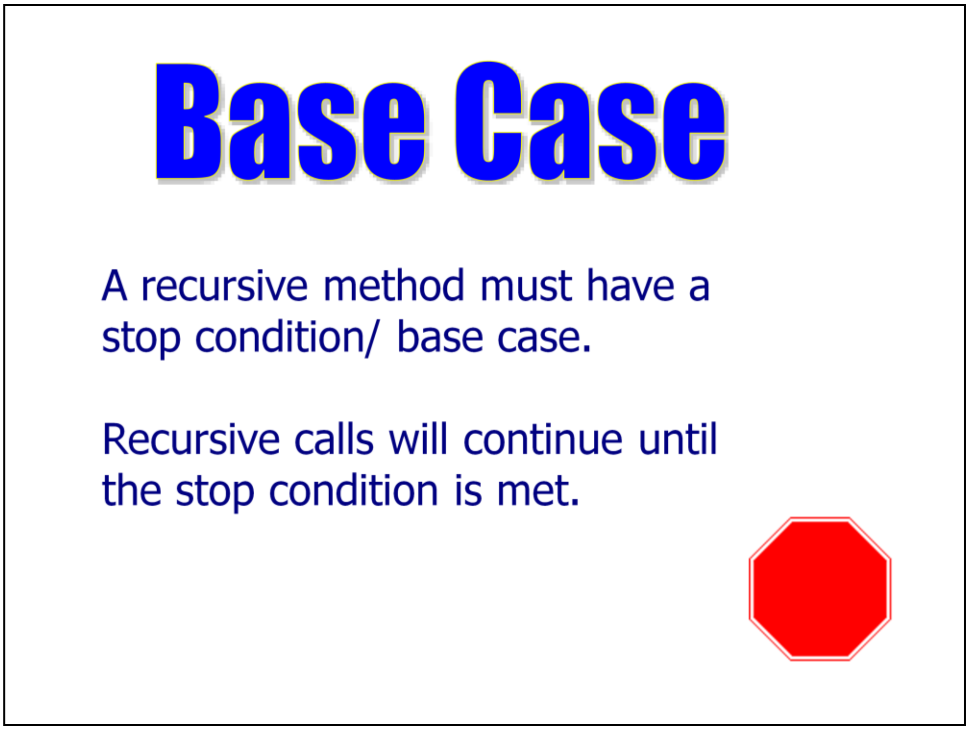
open loop_with_if.jsc



Recursion

Recursion occurs when a method calls itself.

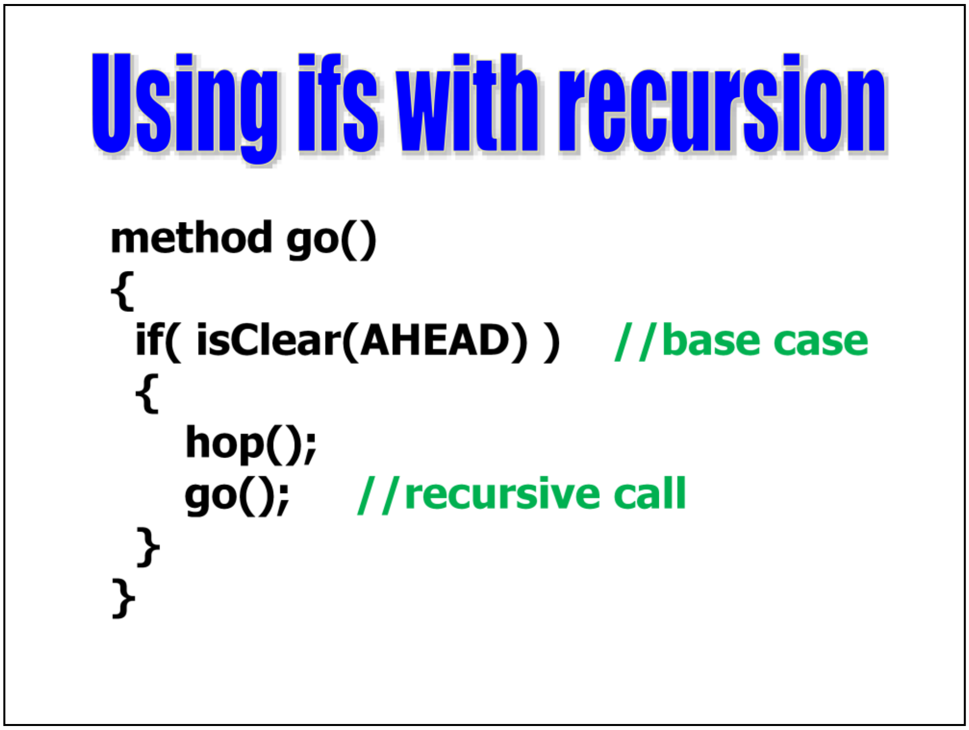
If a method contains a call to itself, that method is recursive. Recursion is a very useful programming tool if used properly.



Base Case

A recursive method must have a stop condition/ base case.

Recursive calls will continue until the stop condition is met.



Using ifs with recursion

```
method go()
{
 if( isClear(AHEAD) )    //base case
 {
    hop();
    go();     //recursive call
 }
}
```

open recursion_with_if.jsc

Start work on If Labs