

Introduction to Python

Conditionals (part 1)

Boolean expressions

A boolean expression is an expression that is either true or false.

Example using the equal operator `==` the expressions compare two operands and produces `True` if they are equal and `False` otherwise:

```
>>> 3 == 3
True
>>> 3 == 4
False
```

`True` and `False` are special values that belong to the class `bool` in Python.

```
>>> type(True)
<class 'bool'>
>>> type(False)
<class 'bool'>
```

More comparison operators

The `print()` function prints the specified message to the screen. Let's try:

```
x != y          # x is not equal to y
x > y           # x is greater than y
x < y           # x is less than y
x >= y          # x is greater than or equal to y
x <= y          # x is less than or equal to y
x is y          # x is the same as y
x is not y      # x is not the same as y
```

Logical Operators

There are three logical operators: and, or, and not. The semantics (meaning) of these operators is similar to their meaning in English.

Example:

```
x > 0 and x < 10
```

```
n%2 == 0 or n%3 == 0
```

When do you think the result of these expressions is True?

For Python any nonzero number is interpreted as "true"

```
>>> 15 and True  
True
```

Conditional execution

Conditional statements allows to check conditions and change the behavior of our programs. The simplest form is the if statement:

```
if x > 0 :
```

```
    print('x is positive')
```

CONDITION

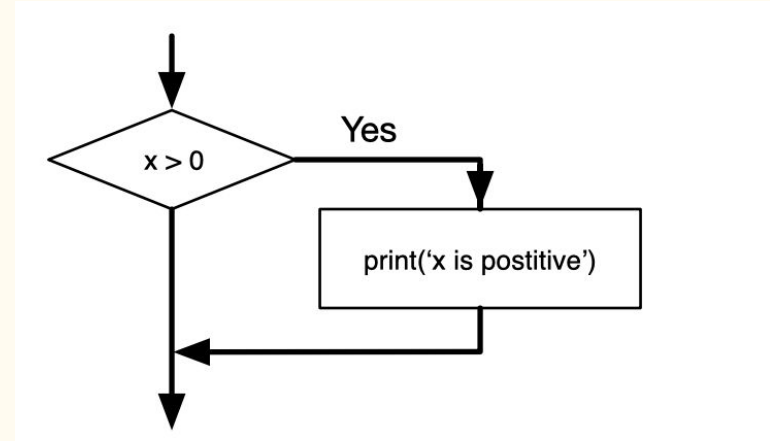
The boolean expression after the if statement is called the condition. We end the if statement with a colon character (:) and the line(s) after the if statement are indented.

Statements like this are called compound statements because they stretch across more than one line.

There is no limit on the number of statements that can appear in the body, but there must be at least one.

if Logic

```
if x > 0 :  
    print('x is positive')
```

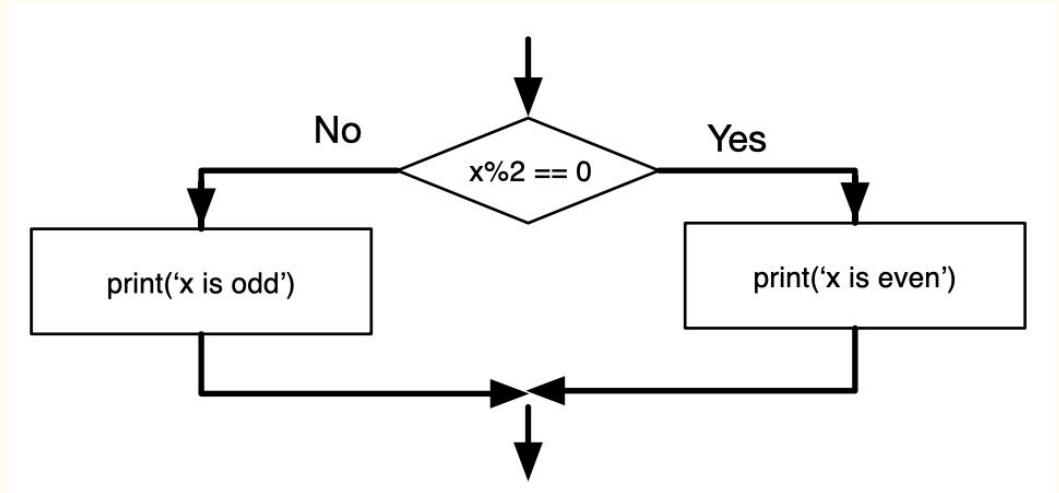


If the logical condition is true, then the indented statement gets executed. If the logical condition is false, the indented statement is skipped

Alternative execution

When there are two possibilities and the condition determines which one gets executed.

```
if x%2 == 0 :  
    print('x is even')  
else :  
    print('x is odd')
```



Here we have 2 branches

The alternatives are called branches

PAIR - PROGRAMMING

1. WORK ON “02 LAB - CONDITIONALS 1”
2. COMPLETE YOUR DAILY LOG
PROGRAMMING

QUESTIONS?