Introduction to Computer Programming

1.3 Operators

Comparison Operators

Comparison operators (==, !=, <, > ....) compare values and return a *Boolean* value: `True` or `False`

Commonly used comparison operators:

==	Equality
! =	Inequality
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

Examples:

In [18]:

```
print(10 < 9)
```

```
False
```

In [2]:

```
print(15 < 20)
```

```
True
```

In [4]:

```
print(20 <= 20)
```

```
True
```

In [2]:

```
A = 1

B = 2

C = type(A) == type(B)

print(C)
```

```
True
```

Logical Operators

Comparison operators compare two values.

Logical operators combine *multiple* expressions/variables with `True` / `False` (boolean) values and outout a *single* `True` / `False` (boolean) value.

Logical operators:
`and`
`or`
`not`

`X and Y`

Output:
`True` if statement `X` **and** statement `Y` *both* true.
Otherwise `False` .

`X or Y`

Output:
`True` if statement `X` **or** statement `Y` true.
Otherwise `False` .

Examples:

$10 < 9$ `False`

$20 <= 20$ `True`

In [20]:

```
print(10 < 9 and 20 <= 20)
```

```
False
```

In [30]:

```
print(10 < 9 or 20 <= 20)
```

```
True
```

In Python, the `not` operator negates a statement, e.g.:

In [6]:

```
a = 12
print(a < 0)
print(not a < 0)
```

```
False
True
```

Operator Precedence

1. Parentheses
2. Exponents
3. Multiplication, Division, Floor Division and Modulo (left to right)
4. Addition and Subtraction (left to right)
5. Comparison Operators (left to right)
6. Boolean not
7. Boolean and
8. Boolean or

Note that the comparison operators (`>=` , `<=` , `<` and `>`) are evaluated before the logical operators (`and` , `or`).

Example: Write a program, using comparison and logical operators, that answers questions based on the current time of day:

> **Is it lunchtime?**
> `True` if current time is between lunch start and end times.
> `False` if not.

In [11]:

```
# Variables
time = 9.00           # current time
lunch_starts = 13.00  # time lunch starts
lunch_ends =   14.00  # time lunch ends


# Logical and comparsion operators to lunchtime True / False value
lunchtime = time >= lunch_starts and time < lunch_ends


print("Is it lunchtime?", lunchtime)
```

```
Is it lunchtime? False
```

If we change the value of `time` , the program output changes.

Stacking Comparison Operators

Extract from example program:

```
lunchtime = time >= lunch_starts and time < lunch_ends
```

We can rewrite *stacking* the comparison operators:

```
time >= lunch_starts and time < lunch_ends
```

is the same as

```
lunch_starts <= time < lunch_ends
```

Summary

- **Arithmetic operators** (+, -, /, * ....) are used with numeric values to perform common mathematical operations (behave differently with strings).
- **Comparison operators** (==, !=, <, > ....) compare two *variables*.
  - The outcome of a comparison is a *Boolean* (True or False) value.
- **Logical operators** (`and` , `or`) compare Boolean `True` or `False` values, such as the outcomes of two *comparison operations*, to form logic statements.
  - The outcome of a logical operation is a *Boolean* (True or False) value.
  - The logical `not` operator returns the inverse Boolean value of a comparison.

In-class Demos

Example 1:
Write a program that creates 3 variables, `A` , `B` and `C` , with numerical values, and outputs a statement that tells the user if the values include any negative numbers.

tells the user if the values include any negative numbers.

In [20]:

```
A = 1
B = -1
C = 2

answer = A < 0 or B < 0 or C < 0

print('Values include negative numbers: ',answer)

print('Values are only positive numbers: ', not answer)
```

```
Values include negative numbers:  True
Values are only positive numbers:  False
```

Example 2:
Write a program that answers two questions based on the current time of day:

> **Is it lunchtime?**
> `True` if time between lunch start and end times.
> `False` if not.

> **Is it time for work?**
> `True` if time between work start and end times *and not* lunchtime.
> `False` if not.

In [21]:

```
# Variables
t = 9.00           # current time
Ls = 13.00         # lunch starts
Le = 14.00         # lunch ends
Ws = 8.00          # work starts
We = 17.00         # work ends

# lunchtime
lunchtime = Ls <= t < Le

# work_time
# between work start and end times and not lunchtime
work_time = Ws <= t < We and not lunchtime

# not before work or after work or lunchtime
# work_time = not ( t < Ws or t > We or lunchtime)




print("Is it work time?", work_time)
print("Is it lunchtime?", lunchtime)
```

```
Is it work time? True
Is it lunchtime? False
```