

# 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:**

<code>==</code>	Equality
<code>!=</code>	Inequality
<code>&gt;</code>	Greater than
<code>&lt;</code>	Less than
<code>&gt;=</code>	Greater than or equal to
<code>&lt;=</code>	Less than or equal to

**Examples:**

In [18]:

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

False

In [2]:

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

True

In [1]:

```
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 output 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 [3]:

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

False

In [4]:

```
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. Arithmetic operators (top to bottom)
  - \*\* Exponent
  - \* Multiplication
  - / Division
  - // Floor division (round down to the next integer)
  - % Modulo  $a \% b = a - b * (a // b)$  (remainder)
  - + Addition
  - Subtraction
3. Comparison operators: `<` , `<=` , `>` , `>=` , `!=` , `==`
4. Logical operators: `not` , `and` , `or` (left to right)

**Example:** Write a program, using comparison and logical operators, that answers a question 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 comparison 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** (+, -, /, \* ....)  
Used with numeric values to perform mathematical operations (behave differently with strings).
- **Comparison operators** (==, !=, <, > ....)  
Compare two *variables*.  
Outcome of a comparison is a *Boolean* (True or False) value.
- **Logical operators** ( and , or )  
Compare Boolean True or False values (e.g. outcomes of two *comparison operations*) to form logic statements.  
Outcome of a logical operation is a *Boolean* (True or False) value.  
Logical not operator returns the inverse Boolean value of a comparison.

## In-class Demos

**Example 1:**

Write a program that:

1. creates 3 variables, A , B and C , with numerical values
2. outputs a statement that tells the user if the values include *any* negative numbers.

In [ ]:

**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 [ ]: