Boolean Logic

Boolean values

- Are **TRUE** and **FALSE**

Boolean values

- Are **TRUE** and **FALSE**
  - » **Named after the English mathematician George Boole (1815-64) who developed Boolean algebra**

Boolean values

- Are **TRUE** and **FALSE**
- They can be established in a number of ways

Boolean values

- Are **TRUE** and **FALSE**
- They can be established in a number of ways
  - » **Declaration**

Boolean values

- Are **TRUE** and **FALSE**
- They can be established in a number of ways
  - » **Declaration**
  - » **Comparison**

Boolean values

- Are **TRUE** and **FALSE**
- They can be established in a number of ways
  - » **Declaration**
  - » **Comparison**
  - » **Boolean expression**

Declaration

- To **declare a variable** means to create it

Declaration

- To **declare a variable** means to create it
- Example:
  - » **In JavaScript declare the variable doorIsOpen and assert its value is true**

```
var doorIsOpen = true;
```

Comparison

- **Comparison operators** are used to establish a relationship between objects of the same type

Comparison operators

- < less than
- <= less than or equal to
- === equal to (strict includes operand types)
- !== not equal to (strict includes operand types)
- >= greater than or equal to
- > greater than

Comparison operators

- < less than
- <= less than or equal to
- === equal to (strict includes operand types)
- !== not equal to (strict includes operand types)
- >= greater than or equal to
- > greater than

  » **Comparison operators are infix and binary**

Comparison operators

- < less than
- <= less than or equal to
- === equal to (strict includes operand types)
- !== not equal to (strict includes operand types)
- >= greater than or equal to
- > greater than

» **Comparison operators are infix and binary**

> **The operator is between (infix) its two operands (binary)**

Examples

- maryAge < aliAge
- year > 1582
- elephantWeight <= mouseWeight
- thisYear === 2017
- thisYear !== leapYear

Boolean expression

- **Logical operators** can be used to construct Boolean expressions from Boolean values

Boolean expression

- **Logical operators** can be used to construct Boolean expressions from Boolean values
  - » **The operands of the operators must be Boolean**

Logical operators

- && and
- | | or
- ! not

Boolean expression example

- Assume the value of year is a calendar year such as 2017
- The following expression is true if year is a leap year in the Gregorian calendar

```
( year % 4 === 0        // Remainder of year is 0
                        // on division by 4
&&                      // AND
year % 100 !==0 )       // Remainder of year is not 0
                        // on division by 100
| |                     // OR
year % 400 === 0        // Remainder of year is 0
                        // on division by 400
```

Using the not operator

```
! ( year % 100 === 0 )

        ===

  ( year % 100 !== 0 )
```

Using the not operator

- The following expression is true if year is a leap year in the Gregorian calendar

```
( year % 4 === 0          // Remainder of year is 0
                          // on division by 4
&&                        // AND
! ( year % 100 === 0 ) )  // Remainder of year is not 0
                          // on division by 100
| |                       // OR
year % 400 === 0          // Remainder of year is 0
                          // on division by 400
```