Python: Operators

1405 Instructor: Ruiqing (Sam) Cao

Operators (focus on the highlighted)

Type	Description
Arithmetic	Mathematical calculations (on `bool`, `int`, and `float`)
Comparison	Compare two values and return a `bool` type
Logical	Combine two `True`/`False` values and return a `bool` type
Assignment	Store a value in a variable (works on all types)
Bitwise	Common use cases very similar to logical operators &, |, ~
Membership	Check if a value is a member of a sequence or collection
Identity	Check if two objects are the same (not just equal ==, but identical in memory)

Arithmetic Operators (math calculations)

Operator	Description	Example	Result
+	Addition	5 + 3	8
-	Subtraction	5 - 3	2
*	Multiplication	5 * 3	15
/	Division	5 / 3	1.666...
//	Floor Division	5 // 3	1
%	Modulus (remainder)	5 % 3	2
**	Exponentiation	5 ** 3	125

Comparison Operators (compare values)

Operator	Description	Example	Result
==	Equal to	`5 == 3`	`False`
!=	Not equal to	`5 != 3`	`True`
>	Greater than	`5 > 3`	`True`
<	Less than	`5 < 3`	`False`
>=	Greater than or equal to	`5 >= 3`	`True`
<=	Less than or equal to	`5 <= 3`	`False`

Logical Operators (combine Boolean)

Operator	Description	Example	Result
`and`	Returns `True` if both are `True`	`True and False`	`False`
`or`	Returns `True` if at least one is `True`	`True or False`	`True`
`not`	Inverts the `Boolean` value	`not True`	`False`

Assignment Operators (store value)

Operator	Description	Example	Equivalent To
=	Assign	x = 5	x = 5
+=	Add and assign	x += 3	x = x + 3
-=	Subtract and assign	x -= 3	x = x - 3
*=	Multiply and assign	x *= 3	x = x * 3
/=	Divide and assign	x /= 3	x = x / 3
//=	Floor-divide and assign	x //= 3	x = x // 3
%=	Modulus and assign	x %= 3	x = x % 3
**=	Exponent and assign	x **= 3	x = x ** 3

Bitwise Operators (binary operation)

Operator	Description	Example	Result
&	Bitwise AND	`5 & 3`	`1`
|	Bitwise OR	`5 | 3`	`7`
~	Bitwise NOT	`~5`	`-6`
^	Bitwise XOR	`5 ^ 3`	`6`
<<	Left Shift	`5 << 1`	`10`
>>	Right Shift	`5 >> 1`	`2`

- For Boolean indexing in NumPy and Pandas: and (&) or (|) not(~)
- If you know Stata, the only different one here is not (~)

Membership and Identity Operators

Operator	Description	Example	Result
`in`	True if value exists	`'a' in 'apple'`	True
`not in`	True if value does not exist	`'b' not in 'apple'`	True

Operator	Description	Example	Result
`is`	True if same object	`x is y`	
`is not`	True if not same object	`x is not y`	

Exercise: Variables & Operators

1. `x = 10.5 y = x`

   What do the following expressions evaluate to, and why?

   `x==10.5` `x!=y` `int(x)`

2. Write the expressions or statements for the following tasks

   x is no less than y or 2*y is greater than -x

   Assign 3 to x, then assign `True` to x. Print whether x equals 3

3. Create two variables x and y and assign `10` and `20` as their values in one statement (hint: use value unpacking). Now swap the values of x and y without re-assigning these values directly

Exercise: Variables & Operators

1. `x = 10.5 y = x`

   What do the following expressions evaluate to, and why?

   `x==10.5` *True* `x!=y` *False* `int(x)` *10*

2. Write the expressions or statements for the following tasks

   x is no less than y or 2*y is greater than -x *(x>=y) or (2*y>-x)*

   Assign 3 to x, then assign True to x. Print whether x equals 3

```
x= 3
x= True
x==3
```

3. Create two variables x and y and assign 10 and 20 as their values in one statement (hint: use value unpacking). Now swap the values of x and y without re-assigning these values directly