

# ITP 115

# Programming in Python

## Expressions and Operators

# Numbers

- Don't panic! This isn't a math class
- But...sometimes we need to do mathematical operations with numbers (like add, multiply, etc.)

# Recall

- Python has two types of variables to store numbers
- **int**                      3        -1        0        2011
- **float**                    3.14            0.094        -12.0
- We can do mathematical operations with each

# Arithmetic Operations

## `int` (integers)

Operator	Description	Example	Evaluates To
<code>+</code>	Addition	<code>7 + 3</code>	<code>10</code>
<code>-</code>	Subtraction	<code>7 - 3</code>	<code>4</code>
<code>*</code>	Multiplication	<code>7 * 3</code>	<code>21</code>
<code>/</code>	Division (True)	<code>7 / 3</code>	<code>2.333333</code>
<code>//</code>	Division (Integer)	<code>7 // 3</code>	<code>2</code>
<code>%</code>	Modulus	<code>7 % 3</code>	<code>1</code>
<code>**</code>	Exponent	<code>7 ** 3</code>	<code>343</code>

# Arithmetic Operations

## **float** (real numbers)

Operator	Description	Example	Evaluates To
<b>+</b>	Addition	<b>7.0 + 3.0</b>	<b>10.0</b>
<b>-</b>	Subtraction	<b>7.0 - 3.0</b>	<b>4.0</b>
<b>*</b>	Multiplication	<b>7.0 * 3.0</b>	<b>21.0</b>
<b>/</b>	Division (True)	<b>7.0 / 3.0</b>	<b>2.333333</b>
<b>//</b>	Division (Integer)	<b>7.0 // 3.0</b>	<b>2.0</b>
<b>%</b>	Modulus	<b>7.0 % 3.0</b>	<b>1.0</b>
<b>**</b>	Exponent	<b>7.0 ** 3.0</b>	<b>343</b>

# General Rules

- With many operations, they work as you would expect

$$\begin{array}{ccccc} 4 & + & 3 & \rightarrow & 7 \\ \text{int} & & \text{int} & & \text{int} \end{array}$$

$$\begin{array}{ccccc} 4 & * & 3 & \rightarrow & 12 \\ \text{int} & & \text{int} & & \text{int} \end{array}$$

$$\begin{array}{ccccc} 4.0 & + & 3.0 & \rightarrow & 7.0 \\ \text{float} & & \text{float} & & \text{float} \end{array}$$

$$\begin{array}{ccccc} 4.0 & * & 3.0 & \rightarrow & 12.0 \\ \text{float} & & \text{float} & & \text{float} \end{array}$$

# General Rules

- What about combining **float** and **int**?

$$4 + 3.0 \rightarrow 7.0$$

int

float

float

$$4 * 3.0 \rightarrow 12.0$$

int

float

float

$$4.0 + 3 \rightarrow 7.0$$

float

int

float

$$4.0 * 3 \rightarrow 12.0$$

float

int

float

# What about Division?

4 / 3 → 1.33333

int

int

float

**That's odd...we  
expected an *int***

4 // 3 → 1

int

int

int

**Wait? What?**



# Two Types of Division

- True Division /
  - This is what think we usually think of as "division"
  - Result will always be a **float** regardless of the input types

- Examples

**10 / 5 → 2.0**

**10.0 / 5 → 2.0**

**4 / 3 → 1.3333**

**99 / 100.0 → 0.99**

# Two Types of Division

- Integer Division `//`
  - Gives you only the integer part of division
  - **Truncates** (or removes) the decimal part of the answer

`int // int → int`  
`float // float → float`  
`float // int → float`

- Examples

`10 // 5 → 2`  
`4 // 3 → 1`      (*not 1.333*)  
`99 // 100 → 0`      (*not 0.9999*)

# The modulo operator (%)

- *Modulo* (modulus – or *mod* for short) gives you the remainder from division
- Example:  
14 divided by 4 is 3 *with a remainder of 2*  
**14 % 4 = 2**
- Uses
  - Determining if an integer is odd or even
  - Determining if one integer is evenly divisible by another integer

# Arithmetic operations

- Arithmetic operators only work in pairs
  - Expressions with more than 2 operators are really a series of steps of only 2 operands
  - Results are compounded and used as next operand

**item1 + item2 + item3 + item4**

**((item1 + item2) + item3) + item4**

# Operator precedence

- **PEMDAS**
- Can prioritize with parenthesis  
 $(\text{cost} + \text{tax}) * \text{discount}$   
 $\text{cost} + (\text{tax} * \text{discount})$
- Without parentheses, expressions are evaluated according to the rules of precedence

# Partial List of Operators

*Evaluated sooner (higher precedence)*

Category	Operators
Parentheses (grouping)	()
Exponent	$a^{**}b$
Positive, Negative	$+a$ , $-a$
Multiplication, Division, Modulus	$a * b$ , $a / b$ , $a // b$ , $a \% b$
Addition, Subtraction	$a + b$ , $a - b$

*Evaluated later (lower precedence)*

# Specialized assignment operators

- Assignment operators can be combined with arithmetic operators (including  $-$ ,  $*$ ,  $/$ , and  $\%$ )

- Example

**amount = amount \* 2**

Can be written as

**amount \*= 2**

Other examples

**a -= 2**

**rate /= 59**

**age += 1**

**hours %= 24**