

CS 152: Operations, Input, Types

CS 152: Python for STEM

Colorado State University
Computer Science Department

Slides Originally Created by Albert Lionelle and Updated by Marcia Moraes



Colorado State University

Weekly Announcements!

TODO Reminders:

- Setup MS Teams
- Reading 1 (zyBooks)
- Reading 2 (zyBooks)

**THE EXPERT
AT ANYTHING
WAS ONCE A
BEGINNER.**

- HELEN HAYES



DAILY INSPIRATIONAL QUOTES
facebook.com/d.i.quotes

Recall Activity

- Individually
 - Grab a paper and write at least three concepts that you can remember from our last class
- With your neighbor(s)
 - Discuss what each other could remember. Did you remember the same things? What did you learn from each other?
- Turn your paper to the TAs or myself at the end of the class

What is missing?

- What do we need to do in order to make the following program generic, meaning that it could work for any values of x and y?

```
1  x = 10
2  print(x)
3  x = x + 10
4  print(x)
5  print(x + 10)
6  print('value of x = ', x, ' ', 'value of x + 10 = ', x + 10)
7  x = x/10
8  y = 2
9  x = x * y
10 y = x
11 print(x, end=' ')
12 print(y)
```

Input: Reading

- `input()` function
 - reads a string (sequence of characters) that the user typed
 - `text = input()`

```
print("Enter your name: ")  
name = input()  
print("You name is: ", name)
```

OR

```
name = input("Enter your name: ")  
print("You name is: ", name)
```

Input: Reading

- Analyze the program below
 - Considering that input() reads strings, do you think we are going to have a problem when we run this program?

```
print("Enter your name: ")
name = input()
print("You name is: ", name)
print("Enter your age: ")
age = input()
print("Your age is: ", age, " In 10 years you will have: ", age + 10)
```



```
print("Your age is: ", age, " In 10 years you will have: ",
age + 10)
```

TypeError: can only concatenate str (not "int") to str

Types in Python

- We already saw that Python uses “implicit”/weak typing
 - figures out types for you!
- Somethings you want to specify type
 - int - whole numbers only
 - float – floating point/decimal numbers
 - str – strings (sequence of characters)
- Most useful on getting client input
 - `answer = int(input(“get the answer”))`

Input: Reading – Class Activity 1

- Discuss with your peers and change the program below in order to read an int value for age.

```
print("Enter your name: ")
name = input()
print("You name is: ", name)
print("Enter your age: ")
age = input()
print("Your age is: ", age, " In 10 years you will have: ", age + 10)
```


Arithmetic Expressions

- combination of items, like variables, literals, operators, and parentheses, that evaluates to a value

Arithmetic operator	Description
+	The addition operator is + , as in $x + y$.
-	The subtraction operator is - , as in $x - y$. Also, the - operator is for negation , as in $-x + y$, or $x + -y$.
*	The multiplication operator is * , as in $x * y$.
/	The division operator is / , as in x / y .
**	The exponent operator is ** , as in $x ** y$ (x to the power of y).

// - returns an integer result (the floor)

% - (modulo) operator returns the remainder

Compound Operators

- provide a shorthand way to update a variable

Compound operator	Expression with compound operator	Equivalent expression
Addition assignment	<code>age += 1</code>	<code>age = age + 1</code>
Subtraction assignment	<code>age -= 1</code>	<code>age = age - 1</code>
Multiplication assignment	<code>age *= 1</code>	<code>age = age * 1</code>
Division assignment	<code>age /= 1</code>	<code>age = age / 1</code>
Modulo (operator further discussed elsewhere) assignment	<code>age %= 1</code>	<code>age = age % 1</code>

Arithmetic Expressions and Compound Operators – Class Activity 2

- Write exactly what will be printed in the program below:

```
x = 15
y = 12
print("Line 1:", x/y)
print('Line 2:', x%y)
print("Line 3:", x//y)
x = 4
y = 2
print('Line 4:', x**y)
y-=1
print('Line 5:', x**y)
print("Line 6:", 16 - 2 * 5 // 3 + 1)
x = 15
y += x
x = 22
print("Line 7:", x)
print('Line 8:', y)
```

Objects

- are used to represent everything in a Python program, including integers, strings, functions, lists, etc.
- Each object has:
 - Value: A value such as "20", "abcdef", or 55.
 - Type: The type of the object, such as integer or string.
 - Identity: A unique identifier that describes the object (address in memory).

```
x = 10
print(x)
print(type(x))
print(id(x))
```



```
10
<class 'int'>
140706803187040
```

	Memory	
140706803187039		
140706803187040	10	x
140706803187041		

Strings Basic

- Immutable sequence of characters
- `name = "Marcia"`

0	M
1	a
2	r
3	c
4	i
5	a

- `print(name[0])` → 'M'
- `name[0] = 'm'` → error – Strings are immutable, can't be changed by indexes, update the variable by assigning an entirely new string

Strings Basic

- Concatenation → +

```
str1 = "CS"  
str2 = "152"  
str3 = str1 + str2  
print(str3)
```

- Formatting → f-string, allows a programmer to create a string with placeholder expressions that are evaluated as the program executes

```
dollar = 1  
reais = 5.17  
print(f'To buy {dollar} dollar, you need {reais} reais')
```

Formatting floating point numbers – Class Activity 3

- Write exactly what will be printed in the program below:

```
x = 4.33333339  
print(f'{x:.2f}')  
print(f'{x:.4f}')
```

Peer Coding – Class Activity 4

- Write a Python program that reads three numbers (num1, num2, num3) and calculates and prints the following:
 - Average of the numbers
 - Result of num1 divided by num2
 - Result of num1 module num2
 - Result of num1 floor num2
 - Area of a rectangle of side num1
 - Area of a triangle of base num1 and height num2
 - Area of a sphere with radius as num3, use math.pi for pi number

Peer Coding – Class Activity 5

- Dr. Green is looking for a bank that will give the most return on her money over the next 5 years. She has P100,000.00 into a savings account. The standard equation to calculate principal plus interest at the end of a period is:
 - $\text{amount} = P * (1 + I/M) ^ (N * M)$
- Where:
 - P – principal (amount of money to invest)
 - I – interest (percentage rate the bank pays to the investor)
 - N – number of years (time for which the principal is invested)
 - M – compound interval (the number of times per year the interest is calculated and added to the principal)
- Think about what problem do you need to solve, how you are doing to solve it (write in English the steps to do that), write a Python code to solve that.