

# Storing numbers

Susan Ibach | Technical Evangelist  
Christopher Harrison | Content Developer

# Many problems we have to solve involve math

- How much will I pay monthly on a mortgage?
- How much will this cost when I add taxes?
- How much milk do I need to use in this recipe if I want to double the recipe?

So it's important to be able to store and manipulate numbers as well as strings

```
age = 42  
print(age)
```

You can perform math operations on numeric values or on variables containing numeric values

```
width = 20
```

```
height = 5
```

```
area = width * height
```

```
perimeter = 2*width + 2*height
```

```
perimeter = 2*(width+height)
```

# These are the most common math operations

Symbol	Operation	Example
+	Addition	$5+2 = 7$
-	Subtraction	$5-2 = 3$
*	Multiplication	$5*2 = 10$
/	Division	$5/2 = 2.5$
**	Exponent	$5**2 = 25$
%	Modulo	$5\%2 = 1$

# Math rules haven't changed since school

## Order of operations

( ) parentheses

\*\* exponent (e.g. \*\*2 squared \*\*3 cubed)

\*/ multiplication and division

+ - addition and subtraction

# DEMO

---

Getting a computer to do your math homework

# Formatting numbers



Sometimes you will need to format the numbers when you display them to users

Syntax	Output
<code>print('I have %d cats' % 6)</code>	I have 6 cats
<code>print('I have %3d cats' % 6)</code>	I have 6 cats
<code>print('I have %03d cats' % 6)</code>	I have 006 cats
<code>print('I have %f cats' % 6)</code>	I have 6.000000 cats
<code>print('I have %.2f cats' % 6)</code>	I have 6.00 cats

You can also use a format method to format numeric values

Syntax	Output
<code>print("I have {0:d} cats".format(6))</code>	I have 6 cats
<code>print("I have {0:3d} cats".format(6))</code>	I have 6 cats
<code>print("I have {0:03d} cats".format(6))</code>	I have 006 cats
<code>print("I have {0:f} cats".format(6))</code>	I have 6.000000 cats
<code>print("I have {0:.2f} cats".format(6))</code>	I have 6.00 cats

# DEMO

---

Formatting numeric values

# Geek Tip!



- Sometimes commands are too long to fit on a single line
- You can use a “\” to indicate a command continues on the next line

```
total = 5 + 6 + 8 \  
        + 6 + 2
```

# Inputting numbers

# DEMO

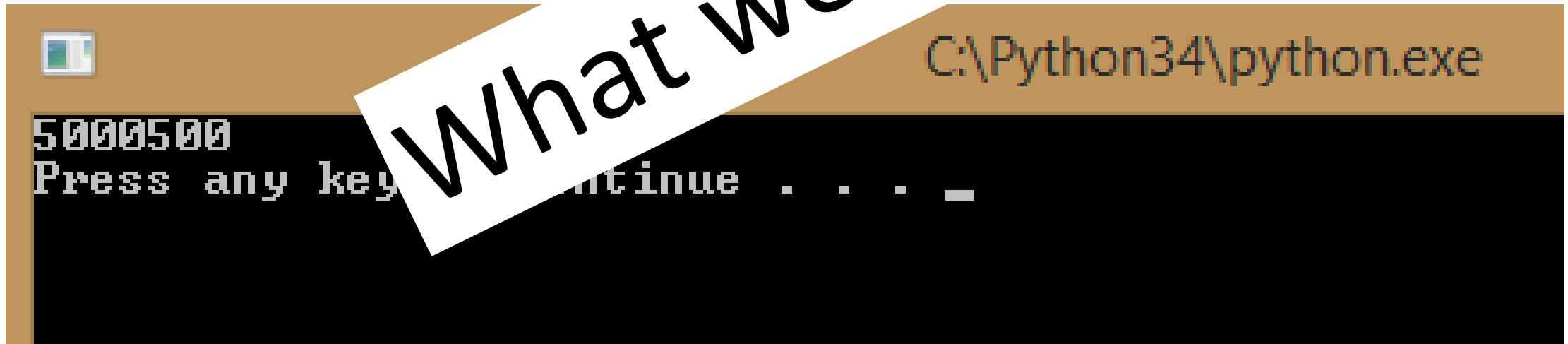
---

Asking a user to enter the numbers

Why did we get the wrong answer when we ask the user to enter their bonus and salary values?

```
salary = input("Please enter your salary")  
bonus = input("Please enter your bonus")  
payCheck = salary + bonus  
print(payCheck)
```

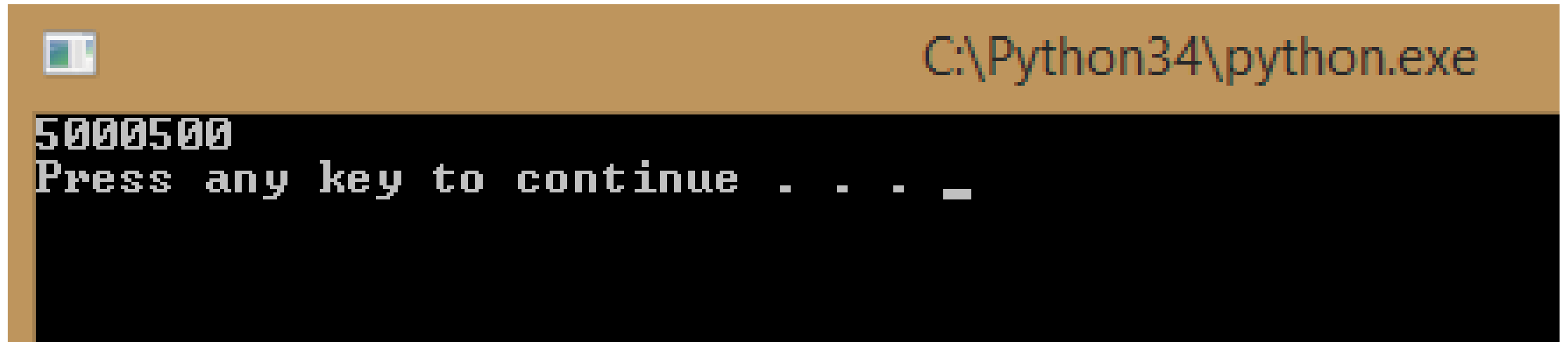
What went wrong?



The screenshot shows a Windows command prompt window titled "C:\Python34\python.exe". The prompt displays the output of the Python script: "5000500". Below the output, it says "Press any key to continue . . . \_". The text "What went wrong?" is overlaid on the image, pointing to the output. The error occurs because the `input()` function returns a string, and adding two strings results in concatenation rather than arithmetic addition.

Here is a hint: The input statement returns strings

```
salary = '5000'  
bonus = '500'  
payCheck = salary + bonus  
print(payCheck)
```



A screenshot of a Windows command prompt window. The title bar is brown and contains a small icon on the left and the text "C:\Python34\python.exe" on the right. The main area has a black background with white text. The first line of output is "5000500". The second line is "Press any key to continue . . . \_", where the underscore indicates a key has been pressed.

```
C:\Python34\python.exe  
5000500  
Press any key to continue . . . _
```



The program thought salary and bonus were strings so it concatenated instead of adding

```
salary = 5000  
bonus = 500  
payCheck = salary + bonus  
print(payCheck)
```



A screenshot of a Windows command prompt window with a brown title bar. The title bar contains a small icon on the left and the text "C:\Python34\p" on the right, with a mouse cursor hovering over it. The command prompt area has a black background with white text. It displays the output "5500" on the first line and "Press any key to continue . . ." on the second line.

We need a way to tell our program we want to treat values as a number instead of a string

There are functions to convert from one datatype to another.

<code>int(value)</code>	converts to an integer
<code>long(value)</code>	converts to a long integer
<code>float(value)</code>	converts to a floating number (i.e. a number that can hold decimal places)
<code>str(value)</code>	converts to a string

Which function should we use to fix our code?

If we convert the string to a float we get the desired result

```
salary = input("Please enter your salary: ")
bonus = input("Please enter your bonus: ")
payCheck = salary + bonus
payCheck = float(salary) + float(bonus)
print(payCheck)
```

What do you think will happen if someone types “BOB” as their salary?

The code crashes because we can’t convert the string “BOB” into a numeric value. We will learn how to handle errors later!

# DEMO

---

Changing the datatype

# Your Challenge – create a loan calculator

- Have the user enter the cost of the loan, the interest rate, and the number of years for the loan
- Calculate monthly payments with the following formula

$$M = L[i(1+i)^n] / [(1+i)^n - 1]$$

- M = monthly payment
- L = Loan amount
- i = interest rate (for an interest rate of 5%, i = 0.05)
- n = number of payments

# Congratulations!

- You can now solve mathematical problems with code!





# Microsoft

©2013 Microsoft Corporation. All rights reserved. Microsoft, Windows, Office, Azure, System Center, Dynamics and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries. The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.