

String variables and asking a user to enter a value

input

Christopher Harrison | Content Developer
Susan Ibach | Technical Evangelist

Two way conversations allow you to do more with computers

- Websites need your address and payment information so they can ship you products
- Insurance companies need information to calculate how much you would pay for car insurance
- Even calculators need you to enter the numbers before they can tell you the answer
- Cortana will tell you a joke if you ask her

How can we ask a user for information?

```
name = input("What is your name? ")
```

The **input** function allows you to specify a message to display and returns the value typed in by the user.

We use a variable to remember the value entered by the user.

We called our variable "name" but you can call it just about anything as long as the variable name doesn't contain spaces

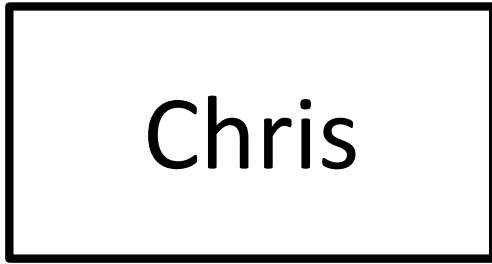
DEMO

Asking a user for input

Where do we store values?

Think of a variable as a box where you can store something and come back to get it later.

name



If you need to remember more than one value, just create more variables

name

Chris

favoriteMovie

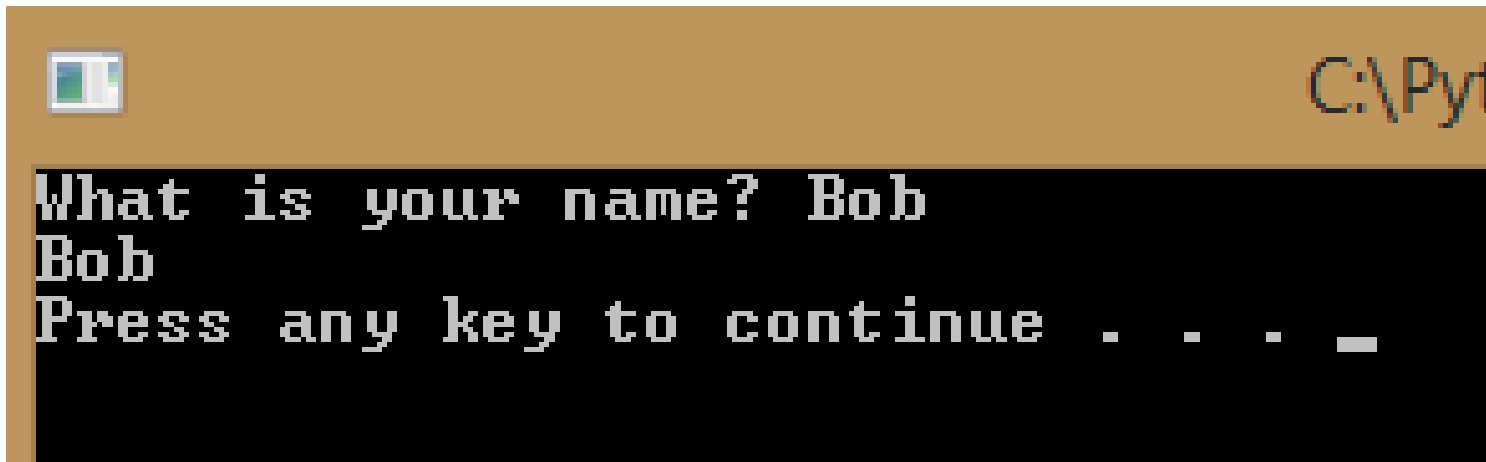
Real
Genius

city

Pasadena

You can access the value you stored later in your code

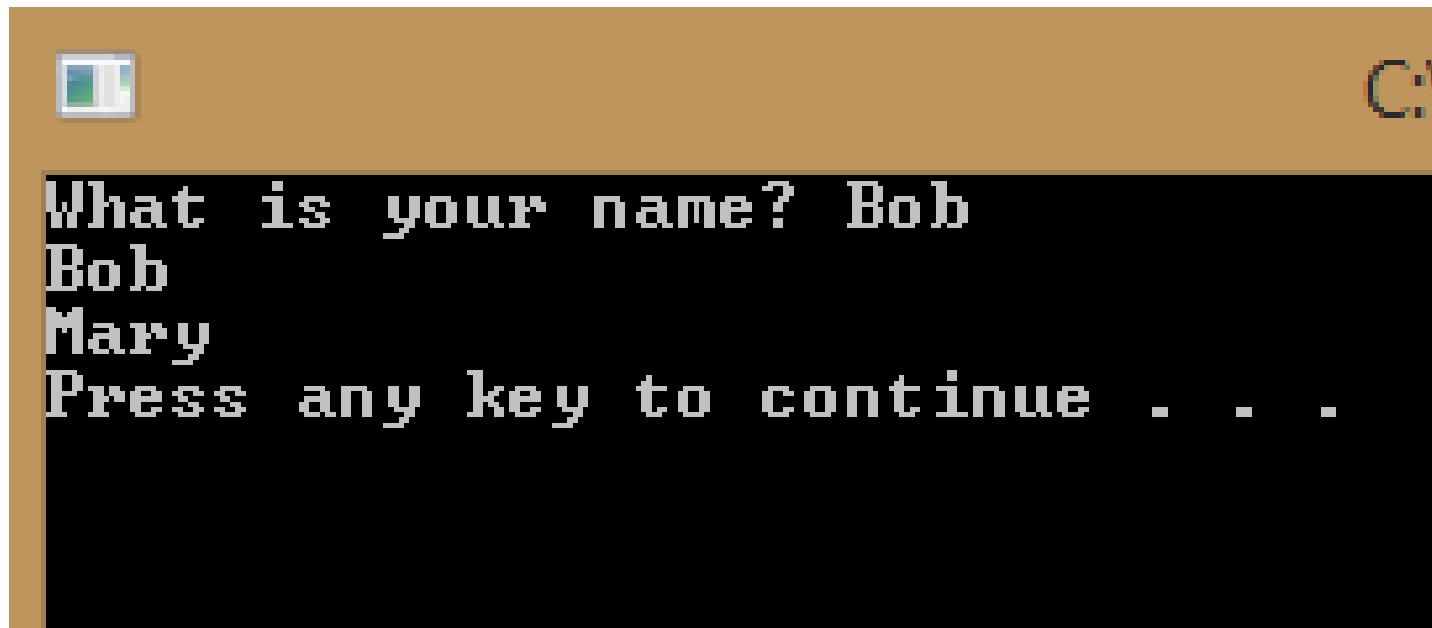
```
name = input("What is your name? ")  
print(name)
```

A screenshot of a Python terminal window. The window has a brown title bar with a small icon on the left and the text "C:\Pyt" on the right. The terminal area has a black background with white text. It shows the prompt "What is your name? Bob", the input "Bob", and the instruction "Press any key to continue . . . _".

```
C:\Pyt  
What is your name? Bob  
Bob  
Press any key to continue . . . _
```


You can also change the value of a variable later in the code

```
name = input("What is your name? ")  
print(name)  
name = "Mary"  
print(name)
```



A screenshot of a Windows command prompt window. The title bar is brown and shows a small icon on the left and 'C:\' on the right. The command prompt has a black background with white text. It displays the prompt 'What is your name? Bob', followed by the user input 'Bob', then the prompt 'Mary', and finally the text 'Press any key to continue . . .'.

DEMO

Accessing a value entered by a user

What should I call them?

Variable names

- Rules
 - Can not contain spaces
 - Are case sensitive
 - firstName and firstname would be two different variables
 - Cannot start with a number
- Guidelines
 - Should be descriptive but not too long (favoriteSign not yourFavoriteSignInTheHoroscope)
 - Use a casing "scheme"
 - camelCasing or PascalCasing

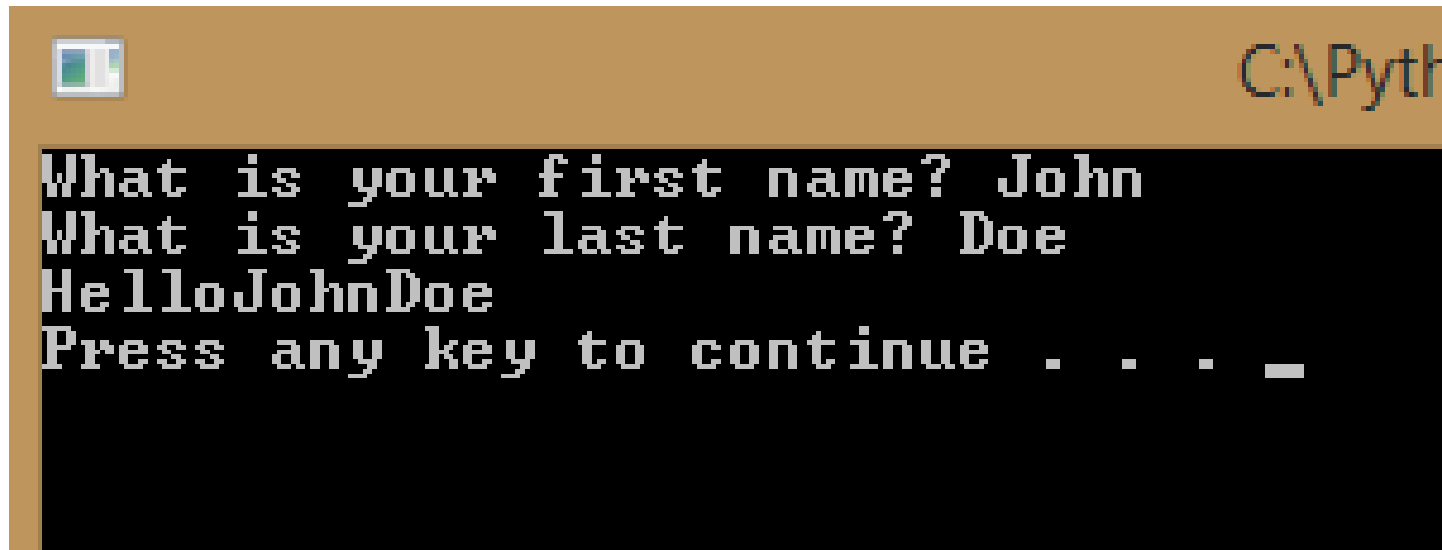
Which of the following do you think would be good names for variables?

- Variable1
- First Name
- Date
- 3Name
- DOB
- DateOfBirth
- YourFavoriteSignInTheHoroscope

Manipulating variables

You can combine variables and strings with the + symbol

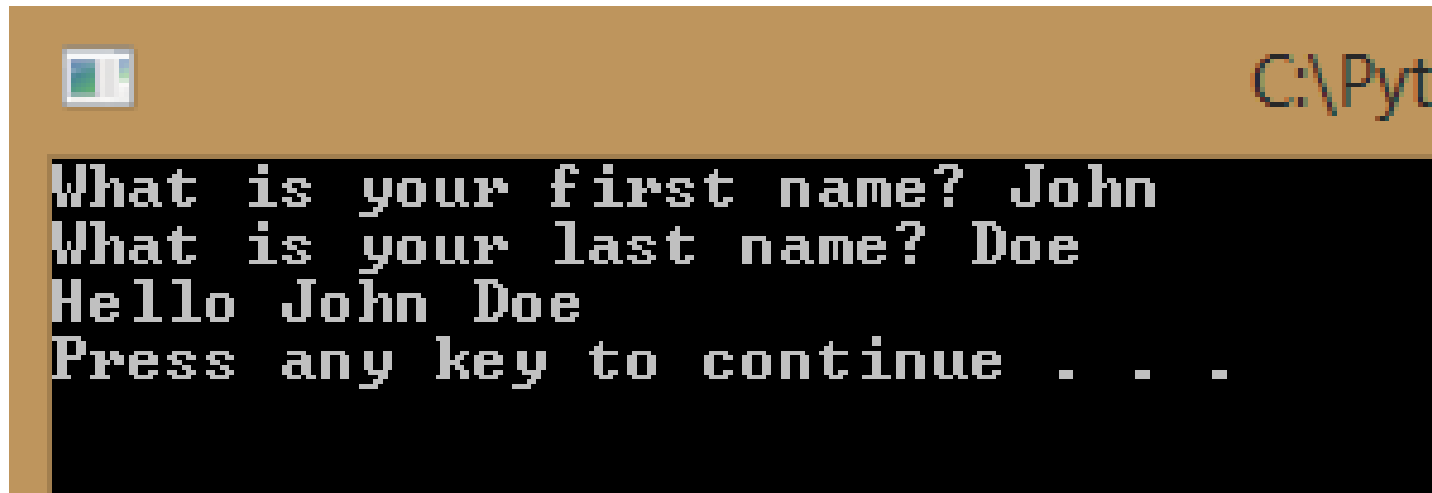
```
firstName = input("What is your first name? ")
lastName = input("What is your last name? ")
print("Hello" + firstName + lastName)
```

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:\Pyth" on the right. The command prompt has a black background with white text. It shows the execution of a Python script. The first two lines are prompts: "What is your first name? John" and "What is your last name? Doe". The third line shows the output: "HelloJohnDoe". The fourth line is a prompt: "Press any key to continue . . . _".

```
C:\Pyth
What is your first name? John
What is your last name? Doe
HelloJohnDoe
Press any key to continue . . . _
```

Often you need to add punctuation or spaces to format the output correctly

```
firstName = input("What is your first name? ")  
lastName = input("What is your last name? ")  
print("Hello " + firstName + " " + lastName)
```



A screenshot of a Python terminal window. The window has a title bar with a small icon on the left and the path 'C:\Pyth' on the right. The terminal content shows the execution of the code from the previous block. The prompts 'What is your first name?' and 'What is your last name?' are followed by the user inputs 'John' and 'Doe' respectively. The output is 'Hello John Doe', and the prompt 'Press any key to continue . . .' is shown at the bottom.

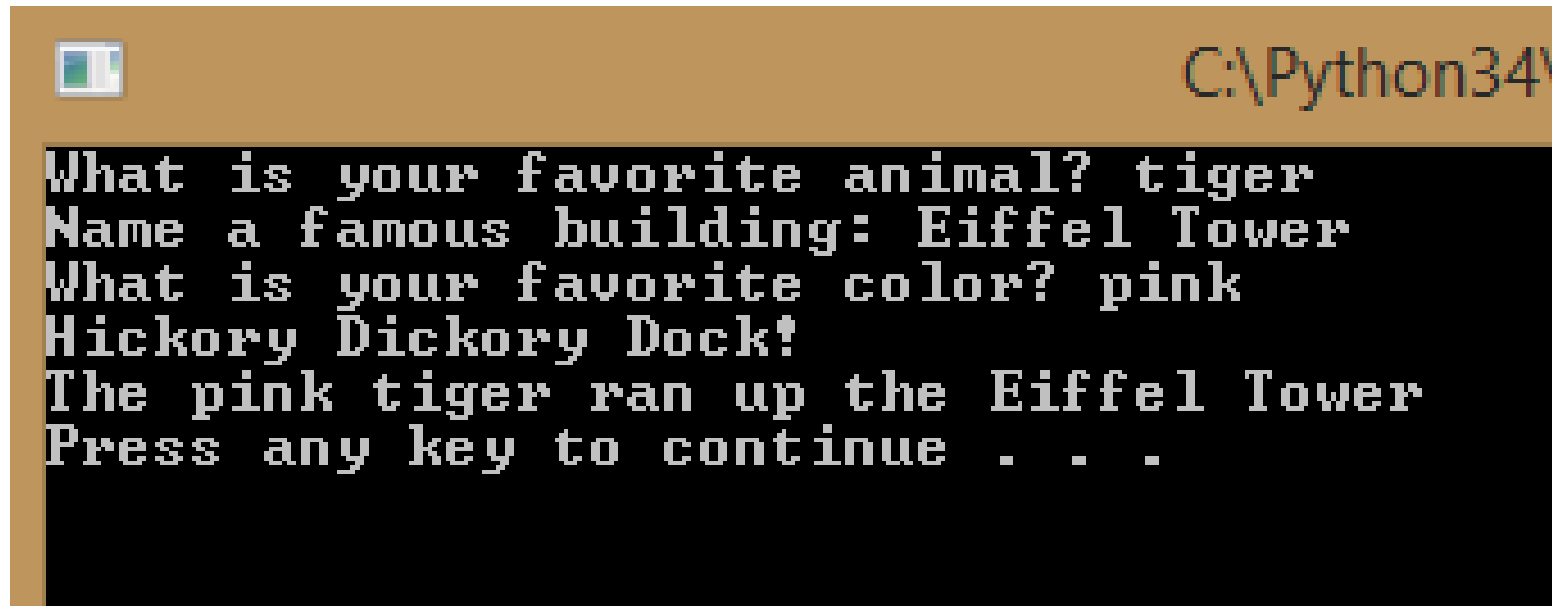
```
What is your first name? John  
What is your last name? Doe  
Hello John Doe  
Press any key to continue . . .
```


DEMO

Formatting output

Now you can create a story teller program!

```
animal = input("What is your favorite animal? ")
building = input("Name a famous building: ")
color = input("What is your favorite color? ")
print("Hickory Dickory Dock!")
print("The "+color+" "+animal+" ran up the "+building)
```



```
C:\Python34\
What is your favorite animal? tiger
Name a famous building: Eiffel Tower
What is your favorite color? pink
Hickory Dickory Dock!
The pink tiger ran up the Eiffel Tower
Press any key to continue . . .
```

Variables also allow you to manipulate the contents of the variable

```
message = 'Hello world'  
print(message.lower())  
print(message.upper())  
print(message.swapcase())
```



C:\Python34\py

```
hello world  
HELLO WORLD  
hELLO WORLD  
Press any key to continue . . .
```

DEMO

Manipulating values with string functions

Geek Tip!



- Lower, upper, and swapcase are different string functions
- Because we are storing a string in the variable, we can use any of the Python string functions to manipulate the string

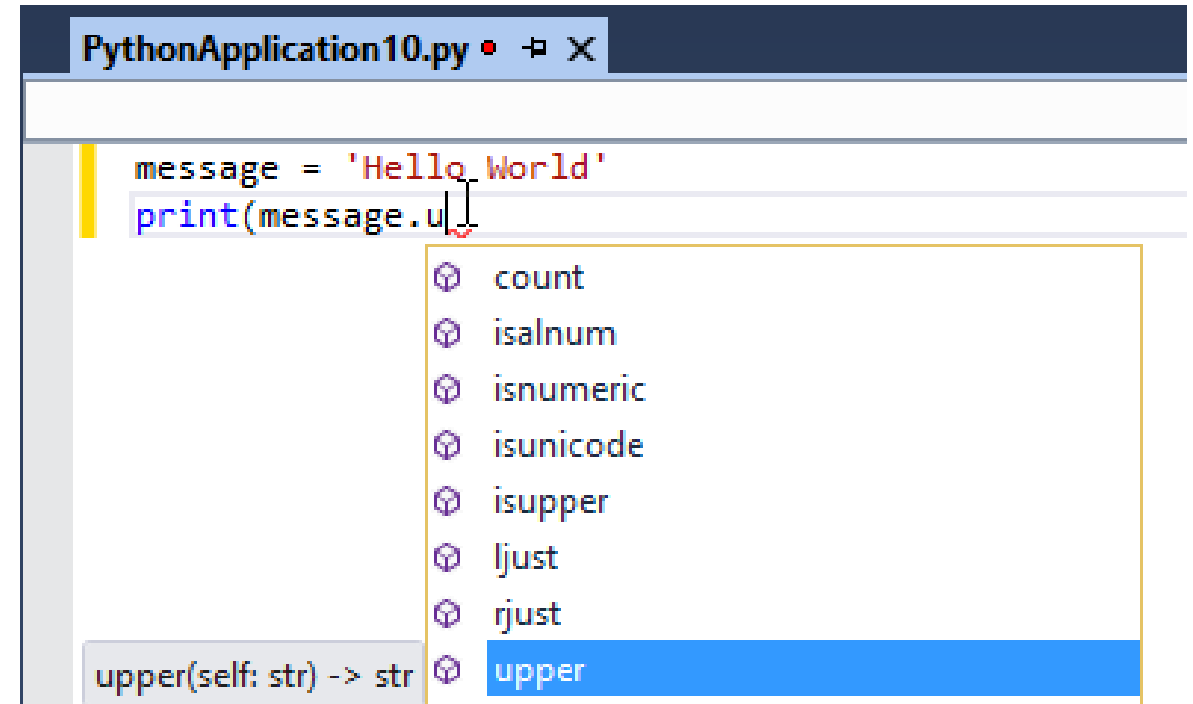
Visual Studio awesomeness

Did you notice the pop up list?

That's IntelliSense.

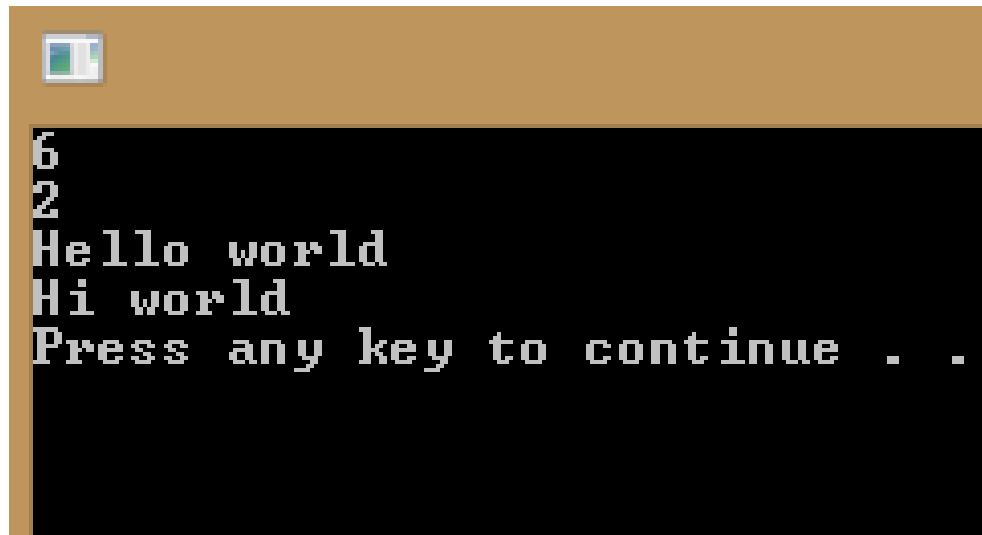
Visual Studio will suggest possible functions that you can call automatically after you type the '.'

You can also use CTRL+J or CTRL+SPACE to launch IntelliSense



What do you think these functions will do?

```
message = 'Hello world'  
print(message.find('world'))  
print(message.count('o'))  
print(message.capitalize())  
print(message.replace('Hello', 'Hi'))
```



```
6  
2  
Hello world  
Hi world  
Press any key to continue . .
```


Programmers do not memorize all these functions!!

So how do programmers find them when they need them?

- IntelliSense
- Documentation
- Internet searches

How could we...

Have a user enter their postal code and then display that postal code in upper case letters even if the user typed it in lowercase?

```
postalCode = input("Please enter your postal code: ")  
print(postalCode.upper())
```

DEMO

Converting to uppercase

Did you notice?

The intellisense didn't appear to help us select the **upper()** function.

That's because our program didn't know we were going to store a string value in the `postalCode` variable. The **upper()** function is only for strings.

A good habit when coding in any language is to initialize your variables. That means when you create them you give them an initial value.

```
postalCode = " "  
postalCode = input("Please enter your postal code: ")  
print(postalCode.upper())
```

How could we...

Ask someone for their name and then display the name
someone with the first letter of their first and last name
uppercase and the rest of their name lowercase?

```
name = ""  
name = input("Please enter your name: ")  
print(name.capitalize())
```

Functions and variables allow us to make new mistakes in our code...

Each line of code below has a mistake...

```
message = Hello world
23message = 'Hello world'
New message = 'Hi there'
print(message.upper)
print(message.lower())
print(message.count())
```

```
message = 'Hello world'
23message = 'Hello world'
New message = 'Hi there'
print(message.upper())
print(message.lower())
print(message.count('H'))
```

Your challenge

- Write a program that allows a person to personalize a story
- Take a page from a book or make up a story. Ask the user to enter information you can replace in the story such as their name, a place, or insert adjectives or adverbs into the story. Then display the personalized story to the user
- For extra credit make sure you correct anything they type in with the incorrect case (e.g. if they type an adjective in uppercase you may want to display it in lowercase)

Congratulations!



- You can now write a computer program that will interact with a user



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.