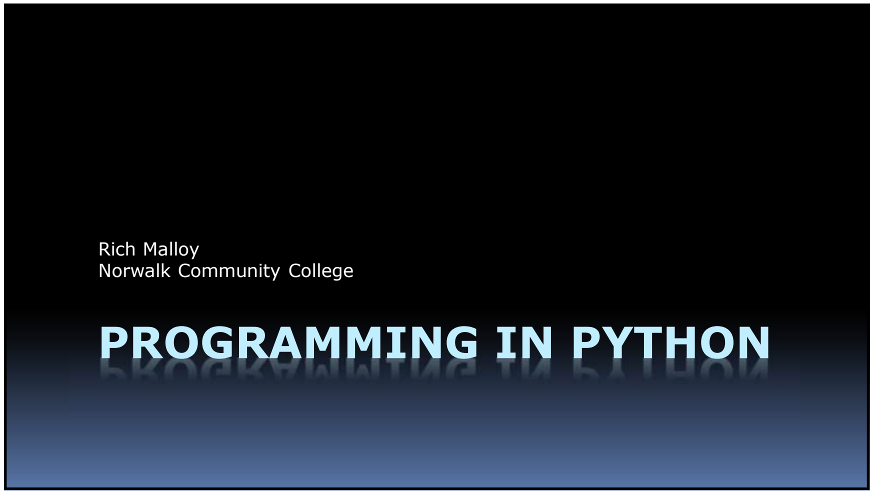


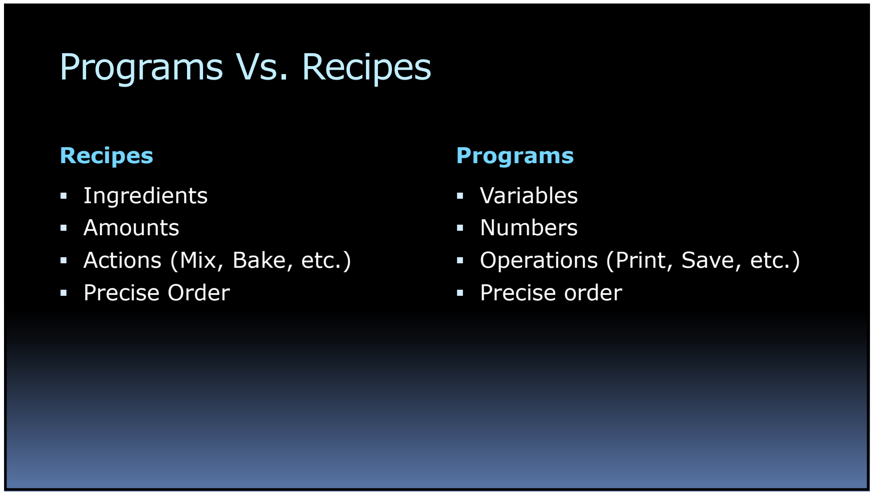
Rich Malloy
Norwalk Community College

PROGRAMMING IN PYTHON

1

COMPUTER PROGRAMS

2



Programs Vs. Recipes

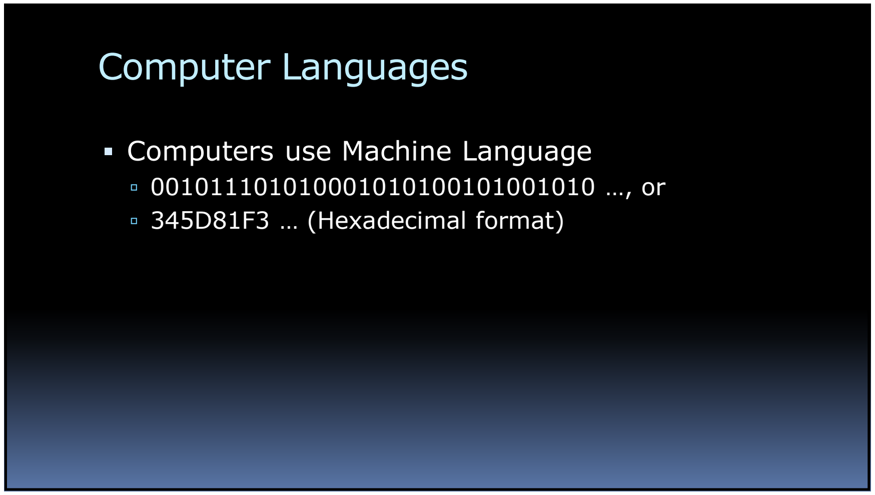
Recipes

- Ingredients
- Amounts
- Actions (Mix, Bake, etc.)
- Precise Order

Programs

- Variables
- Numbers
- Operations (Print, Save, etc.)
- Precise order

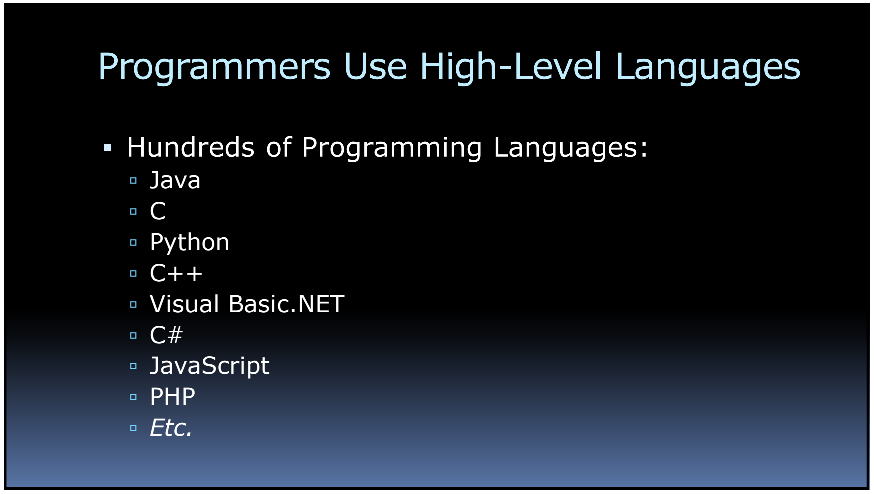
3



Computer Languages

- Computers use Machine Language
    - 0010111010100010101001010010 10 …, or
    - 345D81F3 … (Hexadecimal format)

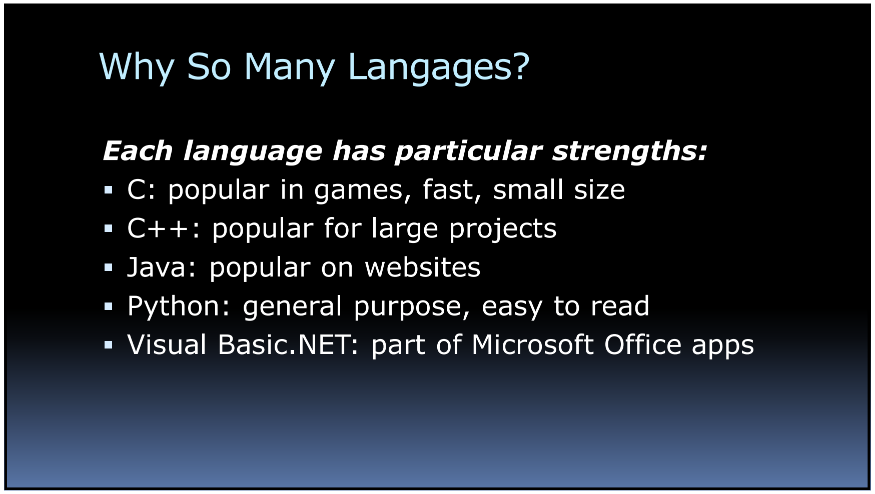
4



Programmers Use High-Level Languages

- Hundreds of Programming Languages:
  - Java
  - C
  - Python
  - C++
  - Visual Basic.NET
  - C#
  - JavaScript
  - PHP
  - *Etc.*

5

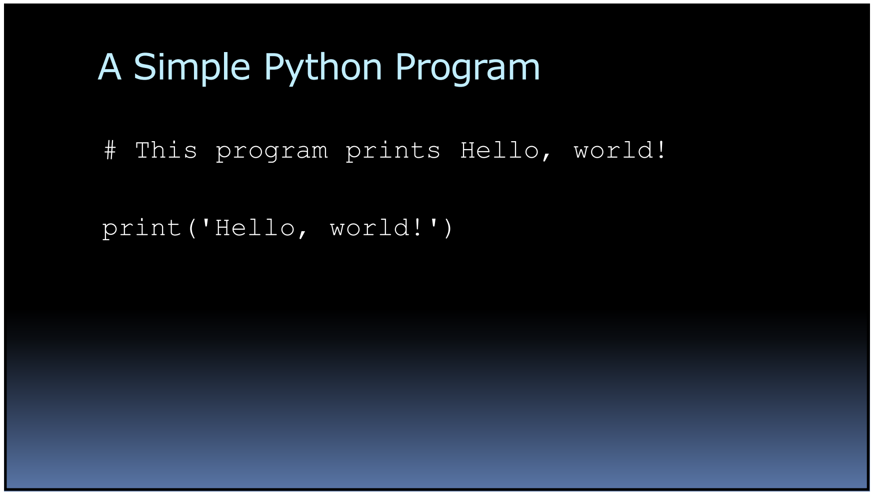


Why So Many Langages?

Each language has particular strengths:

- C: popular in games, fast, small size
- C++: popular for large projects
- Java: popular on websites
- Python: general purpose, easy to read
- Visual Basic.NET: part of Microsoft Office apps

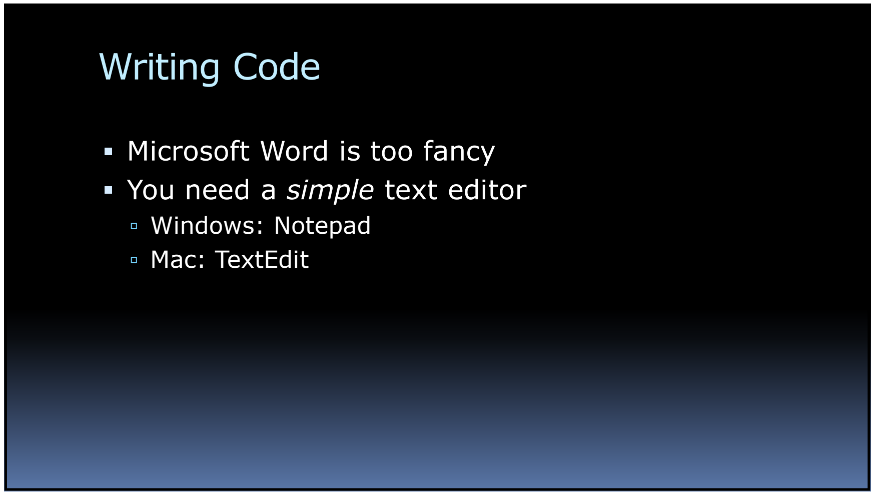
6



A Simple Python Program

```
# This program prints Hello, world!

print('Hello, world!')
```

7



Writing Code

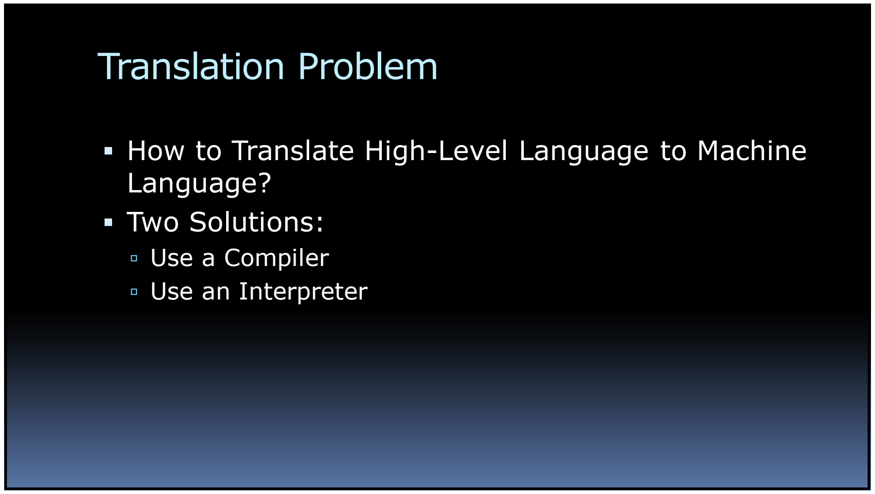
- Microsoft Word is too fancy
- You need a *simple* text editor
  - Windows: Notepad
  - Mac: TextEdit

8

Text Editors for Programmers

- Have special features for programmers
  - E.g., Syntax highlighting
    - Makes it easy to distinguish comments from code
- Popular Text Editors
  - TextWrangler
  - Notepad++

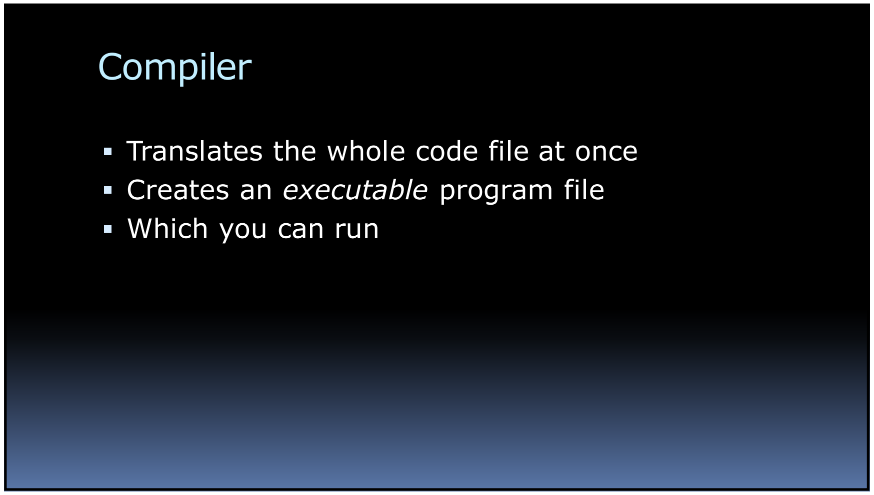
9



Translation Problem

- How to Translate High-Level Language to Machine Language?
- Two Solutions:
  - Use a Compiler
  - Use an Interpreter

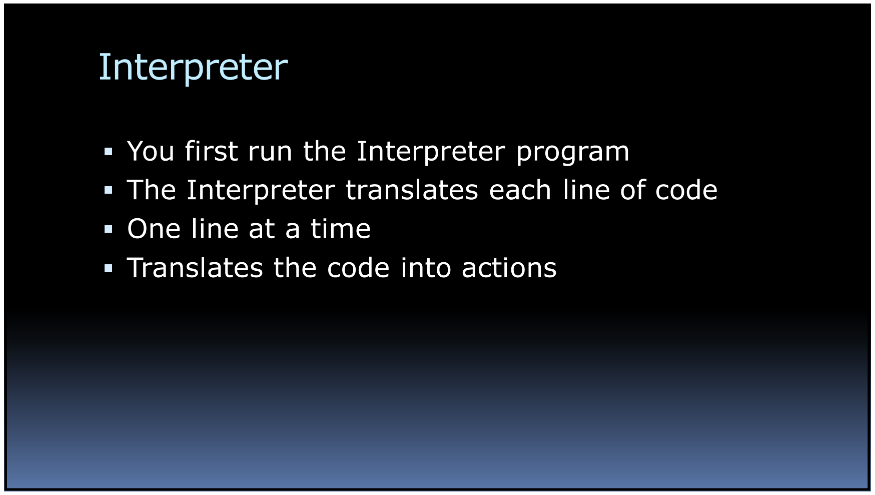
10



Compiler

- Translates the whole code file at once
- Creates an *executable* program file
- Which you can run

11



Interpreter

- You first run the Interpreter program
- The Interpreter translates each line of code
- One line at a time
- Translates the code into actions

12

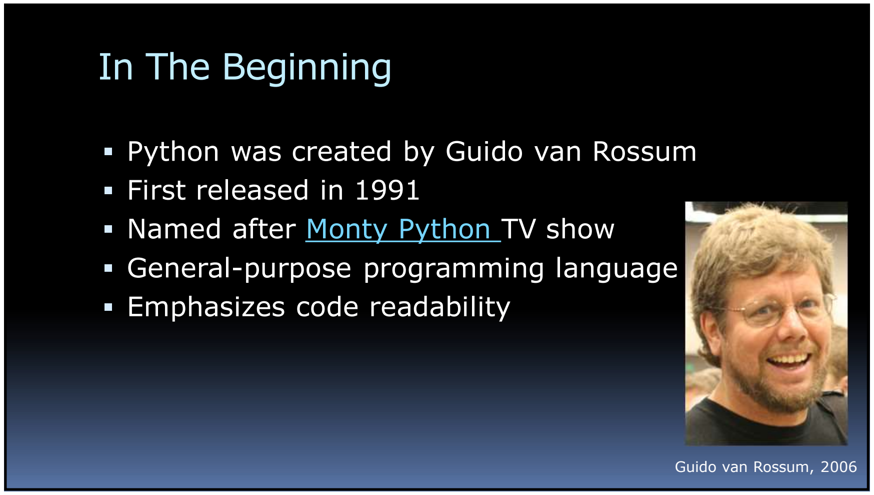
IDE

- Integrated Development Environment
- Combines text editor with compiler or interpreter
- Makes it easy to write, check, and run code
- Examples:
  - Visual Studio Code
  - IDLE (Python GUI)

13

PYTHON

14

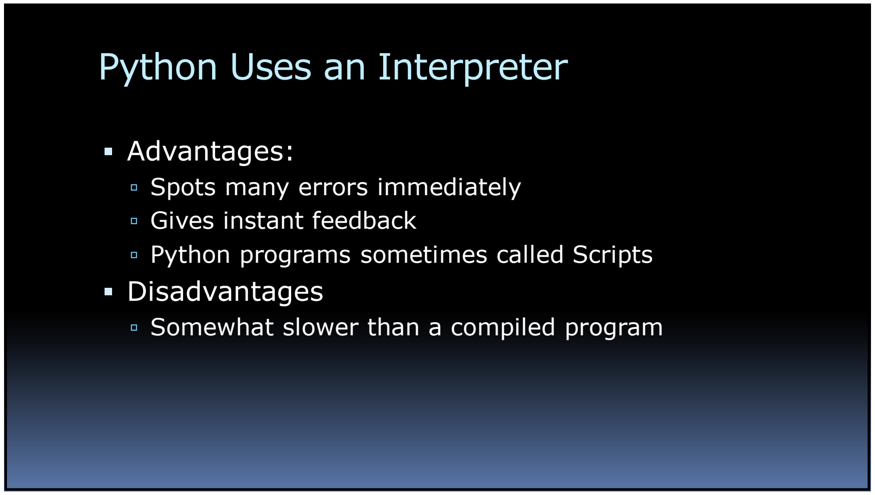


In The Beginning

- Python was created by Guido van Rossum
- First released in 1991
- Named after Monty Python TV show
- General-purpose programming language
- Emphasizes code readability

Guido van Rossum, 2006

15



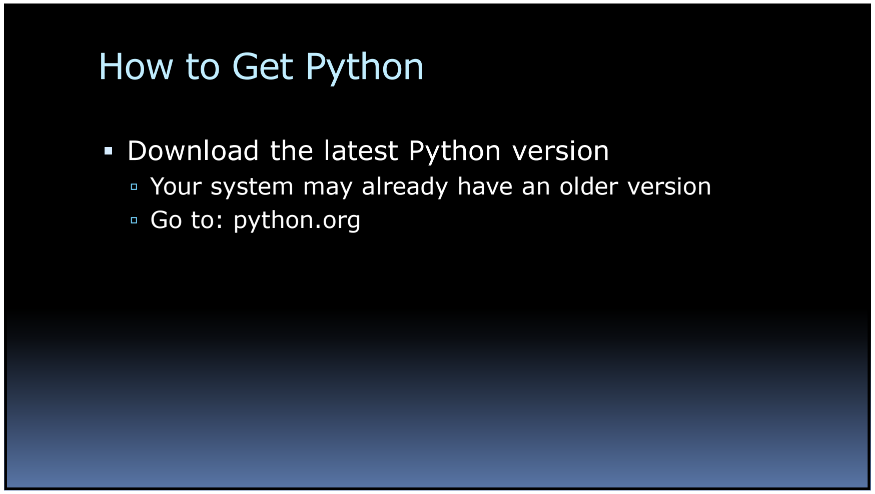
Python Uses an Interpreter

- Advantages:
  - Spots many errors immediately
  - Gives instant feedback
  - Python programs sometimes called Scripts
- Disadvantages
  - Somewhat slower than a compiled program

16

SETTING UP PYTHON

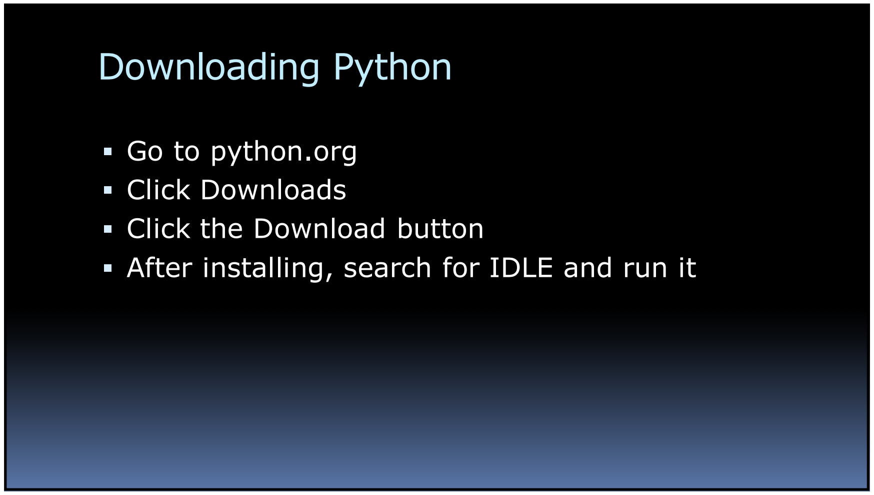
17



How to Get Python

- Download the latest Python version
  - Your system may already have an older version
  - Go to: python.org

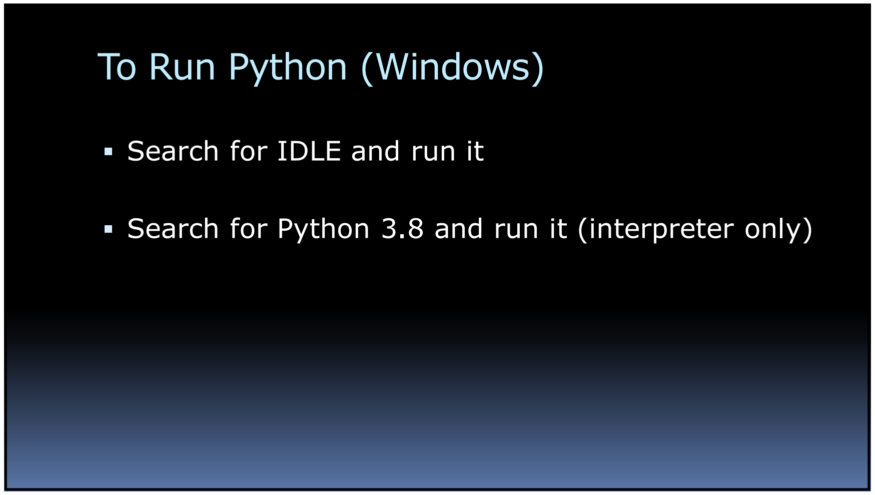
18



Downloading Python

- Go to python.org
- Click Downloads
- Click the Download button
- After installing, search for IDLE and run it

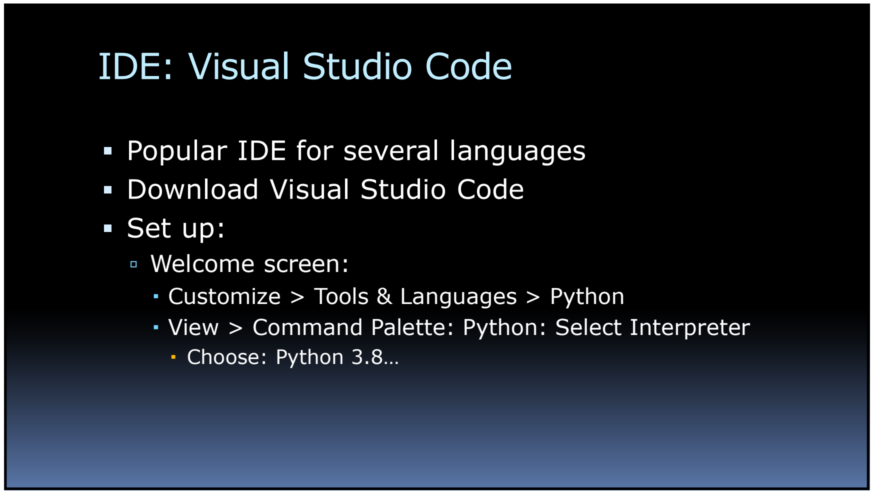
19



To Run Python (Windows)

- Search for IDLE and run it
- Search for Python 3.8 and run it (interpreter only)

20



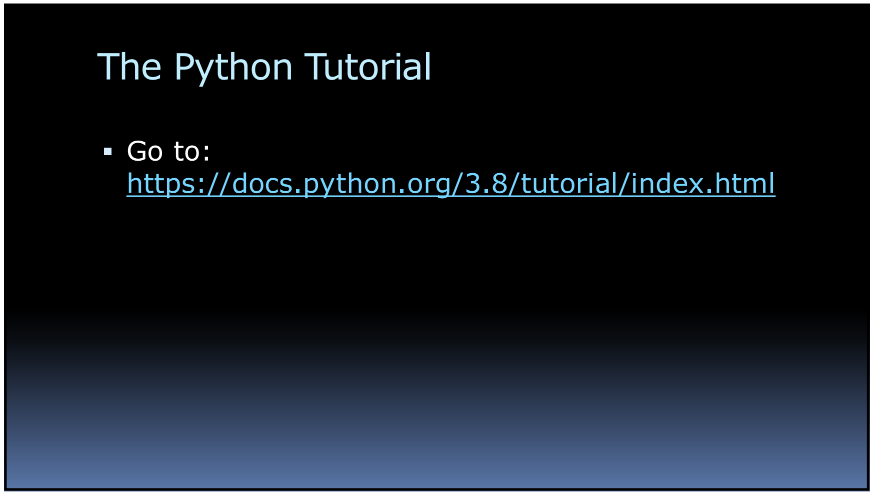
IDE: Visual Studio Code

- Popular IDE for several languages
- Download Visual Studio Code
- Set up:
  - Welcome screen:
    - Customize > Tools & Languages > Python
    - View > Command Palette: Python: Select Interpreter
      - Choose: Python 3.8…

21

PYTHON BASICS

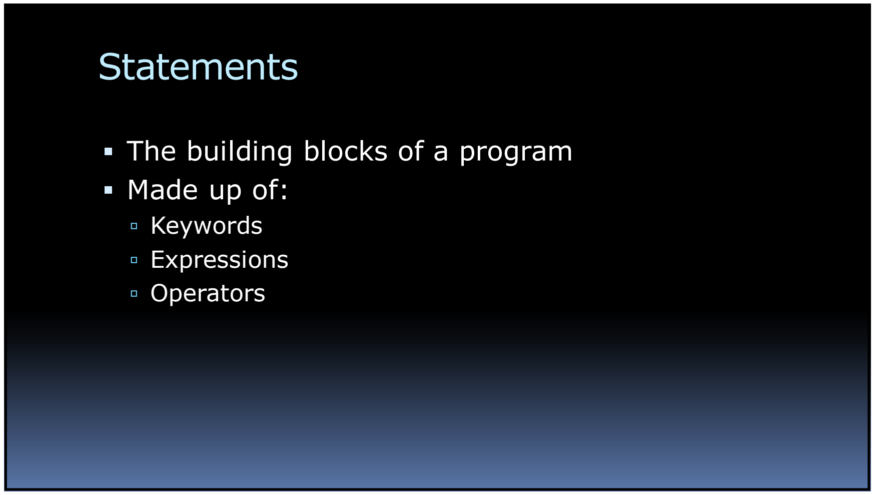
22



The Python Tutorial

- Go to: https://docs.python.org/3.8/tutorial/index.html

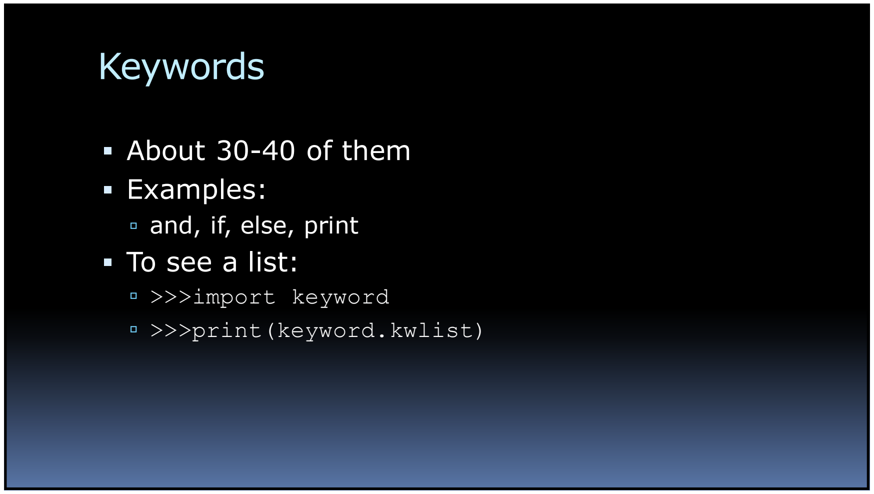
23



Statements

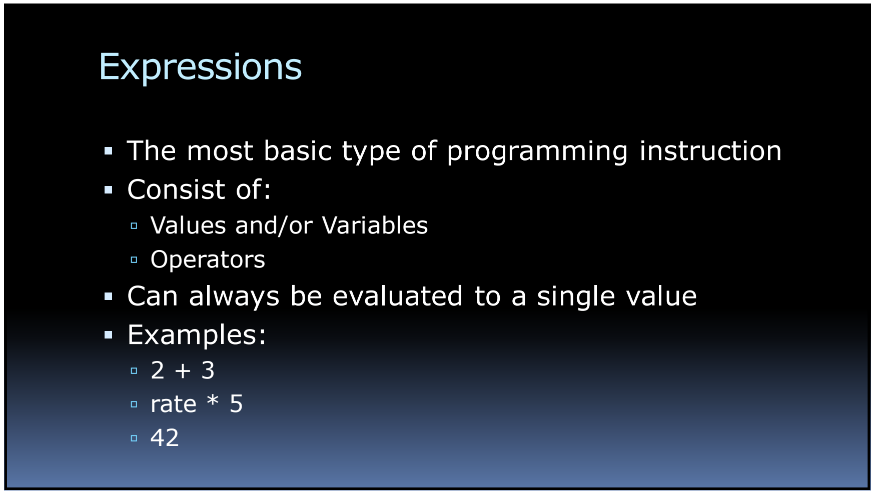
- The building blocks of a program
- Made up of:
  - Keywords
  - Expressions
  - Operators

24



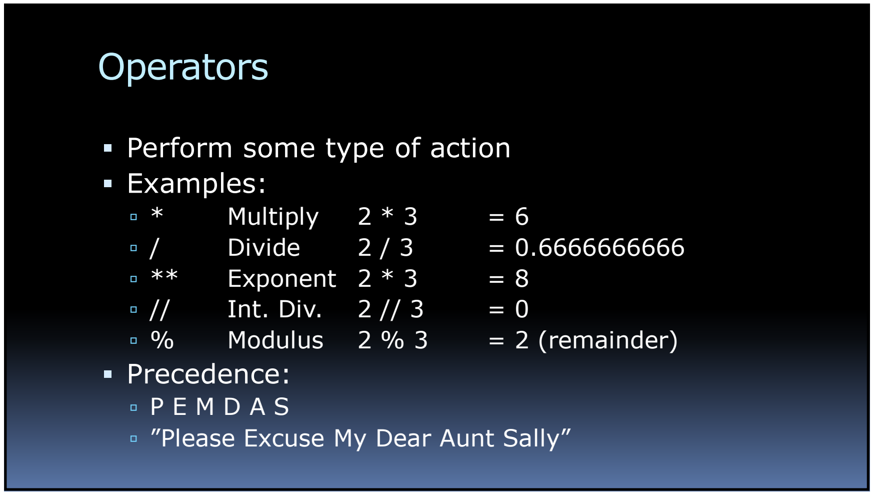
Keywords

- About 30-40 of them
- Examples:
  - and, if, else, print
- To see a list:
  - `>>>import keyword`
  - `>>>print(keyword.kwlist)`



Expressions

- The most basic type of programming instruction
- Consist of:
  - Values and/or Variables
  - Operators
- Can always be evaluated to a single value
- Examples:
  - 2 + 3
  - rate * 5
  - 42

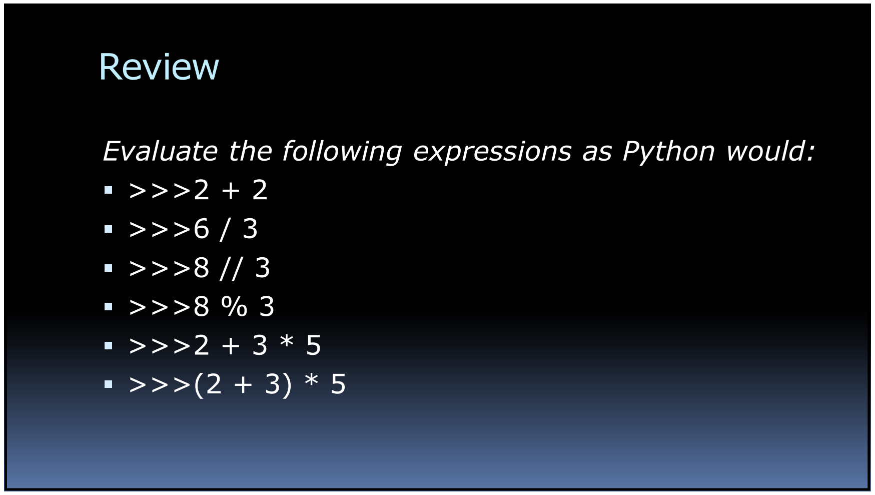


Operators

- Perform some type of action
- Examples:

*	Multiply	2 * 3	= 6
/	Divide	2 / 3	= 0.6666666666
**	Exponent	2 * 3	= 8
//	Int. Div.	2 // 3	= 0
%	Modulus	2 % 3	= 2 (remainder)

- Precedence:
  - PEMDAS
  - ”Please Excuse My Dear Aunt Sally”



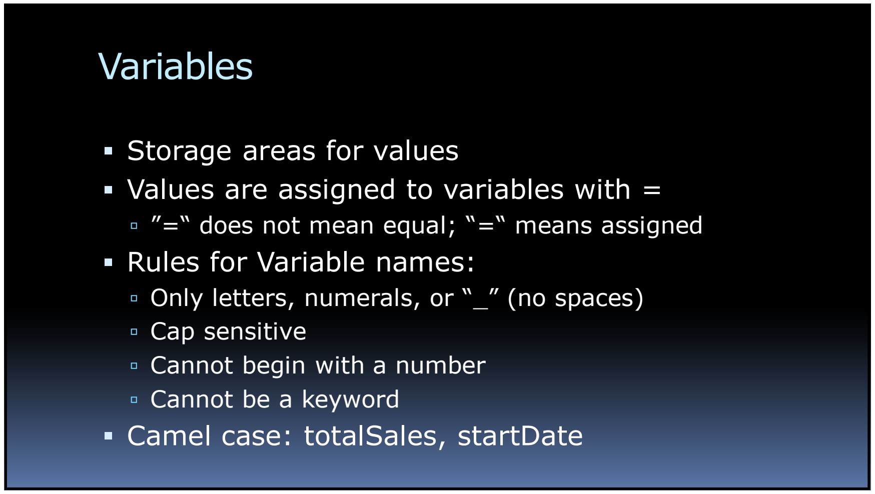
Review

Evaluate the following expressions as Python would:

- >>>2 + 2
- >>>6 / 3
- >>>8 // 3
- >>>8 % 3
- >>>2 + 3 * 5
- >>>(2 + 3) * 5

VARIABLES

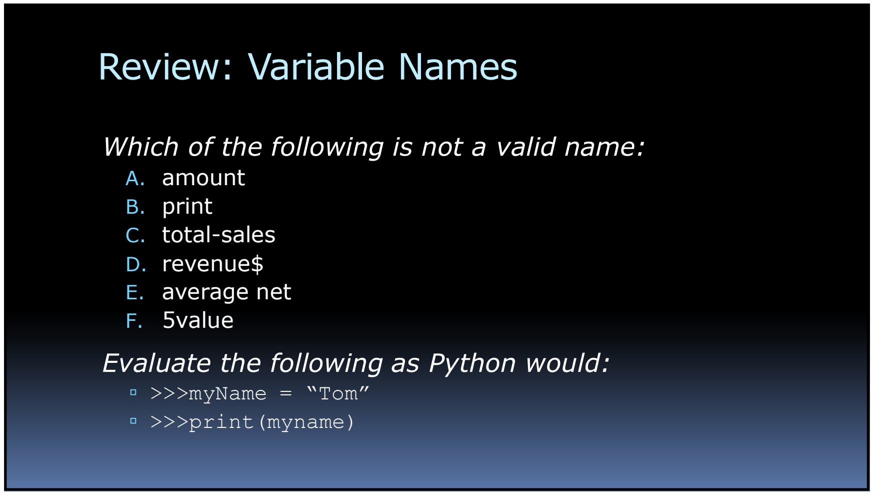
29



Variables

- Storage areas for values
- Values are assigned to variables with =
  - ”=“ does not mean equal; “=“ means assigned
- Rules for Variable names:
  - Only letters, numerals, or “_” (no spaces)
  - Cap sensitive
  - Cannot begin with a number
  - Cannot be a keyword
- Camel case: totalSales, startDate

30



Review: Variable Names

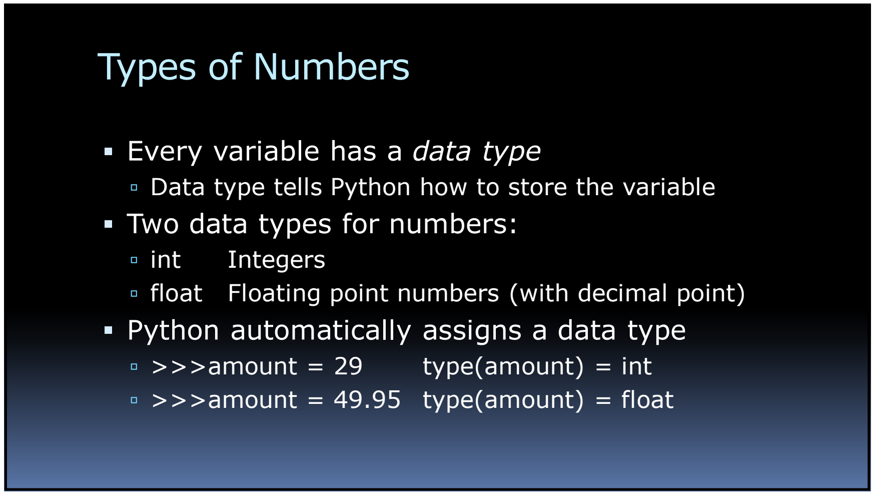
Which of the following is not a valid name:

A. amount
B. print
C. total-sales
D. revenue$
E. average net
F. 5value

Evaluate the following as Python would:

```
>>>myName = “Tom”
>>>print(myname)
```

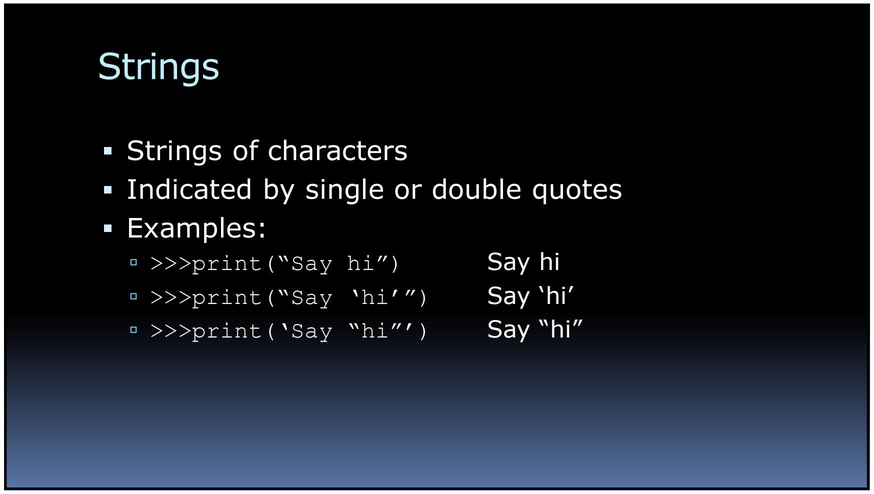
31



Types of Numbers

- Every variable has a *data type*
  - Data type tells Python how to store the variable
- Two data types for numbers:
  - int	Integers
  - float	Floating point numbers (with decimal point)
- Python automatically assigns a data type
  - >>>amount = 29	type(amount) = int
  - >>>amount = 49.95	type(amount) = float

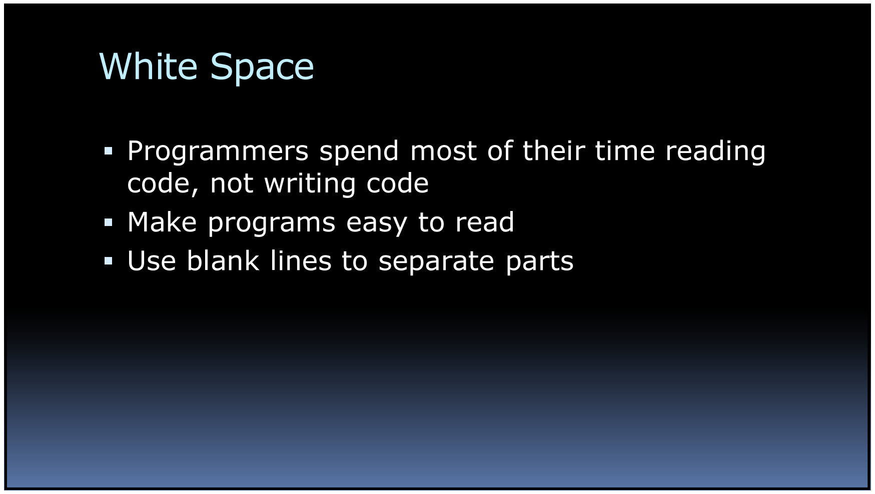
32



Strings

- Strings of characters
- Indicated by single or double quotes
- Examples:
  - `>>>print(“Say hi”)` Say hi
  - `>>>print(“Say ‘hi’”)` Say ‘hi’
  - `>>>print(‘Say “hi”’)` Say “hi”

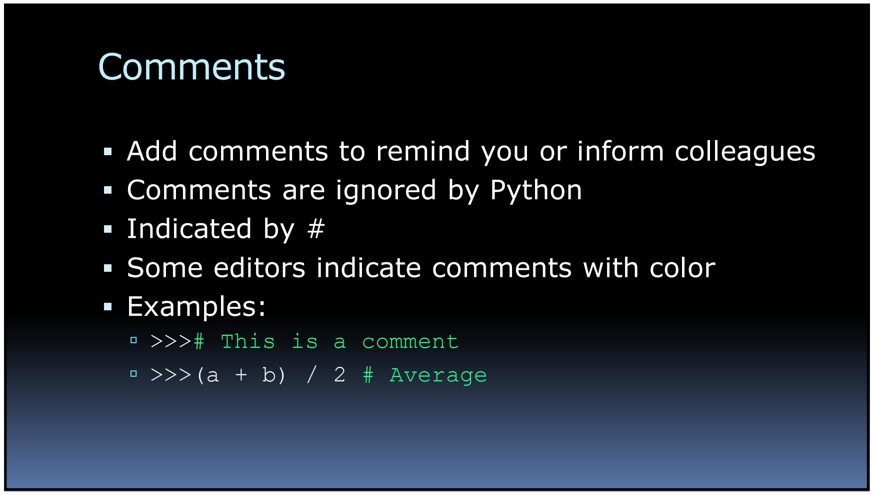
33



White Space

- Programmers spend most of their time reading code, not writing code
- Make programs easy to read
- Use blank lines to separate parts

34



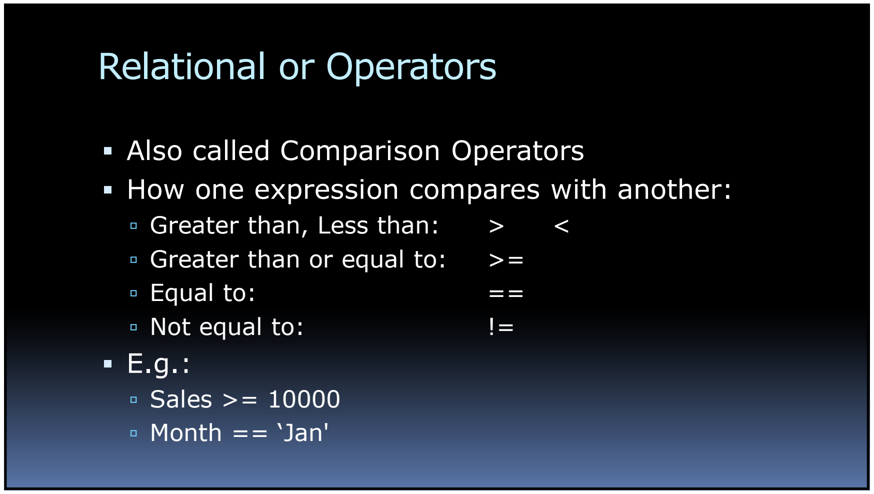
Comments

- Add comments to remind you or inform colleagues
- Comments are ignored by Python
- Indicated by #
- Some editors indicate comments with color
- Examples:
  - `>>># This is a comment`
  - `>>>(a + b) / 2 # Average`

35

CONDITIONAL CODE

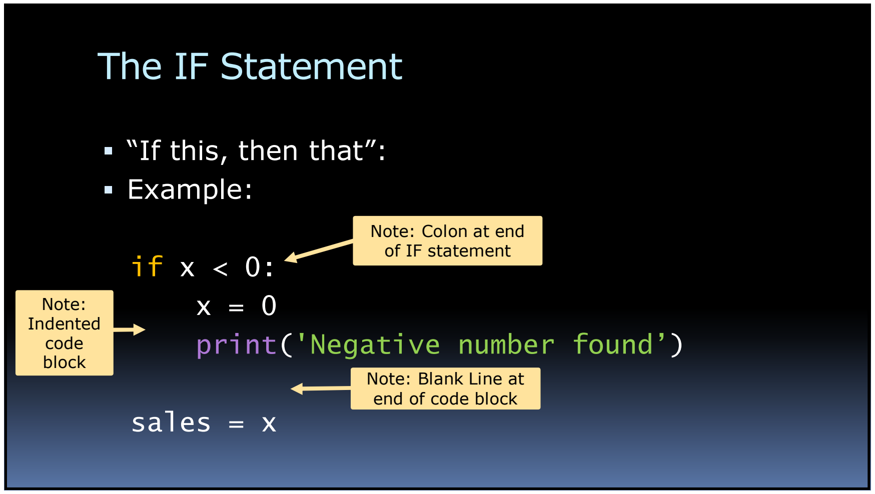
36



Relational or Operators

- Also called Comparison Operators
- How one expression compares with another:
  - Greater than, Less than: > <
  - Greater than or equal to: >=
  - Equal to: ==
  - Not equal to: !=
- E.g.:
  - Sales >= 10000
  - Month == ‵Jan'

37



The IF Statement

- “If this, then that”:
- Example:

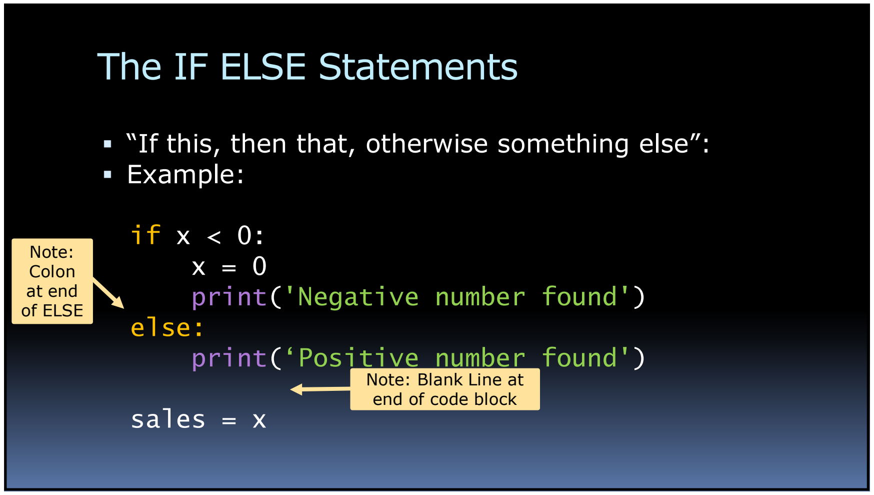
```
if x < 0:
    x = 0
    print('Negative number found’)

sales = x
```

Note: Colon at end of IF statement

Note: Indented code block

Note: Blank Line at end of code block

38



The IF ELSE Statements

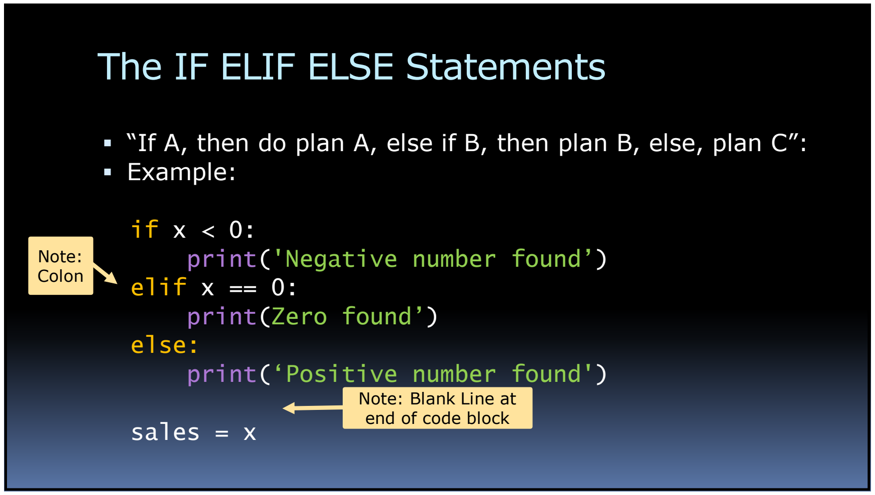
- “If this, then that, otherwise something else”:
- Example:

```
if x < 0:
    x = 0
    print('Negative number found')
else:
    print(‘Positive number found')

sales = x
```

Note: Colon at end of ELSE

Note: Blank Line at end of code block

39



The IF ELIF ELSE Statements

- “If A, then do plan A, else if B, then plan B, else, plan C”:
- Example:

```
if x < 0:
    print('Negative number found’)
elif x == 0:
    print(Zero found’)
else:
    print(‘Positive number found')

sales = x
```

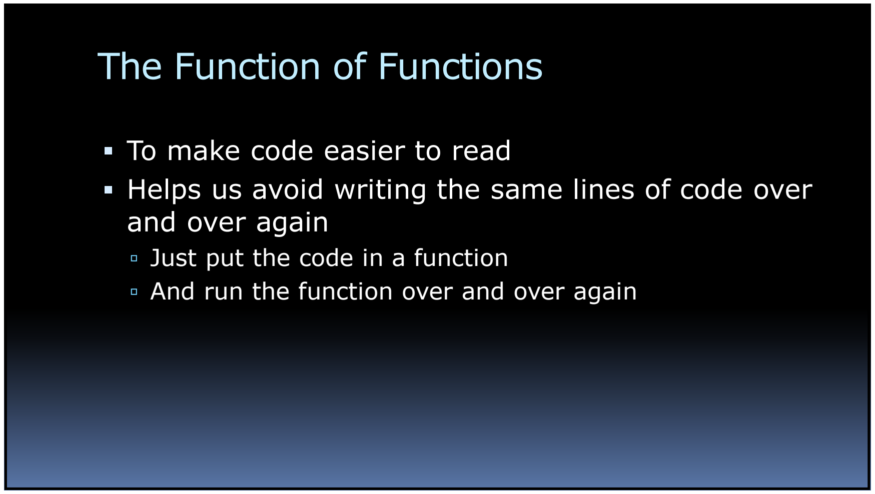
Note: Colon

Note: Blank Line at end of code block

40

FUNCTIONS

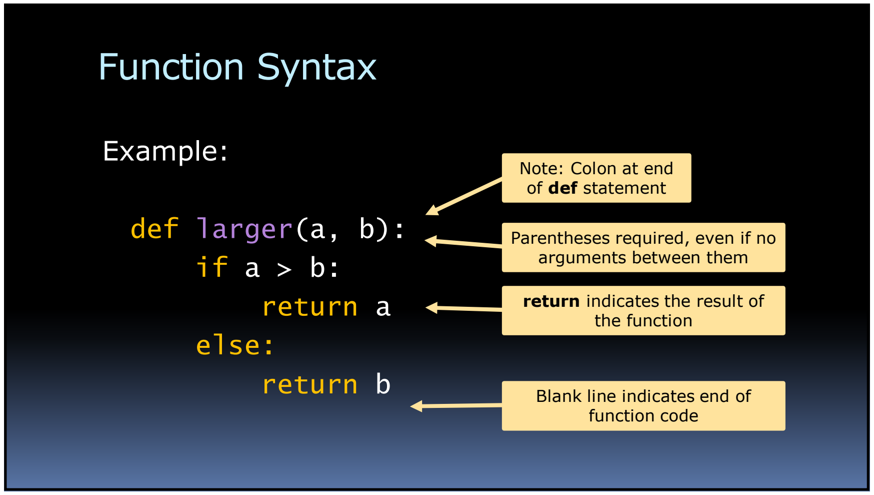
41



The Function of Functions

- To make code easier to read
- Helps us avoid writing the same lines of code over and over again
  - Just put the code in a function
  - And run the function over and over again

42



Function Syntax

Example:

```
def larger(a, b):
    if a > b:
        return a
    else:
        return b
```

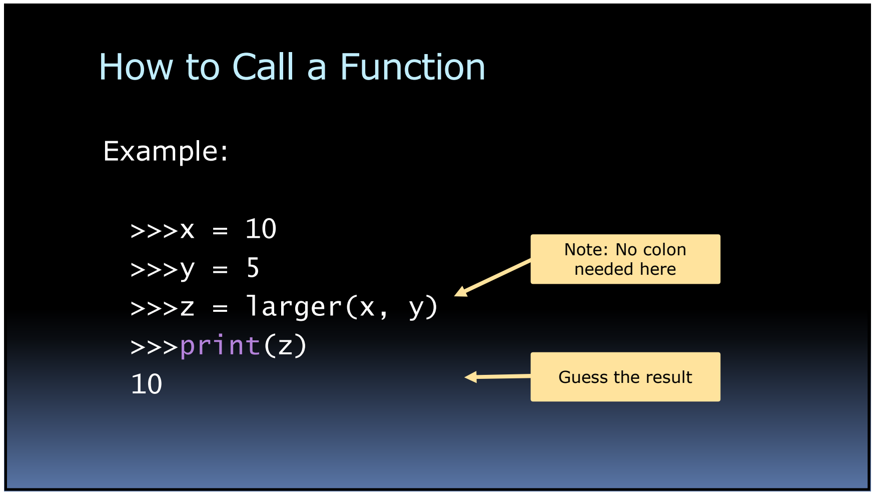
Note: Colon at end of **def** statement

Parentheses required, even if no arguments between them

return indicates the result of the function

Blank line indicates end of function code

43



How to Call a Function

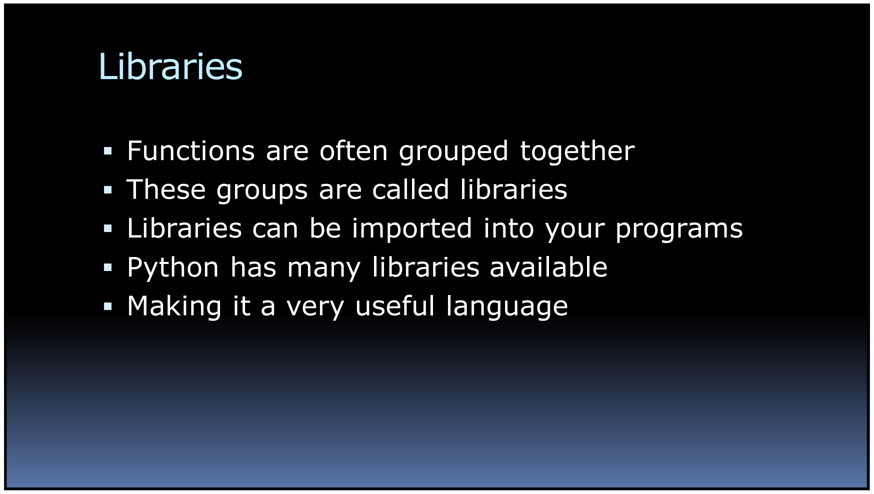
Example:

```
>>>x = 10
>>>y = 5
>>>z = larger(x, y)
>>>print(z)
10
```

Note: No colon needed here

Guess the result

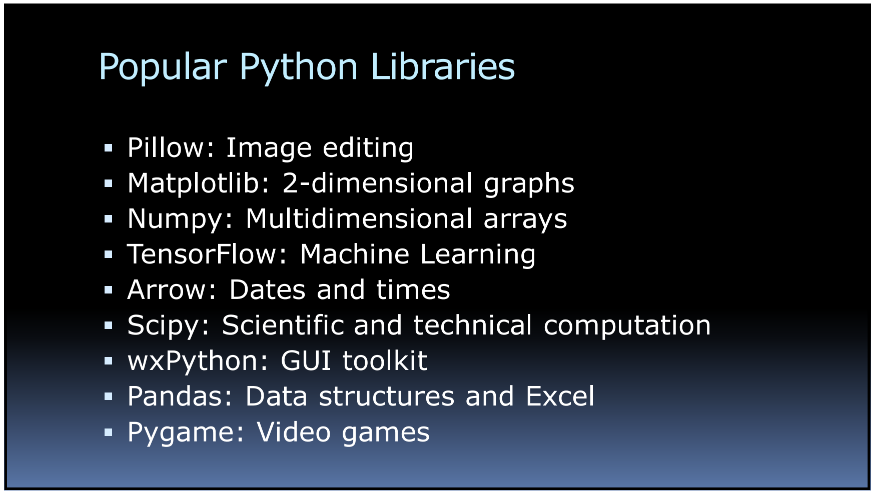
44



Libraries

- Functions are often grouped together
- These groups are called libraries
- Libraries can be imported into your programs
- Python has many libraries available
- Making it a very useful language

45



Popular Python Libraries

- Pillow: Image editing
- Matplotlib: 2-dimensional graphs
- Numpy: Multidimensional arrays
- TensorFlow: Machine Learning
- Arrow: Dates and times
- Scipy: Scientific and technical computation
- wxPython: GUI toolkit
- Pandas: Data structures and Excel
- Pygame: Video games

46

HAVE FUN!

47

48