



58th Annual CTEBVI 2017 Conference

LEARNING & TEACHING BASICS OF

PYTHON

Presented by
Radhika Grover and Komal Shirore
SPEL Technologies, Inc.



SPEL Technologies, Inc



LEVELING THE PLAYING FIELD

- ▶ Lots of commercial animation-based apps and games to teach programming to visual learners
- ▶ Lack of commercial software for learners with visual impairments and blindness
- ▶ Reasons?
- ▶ Create an educational platform to teach programming that is designed for accessibility.

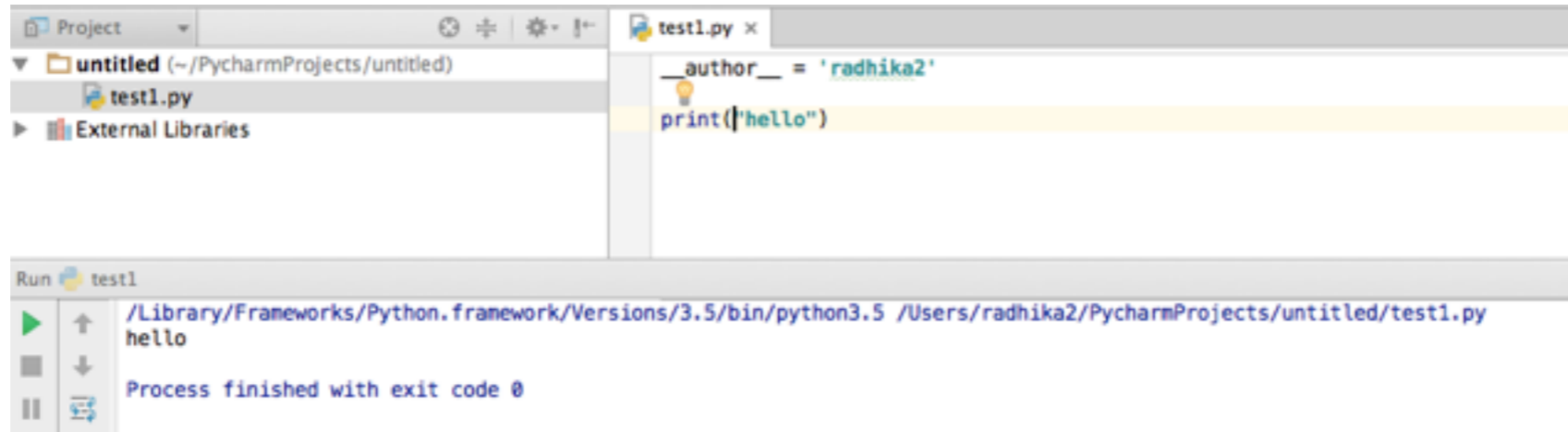


MOTIVATION

- ▶ US Department of Labor projects increase in the number of programming jobs over the next decade.
- ▶ Individuals with visual impairments and blindness can program when they are given the right tools
 - ▶ There are programmers with visual disabilities in software companies, but the number is very small
- ▶ Provide employment opportunities for sought after programming jobs
- ▶ Enable individuals to become independent



CHALLENGES



- ▶ Programming is a visual activity - confusing to switch with a screen reader between different quadrants (Navigation, Code, Error message, Program output)
- ▶ Pop-up windows in some tools are not navigable by screen readers
- ▶ Difficult to locate errors in the program
- ▶ Program may be waiting for some input from user - how does the user know?
- ▶ How to quickly locate the pertinent code segment in a large program?
- ▶ Semantic information, program navigation



OUR SOLUTION

- ▶ Accessibility panel to user preferences for fonts, colors, contrasts, etc.
- ▶ IDE with speech integration to make programming easier
- ▶ Intelligent built-in screen reader
- ▶ Accessible Learning Management system



Accessibility

Help

Contrast

100%

Saturation

100%

Magnification

- 50%
- 100%
- 200%
- 400%
- 1000%
- 1500%
- 2000%
- 2500%
- 3000%
- 3500%

- Invert
- Grayscale
- Text-to-Speech

Color Scheme

- Gray on White
- Blue on White
- Yellow on Black
- Green on Black

Reset

Save Settings



ACCESSIBILITY PANEL

- ▶ Set font size, contrast, color
- ▶ Set the speed and voice
- ▶ Save settings so they are automatically set when user logs in next



LOQUACIOUS - INTEGRATED DEVELOPMENT ENVIRONMENT (IDE) WITH SPEECH INTEGRATION

Exercise Section

Loquacious Python Editor

[Run](#) [Reset](#) [Say](#) [Overview](#) [Download](#)

Solution :

[View Source Code on Github](#)



LOQUACIOUS

- ▶ Cursor moves as each line in program is read out. Stop it on the desired line.
- ▶ Provides a high level overview of the program so user can quickly find the desired information
- ▶ Cursor automatically positioned on line with error and error message is read out.
- ▶ Read out program output
- ▶ Prompts the user to enter input



LEARNING MANAGEMENT SYSTEM (LMS)

- ▶ Designed for accessibility
- ▶ Integrated with the digital textbook

The screenshot displays the LMS interface for SPEL Technologies, Inc. The header includes the company logo, name, and tagline 'Smart Products for Everyday Living'. Navigation links for 'Home', 'Dashboard', 'Shannon Jones', and 'Logout' are present, along with 'Accessibility' and 'Search' icons. The main content area features three class cards:

- Class 103:** Mon 5 pm. Advanced Python Programming. This class covers patterns and numerical computation. Buttons: ASSIGNMENTS, GRADES, VIDEO CLASS. Links: BOOK, TUTORIALS, EDIT CLASS DETAILS.
- Class 102:** Thursday 1PM - 3PM. Intermediate Python. This course covers class creation, file I/O and data structures. Buttons: ASSIGNMENTS, GRADES, VIDEO CLASS. Links: BOOK, TUTORIALS, EDIT CLASS DETAILS.
- Class 101:** Tuesdays, Fridays from 11 am to 5 pm. This is an introductory class for learning Python programming. Buttons: ASSIGNMENTS, GRADES, VIDEO CLASS. Links: BOOK, TUTORIALS, EDIT CLASS DETAILS.



LMS FEATURES

- ▶ Post assignments that can be read out by built-in reader
- ▶ Student can write and test the solution and submit it in the same window - more convenient than using a separate LMS and programming IDE
- ▶ Teacher can grade and provide input in a single window
- ▶ Sortable menus for retrieving information quickly



INTELLIGENT SCREEN READER

- ▶ Understands the content on the website : provides only the information needed to navigate with ease
- ▶ Works with Loquacious (external screen readers will speak at the same time as Loquacious)
- ▶ Change speed/voices



AUDIO-BASED GAME

- ▶ **Merscythe: Adventures with the Codue**
- ▶ **Goal is to engage the learner**
- ▶ **Learner reads a story chapter, unlocks the next chapter by answering a programming question.**
- ▶ **Built-in tutorials for teacher to use.**



WHY LEARN PYTHON?

- ▶ Great first language for beginners
- ▶ Open source - lots of packages available
- ▶ Widely used in industry



TRAINING WEBINAR

We provide training for teachers

Teacher Training Web Conference



Three-hour training :
Two-hour webinar covering Python basics.
One-hour question and answer session.



Cost \$150.
Training fee is waived with purchase of subscription worth at least \$150.



Contact us to attend the one-hour conference:
December 2nd: 2 PM - 3 PM PST
More dates coming soon!

Please Note: This training is available to teachers only.



APPLICATIONS

- ▶ Create Websites
- ▶ System Administration
- ▶ Data Analytics
- ▶ Game Development



PYTHON IS USED BY

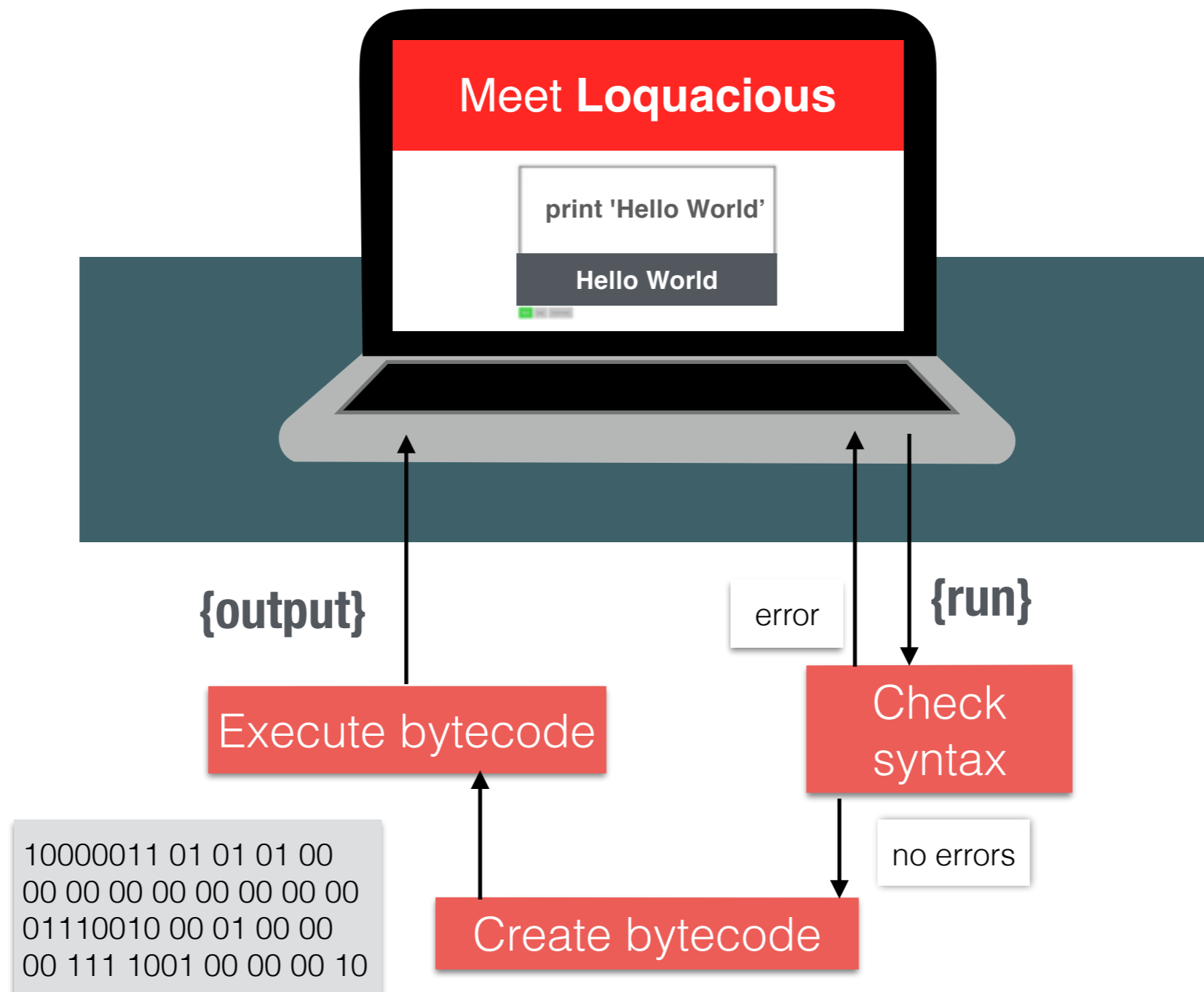
Quora

Google



YAHOO!

HOW PYTHON WORKS





TUTORIAL 1

- ▶ Objects
- ▶ Classes
- ▶ Methods
- ▶ Variables
- ▶ Syntax



OBJECTS

- ▶ Objects are Python's abstraction for data
- ▶ Doors, cars, lamps - these are all objects
- ▶ Objects have attributes (fields) and behaviors (methods)



CLASSES

- ▶ A class is used to create an object and control it
- ▶ Creates new object
- ▶ Example: class ScienceFacts

#create two objects of class

Door: door

door = Door()



OBJECTS & CLASSES

- ▶ Like real life Python objects have properties.
- ▶ Door can be opened and closed.
- ▶ How can these objects be controlled?



METHODS

- ▶ Each object has specific behaviors that can be controlled by methods
- ▶ Example: class `Door` has these methods: `open()`, `close()`
- ▶ `#create two objects of class Door: door1 and door2`

```
door1 = Door()
```

```
door2 = Door()
```

```
#open door1  
door1.open()
```

```
#close door2  
door2.close()
```



VARIABLES

- ▶ Variable stores data in the program
- ▶ `counter = 100` # An integer assignment
- ▶ `name = "John"` # A string
- ▶ Python also supports multiple assignments
 - ▶ `a=b=c=10`
 - ▶ `a,b,c=1,20,"John"`



SYNTAX

- ▶ Strings in Python
single ('), double (") and triple (''' or ''')
quotes are used to denote strings.
'a few words'
"This is a sentence."
"""This is another sentence."""
- ▶ Comments in Python -
A hash sign (#) that is not inside a string
literal begins a comment.
First comment



SYNTAX ERRORS

- ▶ Syntax Error
 - ▶ Spelling Error
 - ▶ Missing Parenthesis “()”
 - ▶ Missing Quotes or Colons “:”
 - ▶ Missing Keyword
- ▶ Indentation

SYNTAX ERRORS

EXAMPLES

Loquacious Python Editor

```
import Door from Door  
door = Door()
```

ParseError: bad input on line 1

Missing keyword

&

Spelling Error

```
from ScienceFacts ffrom ScienceFacts  
science = ScienceFacts()  
science.display_speed_of_lightd()
```

ParseError: bad input on line 1
This line has error 1

SYNTAX ERRORS

EXAMPLES

```
from ScienceFacts import ScienceFacts  
science =ScienceFacts()
```

Missing parenthesis



TUTORIAL 1 PRACTICE EXERCISES



THANK YOU
