POORNIMA
COLLEGE OF ENGINEERING

Approved by AICTE
Affiliated to Rajasthan Technical University, Kota
Recognized by UGC under Section 2(f) of the UGC Act, 1956

SPOKEN TUTORIAL BROCHURES AND MODULES

ISI-6, RIICO Institutional Area, Sitapura, Jaipur-302022 (Rajasthan)
• Phone: +91-9829255102, +91-9414728922 • E-mail: principal.pce@poornima.org
• Website: www.pce.poornima.org

The Spoken Tutorial Project

- Self-explanatory: uses simple language
- Audio-video: uses multisensory approach
- Small duration: has better retention
- Learner-centered: learn at your own pace
- Learning by doing: learn and practise simultaneously
- Empowerment: learn a new **FLOSS** (**F**ree/**L**ibre and **O**pen **S**ource **S**oftware)

Target Group

- Students- High School and College
- Working professional- Software users, developers and trainers
- Research scholars
- Community at large

Workshops

The Spoken Tutorial Project Team conducts workshops on C and C++ and other FLOSS using spoken tutorials and gives certificates to those who pass an online test.
For more details, please visit
https://spoken-tutorial.org

Forum

We have developed a beginner friendly Forum to answer specific questions pertaining to any part of a particular tutorial.
For more details, please visit
https://forums.spoken-tutorial.org.

The Spoken Tutorial Project
is funded by the
National Mission on Education through
Information and Communication Technology,
Ministry of Human Resource Development,
Government of India.

Contact us

Email: contact@spoken-tutorial.org
Website: https://spoken-tutorial.org

Forum help available to all learners

Content available in 22 Indian languages

IIT Bombay

-

Spoken Tutorial
https://spoken-tutorial.org

Scan the QR code to
visit Spoken Tutorial website

THE C PROGRAMMING LANGUAGE

National Mission on Education through
Information and Communication Technology
(NMEICT)
www.sakshat.ac.in

Funded by MHRD, Government of India.

About C

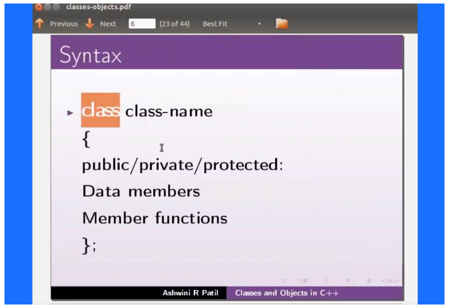
C is a general-purpose programming language, initially developed by Dennis Ritchie between 1969 and 1973 at Bell Labs. Its design provides constructs that map efficiently to typical machine instructions. C is one of the most widely used programming language and there are very few computer architectures for which a C compiler does not exist.

Features

- C has facilities for structured programming and allows lexical variable scope and recursion.
- All executable code is contained within subroutines, called "functions."
- C program source text is free-format, using the semicolon as a statement terminator and curly braces for grouping blocks of statements.
- Typing is static, but weakly enforced: all data has a type, but implicit conversions can be performed; for instance, characters can be used as integers.
- Complex functionality such as I/O, string manipulation, and mathematical functions are easy to implement with library routines.

About C++

- C++ is a statically typed, free-form, compiled, general-purpose programming language. It was developed by Bjarne Stroustrup starting in 1979, at Bell Labs.
- It adds object-oriented features such as classes, and other enhancements to the C programming language.
- The language began as enhancements to C, first adding classes, then virtual functions, operator overloading, multiple inheritances, templates, and exception handling among other features.
- C++ is also one of the most popular programming languages and can be implemented on most hardware and OS platforms.
- As an efficient compiler to native code, its application domains include:
- Systems software
- Application software
- Device drivers
- Embedded software
- High-performance server and client applications
- Entertainment software like video games



Syntax

```
class class-name
{
public/private/protected:
Data members
Member functions
};
```

Ashwini R Patil

Classes and Objects in C++

Features

- Classes: By using classes, we can create user-defined data types. A class is the collection of a set of data and code. An object is the instance of a class.
- Inheritance: Allows one data type to acquire properties of other data types. This provides the idea of reusability, that means we can add new features to an existing class without modifying it.
- Data Abstraction and Encapsulation: Encapsulation means hiding data from the data structures. Here, the data is accessible to only the functions that are allowed to access it. Abstraction means representing essential features without including background details.
- Polymorphism: means one interface can be used for multiple implementations, so that object can behave differently for each implementation.
- Dynamic Binding: At runtime, the code matching the object under the current reference will be called.

C and C++ Advantages

- Powerful and flexible: C/C++ are used for developing operating systems, compilers, parsers, interpreters, word processors, search engines and graphic programs.
- Support: C requires less runtime support
- Portable programming language: A variety of C/C++programm written for one computer system can be compiled and run on another system, with little or no change.
- Modular: Written in routines called functions and classes (C++), programs can be used in other applications or programs.
- Preferred by professional programmers: A variety of C/C++ resources and helpful supports are widely available.
- Standardised: Many standards have been documented, maintained and updated for C and C++ as standard references.

Online tutorial for C and CPP

Contents

Basic Level

1) **First C Program**

- -Header Files
  - example: #include <stdio.h>
- main()
- Curly braces { }
- printf()
- semicolon ;
- Compiling a C program
  - example: gcc filename.c -o output parameter
- Executing a C program
  - example: ./output parameter
- Errors

2) **First C++ Program**

- Header files
  - --example: #include <iostream>
- main()
- Curly braces { }
- -cout<<
- semicolon ;
- Compiling a C++ program
  - example: g++ filename.cpp -o output parameter
- Executing a C program
  - example: ./output parameter

3) **Tokens in C and C++**

- Data types, constants, identifiers
- Keywords
  - example: if, break, else
- Constants
- Data types
  - example: int, float, char, double
- Format specifiers
  - example: %d, %f, %c, %lf
- Range of data types
- Variables
- Identifier
- Errors

Dr. Mahesh Bundele
B.E., M.E., Ph.D.
Director
Poornima College of Engineering
ISI-6, RIICO Institutional Area
Sitapura, JAIPUR

4) **Functions in C and C++**

- What is a function
- Syntax for declaration of a function
- Function with arguments
    - example: return-type function-name(parameter);
- Function without arguments
    - example: return-type function-name;
- Calling a function
- Errors

5) **Scope of Variables in C and C++**

- Introduction
- Syntax of declaring a variable
    - example: data-type var-name;
- Syntax for initializing a variable
    - example: data-type var-name = value;
- Scope of variables
- Global variable
- Local variable
- Error

6) **If And Else If statement in C and C++**

- What are Statements.
- Syntax for if and
- If-else Statement
- Errors

7) **Nested if and switch statement in C and C++**

- Nested if statement.
- Switch statement.
- Syntax for nested-if statement
- Syntax for switch statement
- break statement
- Comparison between nested if-else and switch statement
- Errors

8) **Increment and Decrement Operators in C and C++**

- Increment Operator
    - example: ++
- Postfix increment
    - example: a++
- Prefix increment
    - example: ++a
- Decrement Operator
    - example: --

Dr. Mahesh Bundele
B.E., M.E., Ph.D.
Director
Poornima College of Engineering
ISI-6, RIICO Institutional Area
Sitapura, JAIPUR

- Postfix decrement
    - example: a--
- Prefix decrement
    - example: --a
- Typecasting
- Errors

9) **Arithmetic Operators in C and C++**

- Arithmetic Operators
- Addition Operator
    - example: a + b
- Subtraction Operator
    - example: a - b
- Multiplication Operator
    - example: a * b
- Division Operator
    - example: a \ b
- Modulus Operator
    - example: a % b
- Errors

10) **Relational Operators in C and C++**

- Double Equal to
    - example: a == b
- Not Equal to
    - example: a != b
- Greater Than
    - example: a > b
- Less Than
    - example: a < b
- Greater than Equal To
    - example: a >= b
- Less Than Equal To
    - example: a <= b
- Errors

11) **Logical Operators in C and C++**

- And &&
- Or ||
- Not !
- Errors

Intermediate level

---

12) **Loops in C and C++**

- Loops
- Syntax for while and do-while loop

Dr. Mahesh Bundele
B.E., M.E., Ph.D.
Director
Poornima College of Engineering
ISI-6, RIICO Institutional Area
Sitapura, JAIPUR

- Comparison of while and do-while loop
- Syntax for
- for loop
- Errors

13) **Arrays in C and C++**

- What are arrays
- 1-D Arrays
- Syntax for Declaration of arrays
  - example: data type array_name [size];
- Syntax for Initialization of arrays
  - example: data type array_name [size]=value;
- Accepting values from the user
- Errors

14) **Working with 2-D Arrays in C and C++**

- What are 2-D Arrays.
- Range of arrays
- Syntax for Declaration of 2-D arrays
  - example: data type array_name[row][column];
- Syntax for initialization of 2-D arrays
  - example: data type array_name[row][column]=

{
{row-val},{col-val}
};

- Errors

15) **Strings in C and C++**

- What is a string
- Syntax for declaring a string
- Syntax for initializing a string
- To read a string from keyboard
- Errors

16) **String Library Functions in C and C++**

- What are string library functions.
- Types of string library functions
  - Strcpy
  - Strlen
  - Strcmp
  - Strcat
- Errors

Advanced level

17) **Working with Structures in C and C++**

- Introduction
- Syntax of structures

Dr. Mahesh Bundele
B.E., M.E., Ph.D.
Director
Poornima College of Engineering
ISI-6, RIICO Institutional Area
Sitapura, JAIPUR

- Declaration and initialization
- Declaration of structure variable
- Accessing structure variables

18) **Understanding Pointers in C and C++**
- Introduction
- Syntax of Pointer
    - example: int *iptr;
- Declaration
    - example:

int a; (integer a) int *aptr; (pointer to an integer *aptr) aptr = &a; (aptr set to address of a)
Address Pointer

- Errors

19) **Function call in C and C++**
- types of function calls
- function pass by value
- function pass by reference

20) **File Handling in C**
- File handling functions
- Opening a File closing a file
    - example: fopen, fclose
- Reading data from a File

Dr. Mahesh Bundele
B.E., M.E., Ph.D.
Director
Poornima College of Engineering
ISI-6, RIICO Institutional Area
Sitapura, JAIPUR

The Spoken Tutorial project

*Self explanatory - uses simple language

*Audio-video - uses multisensory approach

*Small duration - has better retention

*Learner-centered - learn at your own pace

*Learning by doing - learn and practice simultaneously

*Empowerment - learn a new FOSS

Target Group

*Students - High School and College

*Working professional - Software users, developers and trainers

*Research scholars

*Community at large

Workshops

The Spoken Tutorial Project Team conducts workshops on PHP & MySQL and other FOSS using spoken tutorials and gives certificates to those who pass an online test.

For more details,please write to
contact@spoken-tutorial.org

The Spoken Tutorial Project is funded by the National Mission on Education through Information and Communication Technology, Ministry of Human Resource Development, Government of India.

Contact us

Email: contact@spoken-tutorial.org
Website: http://spoken-tutorial.org

IIT Bombay

Spoken Tutorial

php
MySQL®

National Mission on Education through
Information and Communication Technology
(NMEICT)
www.sakshat.ac.in

Funded by MHRD, Goverment of India

http://spoken-tutorial.org

PHP

Introduction

PHP or "PHP : Hypertext Preprocessor" is a widely-used Open Source general-purpose scripting language that is especially suited for Web development and can be embedded into HTML. Its syntax draws upon C, Java and PERL, and is easy to learn.

The main goal of the language is to allow web developers to write dynamically generated web pages quickly, but you can do much more with PHP.

Uses of PHP •

- To create large websites
- For E-commerce like osCommerce, OpenCart
- To create online discussion forums like phpBB
- To create content management systems like Drupal, Joomla
- To create e-learning management systems like Moodle
- To develop web-based management tools like phpMyAdmin

And many more..

MySQL

Introduction

MySQL is a relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases. The SQL phrase stands for Structured Query Language. Applications which use MySQL data bases include: Joomla, Word Press, MyBB, phpBB, Drupal and other software built on the LAMP software stack.

A third party open source software "phpMyAdmin" is used as a web-based front end for managing MySQL databases easily and effeciently. It is widely installed by Web hosts worldwide. Also it is included in the convenient LAMP, MAMP and WAMP software bundle installers.
MySQL is used in many high-profile, largescale World Wide Web products, including Wiki-pedia, Google and facebook.

Features of PHP & MySQL

- Scalability and flexibility
- High speed and high performance
- Data protection
- Comprehensive Application Development
- Management tools

And many more...

PHP & MySQL

Benefits

- A large chunk of facebook, the world's leading social networking site, has a huge code based in PHP and it uses MySQL as database to store information of 1 billion+users!

- PHP is the most preferred language for web development by free-lance developers across the globe.

- Many free and open source CMS like Drupal, Moodie, etc. are created using PHP & MySQL.

- PHP & MySQL has a large user and develope community.

Links:

Original videos are available at
http://phpacademy.org

PHP Official Website -http://www.php.net

MYSQL Official Website -
http://www.mysgl.com

W3Schools - PHP and MySQL Tutorials -
http://www.w3schools.com/php/default.asp
http://www.w3schools.com/sql/default.asp =

These tutorials will help you get started with PHP programming. In this series we will go through the basics of installing and getting PHP ready for development, the basic syntax and features of the language.

Online tutorial for PHP and MySQL

Contents

PHP Basics: Level 1

Installing a Webserver with PHP and MySQL (XAMPP)

1. XAMPP in Windows
   - Installing XAMPP in Windows
   - XAMPP is a cumulative package consisting of Apache, PHP and MySQL Packages is available for Windows
   - In this tutorial the XAMPP will be installed and the default Webserver directory will be **"htdocs"**.
2. XAMPP in Linux
   - Installing XAMPP in Linux
   - XAMPP is a cumulative package consisting of Apache, PHP and MySQL Packages is available for Linux
   - In this tutorial the XAMPP will be installed and the default Webserver directory will be **"opt"**.

Echo PHP Function, PHP Variables, If and Switch Statements

3. Echo Function
   - The echo() function outputs one or more strings.
   - Syntax: echo(strings);
   - Ex. echo "Hello World!";
4. Variables in PHP
   - Variables are used for storing values, like text strings, numbers or arrays.
   - When a variable is declared, it can be used over and over again in your script.
   - All variables in PHP start with a $ sign symbol.
   - The correct way of declaring a variable in PHP: $var_name = value;
5. If Statement
   - if statement - use this statement to execute some code only if a specified condition is true.
   - if...else statement - use this statement to execute some code if a condition is true and another code if the condition is false.
   - if...elseif....else statement - use this statement to select one of several blocks of code to be executed.
6. Switch Statement
   - switch statement - use this statement to select one of many blocks of code to be executed

PHP Operators

7. Arithmatic Operators
   - Ex. +,-,*,/,%,++,--
8. Comparison Operators
   - Ex. ==,!=,<>,>,<,>=,<=
9. Logical Operators
   - Ex. && (AND),|| (OR),! (NOT)

Dr. Mahesh Bundele
B.E., M.E., Ph.D.
Director
Poornima College of Engineering
ISI-6, RIICO Institutional Area
Sitapura, JAIPUR

Arrays in PHP

10. Arrays
    - An array stores multiple values in one single variable.
    - Numeric array - An array with a numeric index.
    - Associative array - An array where each ID key is associated with a value.
    - Ex. Numeric Array: $fruits=array("Apple","Mango","Banana","Grapes");
11. Multi-Dimensional Arrays
    - In a multidimensional array, each element in the main array can also be an array. And each element in the sub-array can be an array, and so on.

Loops in PHP

- Loops execute a block of code a specified number of times, or while a specified condition is true.

12. Loops - While Statement
    - The while loop executes a block of code while a condition is true.

```
while (condition)
{
code to be executed;
}
```

13. Loops - Do-While Statement
    - The do...while statement will always execute the block of code once, it will then check the condition, and repeat the loop while the condition is true.

```
do
{
code to be executed;
}while (condition);
```

14. Loops - For Statement
    - The for loop is used when you know in advance how many times the script should run.
    - Syntax:

```
for (init; condition; increment)
{
code to be executed;
}
```

15. Loops - Foreach Statement
    - The foreach loop is used to loop through arrays.

```
foreach ($array as $value)
{
code to be executed;
}
```

16.

17. **Functions in PHP**

18. Functions (Basic)
    - To keep the script from being executed when the page loads, you can put it into a function.
    - A function will be executed by a call to the function.

- You may call a function from anywhere within a page.
- Syntax:

```
function functionName()
{
code to be executed;
}
```

19. Functions (Advanced)

- We can also pass parameters to functions during both the declaration and calling time.
- function functionName($param1,$param2); //during function call.
- function functionName($param1,$param2)

```
{
code to be executed
}
```

20.

21. **PHP Special Variables**

22. GET Variable

- The built-in $_GET function is used to collect values from a form sent with method="get".
- Information sent from a form with the GET method is visible to everyone (it will be displayed in the browser's address bar)
- It has limits on the amount of information to send.

23. POST Variable

- The built-in $_POST function is used to collect values from a form sent with method="post".
- Information sent from a form with the POST method is invisible to others and has no limits on the amount of information to send.

PHP and HTML

24. Embedding PHP

- We can embed our PHP code anywhere in the webpage, by enclosing our script within the <?php...... //SCRIPT.......?>

25. Common Way to Display HTML

- We can also use the HTML Code within the PHP Script. Almost each of the HTML Tags can be used within a PHP Script.

Common Errors

- The PHP Engine in the webserver also displays the user the error in case there is something wrong in the code along with the tentative line number where the fault may have occurred. Thus, in this way we can eradicate errors.

26. Common Errors (Part 1)

- Learn how to spot errors and how to fix them
- Common Parse errors
- Parse errors due to missing comma or semicolon
- Parse errors due to not ending single or double quotes correctly

27. Common Errors (Part 2)

- Parse error due to missing or extra brackets
- Matching brackets during complex mathematical operations

Dr. Mahesh Bundele
B.E., M.E., Ph.D.
Director
Poornima College of Engineering
ISI-6, RIICO Institutional Area
Sitapura, JAIPUR

- Purpose and usefulness of correct indentation
- Errors due to missing or extra characters
- Undefined variable and undefined index errors

28. Common Errors (Part 3)
    - "Cannot modify header information - headers already sent by..." errors when using header() function
    - Using ob_start() to turn on output buffering
    - "Failed to open stream; no such file or directory in..." errors when including a invalid file
    - Using a @ symbol to suppress errors

MYSQL Tutorials: Level 2

MySQL is a Relational Database Management System (RDBMS) that runs as a server providing multiuser access to a number of databases. A third party open source software "phpMyAdmin" will be used as a web-based front end for managing MySQL databases easily and efficiently. It is widely installed by Web hosts worldwide, since it is developed in PHP and is included in the convenient LAMP stack, MAMP, and WAMP software bundle installers.

1. MySQL (Part 1)
    - An Introduction to the PHPMyAdmin Interface.
    - Creating a New Database
    - Creating a new Table and entering the value of the field with the requisite datatype.
    - SQL Query displayed in the PHPMyAdmin window.
2. MySQL (Part 2)
    - Connecting to the database and inserting dummy data into the database.
    - mysql_connect("server_addr", "username", "password") - Connect to the Database Server with the authorised user and password.
    - mysql_select_db("database_name") - Selecting a database within a connected database server.
3. MySQL (Part 3)
    - Writing some data into the database (INSERT and UPDATE Queries).
    - mysql_query('TYPE_HERE_YOUR_MYSQL_QUERY') - This function is used to run specific queries on our database.
    - INSERT QUERY - INSERT into *table* values ('att1', 'att2' , 'att3', 'att4' ,'att5') //Inserts Data into the table
    - UPDATE QUERY - UPDATE *table_name* SET att1='xyz' //Updates the Existing values stored in the table of the database.
4. MySQL (Part 4)
    - Getting data from the database table and displaying it.
    - SELECT QUERY - SELECT * FROM table_name WHERE att1='abc' // Query returns the value from the database where att1 = abc
    - mysql_num_rows() - Gives us the number of rows there are in the query we have just given out.
    - ORDER BY - Helps to order the output result as when selecting the values form the database. {Use of DESC for Descending ordering / ASC for Ascending ordering}
5. MySQL (Part 5)
    - mysql_fetch_assoc — Fetch a result row as an associative array.
    - array mysql_fetch_assoc (resource $result) //Returns an associative array that corresponds to the fetched row and moves the internal data pointer ahead.

mysql_fetch_assoc() is equivalent to calling mysql_fetch_array() with MYSQL_ASSOC for the optional second parameter. It only returns an associative array.

6. MySQL (Part 6)
   - Getting data from the database with the help of an HTML form.
   - Creating a FORM where a user can specify a name and selecting the appropriate value from the database.
7. MySQL (Part 7)
   - Changing the existing values of the databse table using HTML Forms.
   - Update unique records using the id than individual values.
8. MySQL (Part 8)
   - DELETE QUERY - To Delete the specific or all the entries of the Database.
   - DELETE FROM table_name WHERE field='xyz' // Deletes the entry from the database where the field = xyz.

PHP Advanced: Level 3

1. Name Splitter(Part 1)
   - We Input a fullname into a form and then splitting it into firstname and lastname
   - Use of : strlen(string) - This function counts total no of characters, including numbers and white spaces in the string
   - Use of: mb_substr(string,starting_position,no_of_characters) - This function takes a specific character from a string and a range of no of characters preceeding it.
2. Name Splitter(Part 2)
   - Divided the string into 2 halves through searching space, first half is stored as firstname and second half as lastname.
   - Use of : substn(string,starting_position,length) - This function results a substring starting from specified position to no of characters required.
3. PHP Dynamic Pages (Part 1)
   - We learn to create website with standard template and when on clicking the link changes only the content of page ,new page is not loaded.
   - Use of : include(variable) - This function is used to brought up all the content of variable page onto the current page.So that by changing the variable content of the website can be altered without reloading a similar page content everytime.
4. PHP Dynamic Pages (Part 2)
   - Making the dynamic linking user-friendly in case an error is obtained by include() function, i.e. checking if the file connected exists or not.
   - Use of : file_exists(variable) ->This function is results boolean value true(1) if the file exists and false(0) if not.
5. Simple Visitor Counter
   - Counts how many users have viewed your page as per count of refresh button clicked
   - fopen("file_name","parameter") opens a file (Creates it if not exists).parameter assigns the mode, w for writting mode, a for append mode
   - file_get_contents("file_name")- This function is used to obtain content from the file.
   - fwrite("file_name",variable) - This function writes into the file value present in variable.
6. Unique Visitor Counter (Part 1)
   - Counts how many users visiting based on their IP addresses. It obtains IP addresses stored in ip-file to match with user's IP
   - count() - This function is used to count no of lines in the file.

Dr. Mahesh Bundele
B.E., M.E., Ph.D.
Director
Poornima College of Engineering
ISI-6, RIICO Institutional Area
Sitapura, JAIPUR

    - $_SERVER[] - *This is an array containing information such as headers, paths and script locations.*
    - $_SERVER['REMOTE_ADDR'] - It informs about the IP address from which the user is viewing the current page.

7. Unique Visitor Counter (Part 2)
    - Retrieves IP addresses stored and compares them with IP of user viewing the current page.
    - fopen("file_name","parameter") opens a file (Creates it if not exists).parameter assigns the mode, w for writing mode, a for append mode
    - fwrite("file_name",variable) - This function writes into the file value present in variable.
    - intval(string) -This function converts an string value into a integer value.
8. Unique Visitor Counter (Part 3)
    - In this video errors have been corrected. Here counter keeps on increases
9. PHP String Functions (Part 1)
    - strlen(string) - This function counts total no of characters, including numbers and white spaces in the string
    - mb_substr(string,starting_position,no_of_characters) - This function takes a specific character from a string and a range of no of characters preceeding it.
    - explode("delimiter",string) -This function breaks down the string into a array. Delimiter is used to know from where to break string.
    - implode(string,"delimiter") -This function joins the array into a string. Delimiter is used to know how to join array elements.
    - nl2br() -This function prints the content in exactly same form as written. Used in case for breaking lines.
10. PHP String Functions (Part 2)
    - strrev(string) -This function is used to reverse the inputed string
    - strtolower(string) -This function is used to convert all alphabatic characters in string to thier small/lower case form.
    - strtoupper(string) -This function is used to convert all alphabatic characters in string to thier capital/upper case form.
    - substr_count(string,sub_string,) -This counts the no of substrings matching the particular value in string. It retuns an integer value.
    - substr_replace(original_string,string_to_replace) -This function replaces the cuntent of substring into original string.
11. Basic PHP Proxy
    - Providing the proxy to our page of a url.
    - foreach() - this loop looks through a block of code for each element in an array.
    - erag_replace(current_content, altered content,page) - This function is used to manipulate content of a proxy page.
12. Basic Advert Rotation (Part 1)
13. Basic Advert Rotation (Part 2)
14. Find and Replace
15. Date and Time (Part 1)
16. Date and Time (Part 2)
17. Creating Images with PHP
18. File Upload (Part 1)

Dr. Mahesh Bundele
B.E., M.E., Ph.D.
Director
Poornima College of Engineering
ISI-6, RIICO Institutional Area
Sitapura, JAIPUR

- Setup html form for file uploading
- Upload file and get file related information like file name, file size, etc
- Check for error messages after uploading file

19. File Upload (Part 2)
    - Move file from temporary area to user specified location
    - Restrict uploading to only specific file type
    - Restrict uploading to a maximum file size
20. Cookies (Part 1)
    - What are cookies
    - Set cookies using setcookie function
    - Understaing how to set expiry time of cookies
    - Read and print values from existing cookies
    - Print every cookie that we have stored
21. Cookies (Part 2)
    - Check if a cookie exists or not using isset
    - Unset a cookie when no longer required
    - Change the value of a existing cookie
22. Sessions
    - A PHP session variable is used to store information about, or change settings for a user session.
    - Session variables hold information about one single user, and are available to all pages in one application.
    - session_start() - Starting a PHP Session
    - $_SESSION['variable_name']=value - Stores the value in the Session variable.
    - session_stop() - Stopping a PHP Session
23. Search Engine Crawler Detection
24. Swear Word Filter (Part 1)
25. Swear Word Filter (Part 2)
26. Rename Function
27. SQL Injection (Part 1)
28. SQL Injection (Part 2)
29. MD5 Encryption
    - Calculates the MD5 hash of str using the RSA Data Security, Inc.'s MD5 Message-Digest Algorithm, and returns that hash (Its a one way encrypting technique).
    - Syntax : string md5 (string $str [, bool $raw_output = false])
    - Used in encrypting passwords and storing them in a database.
30. Sending Email (Part 1)
    - Create HTML form for getting email subject and message from the user
    - Using the mail() function to send email
31. Sending Email (Part 2)
    - Validating whether the name and message have been entered by the user
    - Check the length of the string using the strlen() function.
    - Set up the to, subject and message field of the mail() function
    - Send email and check for any errors
32. Sending Email (Part 3)
    - Fix the "Sendmail from not set in php dot ini" error

Dr. Mahesh Bundele
B.E., M.E., Ph.D.
Director
Poornima College of Engineering
ISI-6, RIICO Institutional Area
Sitapura, JAIPUR

- Create the mail "From:" header
- Using a local or external mail server to send email
- Using the ini_set() and ini_get() functions to set and read internal php configuration options respectively

33. Upload an Avatar Profile Image (Part 1)
34. Upload an Avatar Profile Image (Part 2)
35. Upload an Avatar Profile Image (Part 3)
36. Upload an Avatar Profile Image (Part 4)
37. Form Validation(Part 1)
38. Form Validation(Part 2)
39. Admin Only Pages (Part 1)
40. Admin Only Pages (Part 2)
41. Admin Only Pages (Part 3)
42. Create a news Feature (Part 1)
43. Create a news Feature (Part 2)
44. Create a news Feature (Part 3)
45. Display Images from a Directory
    - Using opendir() to open a directory handle
    - Using readdir() to read a directory that is already opened
    - Printing the directory listing
46. Pagination (Part 1)
47. Pagination (Part 2)
48. Language Chooser
49. PHP/ MYSQL Based Project - Basic Register and Login Module

1. **User Login**
    1. User Login Part 1
        - Collecting information from user in a form & connecting to authorized database.
        - mysql_connect("hostname", "username", "password") - Connect to the Database Server with the authorized user and password.
        - mysql_select_db("database_name") - This selects a database within a connected database server
    2. User Login Part 2
        - retrieves information about inputed username and checks whether given password matches with the password in database.
        - mysql_query('TYPE_HERE_YOUR_MYSQL_QUERY') - It is used to run specific queries on our database.Here it collects information from field username from specified table.
        - mysql_num_rows('query') - This function is user to counts no of rows retrieved from the query given to the database.
        - mysql_fetch_assoc('query')- This function fetches required information from the database in the form of array.
    3. User Login Part 3
        - Creating session for holding value and destroying that value by destroying session.
        - start_session() - Starts session to hold information from one pages to other until the session exists.
        - $_SESSION['variable_name']=value - Stores the value in the session variable.

Dr. Mahesh Bundele
B.E., M.E., Ph.D.
Director
Poornima College of Engineering
ISI-6, RIICO Institutional Area
Sitapura, JAIPUR

        - session_destroy() - destroys the value present in session variable.

2. **User Password Change**
    1. User Password Change Part 1
        - We learn to obtain old existing password and new password from the user.
        - start_session() - Hold information from previous pages to session page.
        - $variable_name=$_SESSION['value'] - to retrieve value containing in PHP variable.
    2. User Password Change Part 2
        - Checking whether encrypted old password matches with the database password and new password is same as confirm password.
        - md5("parameter")- encrypts parameter into irreversible logical code.
        - mysql_connect("hostname", "username", "password") - Connect to the Database Server with the authorized user and password.
        - mysql_select_db("database_name") - This selects a database within a connected database server
        - mysql_query('TYPE_HERE_YOUR_MYSQL_QUERY') - It is used to run specific queries on our database.Here it retrieves password of user logged in.
    3. User Password Change Part 3
        - updating the new password in database.
        - mysql_query('TYPE_HERE_YOUR_MYSQL_QUERY') - It is used to run specific queries on our database. Here it updates new password into database.
3. **User Registration**
    1. User Registration Part 1
        - Creating a form which allows user to input values in page
    2. User Registration Part 2
        - Striping tags of inputed strings and converting password into md5 encryption.
        - Use of : strip_tags(strigs) - cuts down unnecessary spaces,html tags and queries from string.
    3. User Registration Part 3
        - Checking whether the username and password provided meet the required length sizes.
        - Use of : strlen("string") - counts th character length of the string.
    4. User Registration Part 4
        - Inserting inputed information from the user into the database table through query.
        - mysql_connect("hostname", "username", "password") - Connect to the Database Server with the authorized user and password.
        - mysql_select_db("database_name") - This selects a database within a connected database server
        - mysql_query('TYPE_HERE_YOUR_MYSQL_QUERY') - It is used to run specific queries on our database. Here it inserts different fields into the database table.
    5. User Registration Part 5
        - Converting the password inputed from user to md5 encrypt form.
        - md5("parameter")- encrypts parameter into irreversible logical code.
    6. User Registration Part 6

- Checking the username provided so that condition for duplicate username can be avoided.
- mysql_query('TYPE_HERE_YOUR_MYSQL_QUERY') - This is used to run specific queries on our database. Here it checks if username already exists in database.
- mysql_num_rows('query') - This function is used to counts no of rows retieved from the query.
- strtolower(string) - converts all characters of string into lower case.

Dr. Mahesh Bundele
B.E., M.E., Ph.D.
Director
Poornima College of Engineering
ISI-6, RIICO Institutional Area
Sitapura, JAIPUR

The Spoken Tutorial Project

- Self-explanatory: uses simple language
- Audio-video: uses multisensory approach
- Small duration: has better retention
- Learner-centered: learn at your own pace
- Learning by doing: learn and practise simultaneously
- Empowerment: learn a new **FLOSS** (**F**ree/**L**ibre and **O**pen **S**ource **S**oftware)

Target Group

- Students- High School and College
- Working professional- Software users, developers and trainers
- Research scholars
- Community at large

Workshops

The Spoken Tutorial Project Team conducts workshops on Java and other FLOSS using spoken tutorials and gives certificates to those who pass an online test.
For more details, please visit
https://spoken-tutorial.org

Forum

We have developed a beginner friendly Forum to answer specific questions pertaining to any part of a particular tutorial.
For more details, please visit
https://forums.spoken-tutorial.org.

The Spoken Tutorial Project
is funded by the
National Mission on Education through
Information and Communication Technology,
Ministry of Human Resource Development,
Government of India.

Contact us

Email: contact@spoken-tutorial.org
Website: https://spoken-tutorial.org

Forum help available to all learners

Content available in 22 Indian languages

IIT Bombay

Spoken Tutorial

https://spoken-tutorial.org

Scan the QR code to
visit Spoken Tutorial website

Java

National Mission on Education through Information and Communication Technology
(NMEICT)
www.sakshat.ac.in

Funded by MHRD, Government of India.

Introduction

- Java is the most popular class-based, object-oriented, high-level programming language.
- Developed by James Gosling at Sun Microsystems and released in 1995 as a core component of Sun Microsystems' Java platform.
- Derives much of its syntax from C and C++.
- Is typically compiled to bytecode (class file). It can be run on any Java Virtual Machine (JVM) regardless of the architecture.
- Is specifically designed to have few implementation dependencies.
- Is Intended to let application developers write a code that runs on one platform & does not need to be recompiled to run on another.

Java has characteristics of Object-Oriented languages

- **Inheritance:** Creating new classes & extending them to reuse the existing code and adding new features as needed.
- **Encapsulation:** combining the information and providing the abstraction.
- **Polymorphism:** Providing different functionality by the functions having the same name, based on the signatures of the methods.
- **Dynamic binding:** Providing maximum functionality to a program about the specific type at runtime.

Features

Platform independence:
Key feature of Java language is write-once-run-anywhere (WORA) concept. With Java, you can run the code written on any system.

Simplicity:
Programs are easy to write and debug. Java provides a bug-free system due to strong memory management.

Portability: Java feature write-once-run-anywhere makes it portable, provided that the system has an interpreter for JVM.
Also, Java has standard data size irrespective of the OS or the processor.

Performance: Uses native code and lightweight process called threads.
The advance version of JVM uses adaptive and just-in-time compilation technique to improve the total performance.

Distributed: Widely used protocols like HTTP and FTP are developed in Java.
Internet programmers can call functions on these protocols and can access the files from any remote machine on the internet, rather than writing codes on their local system.

Secure:

- Programs in Java run under an area known as the sandbox.
- Security manager determines the accessibility options of a class like reading and writing a file to the local disk.
- Uses public key encryption system to allow the java applications to transmit over the internet, in a secure and encrypted form.
- The bytecode verifier checks the classes after loading.

Robust:

Java has

- Strong memory allocation.
- Automatic garbage collection mechanism.
- Powerful exception handling.
- Type-checking mechanism.
- A compiler that checks the program for any errors and interpreter checks any runtime errors and makes the system secure from crashes.

Online tutorial for Java
Contents

Basic Level

Introduction to Java

1. Getting started with Java – installation
   - Install jdk from Synaptic Package Manager
   - Choose openjdk-6-jdk from the list of packages available
   - Mark it for installation
   - The installation will take a few seconds
   - Verify the installation
   - At the command prompt type java -version, so the version number of the jdk will be displayed
   - Run a simple java program and see if it works
   - Type javac TestProgram.java for compiling the code and java TestProgram for executing the code
2. Java - First program
   - write simple java program
   - print “My First Java Program!” on Console
   - save the file
   - file name given to the java file
   - compile the file
   - run the file
   - correct the errors
   - naming conventions for class
   - naming conventions for method
   - naming conventions for variable

Eclipse

3. Installing Eclipse
   - Install Eclipse on Ubuntu on the Terminal
   - Set up the proxy on the Terminal
   - Then fetch the list of all the available softwares
   - Type sudo apt-get update
   - Then install eclipse on the Terminal
   - Type sudo apt-get install eclipse
   - Verify if Eclipse is installed on the system
   - Installing Eclipse on Debian,Kubuntu,Xubuntu

- Installing Eclipse on Redhat
- Installing Eclipse on Fedora,centos and suse linux

4. Getting started with Eclipse
    - Eclipse is an Integrated Development Environment
    - It is a tool on which one can write, debug and run java programs easily
    - Open Dash Home and type Eclipse in the search box.
    - We get Workspace Launcher
    - On clicking on Workbench we get the Eclipse IDE
    - Go to File->New->Project and select Java Project
    - Create a project named EclipseDemo and create a class inside DemoClass
    - Learn about Package Explorer and Editor portlet
5. Hello World Program
    - Open Eclipse
    - Create a Java Project named DemoProject
    - Create a class named DemoClass
    - Class name and file name will be the same
    - Eclipse suggests various possibilities as we type a command
    - Eclipse also completes the parentheses by automatically adding the closing parentheses
    - Include the statement that we want to print
    - Eclipse also completes the quotes by adding the closing quote
    - Compile and execute the program
    - Change the code to print
6. Errors and Debugging
    - When writing a Java Program, here is a list of typical errors:
    - Missing semicolon(;)
    - Missing double quotes(".")
    - Mis-match of filename and classname
    - Typing the print statement n lower case
    - The line which has the error will be indicated with a red cross mark on the left margin
    - The list of errors is displayed by hovering the mouse over the cross mark
    - Create a class ErrorFree with Errors, debug the code and run it
    - Eclipse also offers intelligent fixes
7. Programming features of Eclipse
    - Auto completion
    - Sets the corresponding closing brace when we open the brace
    - Provides a drop-down list of methods when you start typing the code.
    - Syntax highlighting
    - Classname is highlighted in pink color and method in blue color.
    - Keyboard shortcuts
    - F11 to debug a program and Ctrl plus H to search a specific file.
    - Error highlighting
    - Cross symbol in the program denotes errors

- Remove semicolon and error details are displayed when mouse is hovered over cross symbol.

Fundamental Programming Structures in Java

8. Numerical datatypes
   - Define datatypes and numerical datatypes
   - int
   - float
   - byte
   - short
   - long
   - double
   - range of each numerical datatypes
   - declaration and initializationof numerical datatypes.
   - valid and invalid declaration
9. Arithmetic Operations
   - Define an operator
   - Define arithmetic operators
   - addition
   - subtraction
   - multiplication
   - division
   - modulo
   - simple program to demonstrate arithmetic operators
   - appropriate datatypes for appropriate values
   - save, compile and run the program
10. Strings
    - char datatype
    - letter, digit, punctuation marks, **tab**, or a space are all characters.
    - Program explaining the variable and the character data.
    - Introduction to strings
    - Creating string by Direct Initialization
    - Creating string by using new operator
    - String length()
    - String concat()
    - String toUpperCase()
    - String toLowerCase()
11. Primitive type conversions
    - define type conversion or type casting
    - higher order integer to lower order integer- Explicit type casting
    - program to show explicit type casting
    - common mistake in explicit type casting.

- program to show common mistake in explicit type casting
- lower order integer to higher order integer – Implicit type casting
- program to show implicit type casting
- char to integer
- integer to char
- program to show char to int type casting.

Control flow

12. Relational Operations
    - boolean datatype
    - equal to and not equal to
    - less than and less than or equal to
    - greater than and greater than or equal to
13. Logical Operations
    - use of logical operators
    - and (&&) operator
    - example to explain **and** operator
    - program to demonstrate **and** operator
    - or (||) operator
    - example to explain **or** operator
    - program to demonstrate **or** operator
    - not (!) operator
    - program to demonstrate **or** operator
    - save, compile and run the programs
14. if else construct
    - Conditional Statements and types of Conditional Statements
    - Use of if statement
    - Syntax for if statement
    - Program using if statement
    - Use of if else statement
    - Syntax for if else statement
    - Program using if else statement
    - Use of if else if statement
    - Syntax for if else if statement
    - Program using if else if statement
15. nested if and ternary operator
    - explain nested if
    - nested if syntax
    - program to demonstrate nested if
    - explain the control flow of the program
    - explain ternary operator
    - syntax for ternary operator

Dr. Mahesh Bundele
B.E., M.E., Ph.D.
Director
Poornima College of Engineering
ISI-6, RIICO Institutional Area
Sitapura, JAIPUR

- explain the syntax
- program to demonstrate ternary operator
- comparison between ternary operator and nested if
- save, compile and run the program

16. switch statement
    - define switch case statement
    - compare switch and nested if
    - switch case syntax
    - working of a switch case statement
    - use of keyword switch
    - valid and invalid use of keyword **case**
    - use of keyword default
    - use of keyword break
    - program to demonstrate switch case statement
    - save, compile and run the program to check the output
17. while loop
    - Loop control statement
    - types of loop control statements
    - Introduction to while loop
    - syntax of while loop
    - Program using while loop
    - Check the output.
    - Introduction to infinite loop
    - loop variable modification
    - Check the output
    - How to terminate the infinite loop
18. for loop
    - syntax
    - introduction to for loop
    - for loop syntax
    - loop vaiable
    - loop condition
    - loop variable increment or decrement
    - loop block
    - flow of loop
    - advantage of using loop
19. do while loop
    - define do while
    - do while syntax
    - working of do while loop
    - example of do while loop
    - explain the do while programming
    - save, compile and run the program to check the output

Dr. Mahesh Bundele
B.E., M.E., Ph.D.
Director
Poornima College of Engineering
ISI-6, RIICO Institutional Area
Sitapura, JAIPUR

- how different is it from the while loop
- program to demonstrate the differences

Arrays

20. Introduction to Arrays
    - About Arrays
    - Declare an Array
    - Initialize an array
    - Intilalization using for loop
    - Index of an array elements
    - change values of an array
    - print the value of an array
    - Advantage of an array.
21. Array operations
    - import java.util.Arrays
    - use methods from class Arrays
    - toString() method
    - sort() method
    - fill() method
    - copyof() method
    - copyofRange() method
    - about parameters for each method.

Classes & Objects

22. Creating class
    - Whatever we can see in this world are all objects
    - Objects can be categorized into groups known as class
    - This is class in real world
    - Human Being is an example of class in real world
    - Class in java is the blue print from which individual objects are created
    - Class consists defines a set of properties called variables and a set of behaviors called methods
    - Syntax for creating class
    - Create a simple class Student using Eclipse
    - The Student class can contain properties
23. Creating Object
    - An object is an instance of a class
    - Each object consist of state and behavior
    - Object stores it state in fields or variables
    - It exposes its behavior through methods
    - Reference variables
    - Create a class named TestStudent

Dr. Mahesh Bundele
B.E., M.E., Ph.D.
Director
Poornima College of Engineering
ISI-6, RIICO Institutional Area
Sitapura, JAIPUR

- Create an object of the Student class
- Use new operator
- Check what the reference variable contains
- Create one more object of the Student class and check what the reference variable contains

24. Instance fields
    - Also known as non-static fields
    - Open the TestStudent class which we have created
    - Access the fields roll_number and name using dot operator
    - See the output
    - Initialize the field and see the output
    - Change the modifier of the fields to private
    - Debug the error that you get
    - Change the modifier to protected
    - Each object of a class will have unique values
    - Create two objects of the Student class
25. Methods
    - method definition
    - write simple method
    - method returning value
    - call a method in another method
    - flow of the program
    - call a static method
    - call a method from another class
    - method signature
    - method body

Constructors

26. Default constructor
    - what is a constructor?
    - what is a default constructor?
    - when is it called?
    - define a constructor
    - initialize the variables
    - call the constructor
    - difference between constructor and method
27. Parameterized constructors
    - What is a parameterized constructor?
    - create constructor without parameter
    - create a constructor with parameter
    - assign values to the variables in the constructor
    - pass arguments during the constructor call

- working of parameterized constructor
- show common errors
- resolve the errors
- create another parameterized constructor
- why to use constructor?

28. Using this keyword
    - this is a reference to the current object
    - helps to avoid name conflicts
    - we can use this keyword inside a constructor to call another one
    - the constructors must be in the same class
    - explicit constructor invocation
    - Explain it using the parameterized constructor code
    - Make this statement the last one in the constructor
    - You will get an error
    - this statement should be the first one inside a constructor
29. Non-static block
    - Non-static block
    - Any code written between two curly brackets
    - Executed for each object that is created
    - Executes before constructor's execution
    - can initialize instance member variables of the class
    - create a class named NonStaticTest
    - Create a non-static block and a constructor inside it
    - Check the output
    - Include multiple non-static blocks
    - they will be executed in the sequence in which they appear in the class
    - Check the output
    - Non-static block is not a substitute for constructor
30. Constructor Overloading
    - define multiple constructor
    - what is constructor overloading?
    - constructor with different number of parameters.
    - parameters with different datatypes.
    - how is constructor overloaded?
    - flow of overloading process.
    - advantage of constructor overloading.
31. Method Overloading
    - define multiple methods.
    - methods with same name.
    - methods with different number of parameters.
    - methods with different datatypes of parameter.
    - what is method overloading?
    - example for overloadin method

Dr. Mahesh Bundele
B.E., M.E., Ph.D.
Director
Poornima College of Engineering
ISI-6, RIICO Institutional Area
Sitapura, JAIPUR

- how to overload method?
- advantage of method overloading.
- error in method overloading.

32. Taking user input in Java
    - What is BufferedReader?
    - Importing three classes from Java.io package
    - How to take the input from the user?
    - Syntax to implement BufferedReader
    - What is InputStreamReader?
    - Create object of InputStreamReader
    - Create object of BufferedReader
    - About IOException
    - About throws keyword
    - Typecasting

Intermediate Level

1. Subclassing and Method Overriding
    - Definition of subclassing
    - Demo of subclassing using an Employee and Manager class
    - Single inheritance
    - Use of extends keyword
    - Private members in a super class
    - Definition of method overriding
    - Annotation
    - @Override Annotation
2. Calling methods of the superclass
    - super keyword
    - Call methods of the super class
    - Constructor of the super class
    - Demo of super keyword using an Employee and Manager class
    - Single inheritance
    - Use of extends keyword
    - Private members in a super class
3. Using final keyboard
    - final keyword
    - What is final keyword and its application?
    - Where final keyword can be declared?
    - final variable
    - final static variables
    - static block
    - final variable as parameter
    - final method

Dr. Mahesh Bundele
B.E., M.E., Ph.D.
Director
Poornima College of Engineering
ISI-6, RIICO Institutional Area
Sitapura, JAIPUR

- private final method
- final class

4. Polymorphism
   - Polymorphism in Java
   - Run-time polymorphism
   - Virtual Method Invocation
   - Compile-time polymorphism
   - Role of JVM
   - IS-A test
   - Static binding
   - Dynamic binding
5. Abstract Classes
   - Abstract Classes in Java
   - What are Abstract Methods
   - What are Concrete Methods
   - Properties of Abstract Methods and Abstract Classes
   - How to use Abstract Methods
6. Java Interfaces
   - Java Interfaces
   - Implementing Interface
   - Implementation Classes
   - Interfaces Vs Abstract classes
   - Implementing Multiple Interfaces
   - Usage of Interfaces with an example
7. Static Variables
   - What is Static Variable in Java?
   - Usage of Static Variables with Example
   - Static Variables Vs Instance Variables
   - Final Static Constants
8. Static Methods
   - What is static method in Java?
   - Static methods Vs Instance methods
   - Usage of static method with example
   - Passing object variables in a static method
9. Static Blocks
   - What is a static block
   - Declaring and defining a static block
   - How static blocks are invoked and executed
10. Exception Handling
    - What is an Exception
    - Types of Exceptions
      3. Checked Exceptions
      4. Unchecked Exceptions

Dr. Mahesh Bundele
B.E., M.E., Ph.D.
Director
Poornima College of Engineering
ISI-6, RIICO Institutional Area
Sitapura, JAIPUR

   - Explaining *ArrayIndexOutOfBoundsException*
   - Demonstrating Checked Exceptions with example
   - Handling Exceptions using try-catch blocks
   - Explaining *ArithmeticException*
   - Demonstrating Unchecked Exceptions with example
   - Explaining *FileNotFoundException*
   - Usage of finally block
   - Explaining *NullPointerException*

11. Custom Exceptions
    - What is a Custom Exception
    - Demonstration of custom exception
    - Custom exception example - "InvalidMarkException"
    - Usage of "throw" keyword
    - How to resolve errors in custom exceptions
    - Resolve error using "Add throws declaration" option
    - Usage of "throws" keyword
    - Example for "FileNotFoundException"
    - How to handle multiple exceptions
    - How to use "surround with try/catch" option

Dr. Mahesh Bundele
B.E., M.E., Ph.D.
Director
Poornima College of Engineering
ISI-6, RIICO Institutional Area
Sitapura, JAIPUR

python™

What is Python?

Python is a general purpose, high level, remarkably powerful dynamic programming language used in a wide variety of application domains.

Why Python?

- Easy to read and learn
- Free and Open Source
- Useful for scientific computing
- Powerful interactive interpreter
- Extensive scientific libraries
- Well documented

Where can you use Python?

- Numeric and Symbolic computation
- 2D/3D Plotting
- User interfaces
- Parallel computing
- Machine Learning and Image Processing
- Game development
- Web development
- Much more...

Who uses Python?

- Google
- Yahoo
- Walt Disney
- NASA
- IBM
- YouTube
- nVIDIA
- Software - Blender, Motion Builder, Cinema 4D, etc.
- Games - Battle field 2 by EA sports, Crystal space 3D, etc.

Python is one of the most popular programming languages today, and therefore has been included in the CBSE curriculum. It easily performs tasks that proprietary tools like Matlab and Mathematica offer. Today leading companies are using Python extensively, hence there are better job opportunities. Learn Python, and grab the Opportunity!

matplotlib django pandas SciPy

SymPy IP[y]: IPython Interactive Computing Mayavi NumPy

Sponsored by

fossee better education

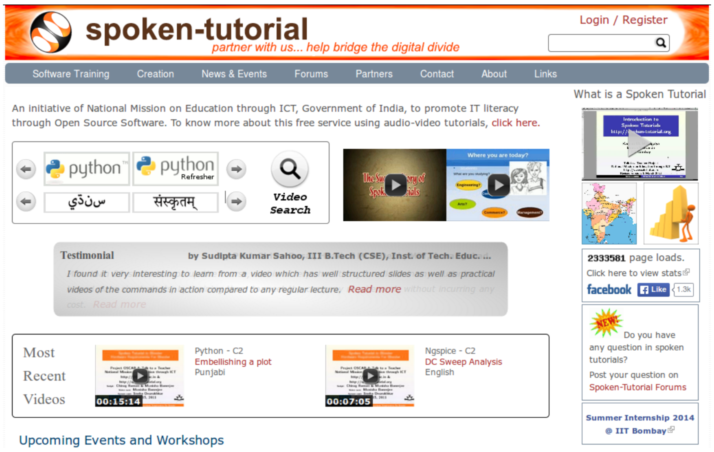
National Mission on Education through ICT
MHRD
http://www.sakshat.ac.in

IIT BOMBAY

How can you learn Python

- **Spoken Tutorial -** The FOSSEE project has created a series of Spoken Tutorials on Python. Theses are available for learning, on the Spoken Tutorial website, free of cost. You can access these tutorials from this link

  python.fossee.in/spoken-tutorials



An initiative of National Mission on Education through ICT, Government of India, to promote IT literacy through Open Source Software. To know more about this free service using audio-video tutorials, click here.

Testimonial by Sudipta Kumar Sahoo, III B.Tech (CSE), Inst. of Tech. Educ. ...

I found it very interesting to learn from a video which has well structured slides as well as practical videos of the commands in action compared to any regular lecture. Read more

Most Recent Videos

00:15:14 Python - C2 Embellishing a plot Punjabi

00:07:05 Ngspice - C2 DC Sweep Analysis English

Upcoming Events and Workshops

2333581 page loads. Click here to view stats

facebook Like 1.3k

NEW! Do you have any question in spoken tutorials? Post your question on Spoken-Tutorial Forums

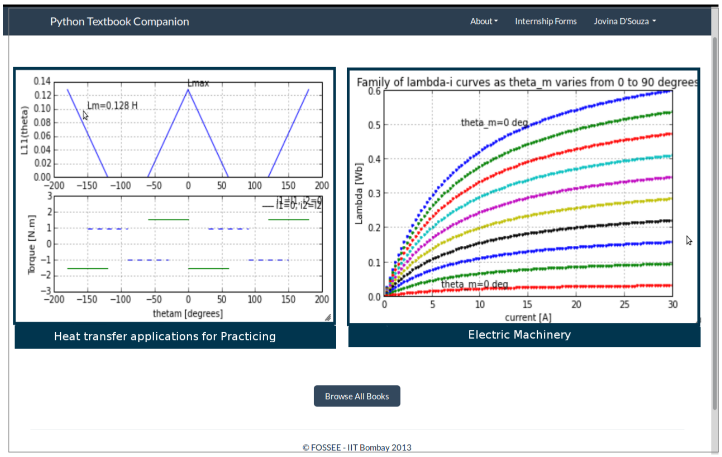
Summer Internship 2014 @ IIT Bombay

Spoken Tutorial website

- **Textbook Companion Internship -** Learn Python in a practical way by contributing to the Python Textbook Companion Internship. It aims to create Companions by coding solved examples from Standard textbooks, using Python. Participate and earn attractive honoarium and Certificate of Internship from FOSSEE, IIT Bombay! For more details, please visit:

  python.fossee.in/textbook-companion-project

Completed Book	: Approx 453 books
Under Progress	: Approx 113 books



Python Textbook Companion website

- **SELF Workshops -** The Spoken Tutorial Team conducts workshops on Python. These are completely free of cost, and are conducted without the need of any domain expert. Learn Python and obtain a certificate from Spoken Tutorial Project, IIT Bombay, upon successful completion of the post-workshop evaluation test. Please visit:

  python.fossee.in/spoken-tutorials

About us

Website:

http://python.fossee.in

Contact us

General help & Queries:

info@fossee.in

python@fossee.in

Sponsored by

National Mission on Education through ICT
MHRD
http://www.sakshat.ac.in

fossee better education

IIT BOMBAY

Online web tutorials for Python
Contents

Module 1: Basic Plotting

Getting started with ipython

1. Use Python 3.5.2
2. Use Ipython version 5.1.0
3. IPython is an enhanced interactive Python interpreter.
4. Invoke the IPython interpreter
5. Quit the IPython interpreter
6. Navigate the IPython session history
7. Use tab-completion to work faster.
8. See the documentation of functions using question mark.
9. Interrupt commands using Ctrl + C when we make an error.
10. round command

Using the plot command interactively

1. Use Python 3.4.3
2. Use Ipython version 5.1.0
3. Start IPython with pylab.
4. ImportError if matplotlib is not installed
5. clf() function
6. Use the linspace function to create equally spaced points in a region.
7. Find the length of sequences using len function.
8. Plot mathematical functions using plot.
9. Clear drawing area using clf.
10. Usage of buttons in the UI of the plot window such as - save, zoom, move axis, back and forward and Home

Embellishing a plot

1. Use Python 3.4.3
2. Use Ipython version 5.1.0
3. Modify the attributes of a plot
4. Pass additional keyword arguments to plot command
5. Add title to a plot using 'title' command.
6. Incorporate LaTeX style formatting by adding a $ sign before and after the string.
7. Label x and y axes using xlabel() and ylabel() commands.
8. Add annotations to a plot using annotate() command.
9. Get the limits of axes using xlim() and ylim() commands.
10. Set the limits of axes using xlim() and ylim() commands.

Saving plots

1. Use Python 3.4.3
2. Use Ipython version 5.1.0
3. Save plots using the savefig() function.
4. Save the plots in different formats like
    - pdf
    - ps

Dr. Mahesh Bundele
B.E., M.E., Ph.D.
Director
Poornima College of Engineering
ISI-6, RIICO Institutional Area
Sitapura, JAIPUR

- png
- svg
- eps

Multiple plots

1. Use Python 3.4.3
2. Use IPython 5.1.0
3. Draw multiple plots which are overlaid.
4. Operations on individual plots.
5. Use the figure command.
6. Distinguish between multiple overlaid plots.
7. Use the legend command.
8. Serial number of the plot to select corresponding plot.
9. Switch between the plots
10. Saving individual plots.

Subplots

1. Use Python 3.4.3
2. Use IPython 5.1.0
3. Creating subplots
4. Switching between subplots.
5. Subplot command
6. Passing arguments to subplot command.
7. First argument is the number of rows of subplots.
8. Second argument is the number of columns of subplots
9. Third argument specifies the serial number for subplot.

Additional features of IPython

1. Use Python 3.4.3
2. Use IPython 5.1.0
3. Retrieve the history using %history command.
4. View only a part of history by passing argument to %history command.
5. Pass arguments to %history to get particular lines of code
6. Save the required lines of code in required order using '%save' command.
7. Use '%run -i' command to run a saved script.

Module 2: Plotting Experimental Data

loading data from files

1. load data from file
2. single column
3. multiple columns separated by delimiter
4. cat command
5. loadtxt()
6. columns separated by spaces
7. columns separated by semi-colon
8. unpack argument
9. delimiter argument
10. three columns of data

Plotting the data

1. plotting data

2. list
3. list element-wise squaring
4. plot data points
5. clear plots
6. errorbar function
7. dots or filled circles in plot
8. plot with red pluses
9. explore documentation in ipython
10. plot with errorbars
11. using format argument

Other types of plots

1. scatter plot
2. scatter function
3. scatter plot with various arguments
4. logarithmic plot
5. loglog function
6. cat command
7. loadtxt function
8. unpack parameter of loadtxt
9. linspace
10. scatter versus plot

Plotting charts

1. Use Python 3.4.3
2. Use IPython 5.1.0
3. To produce scatter plot
4. Plot a pie chart using pie() function
5. Plot a bar chart using bar() function
6. Access the matplotlib online help
7. Charts with line hatching

Module 4: Handling Large Data Files

Getting started with lists

1. What is a list?
   - Define List
   - List index
2. Create:
   - List with elements
   - Empty list
   - List within a list
3. Find out the list length using len function
   - Access elements using their index numbers
   - Append elements to list using the function append
   - Delete element from list using the del and remove function

Getting started with for

1. For loop syntax
2. Example to use For loop
3. Indentation in for loop

Dr. Mahesh Bundele
B.E., M.E., Ph.D.
Director
Poornima College of Engineering
ISI-6, RIICO Institutional Area
Sitapura, JAIPUR

4. Create blocks in python using for
5. Iterate over a list using for loop
6. How to get out of the block
7. use of Range() function
8. Range function in for loop
9. How to use Python interpreter
10. IPython interpreter to specify blocks

Getting started with strings

1. What are strings?
2. How are strings denoted in Python?
3. String concatenation
4. Multiply a string with an integer
5. Accessing individual elements of a string
6. Accessing elements of a string using negative indices
7. Split() function
8. Join() function
9. Define a string in different ways
10. Print a string repeatedly

Getting started with files

1. Open a file
2. Open() function
3. Different Modes of opening a file
4. Read() method
5. Read the content of the file line by line
6. Read the entire content of the file
7. Append the lines of a file to a list
8. Close a file
9. Demonstration using a txt file
10. Splitlines() method

Parsing data

1. What is Parsing data?
2. split function and its syntax
3. What is string tokenizing?
4. How to split a string on whitespace?
5. split function with argument
6. strip function and example
7. Converting string into floats and integers
8. Example to read a huge .txt file line by line and parse each record
9. Perform computations on the .txt file
10. Execute the file using %run command

Statistics

1. Statistical operations in Python
2. Installation of Numpy for mathematical and logical operations
3. Installation of pip to install python libraries
4. loadtxt() function with example
5. Getting the shape of an array

Dr. Mahesh Bundele
B.E., M.E., Ph.D.
Director
Poornima College of Engineering
ISI-6, RIICO Institutional Area
Sitapura, JAIPUR

6. Getting the sum of a column in an array
7. How to calculate mean?
8. Calculate mean across each of the axis of the array
9. How to calculate median?
10. How to calculate standard deviation?

Module 5: Arrays and Matrices

Getting started with arrays

1. Overview of array
2. Usage of numpy library
3. How to create arrays
4. How to create two dimensional array
5. arange() method
6. reshape() method
7. How to find the shape of an array?
8. Create a new array with elements of different datatypes
9. Identity matrix
10. Zeros method

Accessing parts of arrays

1. Create a one-dimensional array
2. Create a two-dimensional array
3. Accessing individual elements of an array
4. How to change the value of an array
5. How to change more than one elements at a time
6. Negative indexing of arrays
7. Slicing of an array
8. Striding of an array
9. Access only the odd rows and columns of an array
10. Examples to demonstrate all the manipulations of arrays

Image manipulation using Arrays

1. Read images into arrays
2. How to access parts of an array?
3. imread command
4. imshow command
5. show command
6. How to check the dimensions of an array?
7. Example to access parts of an image
8. How to stride over an array?
9. Example to access an RGB image
10. Slice an image of different dimension

Basic Matrix Operations

1. Create matrices from lists
2. asmatrix method
3. arange and reshape methods
4. Basic matrix operation
5. Addition, subtraction and multiplication of a matrix
6. Determinant of a matrix

Dr. Mahesh Bundele
B.E., M.E., Ph.D.
Director
Poornima College of Engineering
ISI-6, RIICO Institutional Area
Sitapura, JAIPUR

7. eye(), allclose() functions
8. Inverse of a matrix
9. eigenvalues and eigenvectors of a matrix
10. diag() function

Advanced Matrix Operations

1. flatten() function
2. Example to convert a multidimensional matrix to single dimension matrix
3. Frobenius norm of a matrix
4. Demonstration of Frobenius norm of a matrix
5. Inverse of a matrix
6. Infinity norm of a matrix
7. norm documentation
8. Singular value decomposition
9. svd() function
10. smat function

Least square fit

1. Generating a Least Square fit line
2. Generating L vs t square
3. loadtxt function
4. Usage of numpy library
5. Plotting L vs t square
6. Steps for least square fit line
7. Matrix formulation tsq=A.p
8. Generating the two matrices tsq and A
9. Finding transpose of a matrix
10. lstsq() function

Module 6: Python Language: Basics

Basic datatypes & operators

1. Data types in Python
2. Demonstration of int, float and complex data types with examples
3. Different functions associated with int data type
4. Complex numbers and their functions
5. Boolean operations with examples
6. Operator precedence with parentheses
7. Different operators available in Python3
8. Modulo operator with examples
9. How to do exponent operation in Python?
10. How to find the square root of a number in Python?

Sequence datatypes

1. List, string and tuple sequence data types with examples
2. How to access a list using index numbers?
3. Access the string elements
4. Access the tuple elements
5. How to add different sequence data types?
6. How to find the length of a variable?
7. Find the sum of a list

Dr. Mahesh Bundele
B.E., M.E., Ph.D.
Director
Poornima College of Engineering
ISI-6, RIICO Institutional Area
Sitapura, JAIPUR

8. Convert list to tuple
9. Convert tuple to list
10. Convert string to list and list to string

Input/output

1. Input Output in Python
2. Various output statements
3. Print a string
4. Print a string with newline character
5. How to use format operators?
6. Example for integer format
7. Example for string format
8. Example for float format
9. Getting input from the user using Input()function
10. Display a prompt to get the input
11. Save the script as filename.py and execute using %run command

Conditionals Statements

1. if condition statement
2. Demonstration of if statement with example
3. if/else condition statement with example
4. Importance of indentation in a program
5. Usage of colon in program
6. Condition statement using elif
7. Examples using if/elif/else block
8. Ternary conditional statement
9. Difference between if/else and ternary conditional statements
10. How to use pass statement?

Loops

1. Explanation of while loop
2. Demonstration of while loop with example
3. Print the squares of all the even numbers below 10 using while loop
4. How to use for loop?
5. Print the squares of all the even numbers below 10 using for loop
6. for loop with range function
7. How to use break statement in for loop
8. pass statement in for loop
9. continue statement in for loop
10. Demonstration of pass, break and continue statements

Module 7: Python Language: Datastructures

Manipulating lists

1. Various manipulation in lists
2. Slicing of lists
3. Syntax and demonstration of slicing of lists
4. How to use step value in slicing
5. Striding of list
6. Examples with various parameters in striding of list

Dr. Mahesh Bundele
B.E., M.E., Ph.D.
Director
Poornima College of Engineering
ISI-6, RIICO Institutional Area
Sitapura, JAIPUR

7. sort method in list
8. Usage of sorted() built-in function
9. Reverse a list
10. Striding with negative values
11. How to store a new reversed list in another variable

Manipulating strings

1. How to slice a string
2. Various way to get substrings using index
3. Reverse a string
4. How to check if a given string is a palindrome or not
5. Replace characters in a string
6. Convert a string to uppercase
7. Convert a string to lowercase
8. How to use for loop in a list
9. Join method
10. Join list elements to form a string

Getting started with tuples

1. What are tuples?
2. How to declare tuples?
3. Examples to declare tuples
4. Demonstration of creating tuple
5. Accessing tuples by their index positions
6. Iteration over tuples
7. Demonstration of immutability property of tuples
8. How to swap values in tuples
9. Similarities of tuples with lists
10. Tuple packing and unpacking

Dictionaries

1. Overview of dictionaries
2. Creating an empty dictionary
3. Creating a non empty dictionary
4. About key:value pair
5. How to access the dictionary elements
6. Demonstration of wrong key
7. Add, delete and modify an item in a dictionary
8. Usage of method in
9. Retrieve the keys and values by using the methods keys() and values()
10. Iterate over elements of a dictionary using a for loop

Sets in Python

1. What are sets in python?
2. Input sets
3. Create sets
4. How to create empty sets?
5. Operations on sets
6. Add and remove methods
7. Union and intersection methods

8. Difference and symmetric_difference methods
9. Subset and superset
10. Length and containership on sets

Module 8: Python Language: Advanced

Getting started with functions

1. About Functions
2. How to define a function
3. Example for defining a function
4. Calling a function with arguments
5. Calling a function without arguments
6. Return values from a function
7. Indentation in coding
8. Documenting or commenting code
9. How to use docstrings in python function
10. How to write a function circle to return area and perimeter with radius r

Advanced features of functions

1. Functions with default arguments
2. Various examples for default arguments
3. Interchanging the default and non-default arguments
4. Call a function with keyword arguments
5. Call a function without keyword arguments
6. Functions with positional arguments
7. Functions with arbitrary arguments
8. Demonstration of arbitrary arguments
9. Usage of * and ** in defining a function
10. Python built-in-functions

Using python modules

1. Python modules
2. Run a Python script from command line
3. How to import modules in python scripts?
4. How to import required functions from a module?
5. Usage of namespace
6. Advantages of using import functions
7. Using alias to the module
8. Demonstration of import functions
9. Run python scripts in ipython interpreter
10. Python standard library of modules

Writing python scripts

1. About Python modules
2. What is importing?
3. How to write a function and save it as a script
4. Run a python script
5. Import a module
6. Example to import a module
7. usage of __name__ variable
8. Write test condition using the name variable

Dr. Mahesh Bundele
B.E., M.E., Ph.D.
Director
Poornima College of Engineering
ISI-6, RIICO Institutional Area
Sitapura, JAIPUR

9. How importing works in new IPython console
10. Different ways of running the Python script

Testing and debugging

1. What is software testing?
2. Write a simple function
3. How to write test cases?
4. Create simple tests for a function
5. Run the script and test the code
6. Automate tests
7. Example for test case fail
8. Coding style
9. How to give meaningful names in coding
10. Python coding standards

Handling Errors and Exceptions

1. Errors in Python
2. Syntax errors and exception
3. Exceptions with example
4. Syntax error with example
5. Demonstration of ValueError exception
6. Demonstration of ZeroDivisionError exception
7. try except clause in Python
8. What is debugging?
9. Using %debug for debugging in ipython
10. try except with else clause

Dr. Mahesh Bundele
B.E., M.E., Ph.D.
Director
Poornima College of Engineering
ISI-6, RIICO Institutional Area
Sitapura, JAIPUR

QCad

QCAD is an application for computer aided drafting in two dimensions (2d). It is a simple 2D CAD system for everyone. Using QCAD you can create technical drawings such as plans for buildings, interiors, mechanical parts or schemas and diagrams. QCAD works on Windows, Mac OS X and many Linux and Unix Systems. The source code of the QCAD community edition is released under the GPL (Open Source).

Contents

Basic Level

Dr. Mahesh Bundele
B.E., M.E., Ph.D.
Director
Poornima College of Engineering
ISI-6, RIICO Institutional Area
Sitapura, JAIPUR