

Spring MVC

This **Spring MVC Training Course** will teach you all the exciting features of this successful Framework by building a complete online shopping web application

Prerequisites

You should be at the level of our [Advanced Java](#) and also [SQL](#) courses . This course teaches you the latest tools to use in developing robust enterprise-wide Java Web Applications

Further Training

[Spring Microservices](#)

Contents

DAY 1:Configuring a Spring Development Environment

- Setting up Java
 - Installing JDK
- Setting up environment variables
 - Installing Maven
 - Installing Tomcat
 - Installing Eclipse STS
- Creating our first Spring MVC project
 - Adding Spring jars to the project
 - MVC
 - adding a welcome page
 - The Dispatcher servlet
 - Deploying and running the project

Spring MVC Architecture – Architecting Your Web Store

- Servlet mapping versus request mapping
 - Web application context
 - View resolvers
- Overview of the Spring MVC request flow
 - The web application architecture
 - The Domain layer
 - The Persistence layer
 - The Service layer

DAY 2:Control Your Store with Controllers

- The role of a Controller in Spring MVC
 - Class-level request mapping
 - Default request mapping method
 - Handler mapping
 - Using URI template patterns
- Showing products based on category
 - Request path variable
 - Using matrix variables
- Showing products based on filters
 - Request parameters
 - Adding a product detail page
- Implementing a master detail View
- Adding multiple filters to list products

Working with Spring Tag Libraries

- JSP Standard Tag Library
- Serving and processing forms
 - Customizing data binding
- Whitelisting form fields for binding
 - Externalizing text messages

DAY 3:Working with View Resolver

- Resolving Views
 - RedirectView
 - Flash attribute
 - Serving static resources
- Adding images to the product detail page
 - Multipart requests in action
- Uploading product user manuals to the server
 - ContentNegotiatingViewResolver
 - HandlerExceptionResolver
 - Adding a ResponseStatus exception
 - Adding an exception handler

Internalize Your Store with Interceptor

- Working with interceptors
- LocaleChangeInterceptor – internationalization
 - Mapped interceptors
- Mapped intercepting offer page requests

Incorporating Spring Security

- Spring Security
- Authenticating users based on roles

DAY 4:Validate Your Products with a Validator

- Bean Validation
- Custom validation with JSR-303/Bean Validation
 - Adding custom validation to a category
 - Spring validation
 - Adding Spring validation
- Combining Spring validation and Bean Validation
- Adding Spring validation to a product image
 - **Give REST to Your Application with Ajax**
 - Introduction to REST
 - Implementing RESTful web services
 - Consuming REST web services
 - Handling web services in Ajax
 - **Float Your Application with Web Flow**
 - Spring Web Flow

- Implementing the order processing service
 - Implementing the checkout flow
 - Understanding flow definitions
 - Understanding checkout flow
 - Creating Views for every view state
 - Adding a decision state

DAY 5:Template with Tiles

- Enhancing reusability through Apache Tiles
 - Creating Views for every View state
 - **Testing Your Application**
 - Unit testing
 - Unit testing domain objects
- Integration testing with the Spring Test context framework
 - Testing product validator
 - Testing product Controllers
 - Testing REST Controllers
 - Adding tests for the remaining REST methods
 - **Using the Gradle Build Tool**
 - Installing Gradle
 - The Gradle build script for your project
 - Understanding the Gradle script

Duration and pricing

In [Price Group A](#)

Certificate

[Read about our certificates](#)

Bookings

You can download the course registration form on our home page or by clicking [here](#)

Brochure

You may download a pdf copy of this page by clicking on the pdf icon at the bottom of the page.

Questions

Please [email us](#)

Schedule

On the calendar below. If your browser doesn't display the calendar below, please click on [this link](#) or try using [Google Chrome](#), alternatively please enquire via our [Contact Us](#) page.

Ruby Programming

Be a productive Ruby programmer. With this Ruby programming course, you'll discover how Ruby takes care of all the details for you, so you can simply have fun and get more done with less code.

Intended Audience

- Newcomers / bootcamp apprentices who want to learn Ruby for the first time.
- Programmers who want to re-skill into Ruby:

Prerequisites

This is not a beginner to coding course. You **MUST** be at least at the [Intro To Programming](#) level before attempting this course

After this course you should be able to

- Know the fundamentals of Ruby
- Be a more productive programmer in Ruby
- Know the fundamentals of OO for those who have never done OO before

Course Contents

Day 1

Get Ruby

- Use Ruby
- Use Ruby–interactively
- Your first Ruby expressions
- Math operations and comparisons
- Strings
- Variables
- Everything is an object!
- Calling a method on an object
- Let's build a game
- Input, storage, and output
- Running scripts
- Comments

- “puts” and “print”
- Method arguments
- “gets”
- Parentheses are optional on method calls
- String interpolation
- What’s in that string?
- Inspecting objects with the “inspect” and “p” methods
- Escape sequences in strings
- Calling “chomp” on the string object
- What methods are available on an object?
- Generating a random number
- Converting to strings
- Ruby makes working with strings easy
- Converting strings to numbers
- Conditionals
- The opposite of “if” is “unless”
- Loops
- Let’s try running our game!
- Your Ruby Toolbox

Methods and Classes

- Defining methods
- Calling methods you’ve defined
- Method names
- Parameters
- Optional parameters
- Return values
- Returning from a method early
- Some messy methods
- Too many arguments
- Too many “if” statements
- Designing a class
- What’s the difference between a class and an object?
- Your first class
- Creating new instances (objects)
- Breaking up our giant methods into classes

- Creating instances of our new animal classes
- Updating our class diagram with instance methods
- Our objects don't "know" their names or ages!
- Too many arguments (again)
- Local variables live until the method ends
- Instance variables live as long as the instance does
- Encapsulation
- Attribute accessor methods
- Using accessor methods
- Attribute writers and readers
- Attribute writers and readers in action
- Ensuring data is valid with accessors
- Errors—the "emergency stop" button
- Using "raise" in our attribute writer methods
- Our complete Dog class
- Your Ruby Toolbox

Inheritance

- Copy, paste... Such a waste...
- Mike's code for the virtual test-drive classes
- Inheritance to the rescue!
- Defining a superclass (requires nothing special)
- Defining a subclass (is really easy)
- Adding methods to subclasses
- Subclasses keep inherited methods alongside new ones
- Instance variables belong to the object, not the class!
- Overriding methods
- Bringing our animal classes up to date with inheritance
- Designing the animal class hierarchy
- Code for the Animal class and its subclasses
- Overriding a method in the Animal subclasses
- We need to get at the overridden method!
- The "super" keyword
- A super-powered subclass
- Difficulties displaying Dogs
- The Object class

- Why everything inherits from the Object class
- Overriding the inherited method
- Your Ruby Toolbox

Day 2

Initializing Instances

- Payroll at Chargemore
- An Employee class
- Creating new Employee instances
- A division problem
- Division with Ruby's Fixnum class
- Division with Ruby's Float class
- Fixing the salary rounding error in Employee
- Formatting numbers for printing
- Format sequences
- Format sequence types
- Format sequence width
- Format sequence width with floating-point numbers
- Using "format" to fix our pay stubs
- When we forget to set an object's attributes...
- "nil" stands for nothing
- "/" is a method
- The "initialize" method
- Employee safety with "initialize"
- Arguments to "initialize"
- Using optional parameters with "initialize"
- "initialize" does an end-run around our validation
- "initialize" and validation
- Call other methods on the same instance with "self"
- When "self" is optional
- Implementing hourly employees through inheritance
- Restoring "initialize" methods
- Inheritance and "initialize"
- "super" and "initialize"

- Same class, same attribute values
- An inefficient factory method
- Class methods
- Our complete source code
- Your Ruby Toolbox

Arrays and Blocks

- Arrays
- Accessing arrays
- Arrays are objects, too!
- Looping over the items in an array
- The repeating loop
- Eliminating repetition...the WRONG way...
- Chunks of code?
- Blocks
- Defining a method that takes blocks
- Your first block
- Flow of control between a method and block
- Calling the same method with different blocks
- Calling a block multiple times
- Block parameters
- Using the “yield” keyword
- Block formats
- The “each” method
- The “each” method, step-by-step
- DRYing up our code with “each” and blocks
- Blocks and variable scope
- Using “each” with the “refund” method
- Using “each” with our last method
- Our complete invoicing methods
- We’ve gotten rid of the repetitive loop code!
- Utilities and appliances, blocks and methods
- Your Ruby Toolbox

Block Return Values

- A big collection of words to search through

- Opening the file
- Safely closing the file
- Safely closing the file, with a block
- Don't forget about variable scope!
- Finding array elements we want, with a block
- The verbose way to find array elements, using "each"
- Introducing a faster method...
- Blocks have a return value
- How the method uses a block return value
- Putting it all together
- A closer look at the block return values
- Eliminating elements we don't want, with a block
- The return values for "reject"
- Breaking a string into an array of words
- Finding the index of an array element
- Making one array that's based on another, the hard way
- Making one array based on another, using "map"
- Some additional logic in the "map" block body
- The finished product
- Your Ruby Toolbox

Day 3

Hashes

- Counting votes
- An array of arrays...is not ideal
- Hashes
- Hashes are objects
- Hashes return "nil" by default
- nil (and only nil) is "falsy"
- Returning something other than "nil" by default
- Normalizing hash keys
- Hashes and "each"
- A mess of method arguments
- Using hashes as method parameters
- Hash parameters in our Candidate class
- Leave off the braces!

- Leave out the arrows!
- Making the entire hash optional
- Typos in hash arguments are dangerous
- Keyword arguments
- Using keyword arguments with our Candidate class
- Required keyword arguments
- Your Ruby Toolbox

References

- Some confusing bugs
- The heap
- References
- When references go wrong
- Aliasing
- Fixing the astronomer's program
- Quickly identifying objects with "inspect"
- Problems with a hash default object
- We're actually modifying the hash default object!
- A more detailed look at hash default objects
- Back to the hash of planets and moons
- Our wish list for hash defaults
- Hash default blocks
- Hash default blocks: Assigning to the hash
- Hash default blocks: Block return value
- Hash default blocks: A shortcut
- The astronomer's hash: Our final code
- Using hash default objects safely
- Hash default object rule #1: Don't modify the default object
- Hash default object rule #2: Assign values to the hash
- The rule of thumb for hash defaults
- Your Ruby Toolbox

Mixins

- The media-sharing app
- The media-sharing app...using inheritance

- One of these classes is not (quite) like the others
- Option one: Make Photo a subclass of Clip
- Option two: Copy the methods you want into the Photo class
- Not an option: Multiple inheritance
- Using modules as mixins
- Mixins, behind the scenes
- Creating a mixin for comments
- Using our comments mixin
- A closer look at the revised “comments” method
- Why you shouldn’t add “initialize” to a mixin
- Mixins and method overriding
- Avoid using “initialize” methods in modules
- Using the Boolean “or” operator for assignment
- The conditional assignment operator
- Our complete code
- Your Ruby Toolbox

Day 4

Comparable and Enumerable

- Mixins built into Ruby
- A preview of the Comparable mixin
- Choice (of) beef
- Implementing a greater-than method on the Steak class
- Constants
- We have a lot more methods to define...
- The Comparable mixin
- The spaceship operator
- Implementing the spaceship operator on Steak
- Mixing Comparable into Steak
- How the Comparable methods work
- Our next mixin
- The Enumerable module
- A class to mix Enumerable into

- Mixing Enumerable into our class
- Inside the Enumerable module
- Your Ruby Toolbox

Documentation

- Learning how to learn more
- Ruby's core classes and modules
- Documentation
- HTML documentation
- Listing available classes and modules
- Looking up instance methods
- Instance methods denoted with # in the docs
- Instance method documentation
- Arguments in call signatures
- Blocks in call signatures
- Read the docs for the superclass and mixins, too!
- Read the docs for the superclass and mixins, too!
(continued)
- Looking up class methods
- Class method documentation
- Docs for a class that doesn't exist?!
- The Ruby standard library
- Looking up classes and modules in the standard library
- Where Ruby docs come from: rdoc
- What rdoc can deduce about your classes
- Adding your own documentation, with comments
- The "initialize" instance method appears as the "new" class method
- Your Ruby Toolbox

Exceptions

- Don't use method return values for error messages
- Using "raise" to report errors
- Using "raise" by itself creates new problems
- Exceptions: When something's gone wrong
- Rescue clauses: A chance to fix the problem

- Ruby's search for a rescue clause
- Using a rescue clause with our SmallOven class
- We need a description of the problem from its source
- Exception messages
- Our code so far...
- Different rescue logic for different exceptions
- Exception classes
- Specifying exception class for a rescue clause
- Multiple rescue clauses in one begin/end block
- Updating our oven code with custom exception classes
- Trying again after an error with "retry"
- Updating our oven code with "retry"
- Things you want to do no matter what
- The ensure clause
- Ensuring the oven gets turned off
- Your Ruby Toolbox

Day 5

Unit Testing

- Automated tests find your bugs before someone else does
- A program we should have had automated tests for
- Types of automated tests
- MiniTest: Ruby's standard unit-testing library
- Running a test
- Testing a class
- A closer look at the test code
- Red, green, refactor
- Tests for ListWithCommas
- Getting the test to pass
- Another bug to fix
- Test failure messages
- A better way to assert that two values are equal
- Some other assertion methods
- Removing duplicated code from your tests

- The “setup” method
- The “teardown” method
- Updating our code to use the “setup” method
- Your Ruby Toolbox

Web Apps

- Writing web apps in Ruby
- Our task list
- Project directory structure
- Browsers, requests, servers, and responses
- Sinatra takes requests
- Downloading and installing libraries with RubyGems
- Installing the Sinatra gem
- A simple Sinatra app
- Your computer is talking to itself
- Request type
- Resource path
- Sinatra routes
- Multiple routes in the same Sinatra app
- A route for the list of movies
- Making a movie list in HTML
- Accessing the HTML from Sinatra
- A class to hold our movie data
- Setting up a Movie object in the Sinatra app
- ERB embedding tags
- The ERB output embedding tag
- Embedding a movie title in our HTML
- Pool Puzzle
- Pool Puzzle Solution
- The regular embedding tag
- Looping over several movie titles in our HTML
- Letting users add data with HTML forms
- Getting an HTML form for adding a movie
- HTML tables
- Cleaning up our form with an HTML table
- There’s still more to do

- Your Ruby Toolbox
- **Saving and Loading Data: Keep It Around**
- Saving and retrieving form data
- Our browser can GET the form...
- ... But it needs to POST the response
- Setting the HTML form to send a POST request
- Setting up a Sinatra route for a POST request
- Converting objects to and from strings with YAML
- Saving objects to a file with `YAML::Store`
- Saving movies to a file with `YAML::Store`
- A system for finding Movies in the `YAML::Store`
- Numeric IDs for Movies
- Finding the next available movie ID
- A class to manage our `YAML::Store`
- Using our `MovieStore` class in the Sinatra app
- Testing the `MovieStore`
- Loading all movies from the `MovieStore`
- Loading all movies from the `MovieStore` (continued)
- Loading all movies in the Sinatra app
- Building HTML links to individual movies
- Building HTML links to individual movies (continued)
- Named parameters in Sinatra routes
- Using a named parameter to get a movie's ID
- Defining routes in order of priority
- Finding a `Movie` in the `YAML::Store`
- An ERB template for an individual movie
- Finishing the Sinatra route for individual movies
- Let's try it all out!
- Our complete app code
- Your Ruby Toolbox

Duration and pricing

- *Full-time* over 5 days (R10995)
- *Part-time* over 4 weeks (2 nights per week, 3 hour sessions) (R11995)
- *Part-time* over 8 Saturdays, 3 hour sessions (R11995)

- Please note : For *part-time* courses we do not have a fixed schedule and you will be placed on a waiting list until we get a group of 4+ together. Please book with no dates on the bookings form. This will automatically put you on the waiting list. We will confirm with you as soon as we have a part-time group together.
- [Distance-learning](#) over up to 3 months (R10995)
- International exams are not included in the course price.
- Prices exclude Vat for Vat-registered companies

Certificate

1. Upon completion of this course we will issue you with attendance certificate to certify your attendance and / or completion of the prescribed minimum examples.
2. You may sit for our competency assessment test and on passing you will obtain our competency certificate.
3. Our competency assessment can be booked and taken by someone who has not attended the course at a cost of R2950.

Bookings

You can download the course registration form on our home page or by clicking [here](#)

Brochure

You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

Questions

Please [email us](#)

Schedule

On the calendar below. If your browser doesn't display the calendar below, please click on [this link](#) or try using [Google Chrome](#), alternatively please enquire via our [Contact Us](#) page

Spring 5 Microservices

This course will help you implement the microservice architecture in Spring Framework, Spring Boot, and Spring Cloud. Using the latest specifications of Spring that focuses on Reactive Programming, you'll be able to build modern, internet-scale Java applications in no time. The course starts off with guidelines to implement responsive microservices at scale. You will understand how Spring Boot is used to deploy serverless autonomous services by removing the need to have a heavyweight application server.

You will also learn how to go further by deploying your microservices to Docker and managing them with Mesos. By the end of the book, you will have gained more clarity on the implementation of microservices using Spring Framework and will be able to use them in internet-scale deployments through real-world examples.

Intended Audience;

You should be familiar with [Spring MVC](#)

Day 1

Demystifying Microservices

- Evolution of microservices
- What are Microservices?
- Microservices – The honeycomb analogy
- Principles of microservices
- Characteristics of microservices
- Microservices examples
- Microservices benefits
- Summary

Related Architecture Styles and Use Cases

- Service-Oriented Architecture (SOA)
- Twelve-Factor Apps
- Serverless computing
- Lambda architecture
- DevOps, Cloud, and Containers
- Reactive microservices
- Microservice use cases
- Microservices early adopters – Is there a common theme?
- Microservice frameworks

Day 2

Building Microservices with Spring Boot

- Setting up a development environment
- Spring Boot for building RESTful microservices
- Getting started with Spring Boot
- Developing a Spring Boot microservice
- Developing our first Spring Boot microservice
- HATEOAS-enabled Spring Boot microservice
- Reactive Spring Boot microservices
- Implementing security
- Enabling cross origin for microservices interactions
- Spring Boot actuators for microservices instrumentation
- Documenting microservices
- Putting it all together – Developing a customer registration microservice example

Applying Microservices Concepts

- Microservice design guidelines

Microservices Capability Model

- Microservices capability model
- Core capabilities
- Infrastructure capabilities
- Supporting capabilities

- Process and governance capabilities
- Microservices maturity model
- Entry points for adoption

Day 3

Microservices Evolution – A Case Study

- Understanding the PSS application
- Death of the monolith
- Microservices to the rescue – a planned approach for migration
- Target implementation
- Potential next steps

Scale Microservices with Spring Cloud Components

- What is Spring Cloud?
- Spring Cloud releases
- Setting up the environment for the BrownField PSS
- Spring Cloud Config
- Eureka for registration and discovery
- Zuul proxy as the API Gateway
- Streams for reactive microservices
- Protecting microservices with Spring Cloud Security
- Summarising the BrownField PSS architecture

Day 4

Logging and Monitoring Microservices

- Understanding log management challenges
- Centralized logging solution
- Selection of logging solutions
- Monitoring microservices
- Data analysis using Data Lake

Containerizing Microservices with Docker

- Understanding gaps in the BrownField PSS microservices

- What are containers?
- Difference between VM and containers
- Benefits of containers
- Microservices and containers
- Introduction to Docker
- Deploying microservices into Docker
- Running RabbitMQ on Docker
- Using the Docker registry
- Microservices on Cloud
- Running BrownField services on EC2
- Future of containerization

Day 5

Scaling Dockerized Microservices with Mesos and Marathon

- Scaling microservices
- Container orchestration
- Container orchestration with Mesos and Marathon
- Implementing Mesos and Marathon with DCOS
- Implementing Mesos and Marathon for BrownField microservices
- Preparing BrownField PSS services

Microservice Development Life Cycle

- Practice points for microservice development
- Automating development cycle
- Summary

Duration and pricing

Pricing [Group A](#)

Certificate

- Upon completion of this course we will issue you with attendance certificate to certify your attendance.
- You may sit for our competency assessment test and on passing you will obtain our competency certificate.

- Our competency assessment can be booked and taken by someone who has not attended the course at a cost of R2950.

Bookings

You can download the course registration form on our home page or by clicking [here](#)

Brochure

You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

Questions

Please [email us](#)

Schedule

On the calendar below. If your browser doesn't display the calendar below, please click on [this link](#) or try using [Google Chrome](#), alternatively please enquire via our [Contact Us](#) page

Angular

Our **Angular Training Course** is intended for front-end developers who are familiar with Javascript and want to master Angular to move a gear up. Angular is the leading framework for building dynamic JavaScript applications that take advantage of the capabilities of modern browsers and devices. The course is updated for Angular4+ .

Prerequisites

[JavaScript](#)

Further training

[Node.JS](#)

[MEAN Stack](#)

[Ionic](#)

Intended Audience

Front end developers who found that JavaScript is inadequate to develop a rich, fully-fledged front-end. Our Angular Training Course starts with the MVC pattern and the many benefits that can be gained from separating your logic and presentation code. We then start from the nuts-and-bolts and building up to the most advanced and sophisticated features Angular, going in-depth to

Course material:

DAY 1

Getting Started

- Plunker
- Intro to TypeScript .
- Writing our first app.
- String Interpolation
- Looping
- Property & Event Binding
- Domain Model
- Nesting Components & Inputs .
- User Interaction & Outputs
- ES6 JavaScript & TypeScript Overview

DAY 2

Angular CLI

Components

- Architecting with Components
- Templates, Styles & View Encapsulation

- Content Projection
- Lifecycle Hooks
- ViewChildren & ContentChildren

Built-in Directives

- NgIf & NgSwitch.
- NgStyle & NgClass .
- NgNonBindable
- Structural Directives.

DAY 3

Custom Directives

- HostListener & HostBinding
- Inputs & Configuration

Reactive Programming with RxJS

- Streams & Reactive Programming
- Observables & RxJS .
- RxJS & Angular

Pipes

- Built-in Pipes
- Async Pipe .
- Custom Pipes

DAY 4

Forms

- Model Driven Forms
- Model Driven Form Validation
- Submitting & Resetting
- Reactive Model Form
- Template Driven Forms

Dependency Injection & Providers.

- Injectors
- Provider
- Tokens
- Configuring Dependency Injection in Angular
- NgModule.providers vs Component.providers vs Component.viewProviders

DAY 5

HTTP

- Core HTTP API
- HTTP Example with Promises .
- HTTP Example with Observables
- JSONP Example with Observables.

Routing

- Route Configuration . .
- Navigation.
- Parameterised Routes
- Nested Routes .
- Router Guards
- Routing Strategies

Unit Testing Overview

Duration and pricing

In [Pricing Group A](#)

Certificate

- Upon completion of this course we will issue you with attendance certificate to certify your attendance and / or completion of the prescribed minimum examples.
- You may sit for our competency assessment test and on passing you will obtain our competency certificate.
- Our competency assessment can be booked and taken by someone who has not attended the course at a cost of

R2950.

Bookings

You can download the course registration form on our home page or by clicking [here](#)

Brochure

You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

Questions

Please [email us](#)

Schedule

On the calendar below. If your browser doesn't display the calendar below, please click on [this link](#) or try using [Google Chrome](#), alternatively please enquire via our [Contact Us](#) page.

Mobile Web Development

Professional Mobile Web Development using mainly HTML5, CSS3 and JavaScript. Learn how to build lean and fast performing web mobile apps like native apps

Prerequisites

Before attempting Mobile Web Development Programming on its own, you should have previous web design and development experience and knowledge of HTML, CSS and basic Javascript.

Contents:

Day 1:

Responsive Web Design

- Get on the mobile bandwagon
- The recipe for Responsive Web Design
- An example of a responsively designed site
- CSS media queries
- Analyze the current CSS
- Steps to creating the mobile-specific CS

Responsible Responsiveness

- Mobile-first Responsive Web Design
- Can I get a proxy to set up my proxy?
- What to do when things aren't blazing fast
- Find the drags on page speed
- It looks mobile friendly, but it isn't
- What is progressive enhancement?
- Mobile-first media queries
- Add the map back using JavaScript
- Move iframe attributes to CSS equivalents
- Breakpoints to the rescue

Day 2:

A Separate Mobile Website

- Creature Comforts has agents in the field
- Getting to know user agents
- Straight talk: Most major sites have a separate mobile website
- When what you really want to do is (re-)direct
- Make a mobile mockup
- Not all phones are smartphones...not by a sight
- Let's keep it basic: Meet XHTML-MP
- Access keys in action
- Mobile-savvy CSS

What devices should we support?

- How do you know where to draw the line?

- Step away from the keyboard for a second
- Things you don't support vs. those you can't support
- Ask questions about your project
- Ingredients for your magic mobile potion
- Draw from your cupboard of tools and data

Day 3:

Device Databases and Classes

- A panic button for freaked-out students
- The button is for mobile phones only
- WURFL and its capabilities
- WURFL: Clever API code
- Steps for building our explore page
- Use WURFL to help differentiate content
- Make the page a bit smarter with WURFL
- The panic button: For phones only
- Expanding a lucrative part of AcedIt!'s business
- Get acquainted with the matching function
- Make something actually happen with device classes
- We need a bigger safety net

Build a Mobile Web App Using a Framework

- HTML5 is a specific thing...
- How "traditional" websites typically behave
- A Games Unlimited mobile HTML5 web app
- The master plan for phase 1 of the Game
- Why use mobile web app frameworks?
- Our choice for the Game: jQuery Mobile
- Build a basic page with jQuery Mobile
- Link to multiple pages with jQuery Mobile
- Make the Game feel more applike: to-dos
- Time to make that tartan-building form
- Build an HTML5 form
- Give jQuery Mobile hints about the fields

Day 4:

Mobile Web Apps in the Real World

- Mobile apps in the real world
- Make a better form
- A widget to manage the list of colors and sizes
- The two sides of generate.php
- Offline is important
- A basic recipe to create a cache manifest
- Dev tools to the rescue
- How to ask W3C-compliant browsers where they are
- Let's integrate geolocation

Build Hybrid Mobile Apps

- How do hybrid apps work?
- Bridge the web-native gap with PhoneGap
- Get acquainted with PhoneGap Build
- Keep track of discovered tartans
- Anatomy of the Tartan Hunt project
- What makes localStorage so special?
- Use a function to show which tartans are found
- Rope in PhoneGap to take pictures
- Now we're ready for the mediaCapture API

Day 5:

How to Be Future Friendly

- Time to dispel our collective illusions of control
- A future-friendly manifesto
- There are no silver bullets
- App today, web page tomorrow
- Remove PhoneGap references

Project

- Develop a mobile app from start to finish

Duration and pricing

Pricing [Group A](#)

Certificate

Please read about [our certificates](#).

Bookings

You can download the course registration form on our home page or by clicking [here](#)

Brochure

You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

Questions

Please [email us](#)

Schedule

On the calendar below. If your browser doesn't display the calendar below, please click on [this link](#) or try using [Google Chrome](#), alternatively please enquire via our [Contact Us](#) page

Python Machine Learning

Python Machine Learning Training Course

Prerequisites and Further Training

You should definitely be very well familiar with our [Python AdvancedTopics](#).

Intended Audience

One of the fastest growing areas in the industry at the moment is Data Science and Machine Learning. Join us now!

Further Training

Also have a look at our [Python Data Science Bootcamp](#) .

Course Material

Supplied in electronic format unless specified otherwise

Course Content

- Giving Computers the Ability to Learn from Data
- Building intelligent machines to transform data into knowledge
- The three different types of machine learning
- An introduction to the basic terminology and notations
- A roadmap for building machine learning systems
- Using Python for machine learning

Training Machine Learning Algorithms for Classification

- Artificial neurons – a brief glimpse into the early history of machine learning
- Implementing a perceptron learning algorithm in Python
- Adaptive linear neurons and the convergence of learning

A Tour of Machine Learning Classifiers Using Scikit-learn

- Choosing a classification algorithm
- First steps with scikit-learn
- Modeling class probabilities via logistic regression
- Maximum margin classification with support vector machines

- Solving nonlinear problems using a kernel SVM
- Decision tree learning
- K-nearest neighbors – a lazy learning algorithm

Building Good Training Sets – Data Preprocessing

- Dealing with missing data
- Handling categorical data
- Partitioning a dataset in training and test sets
- Bringing features onto the same scale
- Selecting meaningful features
- Assessing feature importance with random forests

Compressing Data via Dimensionality Reduction

- Unsupervised dimensionality reduction via principal component analysis
- Supervised data compression via linear discriminant analysis
- Using kernel principal component analysis for nonlinear mappings

Learning Best Practices for Model Evaluation and Hyperparameter Tuning

- Streamlining workflows with pipelines
- Using k-fold cross-validation to assess model performance
- Debugging algorithms with learning and validation curves
- Fine-tuning machine learning models via grid search
- Looking at different performance evaluation metrics

Combining Different Models for Ensemble Learning

- Learning with ensembles
- Implementing a simple majority vote classifier
- Evaluating and tuning the ensemble classifier
- Bagging – building an ensemble of classifiers from bootstrap samples
- Leveraging weak learners via adaptive boosting

Applying Machine Learning to Sentiment Analysis

- Obtaining the IMDb movie review dataset
- Introducing the bag-of-words model
- Training a logistic regression model for document classification
- Working with bigger data – online algorithms and out-of-core learning

9: Embedding a Machine Learning Model into a Web Application

- Serializing fitted scikit-learn estimators
- Setting up a SQLite database for data storage
- Developing a web application with Flask
- Turning the movie classifier into a web application
- Deploying the web application to a public server

Predicting Continuous Target Variables with Regression Analysis

- Introducing a simple linear regression model
- Exploring the Housing Dataset
- Implementing an ordinary least squares linear regression model
- Fitting a robust regression model using RANSAC
- Evaluating the performance of linear regression models
- Using regularized methods for regression
- Turning a linear regression model into a curve – polynomial regression

Working with Unlabeled Data – Clustering Analysis

- Grouping objects by similarity using k-means
- Organizing clusters as a hierarchical tree
- Locating regions of high density via DBSCAN

Training Artificial Neural Networks for Image Recognition

- Modeling complex functions with artificial neural networks
- Classifying handwritten digits
- Training an artificial neural network
- Developing your intuition for backpropagation
- Debugging neural networks with gradient checking
- Convergence in neural networks
- Other neural network architectures
- A few last words about neural network implementation

Parallelizing Neural Network Training with Theano

- Building, compiling, and running expressions with Theano

- Choosing activation functions for feedforward neural networks
- Training neural networks efficiently using Keras

Duration and pricing

- *Full-time* over 10 days (R19 995)
- *Part-time* over 4 weeks (2 nights per week, 3 hour sessions) (R11995)
- *Part-time* over 8 Saturdays, 3 hour sessions (R11995)
- Please note : For *part-time* courses we do not have a fixed schedule and you will be placed on a waiting list until we get a group of 4+ together. Please book with no dates on the bookings form. This will automatically put you on the waiting list. We will confirm with you as soon as we have a part-time group together.
- [Distance-learning](#) over up to 3 months (R9995)
- International exams are not included in the course price.
- Prices exclude Vat

Certificate

1. Upon completion of this course we will issue you with attendance certificate to certify your attendance and / or completion of the prescribed minimum examples.
2. You may sit for our competency assessment test and on passing you will obtain our competency certificate.
3. Our competency assessment can be booked and taken by someone who has not attended the course at a cost of R950.

Bookings

You can download the course registration form on our home page or by clicking [here](#)

Brochure

You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

Questions

Please [email us](#)

Schedule

On the calendar below. If your browser doesn't display the calendar below, please click on [this link](#) or try using [Google Chrome](#), alternatively please enquire via our [Contact Us](#) page.

We are a member of the Python Software Foundation



Python Advanced

Python Advanced Training Course

This course is not for novices in coding, we assume you know the non OO fundamentals of any coding language. A critical course in Python

Prerequisites and Further Training

You should definitely be very well familiar with our [Python Beginner](#) Topics.

Intended Audience

Whether you are new to programming or a professional developer, this course is designed to bring you up to speed on the Python language in ways that more limited approaches cannot.

Further Training

Also have a look at our [Python Data Science Bootcamp](#) .

Course Material

Supplied in electronic format unless specified otherwise

Course Content

DAY 1

Text Processing and Function Topics

- Text Manipulation
- First Class Functions
- Design Patterns with First Class Functions
- Decorators and Closures

DAY 2

00 Review and sequence operations

- Object References, mutability and recycling
 - Pythonic Objects
- Sequence operations (hacking, hashing, slicing)

DAY 3

Coroutines and concurrency

- Coroutines
- Concurrency
 - Asyncio
 - Tornado
 - Trolius

DAY 4

Metaprogramming

- Dynamic attributes and properties
 - Data wrangling with dynamic attributes
- Exploring JSON-like data with dynamic attributes
 - The invalid attribute name problem
 - Flexible object creation with `__new__`
 - Restructuring the OSCON feed with `shelve`
 - Linked record retrieval with `properties`
- Using a property for attribute validation
- `LineItem` take #1: class for an item in an
 - `LineItem` take #2: a validating property
 - A proper look at `properties`
 - `Properties` override instance attributes
 - Property documentation
 - Coding a property factory
 - Handling attribute deletion
- Essential attributes and functions for attribute handling
- Special attributes that affect attribute handling
 - Built-in functions for attribute handling
 - Special methods for attribute handling

Attribute descriptors

- Descriptor example: attribute validation
 - `LineItem` take #3: a simple descriptor
- `LineItem` take #4: automatic storage attribute names
 - `LineItem` take #5: a new descriptor type
- Overriding versus non-overriding descriptors
 - Overriding descriptor
 - Overriding descriptor without `__get__`
 - Non-overriding descriptor
 - Overwriting a descriptor in the class
 - Methods are descriptors
 - Descriptor usage tips

- 1. Use property to keep it simple
- 2. Read-only descriptors require `__set__`
- 3. Validation descriptors can work with `__set__` only
- 4. Caching can be done efficiently with `__get__` only
- 5. Non-special methods can be shadowed by instance attributes
- Descriptor docstring and overriding deletion

Class metaprogramming

- A class factory
- A class decorator for customizing descriptors
- What happens when: import time versus run time
 - The evaluation time exercises
 - Metaclasses
 - The metaclass evaluation time exercise
 - A metaclass for customizing descriptors
 - The metaclass `__prepare__` special method
 - Classes as objects

DAY 5

- Data Science Libraries Overview
- Practicals and Test

Duration and pricing

- *Full-time* over 10 days (R19 995)
- *Part-time* over 4 weeks (2 nights per week, 3 hour sessions) (R11995)
- *Part-time* over 8 Saturdays, 3 hour sessions (R11995)
- Please note : For *part-time* courses we do not have a fixed schedule and you will be placed on a waiting list until we get a group of 4+ together. Please book with no dates on the bookings form. This will automatically put you on the waiting list. We will confirm with you as soon as we have a part-time group together.
- [Distance-learning](#) over up to 3 months (R9995)

- International exams are not included in the course price.
- Prices exclude Vat

Certificate

1. Upon completion of this course we will issue you with attendance certificate to certify your attendance and / or completion of the prescribed minimum examples.
2. You may sit for our competency assessment test and on passing you will obtain our competency certificate.
3. Our competency assessment can be booked and taken by someone who has not attended the course at a cost of R950.

Bookings

You can download the course registration form on our home page or by clicking [here](#)

Brochure

You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

Questions

Please [email us](#)

Schedule

On the calendar below. If your browser doesn't display the calendar below, please click on [this link](#) or try using [Google Chrome](#), alternatively please enquire via our [Contact Us](#) page.

We are a member of the Python Software Foundation



Python Pandas

This Python Pandas training course will teach you all about Pandas, from installing it to creating one- and two-dimensional indexed data structures, indexing and slicing-and-dicing that data to derive results, loading data from local and Internet-based resources, and finally creating effective visualizations to form quick insights.

Prerequisites / Further Training

Recommended sequence	Summary
Introduction to Programming .	Fundamentals of Computer Programming
Beginner Python	Python Language and OO Fundamentals
<i>Python Pandas</i>	<i>Using Pandas Library</i>
Django or Flask	Python Web development in detail

Also have a look at our [Python Bootcamp](#)

Intended Audience

- Analysts who wants to more on data analysis and less on programming
- Programmers who wants to performing data exploration and analysis on Python using pandas

Overview

- pandas and why it is important
 - pandas and IPython Notebooks
- Referencing pandas in the application
 - Primary pandas objects

- The pandas Series object
- The pandas DataFrame object
- Loading data from files and the Web
 - Loading CSV data from files
 - Loading data from the Web
- Simplicity of visualization of pandas data

Installing pandas

- Getting Anaconda
- Installing Anaconda
 - Installing Anaconda on Linux
 - Installing Anaconda on Mac OS X
 - Installing Anaconda on Windows
 - Ensuring pandas is up to date
- Running a small pandas sample in IPython
 - Starting the IPython Notebook server
- Installing and running IPython Notebooks
 - Using Wakari for pandas

NumPy for pandas

- Installing and importing NumPy
- Benefits and characteristics of NumPy arrays
- Creating NumPy arrays and performing basic array operations
 - Selecting array elements
 - Logical operations on arrays
 - Slicing arrays
 - Reshaping arrays
 - Combining arrays
 - Splitting arrays
- Useful numerical methods of NumPy arrays

The pandas Series Object

- The Series object
- Importing pandas
- Creating Series

- Size, shape, uniqueness, and counts of values
- Peeking at data with heads, tails, and take
 - Looking up values in Serie
 - Alignment via index labels
 - Arithmetic operations
- The special case of Not-A-Number (NaN)
 - Boolean selection
 - Reindexing a Series
- Modifying a Series in-place
 - Slicing a Series

The pandas DataFrame Object

- Creating DataFrame from scratch
 - Example data
 - S&P 500
 - Monthly stock historical prices
- Selecting columns of a DataFrame
- Selecting rows and values of a DataFrame using the index
 - Slicing using the [] operator
- Selecting rows by index label and location: .loc[] and .iloc[]
- Selecting rows by index label and/or location: .ix[]
 - Scalar lookup by label or location using .at[] and .iat[]
- Selecting rows of a DataFrame by Boolean selection
 - Modifying the structure and content of DataFrame
 - Renaming columns
 - Adding and inserting columns
 - Replacing the contents of a column
 - Deleting columns in a DataFrame
 - Adding rows to a DataFrame
 - Appending rows with .append()
 - Concatenating DataFrame objects with pd.concat()
- Adding rows (and columns) via setting with enlargement
 - Removing rows from a DataFrame
 - Removing rows using .drop()

- Removing rows using Boolean selection
 - Removing rows using a slice
- Changing scalar values in a DataFrame
 - Arithmetic on a DataFrame
 - Resetting and reindexing
 - Hierarchical indexing
- Summarized data and descriptive statistics

Accessing Data

- Setting up the IPython notebook
 - CSV and Text/Tabular format
 - The sample CSV data set
 - Reading a CSV file into a DataFrame
- Specifying the index column when reading a CSV file
 - Data type inference and specification
 - Specifying column names
 - Specifying specific columns to load
 - Saving DataFrame to a CSV file
 - General field-delimited data
- Handling noise rows in field-delimited data
- Reading and writing data in an Excel format
 - Reading and writing JSON files
 - Reading HTML data from the Web
 - Reading and writing HDF5 format files
- Accessing data on the web and in the cloud
 - Reading and writing from/to SQL databases
 - Reading data from remote data services
- Reading stock data from Yahoo! and Google Finance
 - Retrieving data from Yahoo! Finance Options
- Reading economic data from the Federal Reserve Bank of St. Louis
 - Accessing Kenneth French's data
 - Reading from the World Bank

Tidying Up Your Data

- What is tidying your data?

- Setting up the IPython notebook
 - Working with missing data
- Determining NaN values in Series and DataFrame objects
 - Selecting out or dropping missing data
- How pandas handles NaN values in mathematical operations
 - Filling in missing data
 - Forward and backward filling of missing values
 - Filling using index labels
 - Interpolation of missing values
 - Handling duplicate data
 - Transforming Data
 - Mapping
 - Replacing values
- Applying functions to transform data

Combining and Reshaping Data

- Setting up the IPython notebook
 - Concatenating data
 - Merging and joining data
 - An overview of merges
- Specifying the join semantics of a merge operation
 - Pivoting
 - Stacking and unstacking
- Stacking using nonhierarchical indexes
- Unstacking using hierarchical indexes
 - Melting
- Performance benefits of stacked data

Grouping and Aggregating Data

- Setting up the IPython notebook
- The split, apply, and combine (SAC) pattern
 - Split
- Grouping by a single column's values
 - Accessing the results of grouping
 - Grouping using index levels
 - Apply

- Applying aggregation functions to groups
 - The transformation of group data
 - An overview of transformation
 - Practical examples of transformation
 - Filtering groups
 - Discretization and Binning

Time-series Data

- Setting up the IPython notebook
- Representation of dates, time, and intervals
 - The datetime, day, and time objects
 - Timestamp objects
 - Timedelta
 - Introducing time-series data
 - DatetimeIndex
- Creating time-series data with specific frequencies
 - Calculating new dates using offsets
 - Date offsets
 - Anchored offsets
- Representing durations of time using Period objects
 - The Period object
 - PeriodIndex
 - Handling holidays using calendars
- Normalizing timestamps using time zones
 - Shifting and lagging
 - Frequency conversion
 - Up and down resampling
- Time-series moving-window operations

Visualization

- Setting up the IPython notebook
 - Plotting basics with pandas
- Creating time-series charts with `.plot()`
- Adorning and styling your time-series plot
 - Adding a title and changing axes labels
- Specifying the legend content and position

- Specifying line colors, styles, thickness, and markers
 - Specifying tick mark locations and tick labels
- Formatting axes tick date labels using formatters
 - Common plots used in statistical analyses
 - Bar plots
 - Histograms
 - Box and whisker charts
 - Area plots
 - Scatter plots
 - Density plot
 - The scatter plot matrix
 - Heatmaps
- Multiple plots in a single chart

Applications to Finance

- Setting up the IPython notebook
- Obtaining and organizing stock data from Yahoo!
 - Plotting time-series prices
 - Plotting volume-series data
- Calculating the simple daily percentage change
 - Calculating simple daily cumulative returns
- Resampling data from daily to monthly returns
 - Analyzing distribution of returns
 - Performing a moving-average calculation
- The comparison of average daily returns across stocks
- The correlation of stocks based on the daily percentage
 - change of the closing price
 - Volatility calculation
- Determining risk relative to expected returns

—————

Duration and pricing

In [Price Group A](#)

Certificate

1. Upon completion of this course we will issue you with attendance certificate to certify your attendance and / or completion of the prescribed minimum examples.
2. You have the option to get the competency / academic certificate if you :
hand in a project (pre-approved) covering most of the topics in the book.
3. If you have not enrolled for the course you may opt to sit only for the competency certificate, at a cost of R3500.

Schedule

On the calendar on this page below.

If your browser doesn't display the calendar below, please click on [this link](#) or try using [Google Chrome](#), alternatively please enquire via our '[Contact Us](#)' page.

Bookings

Please click [click here](#) or send us an [email](#).

Questions

Please [email us](#)

Ionic (for Android & iOS)

This Ionic training course will teach you how to create hybrid mobile applications by combining the capabilities of Ionic, Cordova, and AngularJS. Reduce the time to market your application using Ionic, that helps in rapid application

development.

Prerequisites

[HTML5 / CSS3](#), [AngularJS](#) recommended. There is time for a small re-cap of Angular at the beginning of the course, but not enough time to master it.

Course Contents:

Ionic – Powered by AngularJS

- Understanding the separation of concerns
 - AngularJS components
 - AngularJS directives
 - AngularJS services
 - AngularJS resources

Welcome to Ionic

- Mobile Hybrid Architecture
 - What is Apache Cordova?
 - What is Ionic?
 - Software setup
 - Install Node.js
 - Install Git
 - Install Bower
 - Install Gulp
 - Install Sublime Text
 - Install Cordova and Ionic CLI
 - The platform guide
 - Hello Ionic
- The browser developer tools setup
 - Google Chrome
 - Mozilla Firefox
- The Ionic project structure
 - The config.xml file
 - The www folder
- Scaffolding the tabs template

- Scaffolding the side menu template
 - generator-ionic
- Installing generator-ionic

Ionic CSS Components and Navigation

- Ionic CSS components
- The Ionic grid system
 - The page structure
 - Buttons
 - Lists
 - Cards
 - Ionicons
 - Form elements
- Integrating Ionic CSS components with AngularJS
 - The Ionic router
 - A simple two-page app

Ionic and SCSS

- What is Sass?
- Setting up SCSS in our Ionic project
 - The manual setup
 - The Ionic CLI task
- Working with Ionic SCSS
 - Basic swatch
- Understanding the Ionic SCSS setup
 - Using variables and mixins
 - The SCSS workflow
 - Building a swatch

Ionic Directives and Services

- Ionic directives and services
 - The Ionic Platform service
 - registerBackButtonAction
 - The on method
 - Headers and footers
 - Content

- ion-content
- ion-scroll
- ion-refresher
- ion-infinite-scroll
- \$ionicScrollDelegate
 - Navigation
 - ion-view
- Ionic view events
 - ion-nav-bar
 - ion-nav-buttons
- \$ionicNavBarDelegate
 - \$ionicHistory
- Tabs and side menu
 - Ionic loading
- The Action Sheet service
- Popover and Popup services
 - \$ionicPopup
- The ion-list and ion-item directives
 - Gesture directives and services
 - Utilities

Building a Bookstore App

- An introduction to the Bookstore application
 - The Bookstore architecture
 - The server architecture
 - The server-side API documentation
 - The client architecture
 - Code on GitHub
 - A Bookstore demo
 - The development flow
 - Setting up the server
 - Building the application
- Step 1 – Scaffolding the side menu template
 - Step 2 – Refactoring the template
 - Refactoring the menu
 - Refactoring the module name

- Adding a run method and modifying routes
 - Refactoring templates
- Step 3 – Building authentication, localStorage, and the
 - REST API factory
 - The Ionic loading factory
 - The localStorage factory
 - The Authentication factory
 - The REST API factory

Creating controllers for each route and integrating with the factory

- The application controller
 - The browse controller
 - The book controller
 - The cart controller
 - The purchase controller
- Step 5 – Creating templates and integrating with the controller data
 - The Login template
 - The Browse template
 - The Book template
 - The Cart template
 - The Purchase template

Cordova and ngCordova

- Setting up a platform-specific SDK
 - The Android setup
 - The iOS setup
 - Testing the setup
 - Testing for Android
 - Testing for iOS
- Getting started with Cordova plugins
 - The Ionic plugin API
 - Add a plugin
 - Remove a plugin
 - List added plugins

- Search plugins
- The Cordova whitelist plugin
 - ngCordova
- Setting up ngCordova
 - Legend
 - \$cordovaToast
 - \$cordovaDialogs
 - \$cordovaFlashlight
- \$cordovaLocalNotification
- \$cordovaGeolocation

Building a Messaging App

- The Ionic Chat app
 - Firebase
- Setting up a Firebase account
 - AngularFire
- The application architecture
 - Authentication
 - The application flow
 - Previewing the app
 - Data structure
 - Cordova plugins
 - Code on GitHub
 - Developing the application
 - Scaffolding and setting up the app
- Installing the required Cordova plugins
 - Getting the Google API key
- Setting up routes and route authentication
 - Setting up services/factories
 - Setting up a map directive
 - Setting up controllers
 - Setting up templates
 - Setting up SCSS
 - Testing the application

Releasing the Ionic App

- Preparing the app for distribution
- Setting up icons and splash screens
 - Updating the config.xml file
 - The PhoneGap service
- Generating installers using the Cordova CLI
 - Android installer
 - iOS installer
 - The Ionic package
- Uploading the project to Ionic cloud
 - Generating the required keys

Duration and pricing

In [Pricing Group A](#)

Certificate

1. Upon completion of this course we will issue you with attendance certificate to certify your attendance.
2. You may sit for our competency assessment test and on passing you will obtain our competency certificate.
3. Our competency assessment can be booked and taken by someone who has not attended the course at a cost of R3500.

Bookings

You can download the course registration form on our home page or by clicking [here](#)

Brochure

You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

Questions

Please [email us](#)

Schedule

On the calendar below. If your browser doesn't display the calendar below, please click on [this link](#) or try using [Google Chrome](#), alternatively please enquire via our [Contact Us](#) page.

C++ Advanced (with QT)

This Advanced C++ with QT Training course will teach you how to create hybrid desktop/Internet applications, threading, rich text handling, Qt's graphics/view architecture, and Qt's model/view architecture and many other topics are covered. Besides each chapter's main subject, lots of other classes, methods, and techniques are used wherever they make sense, to show as many Qt features as possible.

Prerequisites

You should be solidly at the [C++ Beginner](#) level before doing this course.

Further Training

[HTML5 / CSS3 Responsive Web Design](#) (if you want to learn more about creating responsive front-ends).

How about JavaScript, Angular, Ionic ?

Course Material

Included

Course Contents

Hybrid Desktop/Internet Applications

- Internet-Aware Widgets
 - Using WebKit
- A Generic Web Browser Window Component
- Creating Web Site-Specific Applications
 - Embedding Qt Widgets in Web Pages

Audio and Video

- Using QSound and QMovie
- The Phonon Multimedia Framework
 - Playing Music
 - Playing Videos

Model/View Table Models

- Qt's Model/View Architecture
 - Using QStandardItemModels for Tables
- Changing a Table Model through the User Interface
 - A QStandardItemModel Subclass for Tables
- A QSortFilterProxyModel to Filter Out Duplicate Rows
 - A QSortFilterProxyModel to Filter In Wanted Rows
 - Creating Custom Table Models
- Changing a Table Model through the User Interface
 - A Custom QAbstractTableModel Subclass for Tables
 - The QAbstractItemModel API Methods for Tables
- Methods to Support Saving and Loading Table Items

Model/View Tree Models

- Using QStandardItemModels for Trees
- Changing a Tree Model through the User Interface
 - A QStandardItem Subclass for Tree Items
 - A QStandardItemModel Subclass for Trees
 - Creating Custom Tree Models
- Changing a Tree Model through the User Interface
 - A Custom Item Class for Tree Items
- A Custom QAbstractItemModel Subclass for Trees

- The QAbstractItemModel API for Trees
- The QAbstractItemModel API for Drag and Drop
 - Methods for Saving and Loading Tree Items

Model/View Delegates

- Datatype-Specific Editors
- Datatype-Specific Delegates
- A Read-Only Column or Row Delegate
- An Editable Column or Row Delegate
 - Model-Specific Delegates

Model/View Views

- QAbstractItemView Subclasses
- Model-Specific Visualizing Views
 - The Visualizer Widget
- The Visualizer's Aggregated Header Widget
- The Visualizer's Aggregated View Widget

Threading with QtConcurrent

- Executing Functions in Threads
 - Using QtConcurrent::run()
 - Using QRunnable
- Filtering and Mapping in Threads
 - Using QtConcurrent to Filter
- Using QtConcurrent to Filter and Reduce
 - Using QtConcurrent to Map

Threading with QThread

- Processing Independent Items
 - Processing Shared Items

Creating Rich Text Editors

- Introducing QTextDocument
- Creating Custom Text Editors
- Completion for Line Edits and Comboboxes

- Completion and Syntax Highlighting for Text Editors
 - Completion for Multi-line Editors
 - Syntax Highlighting
 - A Rich Text Single Line Editor
 - Multi-line Rich Text Editing

Creating Rich Text Documents

- Exported QTextDocument File Quality
 - Creating QTextDocuments
 - Creating QTextDocuments with HTML
- Creating QTextDocuments with QTextCursor
 - Exporting and Printing Documents
 - Exporting QTextDocuments
- Exporting in PDF and PostScript Format
 - Exporting in Open Document Format
 - Exporting in HTML Format
 - Exporting in SVG Format
 - Exporting in Pixmap Formats
- Printing and Previewing QTextDocuments
 - Painting Pages
 - Painting PDF or PostScript
 - Painting SVG
 - Painting Pixmap

Creating Graphics/View Windows

- The Graphics/View Architecture
- Graphics/View Widgets and Layouts
 - Introducing Graphics Items

Creating Graphics/View Scenes

- Scenes, Items, and Actions
 - Creating the Main Window
- Saving, Loading, Printing, and Exporting Scenes
 - Saving Scenes
 - Loading Scenes
- Printing and Exporting Scenes

- Manipulating Graphics Items
 - Adding Items
- Copying, Cutting, and Pasting Items
 - Manipulating Selected Items
- Showing and Hiding the Guideline Grid
- Keeping the User Interface Up to Date
 - Enhancing QGraphicsView
 - Creating a Dock Widget Toolbox
 - Creating Custom Graphics Items
 - Enhancing QGraphicsTextItem
 - Graphics Item Transformations
 - Enhancing an Existing Graphics Item
- Creating a Custom Graphics Item from Scratch

The Animation and State Machine Frameworks

- Introducing the Animation Framework
- Introducing the State Machine Framework
- Combining Animations and State Machines

PROJECT

—

Duration and pricing

In [Price Group A](#)

Certificate

1. Upon completion of this course we will issue you with attendance certificate to certify your attendance and / or completion of the prescribed minimum examples.
2. You may sit for our competency assessment test and on passing you will obtain our competency certificate.
3. Our competency assessment can be booked and taken by someone who has not attended the course at a cost of R2950.

Bookings

You can download the course registration form on our home page or by clicking [here](#)

Brochure

You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

Questions

Please [email us](#)

Schedule

On the calendar below. If your browser doesn't display the calendar below, please click on [this link](#) or try using [Google Chrome](#), alternatively please enquire via our [Contact Us](#) page.

Spring Boot

This **Spring MVC Training Course** will teach you all the exciting features of this successful Framework by building a complete online shopping web application

Prerequisites

You should be at the level of our [Advanced Java](#) course . This course teaches you the latest tools to use in developing robust enterprise-wide Java Web Applications

Further Training

[Spring Boot, JEE](#) related courses

Contents

Spring Boot Intro

- Creating Your First Spring Boot Application
 - Spring Initializer
 - Build Tools Overview
 - Maven Build
 - Gradle Build
 - Starter POMs
 - Creating Executable JARs
 - Spring DevTools & Live Reload
- Exercise : Building Your First Real World Application

Spring Boot Fundamentals

- Spring Boot Fundamentals Introduction
 - Spring Beans & Dependency Injection
- Application Properties & YAML Configuration
 - Configuration Properties
 - Profiles
 - Auto Configuration
- Spring Core Fundamentals Exercise

Developing Web Applications

- Spring MVC Intro
- Web Application Demo
 - Static Content
 - Bower
 - Template Engines
 - Thymeleaf Demo
 - GSP Demo
- Static Content & The View Layer
 - Internationalization (i18n)
 - Error Handling
 - Exception Handling
- Exercise : Creating a web application

Data Access with Spring Boot

- Data Access Introduction
- Creating an Application using H2's in-memory database

- Using H2's Web Console
- Creating Entities & Repositories
 - Loading Data
 - Refactor
 - Refactor Blog Part 2
 - Production Database
 - Spring Data JDBC
 - NoSQL – MongoDB
- Exercise : Data Access with JPA
- Spring Data JPA – Creating Queries

Security

- Security Introduction
- Spring Security Intro
- Spring Security – Custom Configuration
 - Spring Security Basics
- Spring Security Custom Login Form
 - Spring Security & JPA
 - Securing you application

Building REST APIs

- REST API Intro
- HTTP Request Methods (Verbs)
 - HTTP Status Codes
 - Content Negotiation
 - HTTP Clients
 - Building A REST API
 - REST Components
 - Error Handling
 - Hypermedia Demo
 - REST Template
 - Building a REST API

CRUD

- Read
- Create

- Validation
 - Update
 - Delete
- [Exercise] Building a CRUD Application

Guides

- Intro
 - Sending Email
- Scheduling Tasks
 - Project Lombok

Spring Boot Intro

- Creating Your First Spring Boot Application
 - Spring Initializer
 - Build Tools Overview
 - Maven Build
 - Gradle Build
 - Starter POMs
 - Creating Executable JARs
 - Spring DevTools & Live Reload
- Exercise : Building Your First Real World Application

Spring Boot Fundamentals

- Spring Boot Fundamentals Introduction
 - Spring Beans & Dependency Injection
- Application Properties & YAML Configuration
 - Configuration Properties
 - Profiles
 - Auto Configuration
- Spring Core Fundamentals Exercise

Developing Web Applications

- Spring MVC Intro
- Web Application Demo
 - Static Content
 - Bower

- Template Engines
 - Thymeleaf Demo
 - GSP Demo
- Static Content & The View Layer
 - Internationalization (i18n)
 - Error Handling
 - Exception Handling
- Exercise : Creating a web application

Data Access with Spring Boot

- Data Access Introduction
- Creating an Application using H2's in-memory database
 - Using H2's Web Console
- Creating Entities & Repositories
 - Loading Data
 - Refactor
 - Refactor Blog Part 2
 - Production Database
 - Spring Data JDBC
 - NoSQL – MongoDB
- Exercise : Data Access with JPA
- Spring Data JPA – Creating Queries

Security

- Security Introduction
- Spring Security Intro
- Spring Security – Custom Configuration
 - Spring Security Basics
- Spring Security Custom Login Form
 - Spring Security & JPA
 - Securing you application

Building REST APIs

- REST API Intro
- HTTP Request Methods (Verbs)
 - HTTP Status Codes

- Content Negotiation
 - HTTP Clients
- Building A REST API
 - REST Components
 - Error Handling
 - Hypermedia Demo
 - REST Template
- Building a REST API

CRUD

- Read
 - Create
 - Validation
 - Update
 - Delete
- [Exercise] Building a CRUD Application

Guides

- Intro
 - Sending Email
 - Scheduling Tasks
 - Project Lombok

Spring Boot Auto-Configuration, Features, and More

- Auto-Configuration
 - Disabling a Specific Auto-Configuration
- @EnableAutoConfiguration and @Enable<Technology> Annotations
 - Spring Boot Features
 - SpringApplication Class
 - SpringApplicationBuilder
 - Application Arguments
 - ApplicationRunner and CommandLineRunner
 - Application Configuration
 - Configuration Properties Examples
 - Custom Properties Prefix

Spring Boot CLI

- Spring Boot CLI
- The run Command
- The test Command
- The grab Command
- The jar Command
- The war Command
- The install Command
- The uninstall Command
 - The init Command
 - The shell Command
 - The help Command

Spring with Spring Boot

- Spring Web Applications
 - J2EE Web Applications
 - Spring MVC Applications
- Spring Boot Web Applications
- Using Spring with Spring Boot
 - XML with Spring Boot
 - Groovy Beans in Spring Boot
- Standalone Spring Apps vs. Spring Boot Apps
 - Using Spring Technologies in Spring Boot

Testing with Spring Boot

- Testing Spring Boot
 - Web Testing

Data Access with Spring Boot

- SQL Databases
- Data Access Using the JDBC Template with Spring Boot
 - Data Access Using JPA with Spring Boot
 - NoSQL Databases

Web Development with Spring Boot

- Spring MVC
- Spring Boot Web Applications
- Playing with the HAL Browser

Security with Spring Boot

- Simple Security for Spring Boot
- Security Using the application.properties File
 - In-Memory Security
 - Security Using a Database
 - Securing Resources
 - Spring Boot with OAuth2

Messaging with Spring Boot

- What Is Messaging?
 - JMS with Spring Boot
 - A Simpler JMS Consumer
- Connect to Remote JMS Server
 - RabbitMQ with Spring Boot
 - Installing RabbitMQ
- RabbitMQ/AMQP: Exchanges, Bindings, and Queues
 - Remote RabbitMQ
- Redis Messaging with Spring Boot
 - Installing Redis
 - Remote Redis
- WebSockets with Spring Boot

Spring Boot Actuator

- Spring Boot Actuator
 - /actuator
 - /autoconfig
 - /beans
 - /configprops
 - /docs
 - /dump
 - /env
 - /flyway

- /health
- /info
- /liquibase
- /logfile
- /metrics
- /mappings
- /shutdown
- /trace
- Sensitive Endpoints
- Changing the Endpoint ID
- Actuator CORS Support
- Changing the Management Endpoints Path
- Using Spring Boot Actuator in a Non-Web Application

Deploying Spring Boot

- Setting Up the Spring Boot Journal App
- Creating the SSL Self-Signed Keystore
 - Testing SSL
 - Creating Executable JARs
 - The Java Way
 - The Spring Boot Way
- Creating Executable and Deployable WARs
 - Deploying to a Tomcat-Based Server
 - Activating Profiles
- Creating Spring Boot Apps as a Service
 - Spring Boot Apps as Windows Service
 - Spring Boot with Docker

Spring Boot in the Cloud

- The Cloud and Cloud-Native Architectures
 - Twelve-Factor Applications
 - Microservices
- Preparing the Spring Boot Journal App as Microservice
 - Cloud Foundry
 - Pivotal Cloud Foundry Features
- Cloud Foundry CLI – Command Line Interface

- Development Environment – PCFDev
 - Pivotal Cloud Foundry
- Deploying to Pivotal Web Services

Extending Spring Boot Apps

- Custom Spring Boot Module
- The spring-boot-journal Project
- The journal-spring-boot-starter Project
- The journal-spring-boot-autoconfigure Project
- Package and Install the Journal Project
 - The spring-boot-calendar Project
 - Custom Health Indicator

Duration and pricing

In [Price Group A](#)

Certificate

[Read about our certificates](#)

Bookings

You can download the course registration form on our home page or by clicking [here](#)

Brochure

You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

Questions

Please [email us](#)

Schedule

On the calendar below. If your browser doesn't display the calendar below, please click on [this link](#) or try using [Google Chrome](#), alternatively please enquire via our [Contact Us](#) page.

Spring Boot

This Spring Developer Training course will help you implement the microservices architecture in Spring Framework, Spring Boot, and Spring Cloud. Using the latest specifications of Spring, you'll be able to build modern, Internet-scale Java applications in no time.

Prerequisites

[Spring Hibernate](#) [Spring Enterprise](#)

Further training

[JEE](#) Courses if you have not done them yet.

Day 1 – Quick Start with Groovy

- Getting started
 - Installing Spring Boot's CLI
 - Testing with Spring Boot's CLI
- Bundling and deploying a Spring Boot application
 - Adding support for templates
 - Modernizing our app with JavaScript
 - Using Bower instead of WebJars
- What about an app that is all frontend with no backend?
 - Adding production-ready support features
 - Pinging our app for general health
 - Gathering metrics
 - Detailed management with CRaSH

Day 2. Quick Start with Java

- Creating an empty project with start.spring.io
 - Spring Boot starters
 - Running a Spring Boot application
 - Adding Spring Social GitHub
 - Digging into GitHub issues
 - Creating a GitHub access token
 - Delving into Spring Boot's property support
 - Adding server-side mobile support with Spring Mobile
 - Creating a mobile frontend with jQuery Mobile
 - Bundling up the application as a runnable JAR
 - Deploying to Cloud Foundry
 - Adding production-ready support

Day 3. Debugging and Managing Your App

- Creating a JMS-based publisher/subscriber app
- Using Spring Boot's auto-configuration report
 - Auto-configuring ActiveMQ
 - Making a change and debugging the results
- Overriding Boot with alternate beans or properties
- Writing a custom health check to ping ActiveMQ
 - Adding customized app data to /info
- Creating custom metrics to track the message traffic
- Tweaking management ports, address, and paths
 - Restricting access only to JMX
- Connecting to the app via JConsole and jmxterm
 - Creating custom CRaSH commands

Day 4 – Data Access with Spring Boot

- Creating an app using H2's in-memory database
 - Defining entities and repositories
 - Loading data using a SQL script

- Loading data programmatically
- Adding a production profile for a MySQL database
 - Adding Spring Data REST and using it to manage teammates
- Reconfiguring our app to use Spring Data MongoDB
 - Running our MongoDB-based app

Day 5 – Securing Your App with Spring Boot

- Getting started
 - Defining our domain
 - Loading the test data
- Creating a server-side controller
 - Crafting our HTML templates
- Running our unsecured application
 - Securing our app
- Navigating with basic authentication
- Enhancing the security model of our app
 - Configuring user data to persist
 - Configuring embedded Tomcat to use SSL
- Spring Security's default web-level protections
 - Navigating our fully secured app

Project:

- Setting up Git and GitHub
 - The Git Workflows
- Setting up Java and Maven
 - Setting up Spring STS
- Setting up IntelliJ Ultimate
 - Setting up an AWS account
 - Setting up IAM
 - Setting up a Stripe account
- Introducing the Project overall architecture
 - Setting up the folder structure
 - Setting up Bootstrap and JQuery as Maven dependencies
 - Creating the index placeholders

- Customizing the index page
- Creating reusable HTML component with Thymeleaf fragments
- Setting up internationalisation with Spring Boot
 - Setting up logging with Logback
 - Creating the About Us page with Bootstrap
- Creating the Contact Us page and Spring Boot Email – Part 1
- Creating the Contact Us page and Spring Boot Email – Part 2
 - Creating the login form
 - Creating the payload page
 - Spring In Memory Security
- The Security Authentication database model
 - Introduction to Spring Data JPA
 - Creating JPA repositories
 - Creating the Service Layer
- Integrating Spring Data JPA with Spring Security User Details
 - Encrypting plain text with BCrypt
- Spring Data JPA – How to create unique indexes
 - Externalising the configuration
- The Forgot My Password Overview and Architecture
 - Creating the Repository Layer
 - Preparing the Service Layer
 - Updating the View Layer – Part 1
 - Updating the View Layer – Part 2
- FormValidation-UI. An excellent framework for Bootstrap form validation
 - Introduction to the User Creation Journey
 - Creating the Front End domain model
 - Creating the Basic Subscription Flow

Duration and pricing

- *Full-time* over 5 days (R9995)
- *Part-time* over 4 weeks (2 nights per week, 3

hour sessions) (R11995)

- *Part-time* over 8 Saturdays, 3 hour sessions (R11995)
 - Please note : For *part-time* courses we do not have a fixed schedule and you will be placed on a waiting list until we get a group of 4+ together. Please book with no dates on the bookings form. This will automatically put you on the waiting list. We will confirm with you as soon as we have a part-time group together.
 - [Distance-learning](#) over up to 3 months (R8995)
- International exams are not included in the course price.
 - Prices exclude Vat for Vat-registered companies

Certificate

1. Upon completion of this course we will issue you with attendance certificate to certify your attendance and / or completion of the prescribed minimum examples.
2. You may sit for our competency assessment test and on passing you will obtain our competency certificate.
3. Our competency assessment can be booked and taken by someone who has not attended the course at a cost of R2950.

Bookings

You can download the course registration form on our home page or by clicking [here](#)

Brochure

You may download a pdf copy of this page by clicking on the

pdf icon at the top of the page.

Questions

Please [email us](#)

Schedule

On the calendar below. If your browser doesn't display the calendar below, please click on [this link](#) or try using [Google Chrome](#), alternatively please enquire via our [Contact Us](#) page.