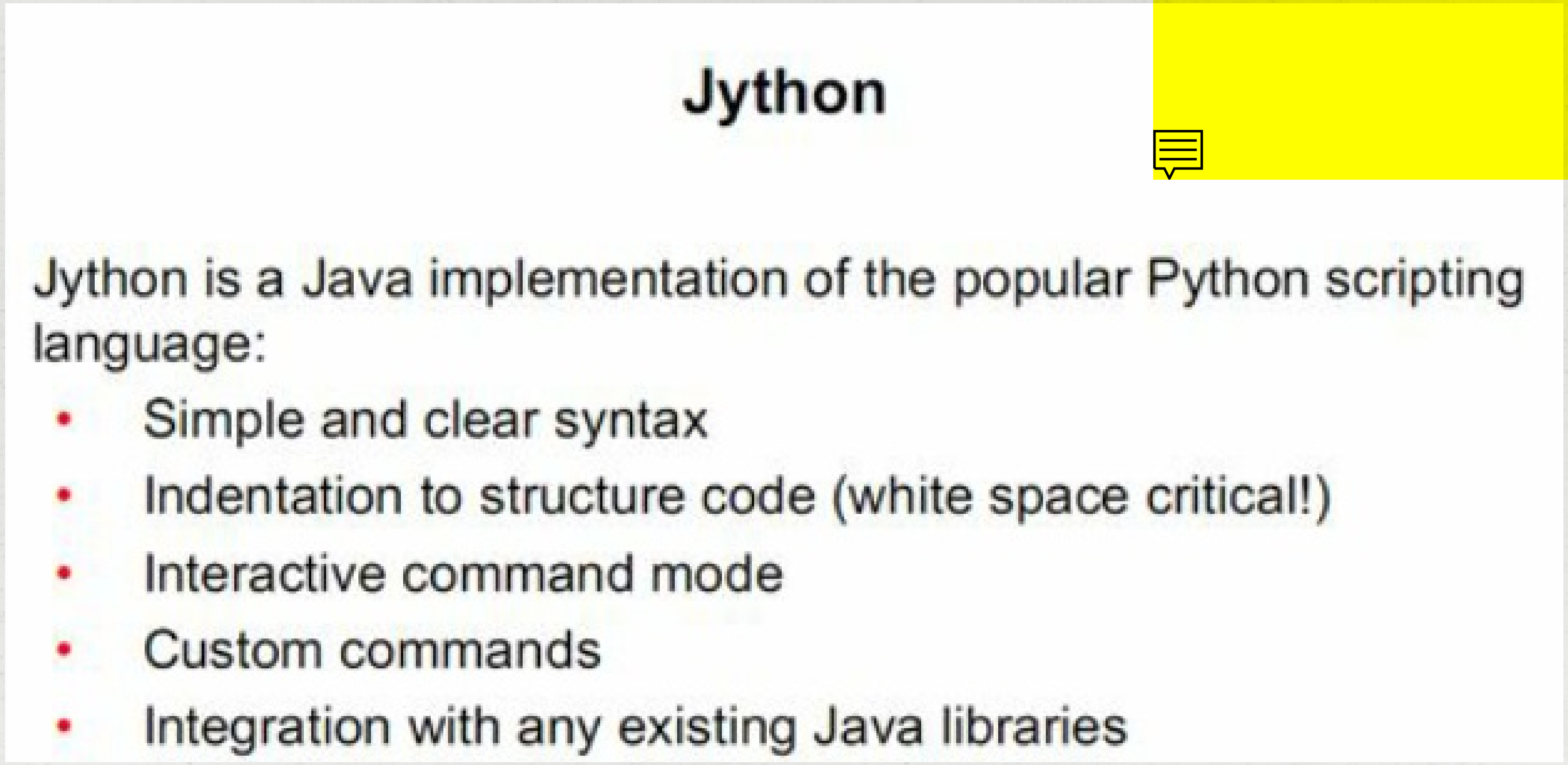
WEBLOGIC SCRIPTING TOOL (WLST)

S & S Technical Training Institute

WebLogic Scripting Tool (WLST)

- Scripting tool for administering a domain (create, configure, manage, monitor, deploy applications)
- Based on Jython - pure Java implementation of Python
- Great for automating repetitive tasks
- Heavily used by customers and within Oracle

Jython

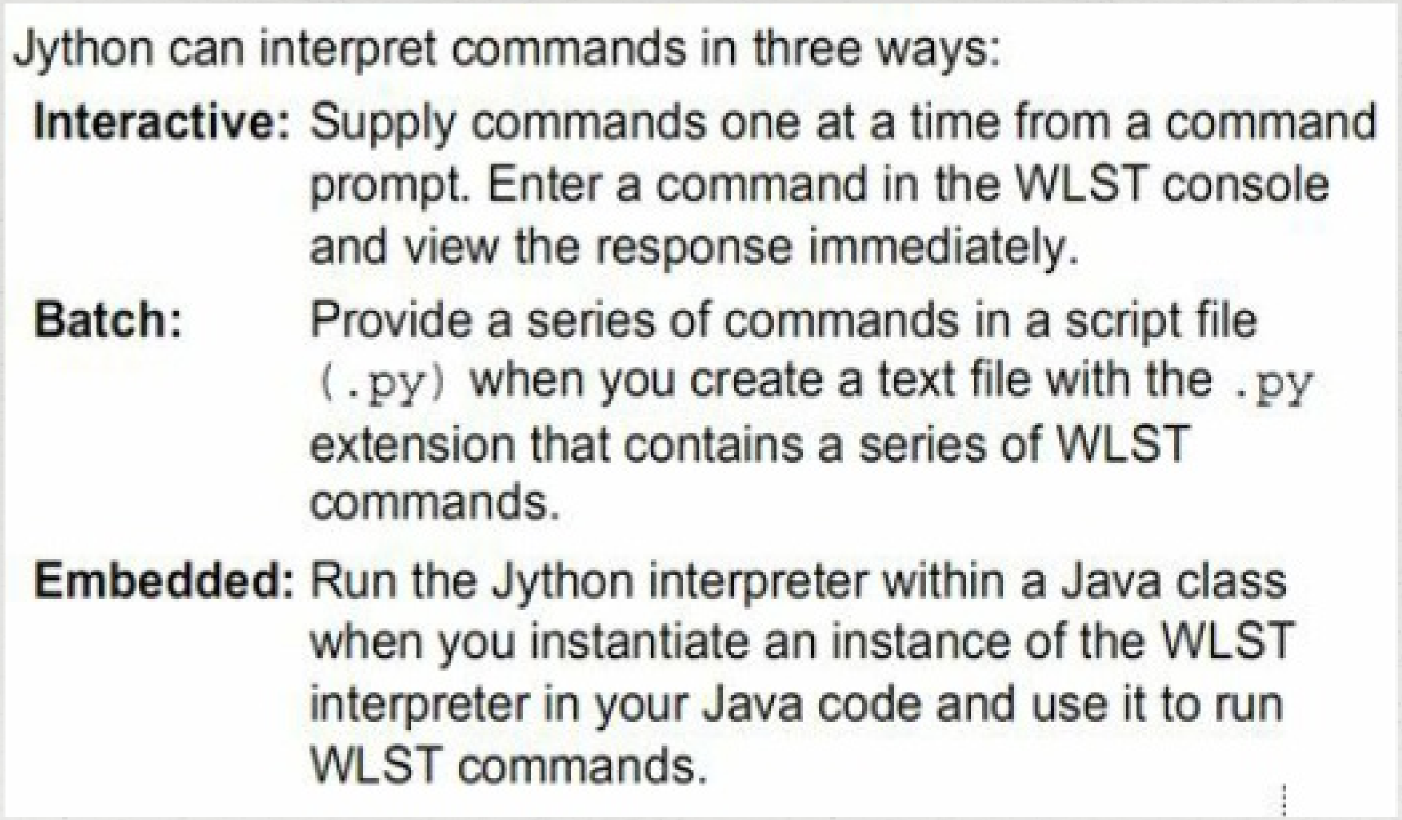


Jython

Jython is a Java implementation of the popular Python scripting language:

- Simple and clear syntax
- Indentation to structure code (white space critical!)
- Interactive command mode
- Custom commands
- Integration with any existing Java libraries

Using Jython



Jython can interpret commands in three ways:

Interactive:	Supply commands one at a time from a command prompt. Enter a command in the WLST console and view the response immediately.
Batch:	Provide a series of commands in a script file (`.py`) when you create a text file with the `.py` extension that contains a series of WLST commands.
Embedded:	Run the Jython interpreter within a Java class when you instantiate an instance of the WLST interpreter in your Java code and use it to run WLST commands.

WLST - How does it work?

- WLST uses T3 [proprietary RMI] protocol to connect with designated server
  - Node Manager
  - Administrative Server
  - Managed Server
- Interacts with Server Runtime MBeans (Runtime server state) Server
- Configuration MBeans (Representation of domain's XML Configuration)

Interaction Modes

> **Interactive**

1. enter a command and view the response at a command-line prompt
2. In online mode: shell maintains a persistent connection to a WLS instance

> **Script text**

1. file with a .py file extension
2. executed using Jython commands for running scripts
3. invoke a sequence of WLST commands without requiring your input

> **Embedded**

1. instantiate the WLST interpreter in your Java code
2. execute WLST commands from a Java program

Connection Modes

- **Offline:**

1. analogous to the Configuration Wizard
2. Uses the Offline Configuration Framework
3. Also used by the Configuration Wizard read and write access to the configuration data that is persisted in the domain's config directory or in a domain template JAR
4. Intended to create a domain or modify a non-running domain
5. Used during WLS install to create samples domains

- **Online:**

1. analogous to the Administration Console
2. JMX client Interacts with a server's MBeans
3. Intended as a runtime management tool: configuration, management, deployment, monitoring

WLST Offline Can/Can't Do

- **Can Do:**

1. Create/modify templates
2. Create domains
3. Extend domains
4. Access and modify the configuration for an offline domain

- **Can't Do:**

1. View runtime performance data
2. Modify security data

WLST Online Can/Can’t Do

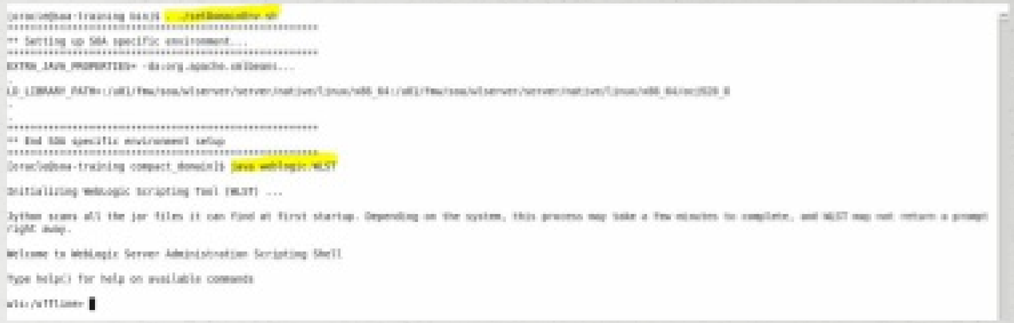
- **Can Do:**

1. Change configuration
2. View runtime data
3. Deploy applications
4. Start and stop servers

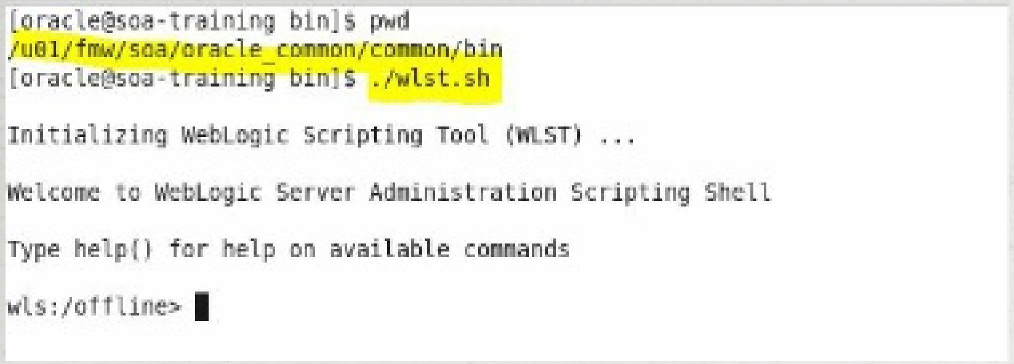
- **Can’t Do:**

1. Create a domain (must be offline mode)

Starting WLST



```
[oracle@soa-training bin]$ . ./setDomainEnv.sh
*****************************************************************
** Setting up SOA specific environment...
*****************************************************************
EXTRA_JAVA_PROPERTIES= -Da:org.apache.xmlbeans...
.
LD_LIBRARY_PATH=:/u01/fmw/soa/wlserver/server/native/linux/x86_64:/u01/fmw/soa/wlserver/server/native/linux/x86_64/oci920_8
.
.
*****************************************************************
** End SOA specific environment setup
*****************************************************************
[oracle@soa-training compact_domain]$ java weblogic.WLST

Initializing WebLogic Scripting Tool (WLST) ...

Jython scans all the jar files it can find at first startup. Depending on the system, this process may take a few minutes to complete, and WLST may not return a prompt
right away.

Welcome to WebLogic Server Administration Scripting Shell

Type help() for help on available commands

wls:/offline>
```



```
[oracle@soa-training bin]$ pwd
/u01/fmw/soa/oracle_common/common/bin
[oracle@soa-training bin]$ ./wlst.sh

Initializing WebLogic Scripting Tool (WLST) ...

Welcome to WebLogic Server Administration Scripting Shell

Type help() for help on available commands

wls:/offline>
```

WLST ONLINE

- Analogous to the Administration Console, but without the GUI
- JMX client; maintains a persistent connection
- Interacts with a server's/domain's MBeans
- Intended as a runtime management tool: configuration, management, deployment, monitoring

What is an MBean

- MBeans are managed beans, Java objects that represent resources to be managed. An MBean has a management interface consisting of:

1. Named and typed attributes that can be read and written
2. Named and typed operations that can be invoked
3. Typed notifications that can be emitted by the Mbean

For example, an MBean representing an application's configuration could have attributes representing the different configuration parameters, such as a cache size. Reading the CacheSize attribute would return the current size of the cache.

What is an MBean Server

- An MBean server is a repository of MBeans that provides management applications access to MBeans.
- Applications do not access MBeans directly, but instead access them through the MBean server with their unique ObjectName.

Traversing MBean Trees

- Simpler than JMX - no need to know the JMX object name
- MBeans are hierarchical, similar to a file system, with the DomainMBean at the top of the tree
- Use commands similar to Unix commands to traverse the tree: cd(), ls() Syntax is the same as with WLST Offline

- Domain MBean (root) |
- - - - MBean type (ServerMBean)
- |- - - MBean instance (ManagedServer1) |- - - MBean attributes & operations (AutoRestart)

Available MBean Trees

❖ **domainConfig**

- configuration hierarchy of the entire domain; represents the configuration MBeans in RuntimeMBeanServer

❖ **serverConfig**

- configuration hierarchy (configuration MBeans) of the server your are connected to

❖ **domainRuntime**

- hierarchy of runtime MBeans for the entire domain read only serverRuntime hierarchy of runtime MBeans for the server you are connected to

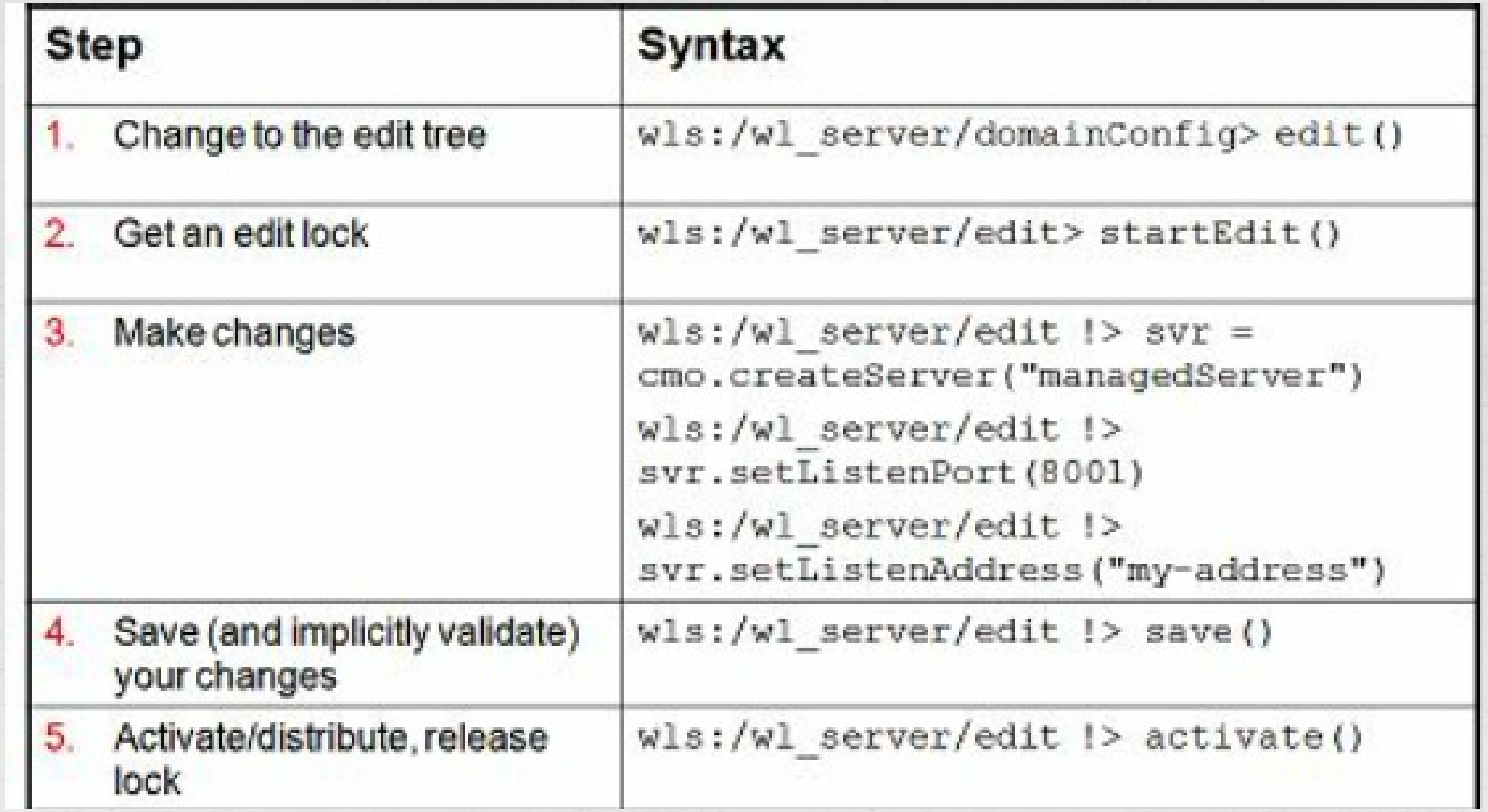
❖ edit

- writable domain configuration with pending changes; represents the configuration MBeans in the EditMBeanServer

Switching Between Trees

- Use the appropriate command to move to a different tree
- domainConfig()
- serverConfig()
- domainRuntime()
- serverRuntime()
- edit()

Changing Configuration in WLST Online



Step	Syntax
1. Change to the edit tree	`wls:/wl_server/domainConfig> edit()`
2. Get an edit lock	`wls:/wl_server/edit> startEdit()`
3. Make changes	`wls:/wl_server/edit !> svr = cmo.createServer("managedServer")` `wls:/wl_server/edit !> svr.setListenPort(8001)` `wls:/wl_server/edit !> svr.setListenAddress("my-address")`
4. Save (and implicitly validate) your changes	`wls:/wl_server/edit !> save()`
5. Activate/distribute, release lock	`wls:/wl_server/edit !> activate()`