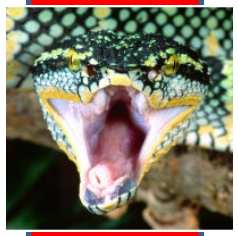
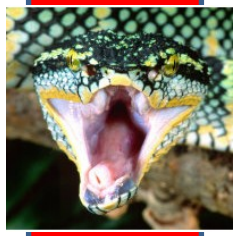


Lab 2b:
Create OpenCPI Project

Objectives

- Create an OpenCPI Project using the IDE
- Modify the Project's Namespace
- Update Project Registry to reflect changes
- Copy “provided” scripts into new Project
- Create Component Library in preparation for next lab

Step 1 - Project Creation using IDE



- Create “training_project” Project
  - **File → New → Other → ANGRYVIPER → OpenCPI Asset Wizard → Project**
  - **Project Name:** *training_project*
  - **Project Prefix:** *ocpi*
  - **Project Dependencies:** *ocpi.assets*

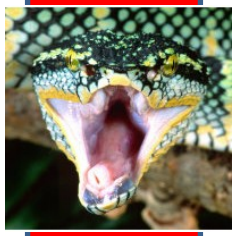
Step 2 – Modify Project's Namespace

- Namespace is constructed by two fields in the Project.mk
  - **PackagePrefix** – ex. `ocpi`
  - **PackageName** – ex. `training_project`
- By default, the IDE sets the PackageName to Project name
  - Currently, "`training_project`"
  - But we desire "`training`"
- **Edit** *Project.mk* and change **PackageName**
  - **"PackageName=training"**
  - **"PackagePrefix=ocpi"** (Simply confirm this is true)

Step 3 – Update Project Registry

- Examine the current state of the Project Registry
  - `$ ocpidev show projects --table`
  - Are the paths correct?
- Un-register a “specific” project from registry
  - `$ ocpidev unregister project ocpi.training_project`
- Re-register **(from within new Project directory)**
  - `$ ocpidev register project`
- Refresh the Project Explorer window in the AV IDE

Step 4 – Copy “provided” scripts into Project



- Copy “scripts/” directory to the top-level of the Project

```
$ cp -rf /home/training/provided/scripts/ /home/training/training_project/
```

Step 5 – Preparation for next lab

- Create a Component Library
  - **File → New → Other → ANGRYVIPER → OpenCPI Asset Wizard → Library**
    - (Verify Project path is /home/training/training_project/)
  - **Library Name:** *components*