Developing modular pipeline components with Viash : : **CHEAT SHEET**

Viash
from scripts to pipelines

Example Viash config (config.vsh.yaml)

```yaml
functionality:
  name: mycomponent
  description: |
    A useful description of the functionality
    provided by this component.
  usage: mycomponent --input foo.txt --output bar.txt
  arguments:
    - name: "--input"
      alternatives: ["-i"]
      type: file
      description: Input text file.
    - name: "--output"
      alternatives: ["-o"]
      type: file
      direction: output
  resources:
    - type: python_script
      path: script.py
  tests:
    - type: python_script
      path: test.py
platforms:
  - type: docker
    image: "python:3.8"
    setup: [...] # further customisation of container
  - type: native
  - type: nextflow
```

Example main script (script.py)

```python
## VIASH START
# this codeblock is for debugging purposes and
# is removed by Viash at runtime.
par = {'input': 'debugging.txt', 'output': 'out.txt'}
## VIASH END

# do something with par dict
print(f"Input: {par['input']}")
print(f"Output: {par['output']}")
```

Common commands during the development cycle

Prototyping and testing

```
viash run config.vsh.yaml -- \
  --help
```
Display CLI help page.
(Auto-generated from config file)

```
viash run config.vsh.yaml -- \
  -i in.txt -o out.txt
```
Run a component.
(CLI auto-generated from config file)

```
viash run config.vsh.yaml \
  -p native -- <args>
```
Run a component natively.

```
viash run config.vsh.yaml \
  -p docker -- <args>
```
Run a component inside a Docker.
(Input files mounted automatically)

```
viash test config.vsh.yaml
```
Run unit tests and display results.

Docker-specific commands

```
viash run config.vsh.yaml \
  -p docker -- ---dockerfile
```
View the Dockerfile used by Viash to build the Docker container.

```
viash run config.vsh.yaml \
  -p docker -- ---setup build
```
Rebuild a Docker container from scratch.

Building and deploying

```
viash build config.vsh.yaml \
  -p docker -o bin
```
Build a component into a containerised standalone executable.

```
bin/method --help
```
Show help page (same as above)

```
bin/method -i in.txt -o out.txt
```
Run a component (same as above)

```
viash build config.vsh.yaml \
  -p nextflow -o modules
```
Build a component into a standalone Nextflow module

Releasing and maintaining

```
viash ns build
```
Build all components under src/

```
viash ns test
```
Test all components under src/