



Hochschule  
Bonn-Rhein-Sieg  
University of Applied Sciences



# ROS Nodes, Topics, and Messages

Foundation Course

August 28, 2019

Hassan Umari

# 1. Recap

## 2. ROS nodes in Python

2.1 A simple ROS node in Python

2.2 Writing a publisher node in Python

2.3 creating a package

# Recap

## *Summary of yesterday's session*

- ROS is a collection of libraries and tools that helps you when you develop software for robots.
- ROS provides several ways to transfer data between nodes:
  1. ROS topics and messages (**publish/subscribe**).
  2. ROS services (**request/reply**).
  3. ROS actions (**request/reply**).
  4. Parameter server.

# Recap

## *Summary of yesterday's session*

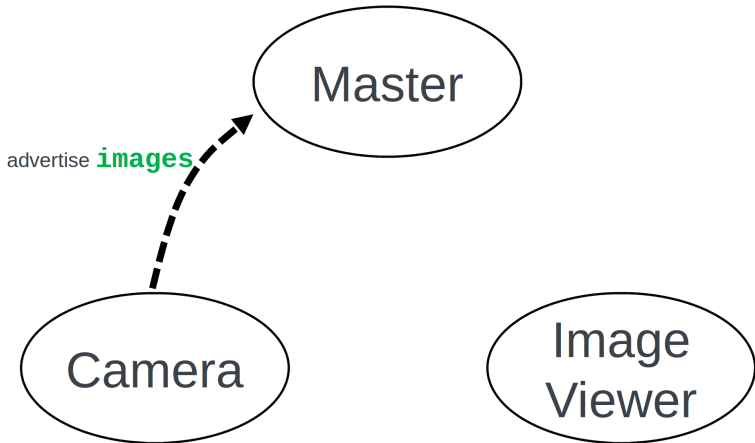
- We will focus today on ROS topics and messages..

```
graph TD; Master([Master]); Camera([Camera]); ImageViewer([Image Viewer]);
```

Master

Camera

Image  
Viewer









## 1. Recap

## 2. ROS nodes in Python

2.1 A simple ROS node in Python

2.2 Writing a publisher node in Python

2.3 creating a package



## 1. Recap

## 2. ROS nodes in Python

### 2.1 A simple ROS node in Python

### 2.2 Writing a publisher node in Python

### 2.3 creating a package



# A simple ROS node

```
#!/usr/bin/env python

import rospy
from time import sleep

rospy.init_node("print_text")

while True:
    print "Hello world!"
    sleep(1)
```

# A simple ROS node

## *ROS Nodes*

```
#!/usr/bin/env python

import rospy

rospy.init_node("print_text")
rate = rospy.Rate(1)

while not rospy.is_shutdown():
    print "Hello world!"
    rate.sleep()
```

# Three ways to run a node

## *ROS Nodes*

There are 3 ways to run a node:

1. Like you normally do (not recommended). Example (in case of python node):

```
python <file name>
```

2. using rosrun command:

```
rosrun <package name> <node name>
```

3. Using launch files. (we'll see it later)

# Let's create a package first!

## *ROS Nodes*

- ROS commands find your files (python scripts, cpp files, launch files, message definitions) if they are located in a package inside the workspace.
- Normally, a package looks like this:



# Let's create a package first!

## *ROS Nodes*

- go to the README and do the steps for **creating a package**.

# ROS commands

## *ROS Nodes*

- Navigate to a ROS package directly:

```
roscd <package name>
```

- run a node without navigating to it's directory:

```
roslaunch <package name> <executable>
```



# Let's create a package first!

## *ROS Nodes*

- go to the README and do the steps for **running a node**.

## 1. Recap

## 2. ROS nodes in Python

2.1 A simple ROS node in Python

2.2 Writing a publisher node in Python

2.3 creating a package



# Writing a publisher node in Python

## *ROS Nodes*

- Let's extend our previous node and make it publish.

# A simple ROS node



## 1. Recap

## 2. ROS nodes in Python

2.1 A simple ROS node in Python

2.2 Writing a publisher node in Python

2.3 creating a package



# ROS Nodes

