======== Humanoid properties ==========
Center of Mass kept at 0.93 meters
Head: 33.0cm above Center of Mass
IMU: 16.0cm below Center of Mass
Lidar: 15.0cm above head
Kinect: 7.0cm above Head

======== How to Load MAT Data Set==========
The data set is given in .mat files. You can use "load_data.py" in order to load the data in python. The file includes four functions - `"get_lidar()", "get_joint()", "get_rgb()", "get_depth()"`. Input for these functions is a file name (string type). For example, in order to log data, "train_joint0.mat". Then, try "get_joint("train_joint0").
The outputs of `"get_lidar()", "get_rgb()", "get_depth()"` are arrays and each element of the arrays is dictionary with corresponding components (described above). The length of each array is the number of steps.
The output of "`get_joint()`" is a dictionary with its corresponding components. Each dictionary key has numpy array and the length of each array is the number of steps.
Here is an example to load the data.

```
>>> import load_data as ld
>>> lidar0 = ld.get_lidar("train_lidar0")
```

======== How to Load PKL Data Set==========
In a directory where your data is:

```
>>> import pickle
>>> load = pickle.load(open("file_name.pkl","rb"))
```

======== P4_UTIL.PY ==========
This file contains functions that you can visualize your data and understand it.
Use corresponding data you loaded using "load_data.py" for the functions - `reply_lidar(), replay_rgb(), replay_depth()`.
For example:

```
>>> import load_data as ld
>>> lidar0 = ld.get_lidar("train_lidar0")
>>> import p4_util as util
>>> util.replay_lidar(lidar0)
```

You can change start_frame, interval, end_frame by yourself at line 20 in the code

```
for i in xrange(start_frame, end_frame, interval):
```

The function, `"get_joint_index()"`, returns a corresponding joint ID for the input joint name. Therefore, the input is a string.

========Joint angle (train_joint*.mat)==========
Joint angles

pos: joint positions (You don't need to use this information in this project)
ft_l, ft_r : (You don't need to use this information in this project)
ts: time stamps
gyro: gyro information
Acc: accelerometer
rpy: rpy angles
head_angles =array([[Neck angle],[Head angle]))

Note : ts for joints is ABSOLUTE time

======== Lidar Information (train_lidar*.mat) ==========

- Data format of the lidar data

  t: 1.4268e+09(absolute time)
  pose: [0 0 0] (absolute odometry)
  res: 0.0044 (radian, resolution)
  rpy: [-0.0120 -0.0164 -0.1107] (IMU roll pitch yaw)
  scan: [1x1081 single] (Radar scan data, range -135deg to 135 deg)

  (You should check the spec of Hukuyo lidar)
  http://www.hokuyo-aut.jp/02sensor/07scanner/download/pdf/UTM-30LX_spec_en.pdf

- Odometry
  lidar_pose: [x, y, theta]

  +x: forward from robot
  +y: left from robot
  +z: up from robot
  theta: rotation around +z

NOTE : lidar.t is Absolute time.

======== RGBD==========
Sensor: Kinect v2 http://smeenk.com/kinect-field-of-view-comparison/
Data 1: DEPTH.depth in millimeter readings
Data 2: RGB.image
*[Note] RGB.image data is chopped into a sequence of files because of its size issue.
    % For example,
    % RGB_3_1.mat has 1 to 300 frames

% RGB_3_2.mat has 301 to 600 frames
% ...
% RGB_3_4.mat has 901 to 996 frames
* RGBD files are provided only for the training set #0, 3, and the test set.

*[Note] DEPTH and RGB has other fields that contain metadata
*[Note] Intrinsic and extrinsic parameters are available in the subfolder 'cameraParam'

======== Calibration Data==========
Data: “IRcam_Calib_result.pkl” and “RGBcamera_Calib_result.pkl”
Instrinsic and Extrinsic Camera Parameters

IMPORTANT: This data contains neither the structure of the calibration objects nor the image coordinates of the calibration points. All those complementary variables are saved in the complete matlab data file Calib_Results.mat. This is converted to .pkl files for this class.
For more information regarding the calibration model visit:
http://www.vision.caltech.edu/bouguetj/calib_doc/

The data is a dictionary with following keys:
‘fc’ : Focal length
‘cc ’ : Principal point
'alpha_c' : Skew coefficient
'kc' : Distortion coefficients:
'fc_err' : Focal length uncertainty
'cc_err' : Principal point uncertainty
'alpha_c_err' : Skew coefficient uncertainty
'kc_err': Distortion coefficients uncertainty
'im_size': Image size