

Notes from web2py Online Lecture #2

```
## Folder Structure #####
```

```
* web2py.py - starts the rocket web server
```

```
* handlers/wsgihandler.py - must be copied to the root of web2py ro  
run with apache instead of rocket
```

```
* parameters_8000.py - port for web2py
```

```
* gluon folder holds web2py framework files
```

```
## DAL
```

```
#####
```

```
* set pool size for mysql database. If you don't set it, connections  
will be closed when done
```

```
* can't set a list of strings for various databases. You can look over  
multiple URIs up to five times in case it cannot connect to first  
database
```

```
db.-uri
```

```
this command will give you the db uri
```

```
## To get a record Given that Database name = person(name,  
phone)#####
```

```
row = db.person(1)
```

```
row.name
```

```
row.phone
```

```
row.first
```

```
row.last
```

```
## select single value
```

```
db.person(phone='112',name='Tim')
```

```
## select multiple values using (| or)
```

```
rows = db((db.person.name=='Max')|(db.person.name=='Tim')).select()
```

```
## select multiple values using (& and)
```

```
rows = db((db.person.name=='Max')& (db.person.name=='Tim')).select()
```

```
## select multiple values using (~ not)
```

```
rows = db(~(db.person.name=='Max')).select()
```

```
## select multiple values using (~ not)
```

```
rows = db(~(db.person.name=='Max')).select()
```

```
## select with (startswith)
```

```
rows = db(db.person.name.startswith('J')).select()
```

```

## select with (contains)
rows = db(db.person.name.contains('J')).select()

## select as part of a set (belongs)
rows = db(db.person.name.belongs(('Max','Tim','Jim'))).select()

## select as part of a sub-set ()
## In the example below we are checking to see if the first
## character begins with M or T
rows = db(db.person.name.[0:1].belongs(('M','T'))).select()

## print db._lastsql - will give you the generate sql from the DAL
syntax you last executed

## If you do not specify, you will return all the columns.
## However you can specify specific columns to return
rows = db(db.person.name.select(db.person.name))

## You can use groupby clause
rows = db(db.person.name.select(db.person.name.groupby=db.person.name))

## You can use groupby clause and count
rows =
db(db.person.name.select(db.person.name.db.person.id.count().groupby=db
b.person.name))

John,1
Max,3
Tim,1

db.define_table('dog',Field('name'),Field('owner','reference person'))

## Using groupby and count, then loop through rows to get owner name
rows =
db(db.dog).select(db.dog.owner,db.dog.id.count(),groupby=db.dog.ownder
)

for row in rows: print row.dog.owner.name, row[db.dog.id.count()]

## Use a join - retrieve all rows from both tables
query = db.dog.owner == db.person.id
rows = db(query).select()

1,Max,111,1,Skipper,1
1,Max,111,2,Snoopy,1
2,Tim,112,3,Wolf,2

## Use a join, but only select specific columns in the query
query = db.dog.owner == db.person.id

```

```

rows = db(query).select(db.person.name, db.dog.name)

person.name, dog.name
Max,1
Max,2
Tim,3

## Use a join, but only select specific columns in the query and
groupby person id
query = db.dog.owner == db.person.id
rows = db(query).select(db.person.name,
db.dog.id.count(),groupby=db.person.id)

person.name,Count(dog.id)
Max,2
Tim,1

>>>print rows[0]
<Row {'person': {'name': 'Max'}, 'dog': {'name': 'Skipper'}}>

## Use orderby on multiple columns
rows = db(db.person).select(orderby=db.person.name|db.person.id)

## Using an outer or left join. This will include all of the owners
along
## with their dogs, but also includes owners names who do not have
dogs.
query = db.dog.owner == db.person.id
rows =
db(db.person).select(db.person.name,db.dog.name,left=db.dog.on((query)
)

>>>print rows
person.name,dog.name
Max,Skipper
Max,Snoopy
Tim,Wolf
John,<NULL>
Max,<NULL>
Max,<NULL>

#####Strategies for creating edit, view, update, callback etc
#####
## at the top default.py
## http://hostname/app/default/index
## http://hostname/app/default/create_post/<category>
## http://hostname/app/default/edit_post/<category>/<id>
## http://hostname/app/default/<func>/<id>

```

```
def index():
    return locals()

def upload_assignment():
    return locals()

def edit_assignment():
    return locals()

def edit_instructor():
    id = request.args(0, cast=int)

def list_assignments_by_datetime():
    page = request.args(0, cast=int)

def list_assignments_by_instructor():
    page = request.args(0, cast=int)
```