

```
SQLFORM(table, record=None, deletable=False,  
linkto=URL(...), upload=URL(...),  
fields=[...], labels={}, col3={},  
submit_button='Submit',  
delete_label='Check to delete:',  
id_label='Record id:', showid=True, **attrs)
```

http://127.0.0.1:8000/app/default/index

### app/models/db.py

```
db=SQLDB('sqlite://data.db',pools=0)  
  
db.define_table('person',  
db.Field('name','string'))  
  
db.person.name.default='Anonymous'  
db.person.name.requires=IS_NOT_EMPTY()
```

### app/controllers/default.py

```
def index():  
    form=SQLFORM(db.person)  
    if form.accepts(request.vars,session):  
        session.flash='record inserted'  
        redirect(URL(r=request,f='index'))  
    elif form.errors:  
        response.flash='there are errors'  
    persons=db(db.person.id>0).select()  
    return dict(form=form,persons=persons)
```

### app/views/default/index.html

```
{{extend 'layout.html'}}  
<h2>Create Person</h2>  
{{=form}}  
<h2>Persons</h2>  
<ul>  
    {{for person in persons:}}  
        {{=LI(person.name)}}  
    {{pass}}  
</ul>
```

```
type='string'  
length=64  
default=None  
required=False  
requires=validators  
ondelete='CASCADE'  
notnull=False  
unique=False  
uploadfield=None
```

```
string  
text  
blob  
password  
upload  
boolean  
integer  
double  
date  
time  
datetime
```

```
request.cookies  
request.args  
request.vars  
request.get_vars  
request.post_vars  
request.client  
request.application  
request.controller  
request.function  
request.folder  
request.env.host_name  
request.env.path_info  
request.env...
```

```
db.table.insert(field=val)  
query=db.table.id>5  
query=query1&query2  
query=query1|query2  
db(query).count()  
db(query).delete()  
rows=db(query).select(  
    field1, field2, ...,  
    left=[db.table.on(query)],  
    orderby=field|~field,  
    groupby=field|field  
    limitby=(0,10),  
    cache=(cache.ram,5000))
```

```
URL(r=request,a='a',c='c',f='f',  
args=[1,2],vars=dict(q=3))  
=> '/a/c/f/1/2?q=3'
```

```
XML(string,sanitize=False,  
permitted_tags=[..],  
allowed_attributes={...})
```

```
BEAUTIFY(object)
```

```
T.force('en-US')  
T('message')  
=> 'translated message'
```

```
DIV('a',_class='b') =>  
<div class="b">a</div>
```

```
CODE('print a',language='web2py'  
or'python' or 'html' or 'cpp')
```

```
HTML BODY SPAN A IMG P CENTER  
HEAD TITLE LINK SCRIPT STYLE  
META HR BR IFRAME H1 H2 H3 H4 H5  
H6 FORM INPUT TEXTAREA LABEL  
FIELDSET SELECT OPTION B EM I TT  
UL OL LI TABLE THEAD TBODY TFOOT  
TH TR TD EMBED OBJECT PRE
```

```
ON=True
```

```
IS_NOT_EMPTY(error_message='...')  
IS_ALPHANUMERIC()  
IS_DATE('%Y-%m-%d')  
IS_DATETIME('%Y-%m-%d %H:%M:%S')  
IS_EXPR('value>3')  
IS_FLOAT_IN_RANGE(0,10)  
IS_INT_IN_RANGE(0,10)  
IS_IN_SET(['a','b','c'])  
IS_LENGTH(32)  
IS_LOWER(), IS_UPPER()  
IS_MATCH('\w{5}')IS_TIME()  
IS_URL(), IS_EMAIL()  
IS_IN_DB(db,'table.field','%(field)s')  
IS_NOT_IN_DB(db,'table.field')  
IS_NULL_OR(validator)  
IS_LIST_OF(validator)  
CLEANUP(),  
CRYPT()
```

```
response.render('view.html',dict(...))  
response.xmlrpc(response,[functions])  
response.stream(filename,chunk_size)  
response.cookies['name']='value'  
response.headers['name']='value'  
response.view='default/index.html'  
response.flash='a flashing message'  
response.status=200  
response.title='this app'  
response.subtitle='the best'  
response.author='you'  
response.keywords='keywords'  
response.description='something'  
response.menu=[  
    ['name',False,URL(...),[]],  
]
```