

OFFLINE STORAGE

Local Storage

TESTING & USING

```
if(typeof(Storage) === "undefined") {  
    alert("This device is not capable of storing local data.");  
}
```

window.localStorage

```
localStorage.myValue = "Value";
```

```
localStorage.setItem("key", "value");
```

```
var value = localStorage.getItem("key");
```

```
if(localStorage.myVale == "Value") { ..... }
```

```
localStorage.removeItem("key");
```

```
localStorage.clear();
```


OFFLINE STORAGE

WebSQL

BASICS

```
openDatabase(DATABASE-NAME, VERSION-NO, DESCRIPTION, SIZE, CALLBACK);
```

```
var db = openDatabase( "database", "1.0", "Employees Index", 2*1024*1024);
```

```
db.transactions(function(tx) {  
    tx.executeSql("..... SQL Query .....");  
});
```

```
db.transactions(function(tx) {  
    tx.executeSql("..... SQL Query .....");  
    tx.executeSql("..... SQL Query .....");  
    tx.executeSql("..... SQL Query .....");  
    tx.executeSql("..... SQL Query .....", [], callbackFunction);  
});
```

```
tx.executeSql("SELECT * FROM people", [], function(tx, results) {  
    var numberOfResults = results.rows.length;  
    var personSurname   = results.rows.item(0).surname;  
});
```


SQL QUERIES - CRUD

```
CREATE TABLE `<<TABLE_NAME>>` (  
  `id` INTEGER PRIMARY KEY AUTOINCREMENT,  
  `category` INTEGER,  
  `brand` TEXT,  
  `logo` BLOB,  
  `cost` REAL,  
  `product_name` TEXT,  
  `food_guide_notes` TEXT,  
)
```

```
CREATE TABLE IF NOT EXISTS '<<TABLE_NAME>>' (.....)
```

```
DROP TABLE <<TABLE_NAME>>
```

```
INSERT INTO <<TABLE_NAME>> (column1, column2, column3)  
VALUES ('val1', 'val2', 'val3')
```

```
SELECT * FROM <<TABLE_NAME>> WHERE id='value' ORDER BY column ASC;
```

```
UPDATE <<TABLE_NAME>>  
SET column1='value1', column2='value2'  
WHERE id='value'
```

```
DELETE FROM <<TABLE_NAME>>  
WHERE id='value'
```

CREATING/OPENING DATABASE

```
var db = openDatabase('mydb', '1.0', 'Test DB', 2 * 1024 * 1024);
```

Above method took following five paramters:

1. Database name
2. Version number
3. Text description
4. Size of database
5. Creation callback

```
var db = openDatabase('mydb', '1.0', 'Test DB', 2 * 1024 * 1024);  
db.transaction(function (tx) {  
    tx.executeSql('CREATE TABLE IF NOT EXISTS LOGS (id unique, log)');  
});
```


WRITING & READING DATA

```
var db = openDatabase('mydb', '1.0', 'Test DB', 2 * 1024 * 1024);
db.transaction(function (tx) {
    tx.executeSql('CREATE TABLE IF NOT EXISTS LOGS (id unique, log)');
    tx.executeSql('INSERT INTO LOGS (id, log) VALUES (1, "foobar")');
    tx.executeSql('INSERT INTO LOGS (id, log) VALUES (2, "logmsg")');
});
```

We can pass dynamic values while creating entering as follows:

```
var db = openDatabase('mydb', '1.0', 'Test DB', 2 * 1024 * 1024);
db.transaction(function (tx) {
    tx.executeSql('CREATE TABLE IF NOT EXISTS LOGS (id unique, log)');
    tx.executeSql('INSERT INTO LOGS
                    (id,log) VALUES (?, ?)', [e_id, e_log];
});
```

```
var db = openDatabase('mydb', '1.0', 'Test DB', 2 * 1024 * 1024);
db.transaction(function (tx) {
    tx.executeSql('CREATE TABLE IF NOT EXISTS LOGS (id unique, log)');
    tx.executeSql('INSERT INTO LOGS (id, log) VALUES (1, "foobar")');
    tx.executeSql('INSERT INTO LOGS (id, log) VALUES (2, "logmsg")');
});
db.transaction(function (tx) {
    tx.executeSql('SELECT * FROM LOGS', [], function (tx, results) {
        var len = results.rows.length, i;
        msg = "<p>Found rows: " + len + "</p>";
        document.querySelector('#status').innerHTML += msg;
        for (i = 0; i < len; i++){
            alert(results.rows.item(i).log );
        }
    }, null);
});
```