```
<summary>
Returns exists save with key
</summary>
<param name="key"></param>
<returns>If save exist return true</returns>
public static bool HasKey(string key)


<summary>
Remove all saves
</summary>
public static void DeleteAll()


<summary>
Remove save with key
</summary>
<param name="key"></param>
<exception cref="ArgumentException">Key not exist</exception>
public static void DeleteKey(string key)


<summary>
Save int with key
</summary>
<param name="key"></param>
<param name="value"></param>
public static void SetInt(string key, int value)


<summary>
Get int with key
</summary>
<param name="key"></param>
<returns>Decrypt int with key</returns>
<exception cref="ArgumentException">Key not exist</exception>
public static int GetInt(string key)


<summary>
Get int with key, if key not exist return default value
</summary>
<param name="key"></param>
<param name="defaultValue">If key not exist return default
value</param>
<returns></returns>
public static int GetInt(string key, int defaultValue)
```

```
<summary>
Save int with choose key size. The larger the key, the longer the
execution.
</summary>
<param name="key"></param>
<param name="value"></param>
<param name="keySize"></param>
public static void SetInt(string key, int value, KeySize keySize)


<summary>
Get int with key size. The larger the key, the longer the execution.
</summary>
<param name="key"></param>
<param name="keySize"></param>
<returns>Decrypt int with key</returns>
<exception cref="ArgumentException"></exception>
public static int GetInt(string key, KeySize keySize)


<summary>
Get int with key size, if key not exist return default value. The
larger the key, the longer the execution.
</summary>
<param name="key"></param>
<param name="keySize"></param>
<param name="defaultValue"></param>
<returns>Decrypt int with key or default value if key is not
exist</returns>
public static int GetInt(string key, KeySize keySize, int
defaultValue)


<summary>
Save float with key.
</summary>
<param name="key"></param>
<param name="value"></param>
public static void SetFloat(string key, float value)


<summary>
Get data
</summary>
<param name="key"></param>
<param name="defaultValue">Will return this, if key is not
exist</param>
<returns></returns>
public static float GetFloat(string key, float defaultValue)
```

```
<summary>
Get data
</summary>
<param name="key"></param>
<returns></returns>
<exception cref="ArgumentException">Key is not exist</exception>
public static float GetFloat(string key)


<summary>
Save float with choose key size. The larger the key, the longer the
execution.
</summary>
<param name="key"></param>
<param name="value"></param>
<param name="keySize"></param>
public static void SetFloat(string key, float value, KeySize
keySize)


<summary>
Get data with key size
</summary>
<param name="key"></param>
<param name="keySize"></param>
<returns></returns>
<exception cref="ArgumentException">Key is not exist</exception>
public static float GetFloat(string key, KeySize keySize)


<summary>
Get data with key size, if key is not exist will return default
value
</summary>
<param name="key"></param>
<param name="keySize"></param>
<param name="defaultValue">Will return this, if key is not
exist</param>
<returns></returns>
public static float GetFloat(string key, KeySize keySize, float
defaultValue)


<summary>
Save bool with key.
</summary>
<param name="key"></param>
```

```
<param name="value"></param>
public static void SetBool(string key, bool value)


<summary>
Get data
</summary>
<param name="key"></param>
<returns></returns>
<exception cref="ArgumentException">Key is not exist</exception>
public static bool GetBool(string key)


<summary>
Get data
</summary>
<param name="key"></param>
<param name="defaultValue">Will return this, if key is not
exist</param>
public static bool GetBool(string key, bool defaultValue)


<summary>
Save bool with choose key size. The larger the key, the longer the
execution.
</summary>
<param name="key"></param>
<param name="value"></param>
<param name="keySize"></param>
public static void SetBool(string key, bool value, KeySize keySize)


<summary>
Get data with key size
</summary>
<param name="key"></param>
<param name="keySize"></param>
<returns></returns>
<exception cref="ArgumentException">Key is not exist</exception>
public static bool GetBool(string key, KeySize keySize)


<summary>
Get data with key size, if key is not exist will return default
value
</summary>
<param name="key"></param>
<param name="keySize"></param>
```

```
<param name="defaultValue">Will return this, if key is not
exist</param>
<returns></returns>
public static bool GetBool(string key, KeySize keySize, bool
defaultValue)


<summary>
Save string with key.
</summary>
<param name="key"></param>
<param name="value"></param>
public static void SetString(string key, string value)


<summary>
Get data
</summary>
<param name="key"></param>
<returns></returns>
<exception cref="ArgumentException">Key is not exist</exception>
public static string GetString(string key)


<summary>
Get data
</summary>
<param name="key"></param>
<param name="defaultValue">Will return this, if key is not
exist</param>
public static string GetString(string key, string defaultValue)


<summary>
Save string with choose key size. The larger the key, the longer the
execution.
</summary>
<param name="key"></param>
<param name="value"></param>
<param name="keySize"></param>
public static void SetString(string key, string value, KeySize
keySize)


<summary>
Get data with key size
</summary>
<param name="key"></param>
<param name="keySize"></param>
```

```
<returns></returns>
<exception cref="ArgumentException">Key is not exist</exception>
public static string GetString(string key, KeySize keySize)


<summary>
Get data with key size, if key is not exist will return default
value
</summary>
<param name="key"></param>
<param name="keySize"></param>
<param name="defaultValue">Will return this, if key is not
exist</param>
<returns></returns>
public static string GetString(string key, KeySize keySize, string
defaultValue)


<summary>
Save serializable object with key.
</summary>
<param name="key"></param>
<param name="value"></param>
<exception cref="ArgumentException"></exception>
public static void SetObject(string key, object value)


<summary>
Get data
</summary>
<param name="key"></param>
<returns></returns>
<typeparam name="T">Serializable class</typeparam>
<exception cref="ArgumentException">Key is not exist</exception>
public static T GetObject<T>(string key)


<summary>
Get data
</summary>
<param name="key"></param>
<param name="defaultValue">Will return this, if key is not
exist</param>
public static T GetObject<T>(string key, T defaultValue)


<summary>
Save serializable object with choose key size. The larger the key,
the longer the execution.
```

```
</summary>
<param name="key"></param>
<param name="value"></param>
<param name="keySize"></param>
<exception cref="ArgumentException">Type need be
serializable</exception>
public static void SetObject(string key, object value, KeySize
keySize)


<summary>
 Get data with key size
 </summary>
 <param name="key"></param>
 <param name="keySize"></param>
 <typeparam name="T">Serializable class</typeparam>
 <returns></returns>
 <exception cref="ArgumentException">Key is not exist</exception>
public static T GetObject<T>(string key, KeySize keySize)


<summary>
Get data with key size, if key is not exist will return default
value
</summary>
<param name="key"></param>
<param name="keySize"></param>
<param name="defaultValue">Will return this, if key is not
exist</param>
<typeparam name="T">Serializable class</typeparam>
<returns></returns>
public static T GetObject<T>(string key, KeySize keySize, T
defaultValue)
```